



AFRL-RY-WP-TR-2022-0171

AMYGDALA MODELING WITH CONTEXT AND MOTIVATION USING SPIKING NEURAL NETWORKS FOR ROBOTICS APPLICATIONS

**Matthew Aaron Zelgen
Wright State University**

**MAY 2022
Final Report**

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

See additional restrictions described on inside pages

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE May 2022		2. REPORT TYPE Thesis		3. DATES COVERED	
				START DATE 19 May 2022	END DATE 19 May 2022
4. TITLE AND SUBTITLE AMYGDALA MODELING WITH CONTEXT AND MOTIVATION USING SPIKING NEURAL NETWORKS FOR ROBOTICS APPLICATIONS					
5a. CONTRACT NUMBER FA8650-18-2-1645		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61102F/62212F/62204F	
5d. PROJECT NUMBER 3001/6095		5e. TASK NUMBER N/A		5f. WORK UNIT NUMBER Y1SH	
6. AUTHOR(S) Matthew Aaron Zelgen					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Wright State University 3640 Colonel Glenn Hwy, Dayton, OH 45435					8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command, United States Air Forces				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/Rydi	11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RY-WP-TR-2022-0171
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES PAO case number AFRL-2022-2370, Clearance Date 19 May. A thesis submitted to Wright State University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering. The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work. Report contains color.					
14. ABSTRACT Cognitive capabilities for robotic applications are furthered by developing an artificial amygdala that mimics biology. The amygdala portion of the brain is commonly understood to control mood and behavior based upon sensory inputs, motivation, and context. This research builds upon prior work in creating artificial intelligence for robotics which focused on mood generated actions. However, recent amygdala research suggests a void in greater functionality. This work developed a computational model of an amygdala, integrated this model into a robot model, developed a comprehensive integration of the robot for simulation, and live embodiment. The developed amygdala, instantiated in the Nengo Brain Maker environment, leveraged spiking neural networks and the semantic pointer architecture to allow abstraction of neuron ensembles into high-level concept vocabularies. Test and validation were performed on a TurtleBot in both simulated (Gazebo) and live testing. Results were compared to a baseline model which has a simplistic, amygdala-like model. Metrics of nearest distance and nearest time were used for assessment. The amygdala model is shown to outperform the baseline in both simulation, with a 70.8% improvement in nearest distance and, 4% improvement in nearest time, and real applications with a 62.4% improvement in nearest distance. Notably, this performance occurred despite a five-fold increase in architecture size and complexity.					
15. SUBJECT TERMS Autonomy, neuromorphic, robotics					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR		18. NUMBER OF PAGES 143
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			
19a. NAME OF RESPONSIBLE PERSON Trevor Bihl				19b. PHONE NUMBER (Include area code)	

AMYGDALA MODELING WITH CONTEXT AND MOTIVATION USING SPIKING NEURAL NETWORKS FOR ROBOTICS APPLICATIONS

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

By

Matthew Aaron Zeglen

B.S., Ohio State University, 2007

2022

Wright State University

iii

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

April 2022

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY
SUPERVISION BY Matthew Aaron Zeglen ENTITLED Amygdala Modeling with
Context and Motivation Using Spiking Neural Networks for Robotic Application BE
ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF Master of Science in Electrical Engineering

Luther Palmer III, Ph.D.
Thesis Director

Brian Rigling, Ph.D.
Chair, Department of Engineering &
Computer Science

Committee on Final Examination:

Luther Palmer III, Ph.D.

Xiaodong Zhang, Ph.D.

Trevor J. Bihl, Ph.D.

Barry Milligan, Ph.D.
Dean of the Graduate School

ABSTRACT

Zeglen, Matthew Aaron. M.S.E.E., Department of Engineering, Electrical Engineering, Wright State University, 2022. *Amygdala Modeling with Context and Motivation Using Spiking Neural Networks for Robotic Application.*

Cognitive capabilities for robotic applications are furthered by developing an artificial amygdala that mimics biology. The amygdala portion of the brain is commonly understood to control mood and behavior based upon sensory inputs, motivation, and context. This research builds upon prior work in creating artificial intelligence for robotics which focused on mood generated actions. However, recent amygdala research suggests a void in greater functionality. This work developed a computational model of an amygdala, integrated this model into a robot model, developed a comprehensive integration of the robot for simulation, and live embodiment. The developed amygdala, instantiated in the Nengo Brain Maker environment, leveraged spiking neural networks and the semantic pointer architecture to allow abstraction of neuron ensembles into high-level concept vocabularies. Test and validation were performed on a TurtleBot in both simulated (Gazebo) and live testing. Results were compared to a baseline model which has a simplistic, amygdala-like model. Metrics of nearest distance and nearest time were

used for assessment. The amygdala model is shown to outperform the baseline in both simulation, with a 70.8% improvement in nearest distance and, 4% improvement in nearest time, and real applications with a 62.4% improvement in nearest distance. Notably, this performance occurred despite a five-fold increase in architecture size and complexity.

TABLE OF CONTENTS

1	INTRODUCTION	11
1.1	Operational Motivation	14
1.2	Technical Motivation	14
1.3	Research Contribution	15
1.4	Research Objectives	16
2	BACKGROUND	18
2.1	Cognition and the Brain	18
2.2	Mimicking the Brain	21
2.2.1	Limitations of Past Instantiations	22
2.2.2	Spiking Neural Networks	23
2.2.3	Neural Engineering Framework	24
2.2.4	SPA Networks	25
2.2.5	Neuromorphic Hardware	26
2.3	Robotic Intelligence	26
2.3.1	Robotic Control Applications	28
3	DEVELOPING A MOOD MODEL FOR ROBOTICS	29
3.1	Background Reference Model	29
3.1.1	Concepts and Extensions	29
3.1.2	Implementation Considerations	30
3.2	Simple Amygdala Model, Model A	31
3.3	Amygdala Model B	32
3.4	Amygdala Model 0	33

3.5	Amygdala Model 1	34
3.6	Amygdala Model 2	35
3.7	Adding Motivation, Amygdala Model 2M	36
3.8	Adding Context, Amygdala Model 2MC.....	38
3.9	Complexity of Models	39
4	PROOF OF CONCEPT DEMONSTRATION.....	40
4.1	Metrics	40
4.2	Guide to Reading Plots	41
4.3	Demonstration for Proof of Concept.....	48
5	SIMULATION AND ROBOTIC EMBODIMENT	70
5.1	TurtleBot 2.0	70
5.2	Integration	70
5.3	Simulation Environment	72
5.4	Live Embodiment.....	76
6	RESULTS AND ANALYSIS	81
6.1	Metrics and Framework	82
6.2	Embodiment: M&S Amygdala vs. Baseline.....	82
6.2.1	Comparing Context and Motivation	89
6.2.2	Simulation Errors.....	94
6.3	Embodiment: Live System Demonstration.....	95
6.3.1	Implementation Challenges	98
6.3.2	Identification Challenges	98
6.3.3	Timing Analysis Challenges.....	99
6.3.4	Robot Movement Challenges	101

6.3.5	Impacts on Run Times	102
7	CONCLUSIONS AND FUTURE RESEARCH	104
7.1	Summary of Model Results	104
7.2	Future research.....	106
8	REFERENCES	109
A	ADDITIONAL INFORMATION	115
A.1	TurtleBot Reference Tables	115
B	ROBOT MOVEMENT PLOTS	119
B.1	Timing Analysis Plots.....	124
C	EQUIPMENT AND SETUP	130
C.1	Computer Equipment.....	130
C.2	Software	130
C.3	TurtleBot 2.0	132
D	APPENDIX - LINUX-ROS-NENGO SETUP	134
D.1	Windows Software.....	134
D.2	Linux and Docker Image	134
D.3	Developed Tools	134
D.4	Windows Setup	134
D.5	Models and Supporting Scripts.....	136
D.6	Fakenet Software	136

LIST OF FIGURES

Figure 2.1: Functional connections of the central nucleus (CEA) of the amygdala [13] .	20
Figure 3.1: Block diagram of the simple amygdala, Model A, displayed in Nengo	31
Figure 3.2: Block diagram of Model B displayed in Nengo	32
Figure 3.3: Block diagram of Model 0 displayed in Nengo	33
Figure 3.4: Block diagram of Model 1 displayed Nengo.....	34
Figure 3.5: Block diagram of Model 2 displayed in Nengo	35
Figure 3.6: Model 2 with the addition of the motivation circuit, outlined in red	36
Figure 3.7: Block diagram for Model 2MC with elements for added context. Binding circuits are outlined in blue and unbinding circuits outlined in red.....	38
Figure 4.1: Model 2MC used to explain probe information, red indicates the location of a probe with its identification number	41
Figure 4.2: Example of single detection input probe location 1.	42
Figure 4.3: Example of two detection inputs at probe location 1.	43
Figure 4.4: Example of applied context input at probe location 2.....	43
Figure 4.5: Example of applied motivation input at probe location 6.	44
Figure 4.6: Example of motive input and associated mood output probe locations 6 (top) and 7 (bottom).....	45
Figure 4.7: Example of context signal flow from input, unbinding, and association. Probes, from top to bottom, are at locations 2, 4, 5, and 9. Arrows and bands show signal routing and contributions.	46
Figure 4.8: Example of context signal flow with multiple contexts shown at once.	47
Figure 4.9: Response from Simple model.....	49

Figure 4.10: Model B input to output response over 5s simulation time with a single input	50
Figure 4.11: Model B output response from two inputs over 5s simulation	51
Figure 4.12: Model 0 selection path signals.	53
Figure 4.13: Model 1 input to output response over 5s simulation time	55
Figure 4.14: Model 2 basal-thalamus path showing WTA behavior with two inputs.	58
Figure 4.15: Model 2 input to output response over 5s simulation time	59
Figure 4.16: Effects of motivation with single detection input. Central and motivation outputs combine to produce the overall mood output.....	62
Figure 4.17: Effects of a detected object, motivation, and context added at the end. Central, motivation, and context outputs combine to produce the final mood output.....	64
Figure 4.18: Multiplexer effect of context unbinding – single detection and a single association table	66
Figure 4.19: Context unbinding with unique association tables	67
Figure 4.20: Increasing thresholds to 0.55 reduces unbinding noise with the noisy data shown in grey	69
Figure 5.1: Amygdala integrated into the robot model	71
Figure 5.2: Top-down view of developed simulated environment in Gazebo with annotation, actual environment is fully 3D.....	73
Figure 5.3: Example of image recognition in simulation	75
Figure 5.4: TurtleBot 2.0 with TX2 board and Xbox Kinect from the side (left) and front (right)	77
Figure 5.5: Live TurtleBot test environment with target marked with a red diamond	78

Figure 5.6: View from the TurtleBot's starting location toward the target	79
Figure 6.1: Multiple ANOVA of model and context for simulation results on nearest distance	86
Figure 6.2 Model performance given context showing least squares mean for the nearest distance	87
Figure 6.3: Multiple ANOVA of model and context for simulation results on nearest time	88
Figure 6.4: Model performance given context showing least squares mean for the nearest time	89
Figure 6.5: Multiple ANOVA for context and motivation on simulation data for nearest distance.	90
Figure 6.6: Model performance given context and motivation on nearest distance.	91
Figure 6.7: Overlay of robot movements given "Practice" context	92
Figure 6.8: Multiple ANOVA for context and motivation on simulation data for nearest time.	93
Figure 6.9: Model performance given context and motivation on nearest time.	94
Figure 6.10: Robot movement of Amygdala Model 2MC with "Living Room" context, all motivations.....	97
Figure 6.11: Console showing the number of detections in the real world exceeded the limit of three.....	98
Figure 6.12: Object and mood timing information from live embodiment Run A6.....	100
Figure 6.13: Object and mood pulses calculated from the live embodiment Run A6....	101

Figure B.1: Robot movement from simulations, both models, “Normal” motivation, all contexts	119
Figure B.2: Robot movement from simulations, arranged by motivation	120
Figure B.3: Robot movement from simulations, arranged by context.....	121
Figure B.4: Robot movement from live embodiment, both models, “Normal” motivation, all contexts	122
Figure B.5: Robot movement from live embodiment, arranged by motivation.....	123
Figure B.6: Robot movement from live embodiment, arranged by context	124
Figure B.7: Object and mood timing information from simulation Run A1	125
Figure B.8: Pulse form of object and mood data from simulation Run A1	125
Figure B.9: Object and mood timing information from live embodiment Run A4	126
Figure B.10: Pulse form of object and mood data from live embodiment Run A4.....	126
Figure B.11: Block diagram of motor control connectivity.....	127
Figure B.12: Overlaying action (scaled, dashed) with mood responses	128
Figure B.13: Linear X motor response with vertical markers for action states	129
Figure D.14: Initial integration with ROS and Fakenet without motivation	138

LIST OF TABLES

Table 1.1: Relational mapping between technical contributions in previous related work and <i>current</i> research contributions. The × symbol denotes areas addressed.	16
Table 3.1: Relating motivation to mood	37
Table 3.2: Neuron count per model	39
Table 4.1: Proof of Concept Results. Simulations are 5s with dt=0.001	48
Table 4.2: Model accuracy overview	48
Table 4.3: Model accuracy from a single input. Yellow items yield close but different results.	51
Table 4.4: Model B accuracy from two inputs.	52
Table 4.5: Model 0 expected versus measured results with a single input. The results in red are incorrect.	54
Table 4.6: Model 0 expected versus measured results with two inputs. The results in red are incorrect.	54
Table 4.7: Model 1 expected versus measured results for a single input. Red entries are incorrect.	56
Table 4.8: Model 1 accuracy results based on two inputs. Red entries are incorrect.	57
Table 4.9: Model 2 expected versus measured results for a single input. Red entries are incorrect.	60
Table 4.10: Model 2 expected versus measured with two inputs. Yellow items yield close but different results.	60
Table 4.11: Accuracy for adding motivation with a single input	63
Table 4.12: Test associations to demonstrate output changes with context	65

Table 4.13: Accuracy measurements for context modeling. Items in red are errors.	68
Table 6.1: Simulation Test Matrix	83
Table 6.2: Oneway ANOVA applied to the nearest distance	85
Table 6.3: Least squares mean given model and context given model and context	87
Table 6.4: Least squares mean given model and context given context and motivation..	92
Table 6.5: Live embodiment test matrix	95
Table 6.6: YX Fit on single factors applied to nearest distance	96
Table 6.7: YX Fit on single factors applied to nearest time	96
Table 6.8: Summary of simulation timing	102
Table A.1: Highway context mapping used for simulation embodiment	115
Table A.2: Neighborhood context mapping used in simulation embodiment	115
Table A.3: Practice context mapping used in simulation embodiment	115
Table A.4: Office context	116
Table A.5: Living room context.....	116
Table A.6: Relating motivation to mood	116
Table A.7: Action to motor effects	116
Table A.8: Simulation Test Matrix	117
Table A.9: Live embodiment test matrix	117
Table A.10: Simple model evaluation points [16]	118
Table A.11: Simple model mood values [16]	118

1 INTRODUCTION

Dreams of automata have been recorded in human history since the 4th century BC. Aristotle conceived “instruments that could listen to commands to perform work” [1]. Hydraulic, steam, and mechanically driven devices have been used for thousands of years. The first century AD documents over 100 automata in books by Hero of Alexandria [2]. In 1206, Ismail al-Jazari created a programmable, humanoid, automaton band with moving peacocks. In 1898, Nikolai Tesla demonstrated the first electrically automated submarine [3]. Inventors over time have created automata but the intelligence to drive them was missing.

The invention of the computer began to change fantasy into reality. Artificial intelligence (AI) started its development in the 1940s with a machine based on abstracted mathematical reasoning. In the late 1940s, Allen Newell, Cliff Shaw, and Herbert Simon created the Logic Theorist, which solved math theorems [4]. Academia continued to push innovation forward during the Golden Age of AI which occurred from 1956 to 1974. Newell, Shaw, and Simon were part of that push. For over a decade, MIT produced several noteworthy AI systems. These include feats of continuous speech recognition, natural language processing for algebra, a virtual counselor, computer vision, and even a tournament-capable chess computer [4].

Statistical analysis and training methods to create accurate predictions gave rise to machine learning [4]. These predictions have been used in programs that recognize speech or determine if the email is spam. Unsupervised, reinforcement, and supervised are common learning algorithms used to train machines and allow algorithm improvement [4]. Robotic applications often use reinforcement learning. Responses to

interactions cause positive or negative learnings to guide the intended behaviors [4].

Machine learning methods are helping advance artificial cognition by training computers how to think and act [5].

Neural networks were created as an early digitally inspired application of neurological processes [6]. Three generations have since been developed from this biologically inspired idea [6]. The first generation considered simplistic neuron models which only provided binary responses; however, such models were difficult to train due to their non-differentiable manner [6]. The second-generation, which includes much of the current state-of-the-art and state-of-the-practice in ANNs, innovated by incorporating differentiable cost functions, e.g., sigmoidal, and novel architectures, including the latest in deep learning [6]. The third generation, which is currently the theoretical state-of-the-art in ANNs, mimics neurological processes with spike train signaling [6]. Neural networks are the basis for deep learning, a subset of machine learning and part of the state-of-the-art in second generation neural networks [6]. The term deep comes from the number of layers in the neural network [4]. The advantage of deep learning is that feature extraction and relationship modeling can often be processed more accurately. Convolutional neural network (CNN), provides visual data exploitation through convolutions, pooling, and nonlinear functions, and are used in applications for computer vision [6].

Cognitive architectures started their development in the 1950s as part of general AI research [5]. Scientists from the fields of biology, computer science, logic, neuroscience, and psychology have been collaborating since then to advance the technology [7]. Cognitive computing research aims to simulate action selection, attention, learning,

knowledge, metacognition, memory, perception, problem-solving, and reasoning [5]. All of which are human and animal capabilities. Different paradigms and hypotheses have been developed around how to implement these concepts [8]. These principles have then been used to create interactive cognitive systems such as a practical algebra tutor and interactive rescue robots [8]. These systems show how applying cognitive concepts produces adaptive, interactive, iterative, and contextual behavior. Decisions can be made without the need for constant changes specific to the task or situation. These are reasons to research and develop models for intelligent automata.

Brains are divided into many sections and perform complex functions. Modern neuroscience and psychology teach us more about how each works, which helps to create better models. The basal ganglia have been shown to drive action selection [9] [10]. The hippocampus has been modeled for its role in memory [11]. Models of the cerebellum are based on its relationship to behavioral learning [12]. The amygdala is best known for mood and emotions [13].

The research in this thesis will model an amygdala to increase cognitive capabilities using SNNs. Existing models largely focus on fear and anxiety training and responses [14]. Recent neuroscience research shows context, motivation, action prediction, and nociception also guide autonomic responses and action selection [13]. The new research shows a gap in existing amygdala model functions either previously unknown or omitted. Nengo-based amygdala models and research performed by Stewart [15] [16] and Fischl [14] [17] are considered as a starting point. Adding these features is expected to influence mood responses due to changing contexts or internal stimuli. These variations will increase model complexity and require evaluation of robot response.

1.1 OPERATIONAL MOTIVATION

Creating cognitive functions for robotic applications will add versatility and adaptability. Existing amygdala models consider different aspects of the brain segment. Nengo models use discrete vectors. This design utilizes Nengo semantic pointer architecture (SPA) to provide an abstraction of neuron vectors into concept vocabularies or symbols. This abstraction also allows methods for symbol manipulation with binding, unbinding, and superposition. These methods are critical elements in this design.

Mood driven responses provided by the amygdala intend to change the way the system interacts with the environment. A cleaning robot typically maps a room, but unexpected items might appear. Identifying a cat laying in the living room might change the direction. Other obstructions might be identified as dirt to vacuum or a toy to avoid. A military drone can identify hostile or friendly territories to determine different flight patterns and actions. Rescue robots can interact uniquely with objects in different situations. Attempting to find a person in a fire might provoke urgency versus the careful sorting of wreckage. These are a few examples of how moods can influence robotic applications.

1.2 TECHNICAL MOTIVATION

Prior research developed a mood-responses-driven robot. [18]. This capability serves as the technical starting point for this effort. However, this prior work was limited by simplistic amygdala-like functions relying only on associative memory. Additionally, the software framework was not transferable to another computer, thus a recreation of the technical baseline was also performed on this effort to enable future use, extension, and reuse of this codebase.

Higher-level cognition can be developed in models by mimicking biology. This model elicits the amygdala features for contextual and motivational influences. These features are expected to provide dynamic interactions with the environment. In this work, new abilities will be added to this framework with demonstration on a TurtleBot in both modeling and simulation (M&S) and live embodiment testing. Context and motivation are expected to alter the way a robot will interact with its environment. This is due to the mood-driven action selection. Successful modeling and testing hope to bring the dream of intelligent automata closer to reality.

1.3 RESEARCH CONTRIBUTION

Table 1.1 provides a summary and mapping of the contributions of this research, “Current Research,” to previous related research, “Prior Work.” In Table 1.1, the × symbol indicates that a technical area was addressed.

Table 1.1: Relational mapping between technical contributions in previous related work and *current* research contributions. The × symbol denotes areas addressed.

Technical Area	Prior Work		Current Research
	Addressed	Ref #	Addressed
TurtleBot2	×	[18] [19]	×
Neuroscience Principles in Robotics			
Basal Ganglia	×	[18] [20]	×
Thalamus	×	[18]	×
Amygdala	×	[18] [16] [15]	×
Context			×
Motivation			×
Additional Contributions			
Ease of Future Development	×	[17] [14]	×
Model and Sim	×	[18]	×
Live Robot	×	[18] [16] [17]	×
Robot Performance Metrics			×

1.4 RESEARCH OBJECTIVES

The primary objective is to create an integrated model adding context and motivational features for robotic decision making and control. For this, an amygdala model was developed which considers moods from visual inputs with influence from context and motivations. Nengo will be used to create a SPA-based model, which allows the important elements to be processed conceptually. Concepts provides abstraction from low-level vectors into symbol vocabularies. Special operations on the symbols allow for binding for data combinations and unbinding for data extraction. Working with symbols and their operations simplifies concepts for motivation and context. The new model is expected to provide mood responses that change with environments and provocation.

The secondary object places the new amygdala into a robotic application. Higher cognitive abilities are expected to cause intelligent interactions with the environment. Motivations are expected to sway the behavior to act more positively or negatively.

Tertiary objectives include integration, documentation, and repeatability. The goal is a repeatable and reusable codebase to address concerns from prior demonstration of [18]. Thus, this work had to recreate much of the technical baseline of [18]. With this expanded integration of framework and integration of documentation, future code modules and research can be performed using this work as a technical baseline.

Comparisons will be run against a baseline model that was used in previous research [18]. Test metrics are defined in section 4.1 for proof-of-concept models and 6.1 for simulation and live testing. The new amygdala model is expected to perform equal to or better than the baseline.

2 BACKGROUND

Creating cognitive functions is an area of research which aims to develop better artificially intelligent (AI) systems by leveraging constructs and principles from biology. While this has seen significant utility in the use and deployment of artificial neural networks (ANNs), further computational brain modeling is performed in hopes of achieving higher levels of cognition, and hence utility, in artificial agents. Increasing levels of cognition is viewed as important for creating the ability for artificial agents to both understand and provide capabilities [21].

2.1 COGNITION AND THE BRAIN

Cognition is considered a process of acquiring knowledge and understanding which allows a system to effectively adapt and improve [21]. Cognition includes brain processes such as perception, attention, focus, learning, decision making, problem-solving, and memory. Intelligence and cognition provide the framework to gain new knowledge from existing knowledge [22]. Many of the process of cognition are theorized to occur due to the complex interaction of brain segments.

The human brain is a significantly interconnected system. In order to understand the application of a biological brain in a robotic application, one must understand how different segments contribute to cognitive functions. The cerebral cortex is relied on for motor control, sensory control, and executive function. The hippocampus is involved in spatial memory and learning. The hypothalamus oversees maintaining homeostasis. The thalamus is a relay center for neural signals to be transmitted from the cortex and brain stem to other parts of the brain for interpretation. The basal ganglia are involved with

reward-based learning, memory, and emotion, and are considered a critical component in decision making [9].

The amygdala is often described as the control center for emotions and fight or flight reactions. New insights into the function and connectivity of the amygdala have been revealed in recent research. While defensive responses are derived from this section of the brain, the amygdala performs more than fear responses [13]. Sensory inputs, context, experience, motivation, and threat intensity all connect to the amygdala. Mood responses are generated in the central nucleus (CEA) and sent to autonomic and action selection segments [13]. Additionally, associative memory is shown to have a strong connection with the amygdala. Figure 2.1 provides a graphical representation of connectivity with the CEA. This figure shows the types of inputs and outputs and where they come from in the brain. Directional arrows indicate signal directionality [13]. Sensory information is provided by the sensory cortex (CTXs) and insular cortex (CTXin). The medial prefrontal cortex (mPFC) provides decision prediction inputs. Memory and context information are provided by the hippocampus (HIP) [13]. The bed nucleus of the stria terminalis (BNST) injects anxiety information while the paraventricular nucleus (PVT) and intercalated cells (ITC) help regulate fear. Nociceptive responses are provided by the parabrachial nucleus (PBN) [13]. The substantia nigra pars compacta/ventral tegmental area (SNc/VTA) and raphe nucleus (RN) provide motivational influences. The central nucleus factors all these inputs with valence, or reaction capacity, from the basolateral amygdala (BLA) to generate output responses [13]. Autonomic regulation responses are provided to the locus coeruleus (LC), nucleus tractus solitaries (NTS), and dorsal motor nucleus of the nervus vagus (DMV). Action

selection responses are directed to the basal forebrain (BF), hypothalamus (HYP), and periaqueductal grey (PAG) [13].

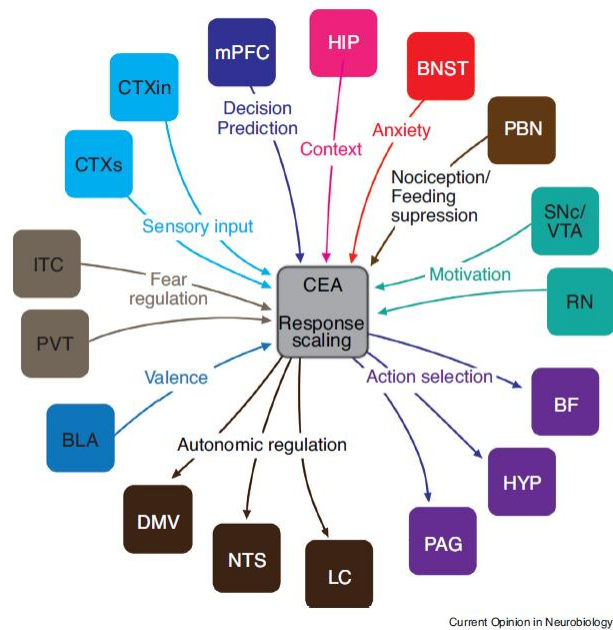


Figure 2.1: Functional connections of the central nucleus (CEA) of the amygdala [13]

The brain's parts are interdependent and require one another to function and create higher-order thinking as a response to complex problems. For instance, a fear response will be mitigated by the amygdala but will involve the cerebral cortex, hypothalamus, and thalamus [13]. While many of the activities in the brain are understood, some of the connections and interactions are beyond present research and scientific capabilities.

2.2 MIMICKING THE BRAIN

Artificial intelligence is used to create systems that have human-like thinking, rational thinking, human-like actions, or rational actions [5]. Training allows similar task repetition without new code creation. Learning can also be used to develop new capabilities for a system or robot. Biology continues to inspire researchers to dive deeper into creating intelligent systems. The brain provides a previously organized system that can be used as a model for effective and efficient design.

Modern AI is frequently designed using Artificial Neural Networks (ANNs). These networks are modeled after neural connections in biology [6]. Spiking Neural Networks (SNNs) are the third generation of neural networks. These networks mimic the spiking nature of neuron action potentials, or spikes [23]. This biological analog allows for better analysis of brain processes, thus better cognitive applications [23]. Nengo AI is a popular neural engineering framework (NEF) for developing SNN applications [24]. A software library for SPA modules is provided in Nengo. This library will allow for the abstracted elements for concept vocabularies. Higher-level cognition is produced by working with concepts.

Spiking neural networks have seen successful use in robotic applications. Control over a robotic hand [25], a six-legged walker [26], and a small insect scale robot [27] have all been achieved using SNNs. Robotic navigation has also been developed this way [28]. The applications are enhanced by neuromorphic hardware. This technology provides a physical application of SNNs, which provides speed and power efficiencies [23].

Increased cognitive capabilities allow robotic applications to obtain a higher order thinking. Developing a brain-like feature within robotics accomplishes this goal. These digital brains allow robots to perform more dynamic tasks like action selection, prioritization, or adaptable responses.

Implanting cognitive capabilities into robotic applications will allow greater autonomy. Keys to actionable artificial cognition are sensing, perception, reasoning, and actuation [21]. Sensing gives the robot the ability to observe the world while reasoning determines the appropriate course of action. Actuation allows for the agent to interact with the world through movement or manipulation of the objects within its environment.

Many AIs have been created using computational neuroscience to replicate functions of cognition. TacAir-Soar is an impressive system that replicates the behavior of combat pilots [8]. Façade provides a simulated apartment in which multiple intelligence agents allowed user interactions with the environment. Sentences, body language, and gestures could be understood and exhibited by the agents. [8].

A SPA Unified Network (SPAUN) is capable of a variety of cognitive tasks [24]. Reprogramming and manual interaction are not required for task switching. One task takes a visual input to determine a pattern and use that knowledge to complete it [24]. This system shows a high level of cognition by incorporating several functions of the brain.

2.2.1 LIMITATIONS OF PAST INSTANTIATIONS

Modeled functions may have specified or limited features. The basal ganglia are more often used for action selection rather than other capabilities. Decision-making-based models have been hypothesized and verified using models of a monkey's basal ganglia

[9] and Gaussian process regression [29]. Some thalamus circuit models are used for enhanced learning performance. Mimicking a dopamine response showed a 36% decrease for a system trying to catch a target [30]. The cortex-basal ganglia-thalamus loop is well-known. The cortex provides, stores, and manipulates representations. The basal ganglia capture current brain states to courses of action. The thalamus applies routing signals to cortical pathways [24].

The fields of biology, computer science, logic, neuroscience, and psychology are being merged to model and create behaviors and responses in robots that closely mimic life. Cognitive model applications are emerging with basal-ganglia and thalamus sections utilized for decision making and creating actions. Connecting the emotions from the amygdala to the basal ganglia allows for behavior-driven controls. These elements can drive a person toward or away from a situation as well as create a sense of survival [18]. Nengo provides modules for the basal-ganglia and thalamus but not an amygdala. Adding this functionality will provide dynamic responses that contribute to higher-order thinking.

2.2.2 SPIKING NEURAL NETWORKS

Neural networks and the concept of interconnected branches and nodes get their inspiration from biology. The first generation only allowed for Boolean functions. The second-generation neural networks introduced sigmoid and other activations. These networks have allowed reasonable approximations of analog signals and are commonly used in other learning algorithms. Generation three introduced SNNs, which continued the biomimicry trend by modeling signals after neuron action potentials. These networks have helped develop better tools for analyzing brain functions [23].

Using SNNs the human brain can process an image, classify it, and make decisions within 100 ms. This has not been true for other CNNs. A human brain

consumes about 20 watts, while supercomputers running brain simulations devour megawatts. Robots have limited power sources and CNNs can push beyond those limits. Lower power consumption can pave a new path for advances in robotic systems [23]

Scalability is one of the largest capabilities for SNNs. Information can be processed in a relatively small number of neurons, but it also supports very large networks [23]. This makes abstraction and detail easier. A higher-level function works with a small number of neurons by sending commands to another set of neurons performing lower-level tasks. The same can be applied in reverse. This creates great versatility for SNNs [24].

Spatial-temporal information is also captured by the pulse-encoded nature of the SNN. These spikes can convey the timing of events with high precision and accuracy. The SNN offers advantages and greater biological plausibility for robotic applications [23]. The interest in this type of network has increased, but further research is still needed.

2.2.3 NEURAL ENGINEERING FRAMEWORK

The Neural Engineering Framework (NEF) used mathematical theory to represent the dynamics of neural systems [24]. There are three key principles to constructing an NEF. Representation is the first element. Spike trains need a nonlinear encoding of vector spaces to allow linear decoding [24]. The abstracted conversion allows for easier work with SNNs. Second, is the ability to compute and transform arbitrary vector functions based on the encoding [24]. Dynamics are the last element. Neurons are non-linear dynamical systems that have theoretical control state variables. The ability to represent dynamic and differential equations is required to have a fully functioning framework [24]. Nengo is a current NEF used to develop cognitive models and accomplishes all

three principles. The tool allows for the creation of SNN neuron ensembles, interconnection, and manipulation. Nengo was chosen because it offers better flexibility, scalability, and robustness while allowing a continuation of prior research [24].

2.2.4 SPA NETWORKS

Semantic pointers use high-dimensional vectors to represent concepts. These vectors provide neural representations vital to complex cognition. Structure representations can be created through linear and non-linear operations on concept vectors [24]. There are three operations: superposition, binding, and unbinding.

Superposition allows a union-like operation on two vectors. It provides an additive property. Elements of the original vectors are found in the resulting vector [24].

Binding produces a dissimilar vector but allows information recovery. A circular convolution operation is performed to achieve this. An example is shown for a simple symbol binding example of two concepts, $Red \otimes Square + Blue \otimes Circle = \vec{v}$, where Red is bound with Square summed with Blue bound to Circle creating a new vector, $\vec{v}$. [24]. Unbinding performs the opposite function of binding. Two concepts bound together can become unbound. An approximate inverse is used in the circular convolution process to make this happen, but often produces noise [24]. An example of unbinding is demonstrated here, $\vec{v} \otimes \sim Square = Red \otimes Square \otimes \sim Square + Blue \otimes Circle \otimes \sim Square = Red + noise$, showing how to acquire color information from $\vec{v}$ data by binding with $\sim Square$ (approximate inverse). A cancellation occurs with

Red $\otimes$ *Square* $\otimes$ $\sim$ *Square*, leaving *Red*, and *Blue* $\otimes$ *Circle* $\otimes$ $\sim$ *Square*, becomes noise.

This research relies on these three operations to create the desired functionality. Developing context and motivation become simplified as a result. These equations also demonstrate the practicality of working with concepts.

2.2.5 NEUROMORPHIC HARDWARE

Memristor-based technologies offer a method that natively represents and communicates in spikes. The physical implementation reduces computational overhead and power consumption [23]. This technology is called neuromorphic hardware. Traditional von Neumann and modified Harvard processors can run neural networks and SNNs but lose efficiency due to constant conversions [23].

2.3 ROBOTIC INTELLIGENCE

Robots are often designed to solve specific problems. This allows a given task to be performed well, but it is poor or incapable of performing anything outside of those parameters. Stepping away from the algorithmic approach provides a framework for systems to make decisions, learn, and develop on their own [21]. Systems capable of self-development should meet the following design principles: the ability to mimic brain functions, the ability to organize sensory information, and it should have a physical presence within an environment for exploration [21]. Given a minimal skill set, it should explore, survive, engage in behavioral tasks, and then learn from its experiences. The system should have a means for adaptation and a sense of connection with its environment [21]. These systems use AI and machine learning techniques to adapt to their surroundings and learn new behaviors.

The ability to recognize oneself in a mirror has often been assumed a mark of intelligence. Humans quickly learn their mirrored identity. The test has been performed on numerous animals including dogs and fish with varying levels of success. A robot learned to identify itself in a mirror without other visual cues, such as a marking dot [31]. This has been done by relating the viewed actions with the movements performed by the machine. The ability of a robot to self-recognize can enhance navigation and environmental interactions [31].

Using perfect single-color stimuli, researchers have taught a robot to understand concepts like horizontal, vertical, left, and right. Learning concepts are considered higher-order cognitive functions. Higher-order functions have previously been reserved for cognition and have been a focus of replication in robotics [32]. In biology and robotics, there are questions about how neural connections are made or modified. The relationship between first order and second-order concepts is one such challenge. For example, by understanding the concept of horizontal, can a system identify above and below? [32]. This kind of thinking will help robots to learn at conceptual levels.

2.3.1 ROBOTIC CONTROL APPLICATIONS

SNNs and neuromorphic hardware are research topics for robotics controls applications [23]. The advantage of spikes and learning allows for a wide range of robotic applications. Uses include low-level controls to high-level cognition with a variety of inputs and outputs. These applications and utilization of neuromorphic hardware have great potential.

One application of SNNs developed motor primitives for a soft-grasping robotic arm. Online learning was used to create the adaptive controls. The research allowed development without the need for traditional inverse kinematics and complex point planning, which avoids engineering complexities [25] .

A six-legged walker intended for space exploration used the SNNs to control motor primitives, local leg behaviors, coordination, and high-level controls [26]. This modeling approach represented varied motions and highlighted the ability of a neurorobotic platform by demonstrating that motor primitives are flexible enough to mimic more than one type of movement [26]. Another example involves a small insect scale robot capable of flight. Neuromorphic hardware is used to receive sensor inputs, control hover, perching, and trajectory-following [27].

A social robot, developed by Fischl, utilizes neuromorphic hardware [17]. The robot uses a vision system to detect the social partner's facial expression. A primate amygdala model detects if the person is smiling, distressed, frowning, or neutral. This information is used to provide a joy score [14] [17]. This research leverages these concepts and builds upon this foundation as described in the subsequent chapters.

3 DEVELOPING A MOOD MODEL FOR ROBOTICS

The interest herein involves developing an amygdala-driven robot. This involves creating autonomous abilities for computational mood responses as a result of detected object inputs, contextual applications, and motivational influences.

3.1 BACKGROUND REFERENCE MODEL

Amygdala models frequently focus on fear responses and conditioning [17]. Fischl developed a primate amygdala model that focused on multiple emotions [14] [17]. This model used a visual input to evaluate a subject's emotional countenance and assign a score. Neuron ensembles were created in Nengo to represent neural subcomponents. Additionally, work by Stewart [16] created a simple amygdala model that receives numeric values representing the facial features of a subject to determine a response. However, the work of [16] used limited mood responses based upon inputs to neuron ensembles. Despite these limitations, the simple amygdala models from [16] served as a basis for the start of this research.

3.1.1 CONCEPTS AND EXTENSIONS

The central nucleus is the key processing unit of the amygdala. Here, numerous inputs create responses for autonomic regulation and drivers for action selection [13]. Sensory inputs come from the sensory cortex. The hippocampus provides a relationship for context-based on memory. The substantia nigra pars compacta inject motivational stimuli [13]. These are the inputs of interest for this research.

Creating and altering contexts allows an amygdala to respond differently to the environment. Meeting a lion at the zoo versus on the Savanah likely evokes different reactions. Likewise, interacting with a grease fire demands a different response than the

open flame of a gas stovetop. The examples show how contextual influences are important and can be used to create higher-order thinking in robots.

Motivations of various kinds can amplify an intended response or influence an agent to behave differently. Simple motivations like encouragement or focus can drive one toward a goal and discouragement can drive one away from it. This concept was added to the model and provided a method to increase or decrease the three primary behaviors of fear, hunger, and curiosity.

The inputs provided for object types, context, and motivation lend themselves to a conceptual nature. The Nengo SPA provides an easy way to allow this abstraction from vector values. This conversion of discrete inputs into symbols allowed the transmission and transformation of concepts, which are human-readable, easier to track in the system and simplify scale with added vocabulary.

The other models use low vector count representations for moods and expressions. The primate amygdala used a five-dimensional vector to represent emotional input and gaze [14]. Using SPAs for this research will create a unique model. Neuron count will increase as a trade-off for abstraction. However, working with concepts allows for the binding and unbinding of context with objects. This will be important to the functionality of this model.

3.1.2 IMPLEMENTATION CONSIDERATIONS

Model and simulation work from previous research, see [18] [19], served as a starting framework. This prior work enabled the leveraging of already developed simple robotic controls and sensor inputs such as image processing. These inputs are then processed by the reasoner portions which implement simple amygdala-like functions for mood, intention, and basal ganglia decision making [18]. Moods of fear, hunger, and curiosity

were connected to the actions of flee, feed, and explore to create a basis for mood-driven decision-making [18]. These elements were kept for simplicity. The system used Nengo for cognitive modeling, ROS communication, and Gazebo for the simulation environment. Real-world applications were implemented on a TurtleBot 2.0 system [18].

3.2 SIMPLE AMYGDALA MODEL, MODEL A

Prior to implementation on robotics or considering sensor data, the behavior of Stewart's model [16] was analyzed. This model contains a lateral segment, simple basal ganglia, an external cerebral cortex, and a central nucleus to represent the amygdala. The block diagram of this model, Model A, from Nengo, is shown in Figure 3.1. The input values for a discrete number of signals, such as eyes, mouth, teeth, and known target are received by the lateral neuron ensemble input [16] [15]. Data is transformed along lateral to the central pathway by evaluating the inputs to a list of known point values and resolving it with a list of output moods. Along the lateral to basal connection, a thresholding function is applied. For instance, if input values for teeth or mouth are below a level then anger is given a strong assertion [16]. These two paths then recombine at the central nucleus to create a summed value.

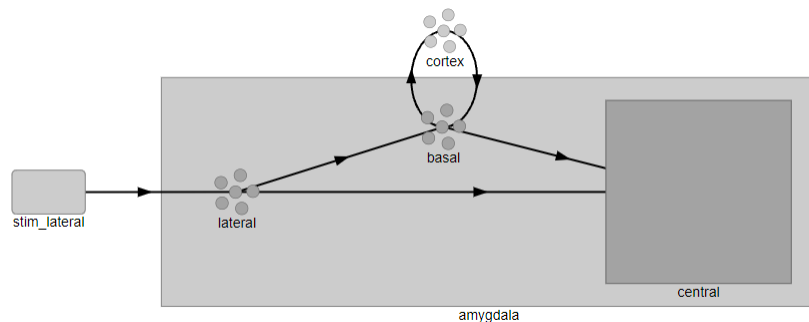


Figure 3.1: Block diagram of the simple amygdala, Model A, displayed in Nengo

This model will serve as a reference. Replication of the lateral and central nucleus is considered the necessary minimum. Conversion to SPA modules is not one-to-one. Transformations performed along signal paths will require different approaches. Path combinations will also require the same vocabulary set. The concepts used to identify objects cannot be interchanged with moods. The conversion is expected to provide similar functionality but not expected to have identical results.

3.3 AMYGDALA MODEL B

The amygdala Model B concept was provided by the previous research on autonomous learning and considered as a baseline [18]. The model is displayed in Figure 3.2. The associative memory block could be considered the simplest stand-alone amygdala. Here, the inputs on the left connect to the associative memory input. This block compares a given input to a Python dictionary to provide the mood output. The function is similar to the lateral basal pathway in the simple model. The output then goes to a SPA state module, “Mood”, which provides a light memory function. The model possesses bio-plausibility [13] and methods for extending motivation.

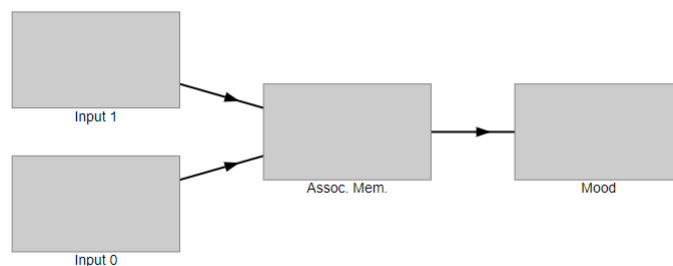


Figure 3.2: Block diagram of Model B displayed in Nengo

3.4 AMYGDALA MODEL 0

The Amygdala Model 0 was developed to convert Model A [15] into the SPA symbolic form. Working with abstracted symbols is the primary objective and will simplify manipulation of concepts. Additionally, instead of input elements derived from images for eyes, mouth, teeth, and a known target, this implementation received vocabulary symbols at the lateral input as detected objects.

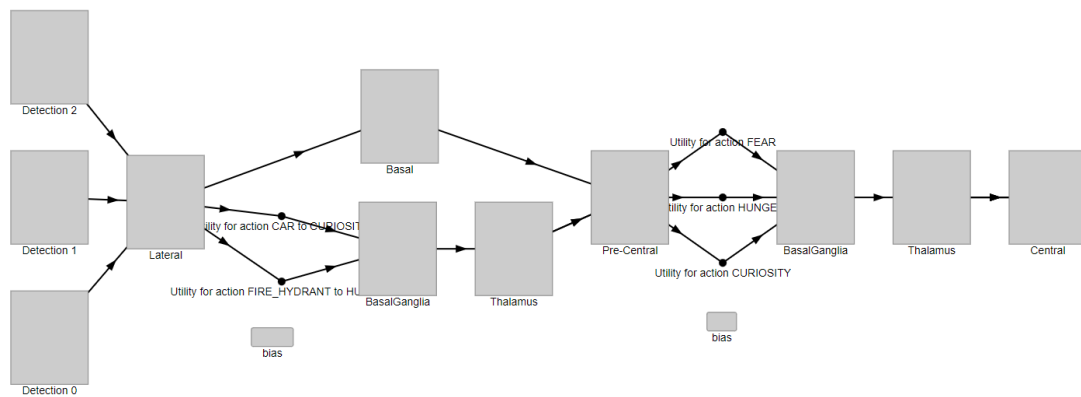


Figure 3.3: Block diagram of Model 0 displayed in Nengo

Figure 3.3 shows the connections between blocks for Model 0. The inputs in Figure 3.3 start on the left side of the diagram and are connected to the lateral inputs. Two paths diverge from the lateral output. One branch to an associative memory block named “Basal”, which converts the object to a mood. The other path goes to a winner-take-all (WTA) basal-thalamus for object selection. The basal-thalamus emphasizes the mood response of specified objects if present. This attempts to replicate the strong influences of certain facial features found in the lateral to central path of the simple model. These branches from the basal and basal-thalamus recombine into the pre-central state before evaluating for the maximum signal. The central portion now requires a WTA approach to

determine the outcome. An attempt was made to use a WTA module, but this altered the output into 32 unique states. Limitations in understanding prevent reasonable adaptation for this. Instead, a second basal-thalamus action selection module was used to provide the maximum output to the CEA.

3.5 AMYGDALA MODEL 1

Model 1 was intended to reduce the complexity of Model 0 due to high neuron count and the number of required components. The key difference was the conversion from signal input to mood output. This was performed in the central stage of the model instead of along the pathways and removed the secondary action selection node used in Model 0. Additionally, the action selection path was replaced with SPA basal ganglia and thalamus blocks, which are expected to perform in a WTA fashion.

Figure 3.4 shows two inputs connecting to the lateral input. The lateral output has one connection to the central and another to a basal-thalamus section. This section includes a SPA state to mimic the cortex memory used in the simple model. Both paths converge back to the central nucleus where the output mood is produced.

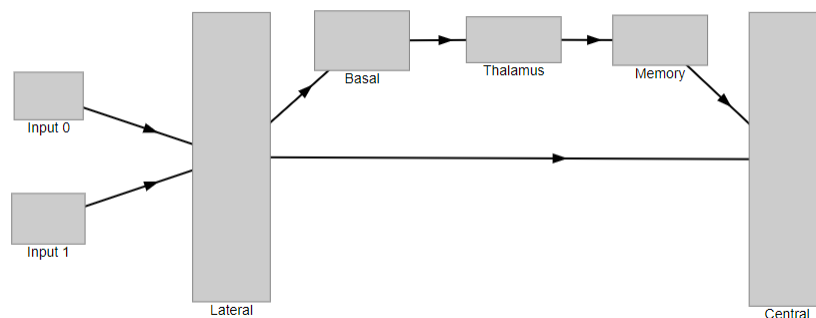


Figure 3.4: Block diagram of Model 1 displayed Nengo

3.6 AMYGDALA MODEL 2

Model 2 was developed to resolve issues in Model 1's WTA pathway discovered during the evaluation. The original path was intended to simplify programming and allow better scalability; information about the related challenges can be found later in the thesis, in Section 4.3. Action selection components were selected to prove the desired WTA functions. This also drove a method for selecting objects of interest.

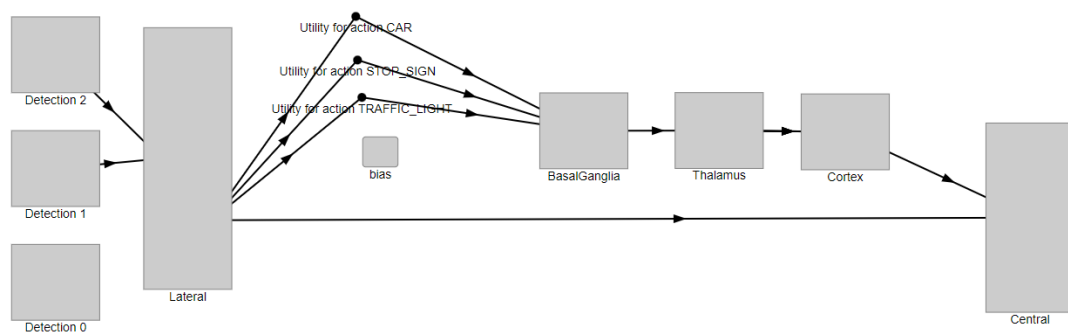


Figure 3.5: Block diagram of Model 2 displayed in Nengo

The connectivity of Model 2 is shown in Figure 3.5. The lateral input (far-left) receives inputs and passes them to both the basal-thalamus circuit and the CEA. The basal-thalamus path identifies objects of interest and emphasizes the signal to the central nucleus. This allows the robot to view an environment and deem some objects of a higher value or importance than another. Responding to a person on the highway may be more important than the traffic sign. The cortex state allows for retention of some values even if the inputs are changed or cleared. The CEA is an associative memory block and will create the mapped mood outputs.

Model 2 became the basis for adding features. While Model B offered simplicity, Model 2 provided additional pathways to influence the response which mimics the simple model. In a modularized form, these features were easily adjusted or omitted.

3.7 ADDING MOTIVATION, AMYGDALA MODEL 2M

Motivational input is a described feature of the central nucleus as well as associative learning [13]. However, prior work [18] did not incorporate this. As discussed in section 2.1 such motivational input influences amygdala responses. Thus, to model this a motivational vocabulary was created. Mood transformations from the motivations were considered with a simple associative memory module added to the output of Model 2 to provide a singular mood output. Figure 3.6 shows the addition of motivation on the far-right side along with the summation with the central output; outlined in red. Thus, the output of the motivation module is combined with the amygdala outputs using the superposition properties in Nengo. The placement of motivation at the end of the path was chosen to reflect an influence on the overall mood and not just a specific input or aspect of it.

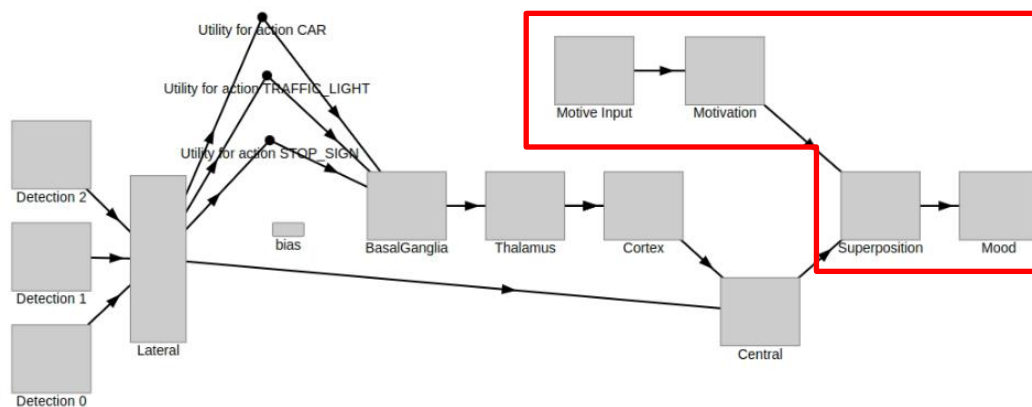


Figure 3.6: Model 2 with the addition of the motivation circuit, outlined in red

Discourage, encourage, focus, and normal symbols were created to represent motivations. These motivators and their effects were decided based on general logic on how individuals might interact with their environment because of the motivator's definition. The represented effects of the motivators are found in Table 3.1. Moods are developed relative to the motivation that they include. For example, the "Normal" motivation provides a simple curiosity so the robot will move and look around if there is no actionable stimulus. Normal has a scaling factor of 0.15 on curiosity to impel movement if no objects drive a mood. A scaling factor of 1.0 was not used to avoid overriding mood responses from object inputs. Further motivations are based on combinations of moods. For example, while the mood hunger drives the robot forward (towards a target), the "Focus" motivations include hunger as a primary effect and curiosity and fear as negations to balance the intent of focusing on a specific object instead of the looking and searching of curiosity and turning away of fear. Discouragement brings about fear and a reduction of hunger and curiosity causing avoidance reflected by an action to flee. Encouragement is used to add courage and reduce the impact of fear.

Table 3.1: Relating motivation to mood

Motivation	Intent	Mood Output
Discourage	Increase fear	$0.2 * \text{FEAR} - 0.1 * \text{HUNGER} - 0.1 * \text{CURIOSITY}$
Encourage	Decrease fear	$0.25 * \text{CURIOSITY} - 0.3 * \text{FEAR}$
Focus	Increase drive	$0.5 * \text{HUNGER} - 0.25 * \text{CURIOSITY} - 0.25 * \text{FEAR}$
Normal	General movement	$0.15 * \text{CURIOSITY}$

3.8 ADDING CONTEXT, AMYGDALA MODEL 2MC

Context is a framework around everything that can be observed. For this application, context is used to describe a given environment, such as a highway or living room. The approach for creating this framework in Nengo relied upon contexts represented as concepts through SPA. Two different models were developed to demonstrate the impact of context. The first mimicked the method used for motivation. This is a very simple, straightforward approach and the results were effective but less realistic. Applying context in a blanketed method overrules unique responses from individual objects.

Principles for binding and unbinding in Nengo are critical for using context. A context is applied, or bound, to the inputs outside of the amygdala model, and is assumed to be unknown. New pathways were created for context unbinding. These additions also created a change in the model. Context unbinding paths obtained localized associative memory units to provide unique mappings. The CEA was changed to be a SPA state as a mood collection point.

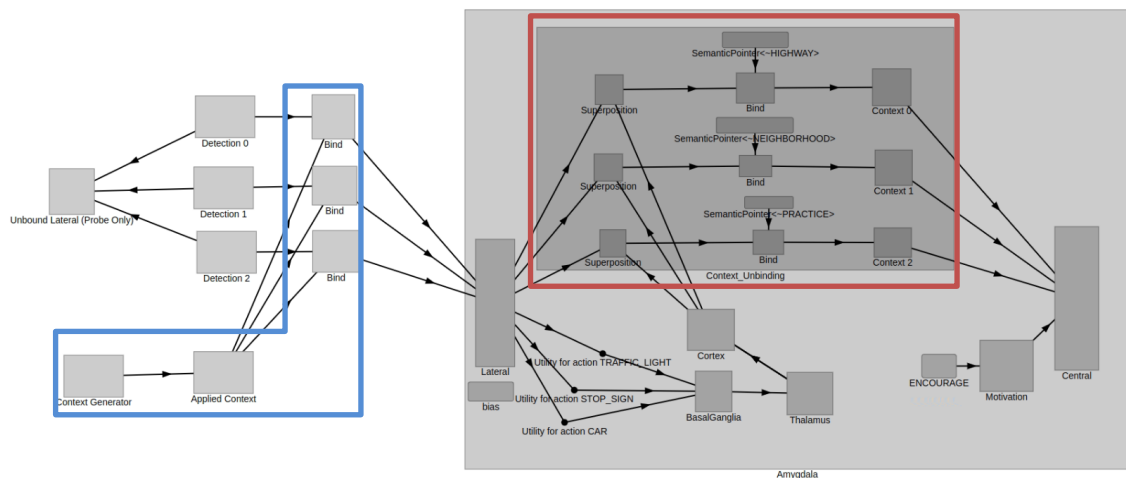


Figure 3.7: Block diagram for Model 2MC with elements for added context. Binding circuits are outlined in blue and unbinding circuits outlined in red.

Figure 3.7 shows binding inputs in a blue box on the left and an encapsulated amygdala model on the right. Inside, a red box labeled “Context_Unbinding” contains the pathways used to unbind the object from the context. The approximate inverse is specified in each pathway providing the unbound information to the respective associative memory block.

Several context symbols were created. A roadway scenario and a building environment were imagined and created as separate sets. The roadways were “Highway”, “Neighborhood”, and “Practice” environments. Buildings were given room names for “Office”, “Kitchen”, and “Living Room.” Each context has a unique mapping for objects to provide different interactions and effects.

3.9 COMPLEXITY OF MODELS

Each element in the model contributes to the overall neuron count. Table 3.2 compares the total number of neurons from one model to another. These counts include input generators. The change of scale was considered more important than an exact count. Model B is the least complex. Model 2 is approximately three times more complex. Adding multiplexer circuits for context dramatically increased neuron count.

Table 3.2: Neuron count per model		
Model	Number of Neurons	Comments
A	1100	Not SPA-based, cannot integrate into robot model [18]
B	1800	Baseline model [18]
0	7750	First attempt
1	22600	Failed reduction
2	6700	Appropriate reduction
2M	8450	Motivation added to Model 2
Context (end)	8600	Concept not used
2MC	64850	Amygdala with context and motivation

4 PROOF OF CONCEPT DEMONSTRATION

The models were developed for proof-of-concept demonstration. A set of metrics were created to perform functional validation. Different phases of development and testing required different metrics. The best-performing model was then selected for expansion and robot integration.

4.1 METRICS

Simulation data was captured in Nengo by using probes. These probes were placed along the signal paths. When observing the final model, one of the probes is placed on a dummy state which collects all inputs without context binding. Once objects have a binding they can no longer be probed in the same manner. Simulation run times were captured during each run and exported to a log file. The log also includes configuration information on the context and motivation. During the proof of concept, the metrics for architecture size, simulation time, and behavioral accuracy were most important.

Network architecture size is based upon the number of neurons and helps to understand complexity. This number is a retrievable property of the Nengo model. Simulation time was assumed to be proportional to the number of neurons. The timing was captured using Python's Time library to determine a run time measurement. This measurement evaluated the complexity and efficiency of the model. Run times were averaged throughout 10 operations. This method helps reduce differences caused by processor loading and other system variables.

Behavioral accuracy was measured by comparing detected inputs to expected outputs. If an object created a "FEAR" response but the output showed a "HUNGER" response, it was considered inaccurate. Correct outputs with lower-than-expected values

are flagged but not considered inaccurate. The accuracy is calculated as a percentage of correct versus the total. Proof of concept models were tested twice. Once with a single detected object and another with two.

4.2 GUIDE TO READING PLOTS

Many of the plots contain large amounts of data. Plots are presented in a way to convey the flow from input to output moving from top to bottom. This section intends to bring clarity on how to read and understand the data.

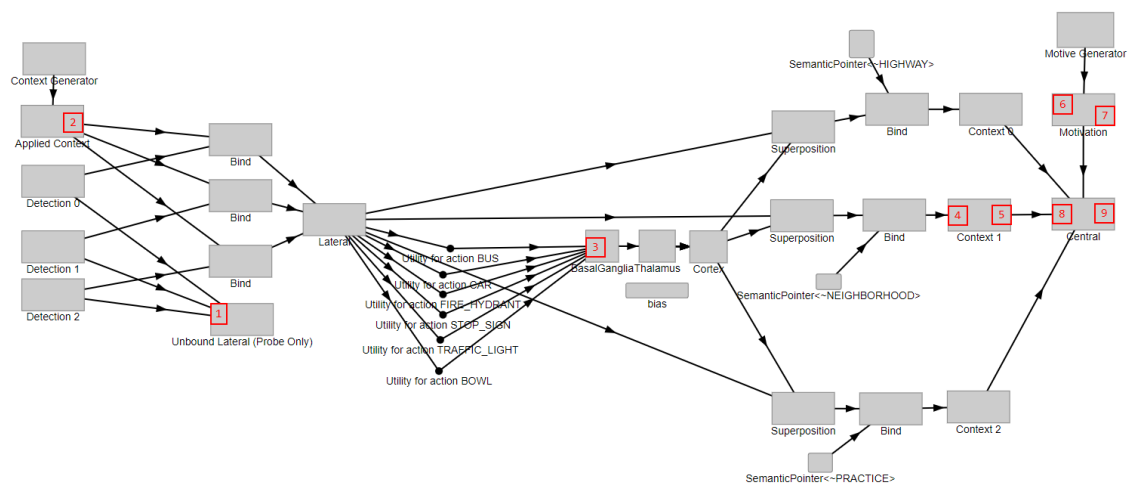


Figure 4.1: Model 2MC used to explain probe information, red indicates the location of a probe with its identification number

Figure 4.1 shows a block diagram of Model 2 with context and motivation. Knowing the progression of signals through the model is important to anticipate and read results while tracking from input to output. Probe locations are numbered in red for reference.

Location 1 (left) is the unbound collection of all detected inputs and measures objects.

Location 2 (left) is the applied context. Location 3 (center) is the input to the basal-

thalamus selection circuit and measures objects. Locations 4 and 5 (right) are the

respective input and output of the context 1 associative memory. The input measures

unbound objects filtered by context 1. The output is the associated mood responses. Location 6 (right) is the input motivation. Location 7 (right) measures the mood associated with the active motivation. Locations 8 and 9 (right) are the respective inputs and outputs of the CEA. These locations are both expected to measure the same mood values with possible exceptions from feedback.

To read plots, like Figure 4.2 and Figure 4.3, the signals of each plot were color-coded according to the defaults provided by the Matplotlib for Python. The number of signals was limited to prevent color overlap and confusion. The signals are represented in time and signal level. Detection inputs combine differently based upon the generated signals and their timing. Figure 4.2, which was created using data from internal cycling generators as inputs to Model 2MC, provides an example of a single input signal where the change is visible every half-second.

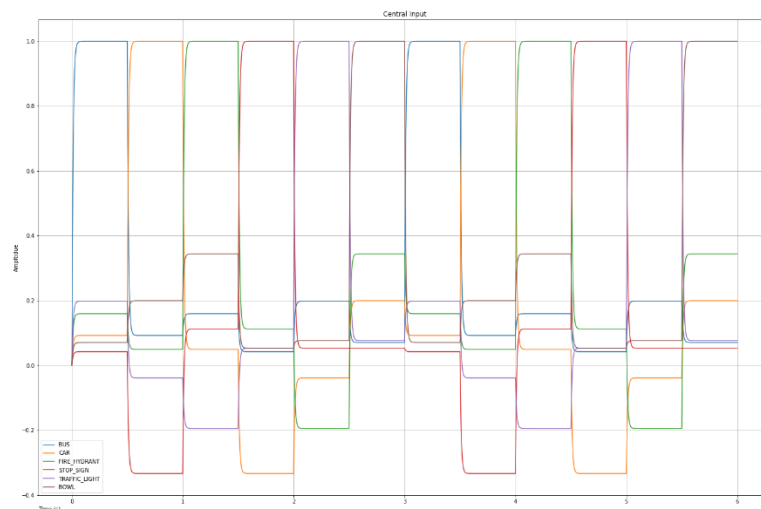


Figure 4.2: Example of single detection input probe location 1.

Figure 4.3 combines two signals with slightly different timing. The signal levels often match, showing both signals as active, but making it more difficult to determine

which is more dominant. Signals of the same object type sum together to produce one larger signal, as found around 1.75s with the “STOP_SIGN”.

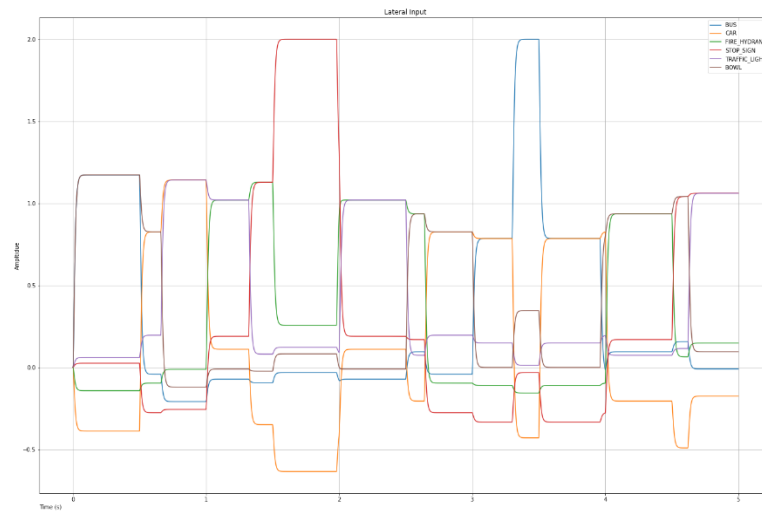


Figure 4.3: Example of two detection inputs at probe location 1.

Context and motivation input signals are easier to track because each one has a single value and cycles over time, so there are no overlapping signals. An example of context is shown in Figure 4.4 and motivation in Figure 4.5.

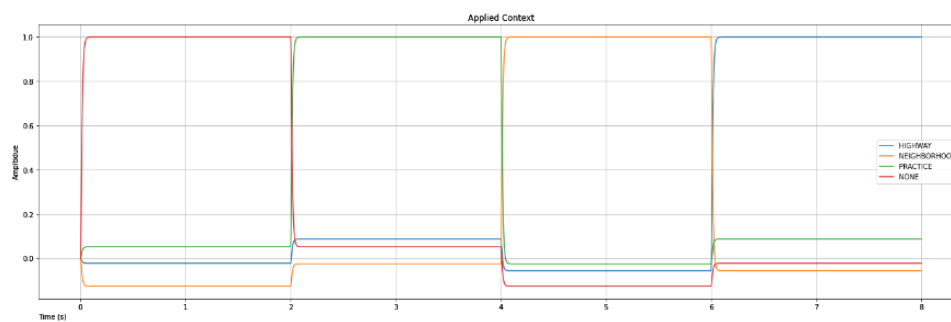


Figure 4.4: Example of applied context input at probe location 2.

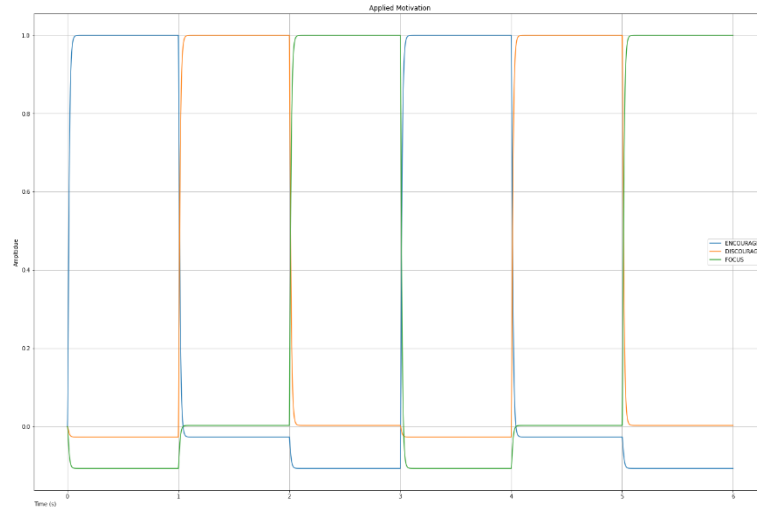


Figure 4.5: Example of applied motivation input at probe location 6.

Part of the model flow is the association of a signal to a corresponding output. The signals are typically synchronous throughout the amygdala portion of the models allowing simple response correlations. Inputs to the robot are denoted by capitalization and quoting to differentiate from discussion around the function. Figure 4.6 uses arrows to show how “FOCUS” corresponds with increased “HUNGER” and suppression of “FEAR” and “CURIOSITY”. Arrows are used in this guide to help show the relationship between signals but are not used in the rest of the document.

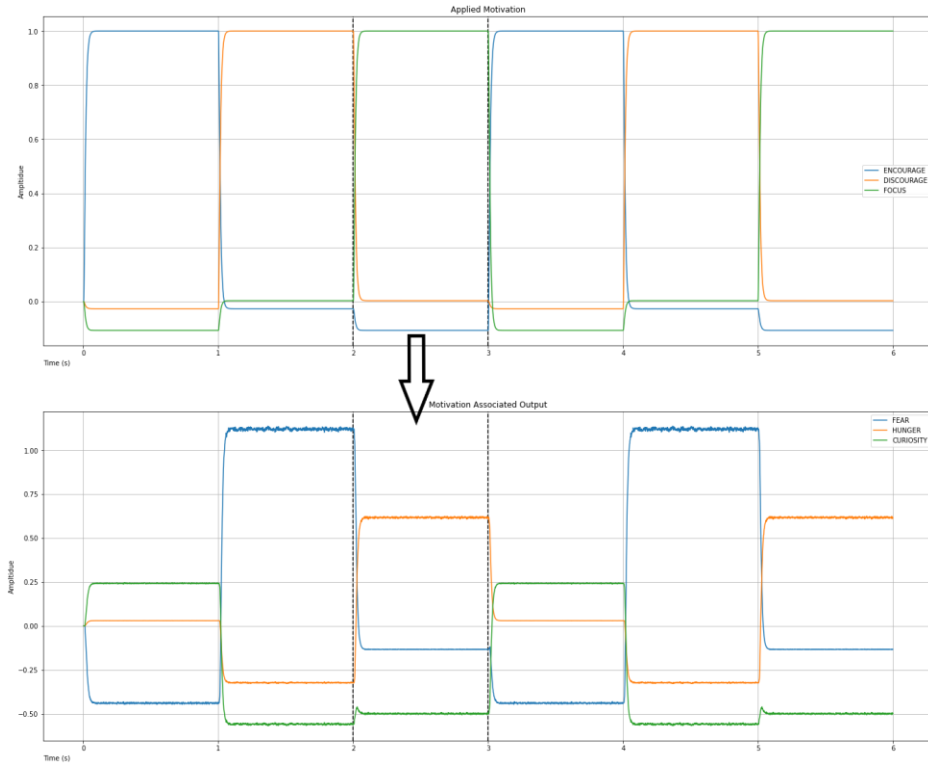


Figure 4.6: Example of motive input and associated mood output probe locations 6 (top) and 7 (bottom).

Figure 4.7 shows signals being traced with a given applied context. Probes are placed at locations 2, 4, 5, and 9 and plotted in that order from top to bottom. Vertical bands are added to help highlight a specific region for observation. This becomes very helpful while following the numerous stages of input with context binding, unbinding, and the eventual outcome. Reading the plots from top to bottom, “NEIGHBORHOOD” is the applied context in the banded region of interest. The second row shows the detected objects related to this context input. The associated mood responses are in row three. The final row shows its contribution to the overall output.

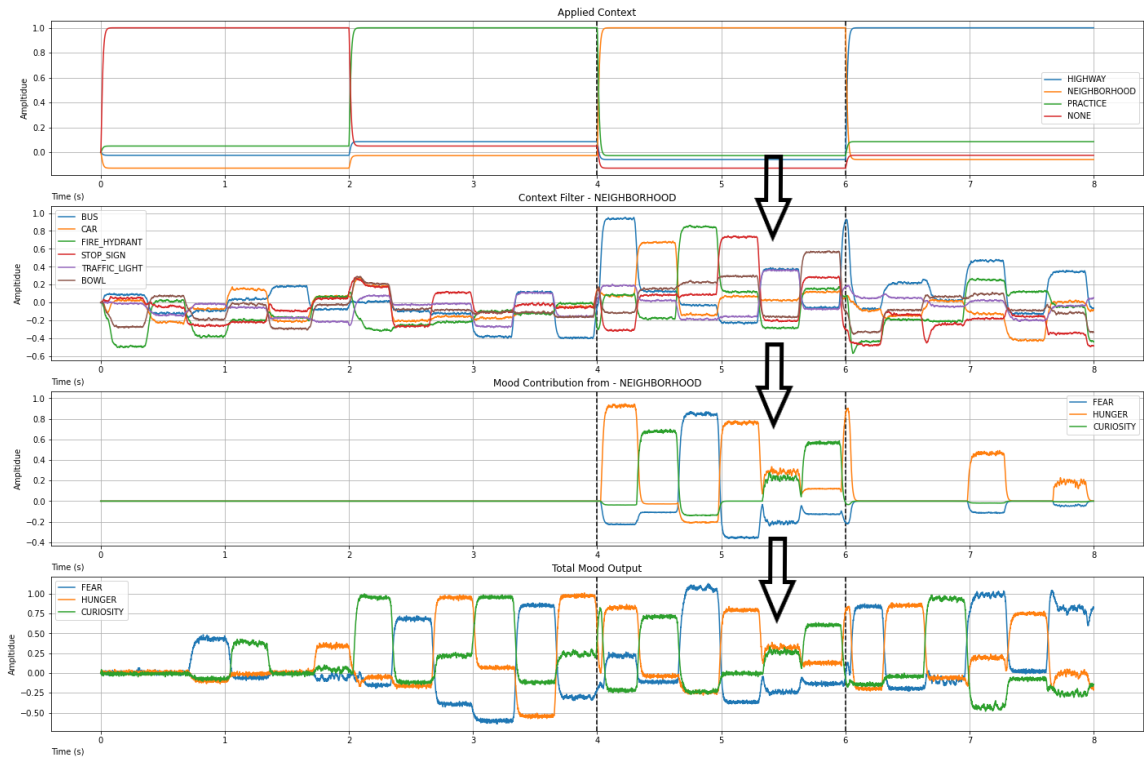


Figure 4.7: Example of context signal flow from input, unbinding, and association. Probes, from top to bottom, are at locations 2, 4, 5, and 9. Arrows and bands show signal routing and contributions.

Showing the flow of multiple signals can be more complicated. In some cases, this is done by plotting each context side by side with the full response in the bottom-most plot. The intention is to show how each path contributes to the overall response. An example of this is shown in Figure 4.8. This plot style shows the propagation and transformation of signals from input to output. The first row shows contexts as they change in time. The second row shows the inputs to each context path as a subplot. The first column is “Practice”, the second is “Neighborhood”, and third is “Highway”. These paths are repeated from Figure 4.7 to Figure 4.8 above and incorporated with the full response for context. The respective mood responses to each context are placed below each in row three. The last row is the overall mood response and shows how each part contributed.



Figure 4.8: Example of context signal flow with multiple contexts shown at once.

4.3 DEMONSTRATION FOR PROOF OF CONCEPT

Proof of concept testing for each model was performed to evaluate performance and accuracy to an expected outcome. The models were compared to each other but not compared to the Model A. These do not operate in the same manner due to the SPA design, and can only be evaluated by similarity in behavior. Simulations were run using a five-second simulation time with a time step of 0.001. Table 4.1 lists the neuron count, average simulation execution time, and the standard deviation between runs for each model. Table 4.2 provides an overview of model accuracy.

Table 4.1: Proof of Concept Results. Simulations are 5s with dt=0.001

Model	Neuron Count	Avg Sim Time (s)	Std. Deviation	Reason
Model 0	7750	26.308	0.873	Initial model
Model 1	22600	18.188	1.065	Model reduction
Model 2	6700	16.960	0.728	
Model B	1800	4.659	0.115	Previous research
Model 2M	8450	19.683	1.365	New feature
Context (end)	8600	22.567	1.172	Test feature
Model 2MC	64850	61.430	1.814	New feature

Table 4.2: Model accuracy overview

Model	Neuron Count	Accuracy (Single Input)	Accuracy (Two Inputs)
Model 0	7750	67%	100%
Model 1	22600	67% (50%)	100%
Model 2	6700	67% (100%)	100%
Model B	1800	100% (92%)	100%
Model 2M	8450	100%	
Context (end)	8600	NA	NA
Model 2MC	64850	83%	

Simulations of the simple model were performed to capture the overall behavior. Inputs for the eyes and known persons used a square wave input. The lateral to central connection path evaluates if the mouth and teeth are below a threshold of 0.8 (changed to

0.6 for simulation) and asserts an angry mood. This value could outweigh the inputs from the basal ganglia portion, which is evaluating all four inputs differently. Inputs were randomized for testing using square waves for eyes and known persons while mouth and teeth were produced with sinusoids.

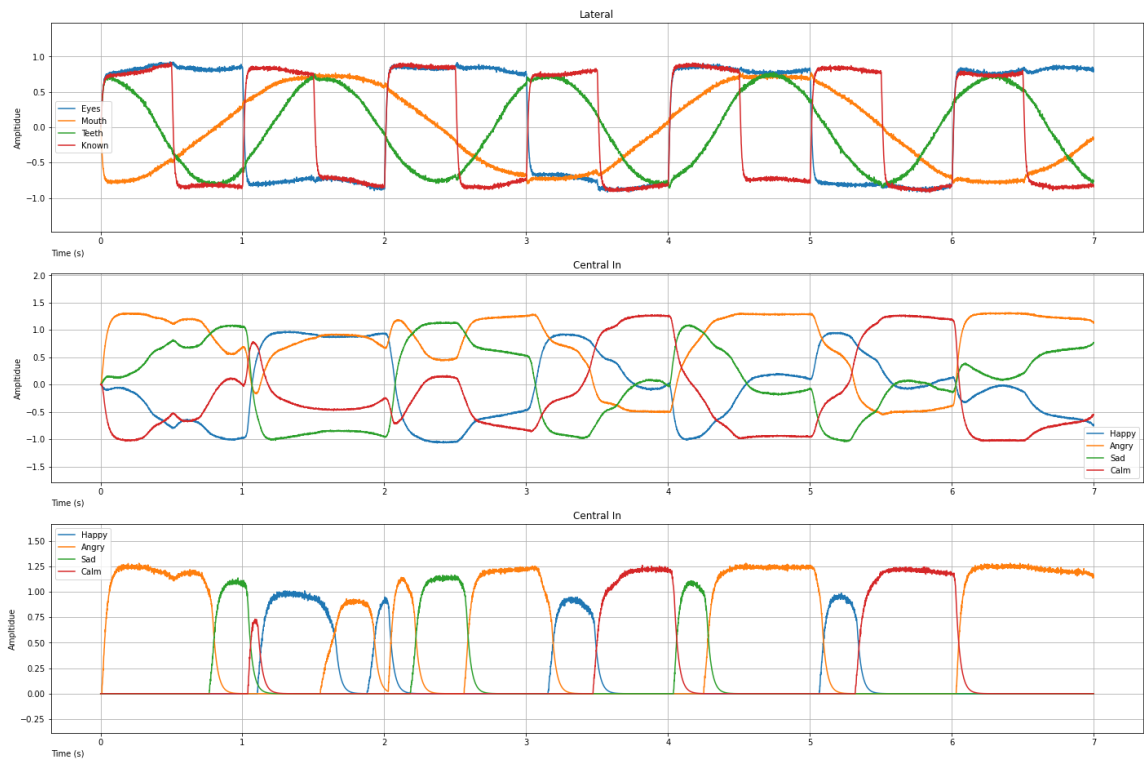


Figure 4.9: Response from Simple model

Figure 4.9 shows the results of the stimulus and output response. The plot for the input to the central nucleus reflects the output of both the basal ganglia and lateral pathways before its final conversion for the output. Stronger assertions of the mood “Anger” are shown when the signals for mouth and teeth meet the threshold requirements.

Model B is highly simplified, consisting of a single associative memory block. The probes capture both input signals and the associated output. Since this block is set up

for threshold and not WTA the output signals expect a dominant signal but do not expect other signals to be suppressed.

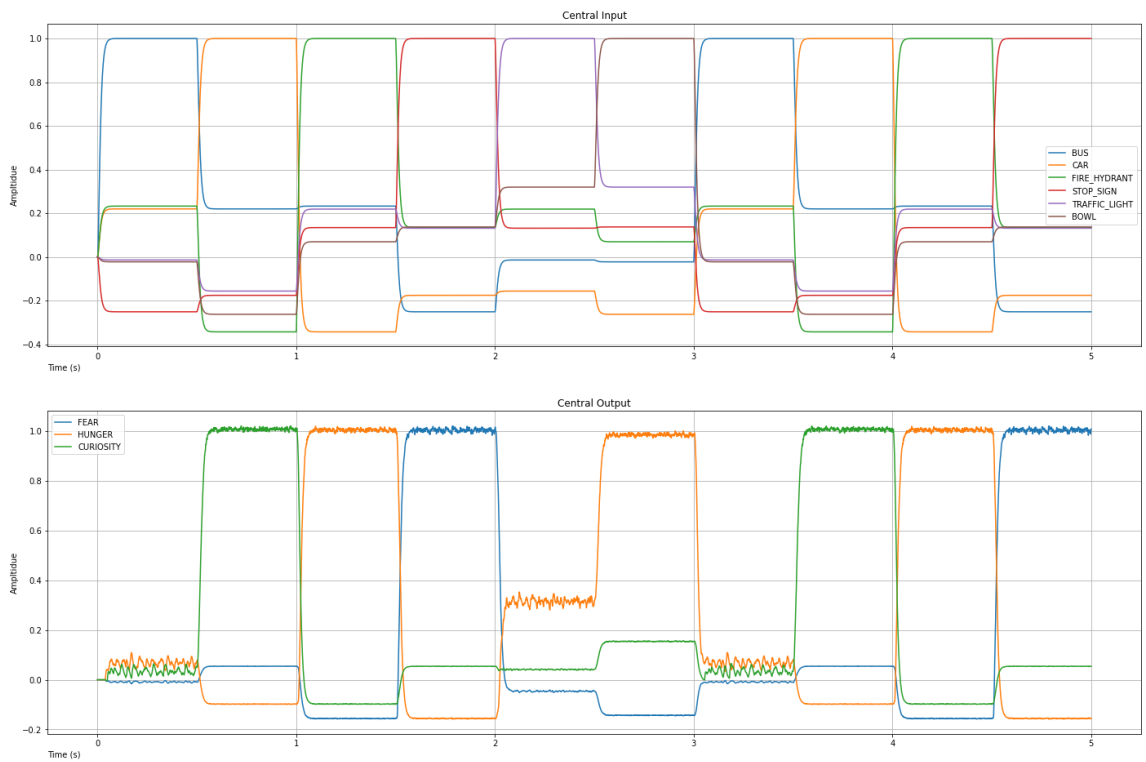


Figure 4.10: Model B input to output response over 5s simulation time with a single input

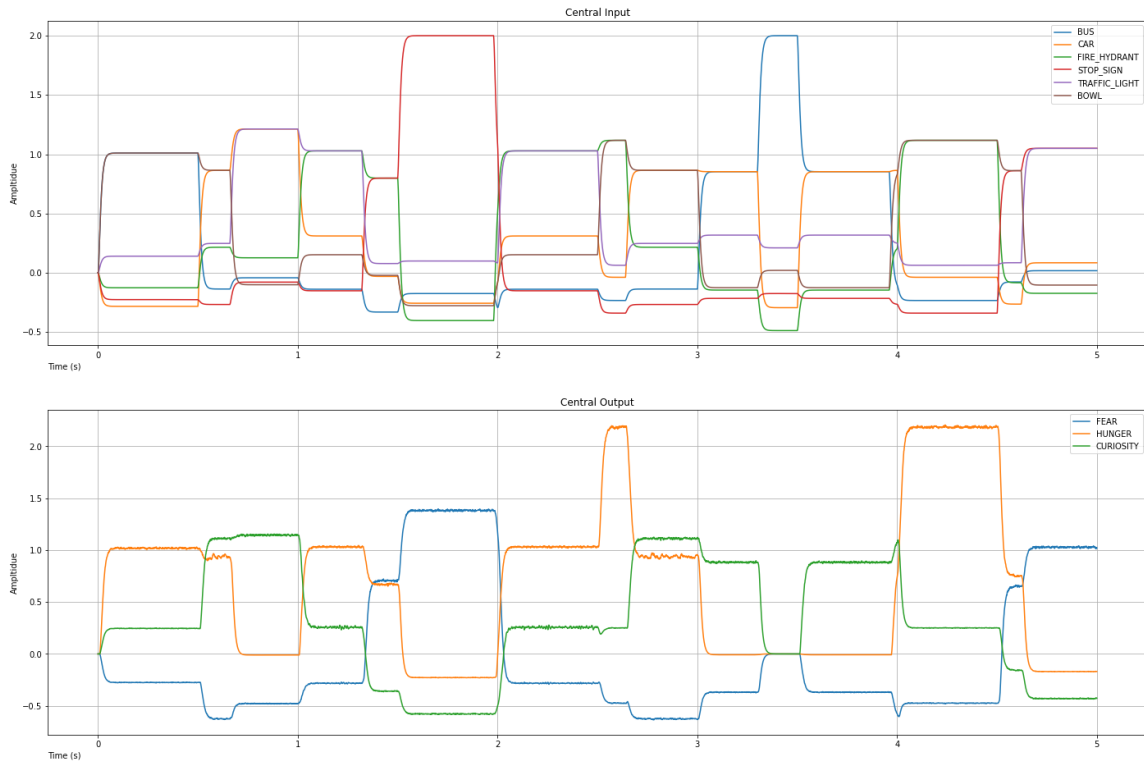


Figure 4.11: Model B output response from two inputs over 5s simulation

Figure 4.10 shows the effectiveness and simplicity of a single-stage module. Figure 4.11 shows how multiple inputs are easily handled and can include multiple mood responses due to the thresholding nature.

Table 4.3: Model accuracy from a single input. Yellow items yield close but different results.

Time	Input	Expected Central Output	Measured Central Output
0.25	0.25	None	None
0.75	0.75	Curiosity	Curiosity
1.25	1.25	Hunger	Hunger
1.75	1.75	Fear	Fear
2.25	2.25	None	None (Hunger 0.3)
2.75	2.75	Hunger	Hunger

Table 4.4: Model B accuracy from two inputs.

Time	Input	Expected Central Output	Measured Central Output
0.25	Bus, Bowl	Hunger	Hunger
0.75	Car, Traffic Light	Curiosity	Curiosity
1.25	Fire Hydrant, Traffic Light	Hunger	Hunger
1.75	Stop Sign	Fear	Fear
2.25	Fire Hydrant, Traffic Light	Hunger	Hunger
2.75	Car, Bowl	Hunger + Curiosity	Hunger (0.8) + Curiosity (1.25)
3.25	Bus, Car	Curiosity	Curiosity
3.75	Bus, Car	Curiosity	Curiosity
4.25	Fire Hydrant, Bowl	Hunger	Hunger
4.75	Stop Sign, Traffic Light	Fear	Fear

Figure 4.3 and Figure 4.4 present the measured versus expected results. The results were sampled every 250 ms to avoid signal transitions. The yellow text reveals a minor inaccuracy which can be considered noise.

Model B provides the lowest number of neurons, fastest result, best accuracy, and the simplest solution. The trade-off is the loss of weighted inputs provided in other models but still holds to principles of the amygdala function.

Model 0 closely mimics the behavior of the simple model. Compared to other models, Model 0 requires 10 seconds more than Model 2 to execute the simulation and is nearly five times slower than Model B. An accuracy test using a single input yielded 67%. The inaccuracies occurred when the input has no associated mood mapping. The action selection path treats the highest signal as the maximum despite its absence as an input. For example, if there is no input, but noise from the object “CAR” is the maximum signal, then the associated output, “CURIOSITY”, is asserted as the response, which causes the output to yield an incorrect result.

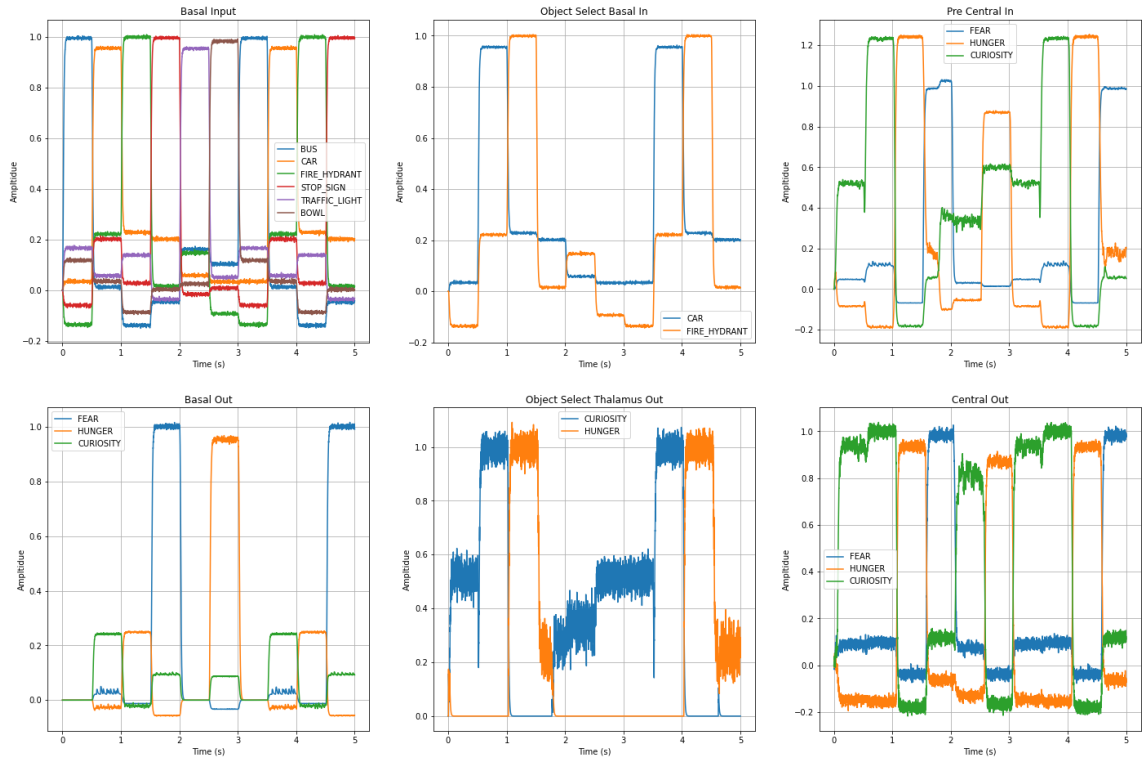


Figure 4.12: Model 0 selection path signals.

Figure 4.12 displays the results using two inputs and the reflected output. The central plots of the figure show how a low signal strength for “CAR” produces a moderate “CURIOSITY” response during the first half-second. The impacts can be viewed in the central output in the bottom right.

Table 4.5: Model 0 expected versus measured results with a single input. The results in red are incorrect.

Time	Input	Expected Pre-Central	Measured Pre-Central	Expected Central (Output)	Measured Central (Output)
0.25	Bus	None	0.5 Curiosity	None	Curiosity
0.75	Car	Curiosity	Curiosity	Curiosity	Curiosity
1.25	Fire Hydrant	Hunger	Hunger	Hunger	Hunger
1.75	Stop Sign	Fear	Fear	Fear	Fear
2.25	Traffic Light	None	0.4*Curiosity	None	Curiosity
2.75	Bowl	Hunger + Curiosity	Hunger + 0.6 Curiosity	Hunger	Hunger

Table 4.6: Model 0 expected versus measured results with two inputs. The results in red are incorrect.

Time	Input	Expected Pre-Central	Measured Pre-Central	Expected Central (Output)	Measured Central (Output)
0.25	Bus, Bowl	Hunger + Curiosity	Hunger + 0.6 Curiosity	Hunger	Hunger
0.75	Car, Traffic Light	Curiosity	Curiosity	Curiosity	Curiosity
1.25	Fire Hydrant, Traffic Light	Hunger	Hunger	Hunger	Hunger
1.75	Stop Sign	Fear	Fear	Fear	Fear
2.25	Fire Hydrant, Traffic Light	Hunger	Hunger	Hunger	Hunger
2.75	Car, Traffic Light	Curiosity	Curiosity + Hunger	Curiosity	Curiosity
3.25	Bus, Car	Curiosity	Curiosity	Curiosity	Curiosity
3.75	Bus, Car	Curiosity	Curiosity	Curiosity	Curiosity
4.25	Fire Hydrant, Bowl	Hunger + Curiosity	Hunger	Hunger	Hunger
4.75	Stop Sign, Traffic Light	Fear	Fear + Hunger	Fear	Fear

Table 4.5 and Table 4.6 present the measured versus expected results. The results were sampled every 250 ms to avoid signal transitions. It highlights in red which tests failed. Despite the closeness of adaptation, this model is deemed impractical due to the slow speed and inaccuracies.

Model 1 relies on a basal ganglia and thalamus circuit for WTA results. The cortex output is the memory stage of the basal-thalamus network. The winner, or maximum signal, is expected to be large and the rest should be suppressed to a value of zero or less. The signals along this path cause the results to be skewed from the expectation.

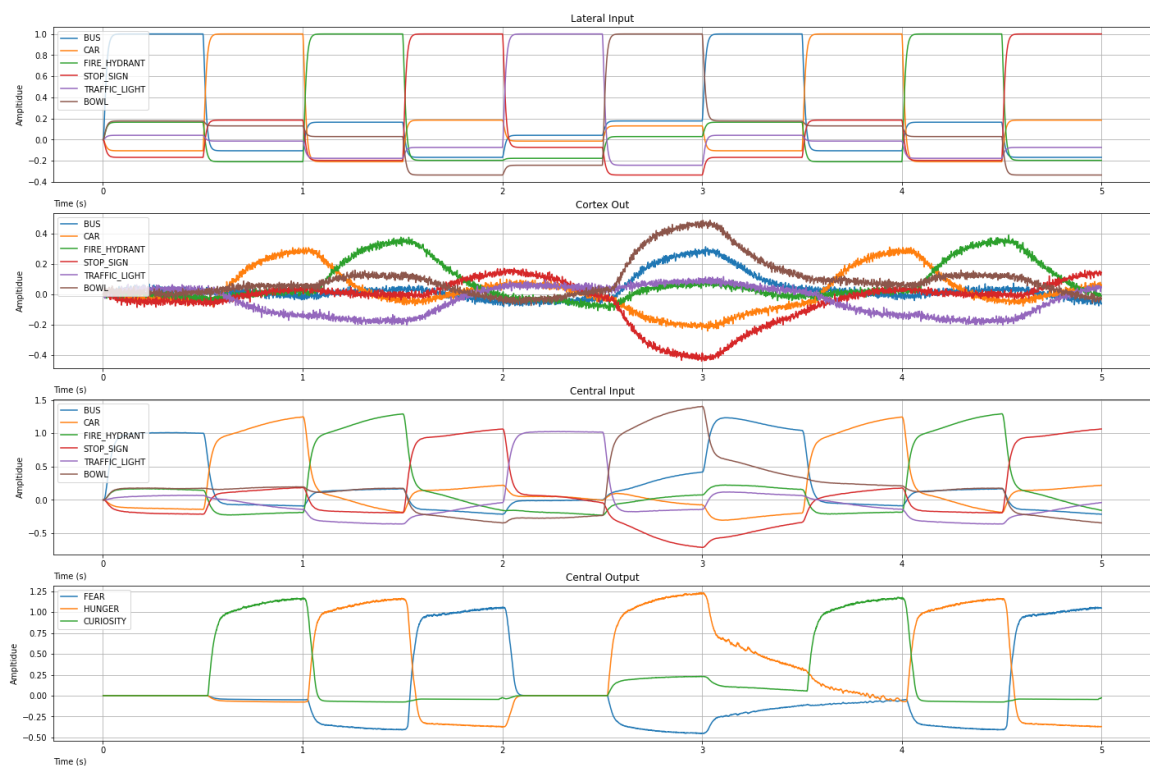


Figure 4.13: Model 1 input to output response over 5s simulation time

Figure 4.13 shows measurements along the Model 1 paths. Object inputs are in the top row. The cortex output is the second row. Only one signal should be asserted

while others are suppressed, but the subplot clearly shows this is false. The figure shows where signals of the central input don't always produce expected signals at the central output. Table 4.7 and Table 4.8 present the measured versus expected results. The results were sampled every 250 ms to avoid signal transitions. It highlights in red which tests failed. Of note, three out of six mood responses were successful in Table 4.7, generating a low accuracy. Mood responses in Table 4.8 were 100% accurate, indicating false mood generation when a single input is present without an expected response.

Table 4.7: Model 1 expected versus measured results for a single input. Red entries are incorrect.

Time	Input	Expected Central Input	Measured Central Input	Expected Central Output	Measured Central Output
0.25	0.25	Bus	Bus	None	Fear (0.2)
0.75	0.75	Car	Car	Curiosity	Curiosity
1.25	1.25	Fire Hydrant	Fire Hydrant	Hunger	Hunger
1.75	1.75	Stop Sign	Stop Sign	Fear	Fear
2.25	2.25	Traffic Light	Traffic Light	None	Fear (0.4)
2.75	2.75	Bowl	Bowl	Hunger + Curiosity	Hunger

Table 4.8: Model 1 accuracy results based on two inputs. Red entries are incorrect.

Time	Input	Expected Central Input	Measured Central Input	Expected Central Output	Measured Central Output
0.25	Bus, Bowl	Bus, Bowl	Bus, Bowl	Hunger	Hunger
0.75	Car, Traffic Light	Car, Traffic Light	Car, Traffic Light	Curiosity	Curiosity
1.25	Fire Hydrant, Traffic Light	Fire Hydrant, Traffic Light	Fire Hydrant, Traffic Light, Bowl	Hunger	Hunger
1.75	Stop Sign	Stop Sign	Stop Sign	Fear	Fear
2.25	Fire Hydrant, Traffic Light	Fire Hydrant, Traffic Light	Fire Hydrant, Traffic Light	Hunger	Hunger
2.75	Car, Bowl	Car, Bowl	Car, Bowl	Hunger + Curiosity	Hunger + Curiosity
3.25	Bus, Car	Bus, Car	Bus, Car, Stop Sign	Curiosity	Curiosity
3.75	Bus, Car	Bus, Car	Bus, Car	Curiosity	Curiosity
4.25	Fire Hydrant, Bowl	Fire Hydrant, Bowl	Fire Hydrant, Bowl	Hunger	Hunger
4.75	Stop Sign, Traffic Light	Stop Sign, Traffic Light	Stop Sign, Traffic Light	Fear	Fear

An underlying element of Nengo caused challenges for this model. The basal ganglia and thalamus modules were expected to function with the SPA inputs. Instead of finding the winner with an abstracted symbol, it appeared to compare each of the 32 dimensions in the underlying vector space and select the winner from each one. Because of this, some results were observed to display outputs that weren't even selected for the inputs. Model 1 was deemed invalid due to the incorrect behavior of the basal-thalamus circuit. This is seen in Table 4.7

Model 2's basal-thalamus path provides the maximum function in an expected manner and yields results that were not found in Model 1. The neuron count is less than

Model 1 by nearly 16,000 neurons, however, its average simulation time is only 5% faster.

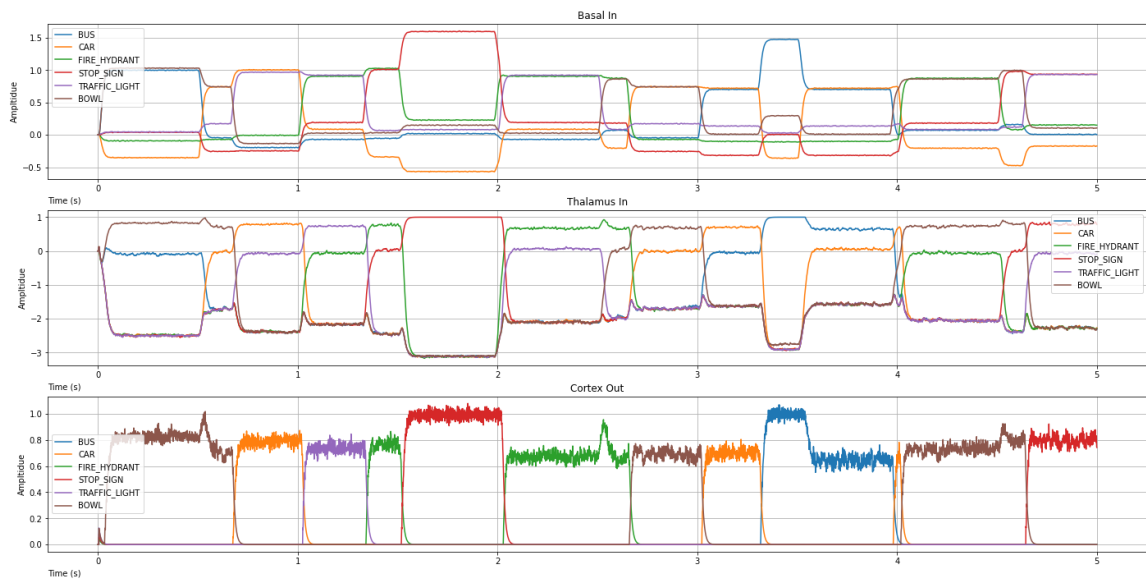


Figure 4.14: Model 2 basal-thalamus path showing WTA behavior with two inputs.

The desired WTA behavior is observed in Figure 4.14. The basal ganglia select the maximum signal and can be viewed in the “Thalamus In” in the center. The thalamus portion suppresses all other signals to provide a single dominant signal. This is viewed in the “Cortex Out” subplot.

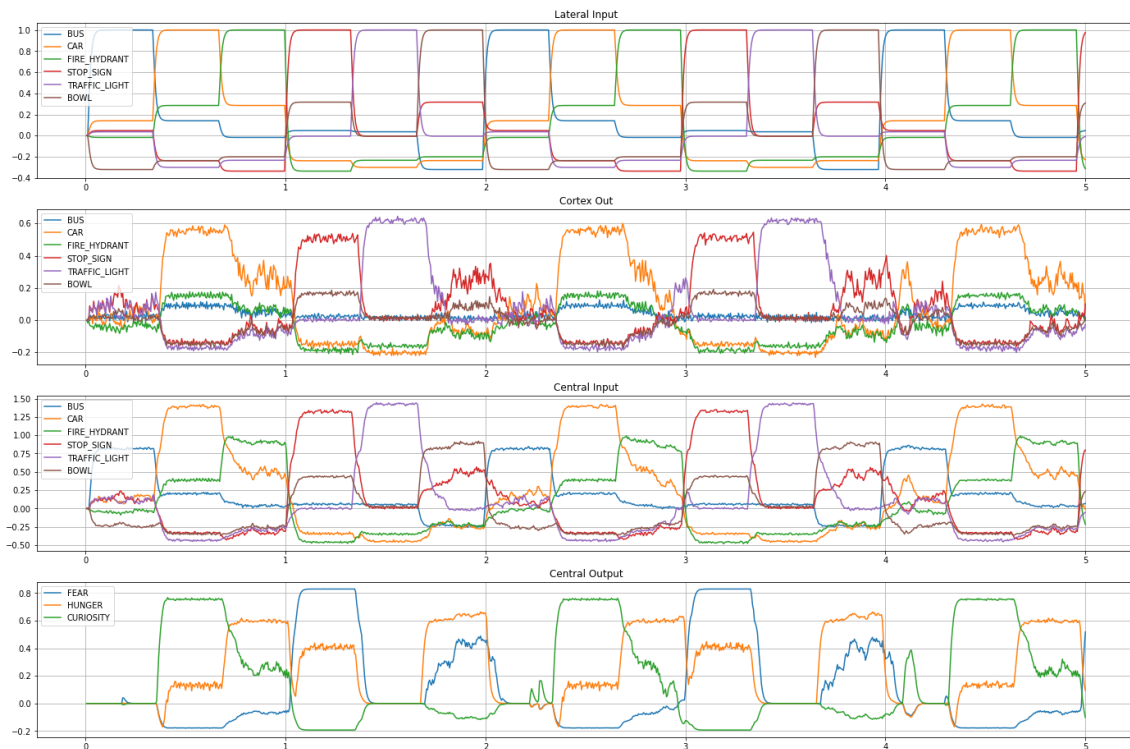


Figure 4.15: Model 2 input to output response over 5s simulation time

Figure 4.15 provides visibility into each stage of the model. The objects “CAR”, “STOP SIGN”, and “TRAFFIC LIGHT” hold additional weight based on the configuration shown in Figure 3.5. This plot shows the influence of weighted items in the “Cortex Output” stage of the basal-thalamus circuit on the “Central Output”.

Single input accuracies follow the trend of Model 0 and Model 1 at 67%. Using thresholding associative memory causes this inaccuracy. Increasing the threshold or changing to a WTA associative memory can resolve the inaccuracies. Changes to the threshold could suppress low signals but would remain effective, however changing the typing of associative memory would only allow a single output. This matches the simple model but would prevent mixed moods or behaviors where other brain segments could

determine the appropriate action. Two or more inputs yield 100% accuracy with signal levels slightly lower than expected at 2.75 and 3.75 seconds.

Table 4.9: Model 2 expected versus measured results for a single input. Red entries are incorrect.

Time	Input	Expected Central Input	Measured Central Input	Expected Central Output	Measured Central Output
0.25	0.25	Bus	Bus	None	Hunger (0.6)
0.75	0.75	Car	Car	Curiosity	Curiosity
1.25	1.25	Fire Hydrant	Fire Hydrant	Hunger	Hunger
1.75	1.75	Stop Sign	Stop Sign	Fear	Fear
2.25	2.25	Traffic Light	Traffic Light	None	Hunger (0.6)
2.75	2.75	Bowl	Bowl	Hunger + Curiosity	Hunger (1.4) + Curiosity (0.25)

Table 4.10: Model 2 expected versus measured with two inputs. Yellow items yield close but different results.

Time	Input	Expected Central Input	Measured Central Input	Expected Central Output	Measured Central Output
0.25	Bus, Bowl	Bus, Bowl	Bus (1.2), Bowl (1.9)	Hunger	Hunger
0.75	Car, Traffic Light	Car, Traffic Light	Car (1.8), Traffic Light (1.1)	Curiosity	Curiosity
1.25	Fire Hydrant, Traffic Light	Fire Hydrant, Traffic Light	Fire Hydrant (0.9), Traffic Light (1.7)	Hunger	Hunger
1.75	Stop Sign	Stop Sign	Stop Sign (2.5)	Fear	Fear
2.25	Fire Hydrant, Traffic Light	Fire Hydrant, Traffic Light	Fire Hydrant (1.8), Traffic Light (0.9)	Hunger	Hunger
2.75	Car, Bowl	Car, Bowl	Bowl (1.5), Car (0.7)	Hunger + Curiosity	Hunger (1.25) + Curiosity (0.75)
3.25	Bus, Car	Bus, Car	Bus (0.5), Car (1.5)	Curiosity	Curiosity
3.75	Bus, Car	Bus, Car	Bus (1.5), Car (0.5)	Curiosity	Curiosity
4.25	Fire Hydrant, Bowl	Fire Hydrant, Bowl	Fire Hydrant (0.8), Bowl (1.6)	Hunger	Hunger
4.75	Stop Sign, Traffic Light	Stop Sign, Traffic Light	Stop Sign (1.8), Traffic Light (1.0)	Fear	Fear

Table 4.9 and Table 4.10 present the measured versus expected results. The results were sampled every 250 ms to avoid signal transitions. It highlights in red which tests failed. Of note two of six inputs were incorrect in Table 4.9. Thus, while Model 2 greatly reduced the neuron count from Model 1, accuracy remained the same but decreased in simulation time of 6.7%. Model B is 72% faster 73% smaller than Model 2. However, Model 2 was selected over Model B because it allowed pathways to emphasize selected object inputs. This feature better reflected the Model A's connection to the basal and was preferred for the design. In a modularized form, these features can be easily adjusted or omitted.

Model 2M adds motivation to Model 2, combining it with the CEA for the complete mood response. A single input was used to highlight the impacts of motivation on the output. The additional circuitry adds 650 neurons and up to six seconds of simulation time. The top row of Figure 4.16 shows the inputs to the central nucleus (top left) and the motivation input (top right). Row two shows the output effects of each of those nodes. The output mood displays the superposed result of the image to mood and motivation to mood signals.

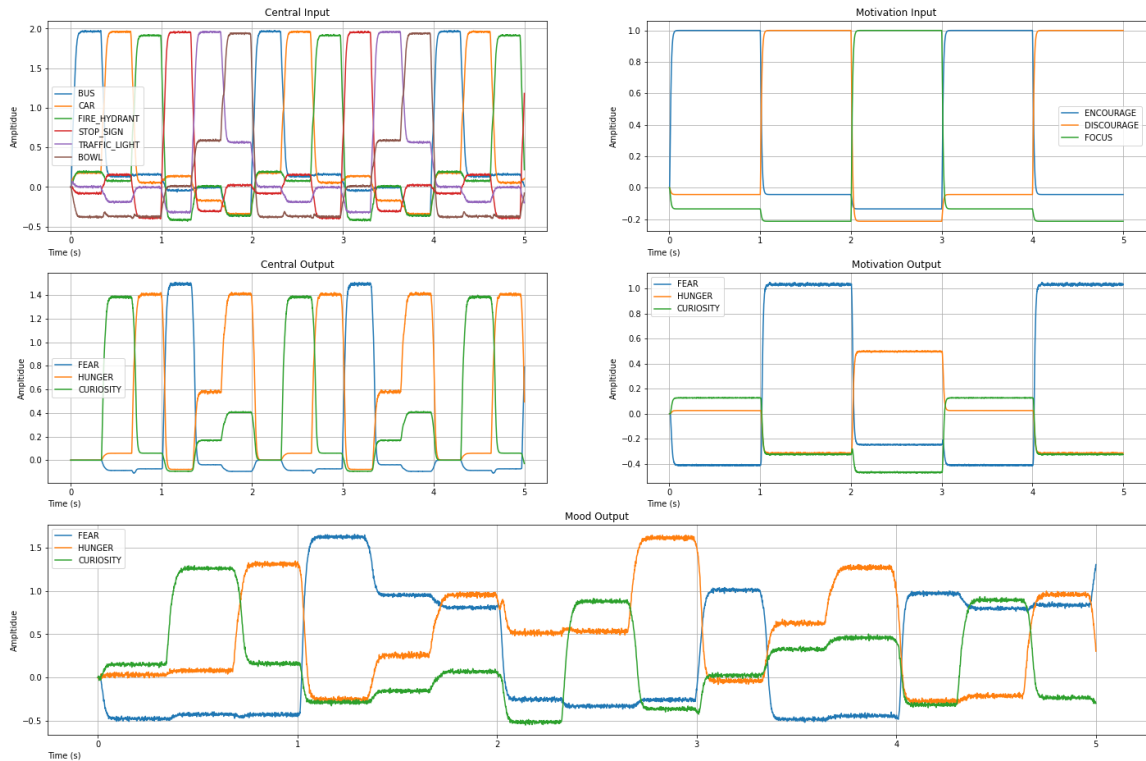


Figure 4.16: Effects of motivation with single detection input. Central and motivation outputs combine to produce the overall mood output.

Table 4.11 present the measured versus expected results for Model 2M as seen in Figure 4.16. The results were sampled every 250 ms to avoid signal transitions. The yellow text reveals a minor inaccuracy in signal level. Adding motivation was an effective influence on behavioral responses. The impact of “FEAR” with the motive “DISCOURAGE” was reduced. Testing revealed the influence overwhelmed other inputs

causing a constant state of fear. Simulation execution time increased up to six seconds by adding this feature to the motivation model.

Table 4.11: Accuracy for adding motivation with a single input

Time	Input	Motivation	Expected Output	Measured Output
0.17	Bus	Encourage	Fear (-0.4)	Fear (-0.4)
0.5	Car	Encourage	Curiosity	Curiosity
0.83	Fire Hydrant	Encourage	Hunger	Hunger
1.16	Stop Sign	Discourage	Fear	Fear
1.49	Traffic Light	Discourage	Fear	Fear
1.82	Bowl	Discourage	Fear + Hunger	Fear (0.5) + Hunger (1.3)
2.15	Bus	Focus	Hunger	Hunger
2.48	Car	Focus	Hunger + Curiosity	Hunger + Curiosity
2.81	Fire Hydrant	Focus	Hunger	Hunger
3.14	Stop Sign	Encourage	Fear (0.5)	Fear (1.0)
3.47	Traffic Light	Encourage	Fear (-0.4)	Fear (-0.4)
3.8	Bowl	Encourage	Hunger + Curiosity (0.4)	Hunger + Curiosity (0.5)
4.13	Bus	Discourage	Fear	Fear
4.46	Car	Discourage	Fear + Curiosity	Fear + Curiosity
4.79	Fire Hydrant	Discourage	Fear + Hunger	Fear (0.5) + Hunger (1.2)
5.12	Stop Sign	Focus	Fear + Hunger	Fear (1.0) + Hunger (0.5)
5.45	Traffic Light	Focus	Hunger	Hunger
5.78	Bowl	Focus	Hunger	Hunger

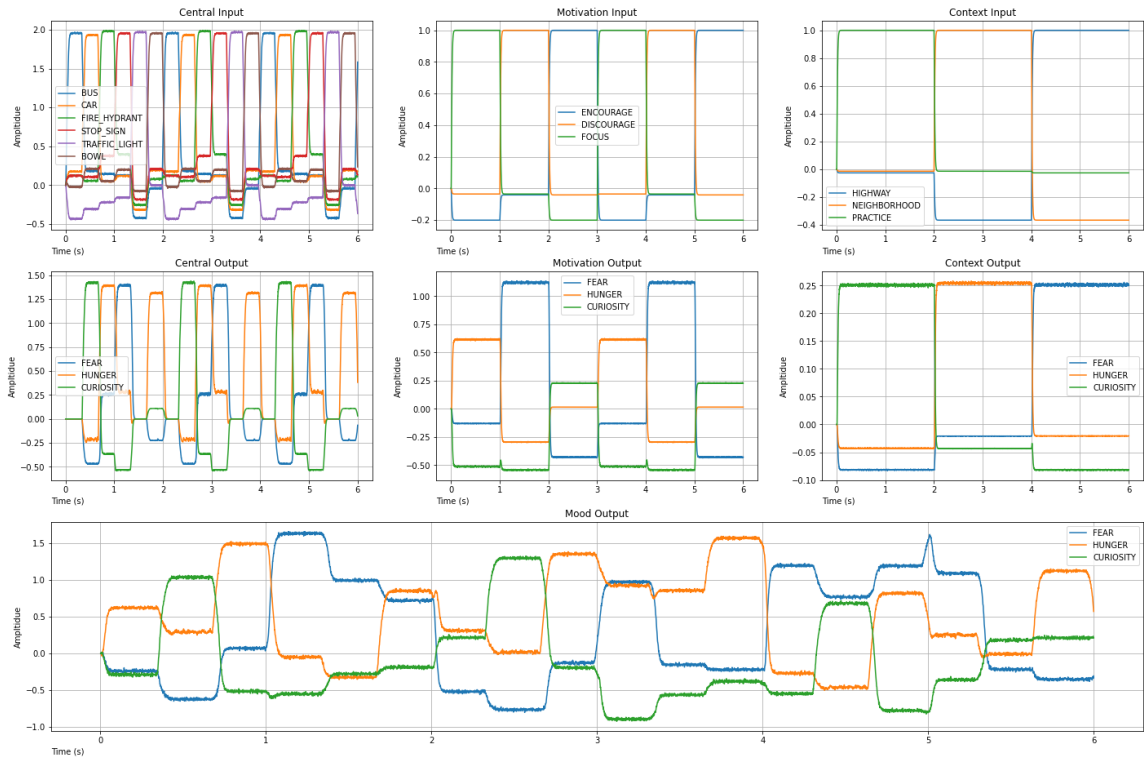


Figure 4.17: Effects of a detected object, motivation, and context added at the end. Central, motivation, and context outputs combine to produce the final mood output.

Model 2MC adds context with binding and unbinding networks creates a multiplexer-like pathway. The model will scale relative to the number of contexts. In this case, the model increases by at least 56,000 neurons. The average simulation time, 61.43s, was approximately three times that of the motivation model. This was not unexpected due to the additional circuits.

Each context has a unique association table allowing a different response to detected objects. A test was created using Table 4.12 for easily identifiable differences. This table describes the context-related mappings of input to mood. The motivation stimulus was nullified for simplification.

Table 4.12: Test associations to demonstrate output changes with context

Context	Input	Mapping
Highway	Bus	Fear
Highway	Car	Hunger
Highway	Fire Hydrant	Curiosity
Highway	Stop Sign	$\text{Fear} - 0.25 * \text{Curiosity}$
Highway	Traffic Light	$\text{Hunger} + 0.25 * \text{Fear}$
Highway	Bowl	$\text{Fear} - 0.5 * \text{Curiosity}$
Neighborhood	Bus	Hunger
Neighborhood	Car	Curiosity
Neighborhood	Fire Hydrant	Fear
Neighborhood	Stop Sign	$\text{Hunger} - 0.25 * \text{Fear}$
Neighborhood	Traffic Light	$\text{Curiosity} - 0.5 * \text{Fear}$
Neighborhood	Bowl	$0.25 * \text{Hunger} + \text{Curiosity}$
Practice	Bus	Curiosity
Practice	Car	Fear
Practice	Fire Hydrant	Hunger
Practice	Stop Sign	$\text{Curiosity} - 0.5 * \text{Fear}$
Practice	Traffic Light	$\text{Fear} - 0.5 * \text{Hunger}$
Practice	Bowl	$\text{Hunger} + 0.25 * \text{Curiosity}$
None	All inputs	(None)

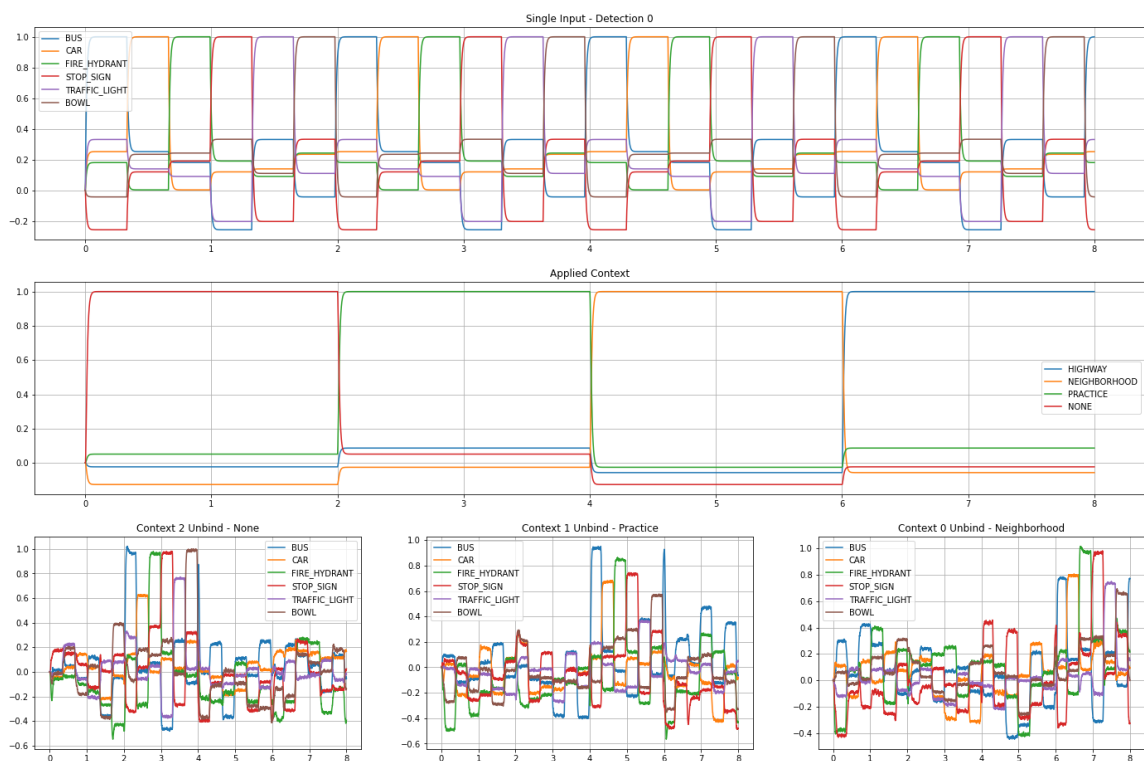


Figure 4.18: Multiplexer effect of context unbinding – single detection and a single association table

The middle subplot of Figure 4.18 shows the applied context. A new context was cycled every two seconds. Unbinding circuits proved the desired multiplexer behaviors. Filtered inputs are shown in the bottom row and labeled with the respective context. Each of the unbound responses also shows signals outside of the filtered context. Noise is found in each of the outputs, which is an artifact of the unbinding process itself. Increasing the threshold levels of the associative memory units will help reduce noise but can also suppress low but valid signals. Symbols for context were created in a vocabulary separate from recognized objects. Symbolic reinterpretation of the context was used for binding and unbinding. This translation may add errors and noise if the generated vector and an existing term are too similar.

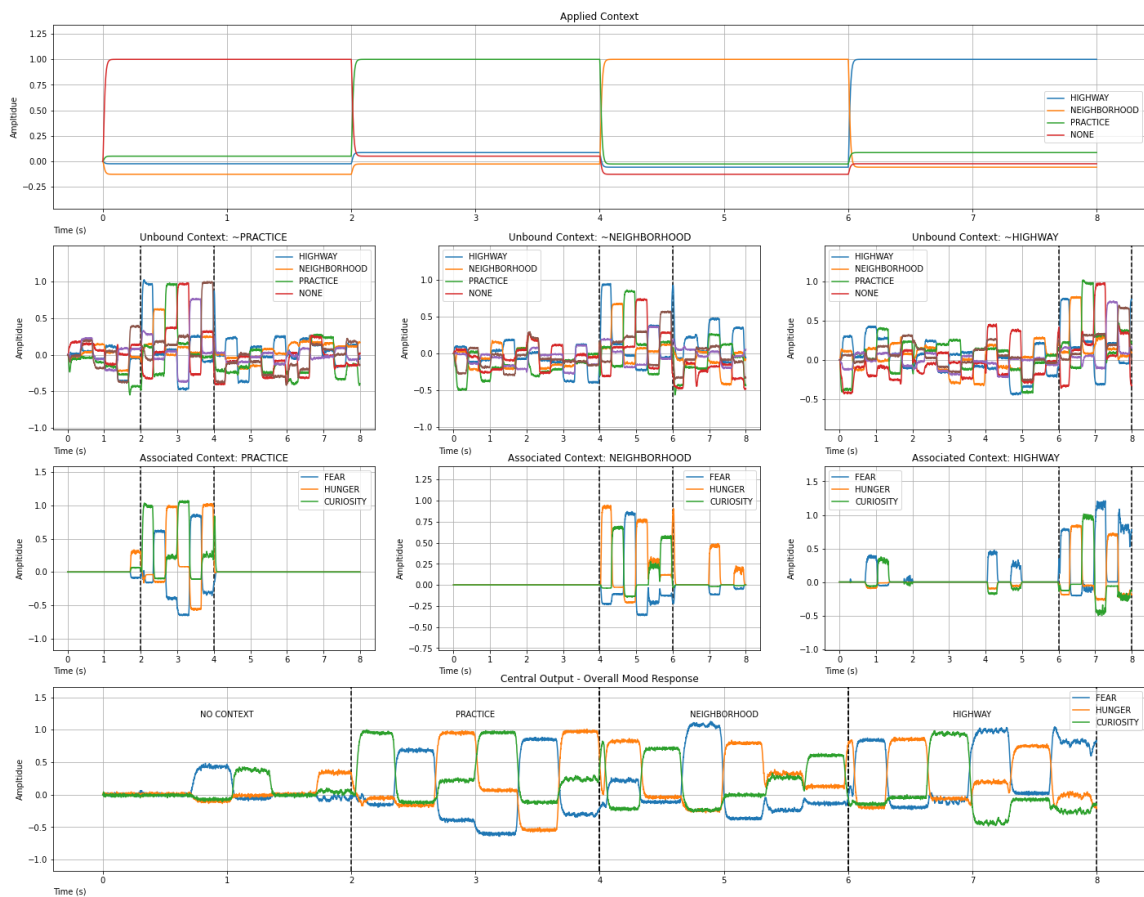


Figure 4.19: Context unbinding with unique association tables

Figure 4.19 shows the relationship between the context filtered inputs and the mood outputs. The object input cycles through six objects. This cycle demonstrates how object responses are altered by context. Noise is shown to carry over into the associated moods, which sometimes causes unintended impacts on the overall responses.

Table 4.13: Accuracy measurements for context modeling. Items in red are errors.

Time	Input	Context	Expected Output	Measured Output
0.17	Bus	None	None	None
0.5	Car	None	None	None
0.83	Fire Hydrant	None	None	Fear (0.4)
1.16	Stop Sign	None	None	Curiosity (0.4)
1.49	Traffic Light	None	None	None
1.82	Bowl	None	None	Hunger (0.3)
2.15	Bus	Practice	Curiosity	Curiosity
2.48	Car	Practice	Fear	Fear
2.81	Fire Hydrant	Practice	Hunger	Hunger
3.14	Stop Sign	Practice	Curiosity – 0.5*Fear	Curiosity - Fear (-0.7)
3.47	Traffic Light	Practice	Fear – 0.5*Hunger	Fear - Hunger (-0.6)
3.8	Bowl	Practice	Hunger + 0.25*Curiosity	Hunger + Curiosity (0.25)
4.13	Bus	Neighborhood	Hunger	Hunger + Fear (0.25)
4.46	Car	Neighborhood	Curiosity	Curiosity
4.79	Fire Hydrant	Neighborhood	Fear	Fear
5.12	Stop Sign	Neighborhood	Hunger – 0.25*Fear	Hunger - Fear (-0.35)
5.45	Traffic Light	Neighborhood	Curiosity – 0.5*Fear	Curiosity (0.25) + Hunger (0.3) - Fear (-0.25)
5.78	Bowl	Neighborhood	0.25*Hunger + Curiosity	Hunger (0.2) + Curiosity
6.11	Bus	Highway	Fear	Fear
6.44	Car	Highway	Hunger	Hunger
6.77	Fire Hydrant	Highway	Curiosity	Curiosity
7.1	Stop Sign	Highway	Fear – 0.25*Curiosity	Fear – Curiosity (-0.4)
7.43	Traffic Light	Highway	Hunger + 0.25*Fear	Hunger
7.76	Bowl	Highway	Fear – 0.5*Curiosity	Fear - Curiosity (-0.25)

Table 4.13 presents the measured versus expected results. The results were sampled every 330 ms to avoid signal transitions. It highlights in red which tests failed.

The failures are related to signal noise and low thresholds. This knowledge allowed for improvements. Figure 4.20 shows noise reduction due to the threshold increase. The original signals are displayed in grayscale to easily show the filtering effects of the applied threshold.

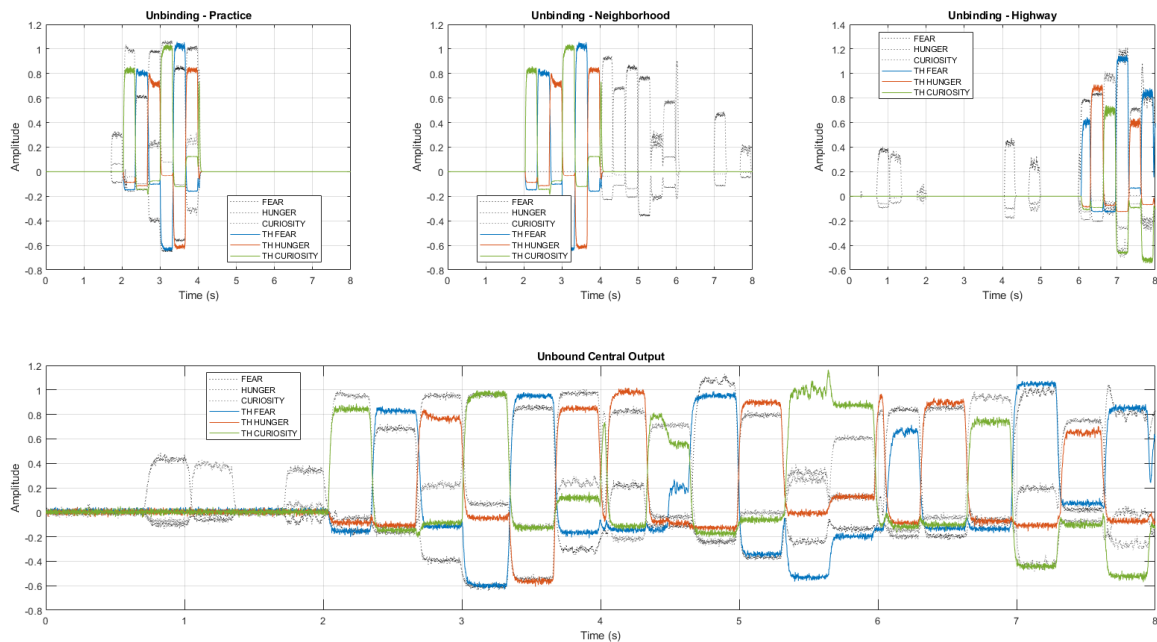


Figure 4.20: Increasing thresholds to 0.55 reduces unbinding noise with the noisy data shown in grey

Adding context offers unique ways to process information. It demonstrated the multiplexed nature of the signals and allows for higher-level thinking. Increasing the threshold effectively removes noise created during the unbinding process. A WTA associative memory would also require an increased threshold and would avoid the results from combined outcomes.

5 SIMULATION AND ROBOTIC EMBODIMENT

5.1 TURTLEBOT 2.0

A TurtleBot 2.0 was selected for the simulated and live embodiment of this research, consistent with, and as an extension of, the work in [18] [19]. The TurtleBot 2.0 is an open source, wheeled, robotic development kit. Platforms stacked vertically above the mobile base allowed for easy additions of sensors and other hardware [33]. Vision for the robot was established using an Xbox 360 Kinect camera and a standard NVIDIA Jetson TX2 was used for image processing. Operating the TurtleBot with the new amygdala model is the primary objective. The features for context and motivation can be evaluated on this platform. Simulations are a quick method to prove functionality before migrating to a real robot.

5.2 INTEGRATION

Integration into a larger robotic system was the next step for the model. After proving the connections and behavior the remainder of the robotic system was included. This allowed integration with the simulated TurtleBot in the Gazebo environment to capture movement, orientation, and response times.

A view of the integrated model is shown in Figure 5.1. Objects in blue are robot functions using neuron ensembles. Motor controls, position, orientation, and target information are among these networks. The robot and CNN are external to the Nengo model and are shown in gray. The amygdala and basal-thalamus sections are marked in orange. These are all cognitive elements. The diagram notes two connections. The robot to CNN link comes from Kinect images provided by ROS. Object detections are provided to the Amygdala from the CNN via ROS. Mood responses are provided to the basal-

thalamus for action selection. These actions are turned into motor selections by another basal-thalamus section.

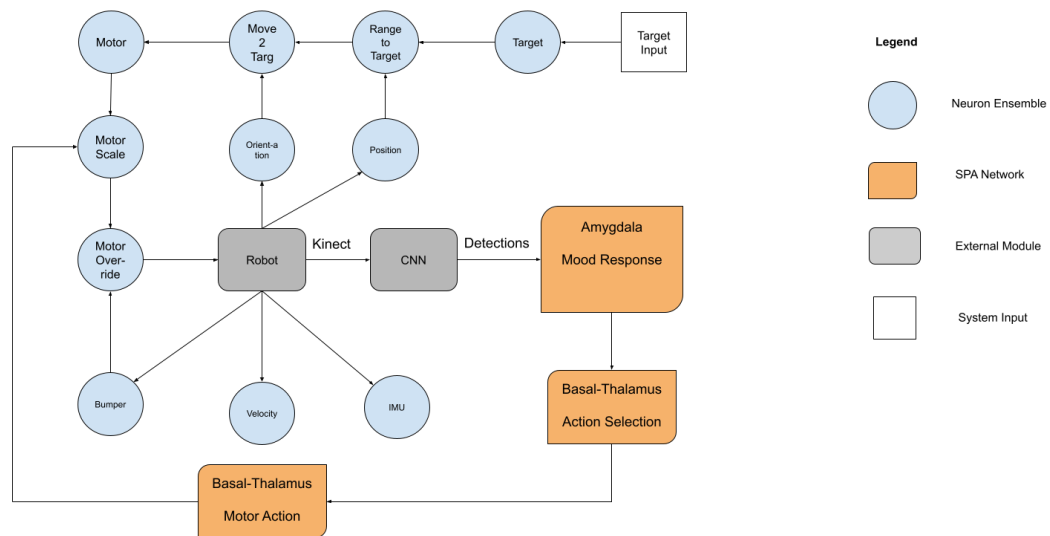


Figure 5.1: Amygdala integrated into the robot model

Changes were made to the radii of neuron ensembles for the target position, robot position, and range to target. Neuron ensembles values are closely bound to the unit circle with a default radius of one. Smaller bounds prevented the robot from knowing the correct information about these parameters. This change allowed for accurate calculations.

Values were adjusted for the action to motor vectors. The values were not orthogonal which created a combined “FEED” and “EXPLORE” response when triggered. Output actions were expected to be only a single item. Vector dimensions were increased, and values changed to fix this and separate the responses. A small script to ensure orthogonality was included in the Nengo robot model should motor scaling values be adjusted in the future.

The integrated model provides a near-seamless migration to a TurtleBot. Messaging from ROS provides most of this simplicity. The Nengo model can operate on the robot or a separate machine. The CNN will operate on the robot instead of a desktop. No code modifications are required for this migration. This provides incredible flexibility and a rapid transition from simulation to live embodiment.

5.3 SIMULATION ENVIRONMENT

Integrated models required a different test environment than proof-of-concept models. Gazebo provides a 3-D simulated environment and allows the robot to operate in created worlds. The virtual TurtleBot operates in Nengo and has subcomponents in Python and C. The CNN image processing unit is also an independent program. It takes images from the virtual camera on the TurtleBot, then classifies and publishes them in ROS messages. The message system from ROS allows the integration of each component. These tools were leveraged from previous research to accelerate testing.

Gazebo worlds were hand-designed to allow for interaction with objects found in the CNN vocabulary. These objects included cars, trucks, and other roadway items. Several environments were created for test purposes. Only one was used for the final simulations.

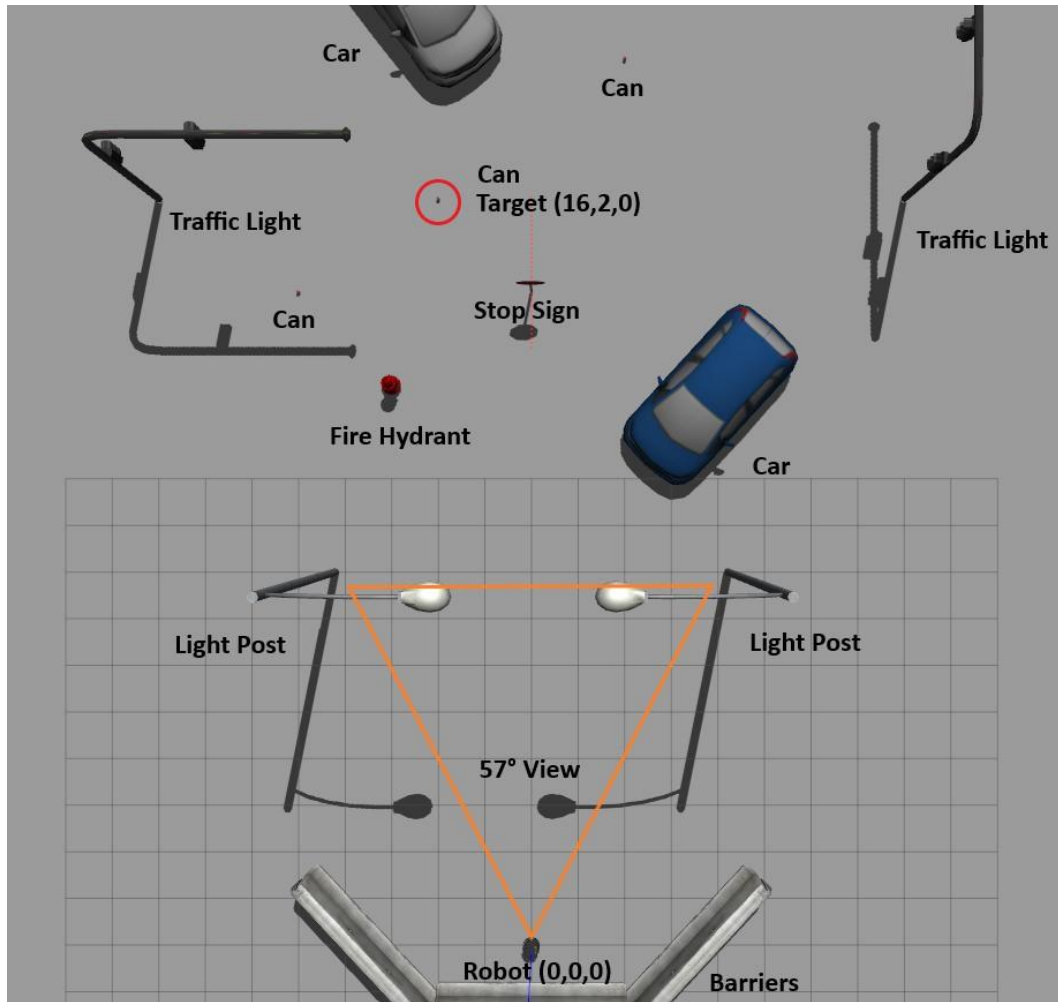


Figure 5.2: Top-down view of developed simulated environment in Gazebo with annotation, actual environment is fully 3D

An overview of the simulation world used for testing is shown in Figure 5.2. The robot is located at the bottom of the figure at the origin. An orange triangle shows the 57° Kinect viewing angle. The triangle also shows the viewing distance. Lamp posts were placed as markers just beyond the viewable limits. All elements of this environment were chosen for a specific purpose. The total number of objects was limited to simplify the interactions and create more predictable behaviors. Objects were spaced with this consideration in mind to prevent immediate interactions. The fire hydrant, stop sign, and

car have unique mappings in each context. This was intentional to observe interactions of “FEAR”, “HUNGER”, and “CURIOSITY”.

One target location was chosen at (16,2,0) for the simulation and hardcoded into the models. This spot allowed for the best interaction with each element. The location is encircled in red in Figure 5.2. This facilitated repeatability across all tests and allowed for testing of expected behaviors. The primary goal for the robot is to reach the target while interacting with the surrounding environment. Mood responses are expected to influence velocities and trajectories.

Context mappings were selected to exhibit different behaviors from the TurtleBot. The context “Highway” was designed to encourage an unhindered response unless the robot would see the fire hydrant. At this point, it was anticipated it might invoke a “FEAR” response. The context “Neighborhood” was created to move quickly to the target. Many of the objects produced a “HUNGER” response, which causes acceleration. The mappings in context “Practice” were intentionally chosen to quickly create “FEAR” responses. These context mapping choices would help elicit behavior changes due to “Motivation”. Tables for contexts maps can be found in appendix section A.1.

Motivations are combined with the output of mood responses from identified objects. Applying the motivation “Encourage” or “Focus” hope to show positive impacts to overcome “FEAR” and continue toward the target. The motivation “Discourage” was excluded from testing due to an increase in “FEAR” that cannot be overcome due to starting objects being out of view. Motivation mappings are found in Table 3.1.

The virtual TurtleBot will move about the environment based upon commands provided by the Nengo model. Gazebo provides information for position, velocity, and

model inputs via ROS. These messages are captured and processed by the Nengo model to provide new motor outputs. The outputs are captured back into Gazebo to create the simulated motion.

Figure 5.3 is an overlay of classified images in Gazebo. The larger image shows a robot and two vehicles. The smaller image shows the classified objects viewed through the Kinect. This image also shows how some objects can be misidentified. In this example, a truck is recognized as a truck and a car. This can create different responses and behaviors based on associative memory mapping.

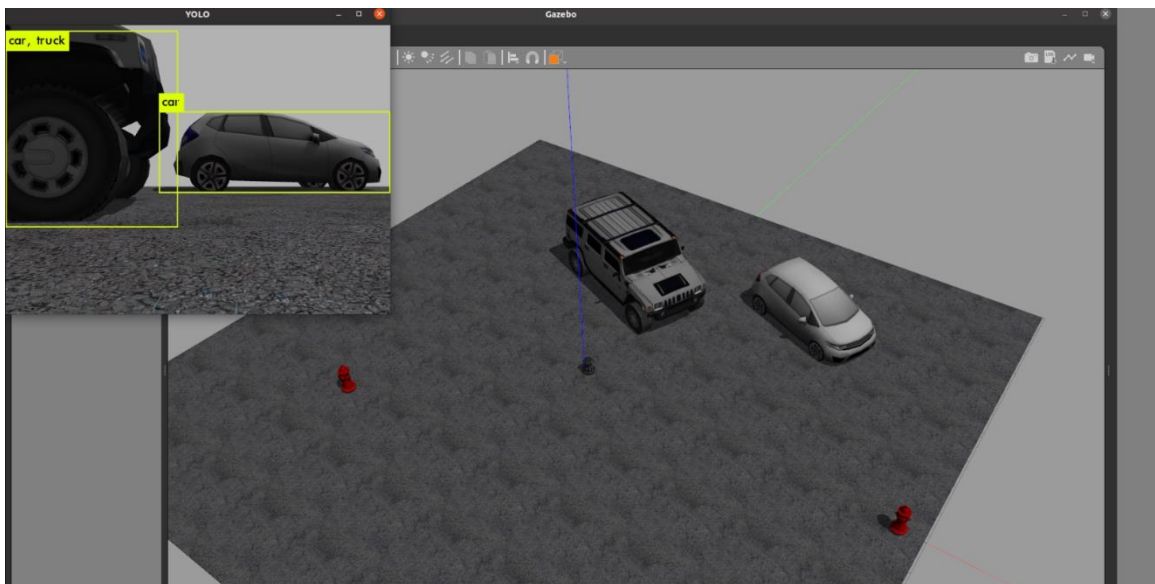


Figure 5.3: Example of image recognition in simulation

Nengo is a slower element in the simulations because spike conversions are calculation intensive. A lock-step method was implemented in the robot model to allow 30ms of Nengo to pass for every 1ms of simulation time. The implementation would pause and unpause the Gazebo environment to allow for the timing differences. This method also created repeatable artifacts. At the start of each simulation, “FEAR” was observed as the largest signal and created an immediate “FLEE” action. When lockstep

was removed, the behavior disappeared. The resolution was to place barriers near the origin in the Gazebo model to prevent the robot from fleeing too far from the beginning.

Creating equal operating conditions between the amygdala and baseline models was important for comparison as well. Since the Model 2 Amygdala can change context, but the Baseline model cannot, the mood mapping in the Baseline model was made equal to the desired context to overcome this challenge for apples-to-apples comparisons. Motivation would remain a feature difference between models. The code bases were made to be identical except for the amygdala-like elements in the baseline.

5.4 LIVE EMBODIMENT

Performing tests given a physical embodiment allows for real-world testing and performance analysis. Figure 5.4 displays the TurtleBot used for these tests. This involved the real TurtleBot 2.0, a Dell workstation for processing Nengo code, an HP laptop to remotely access both devices, and a dedicated router to create an isolated network.

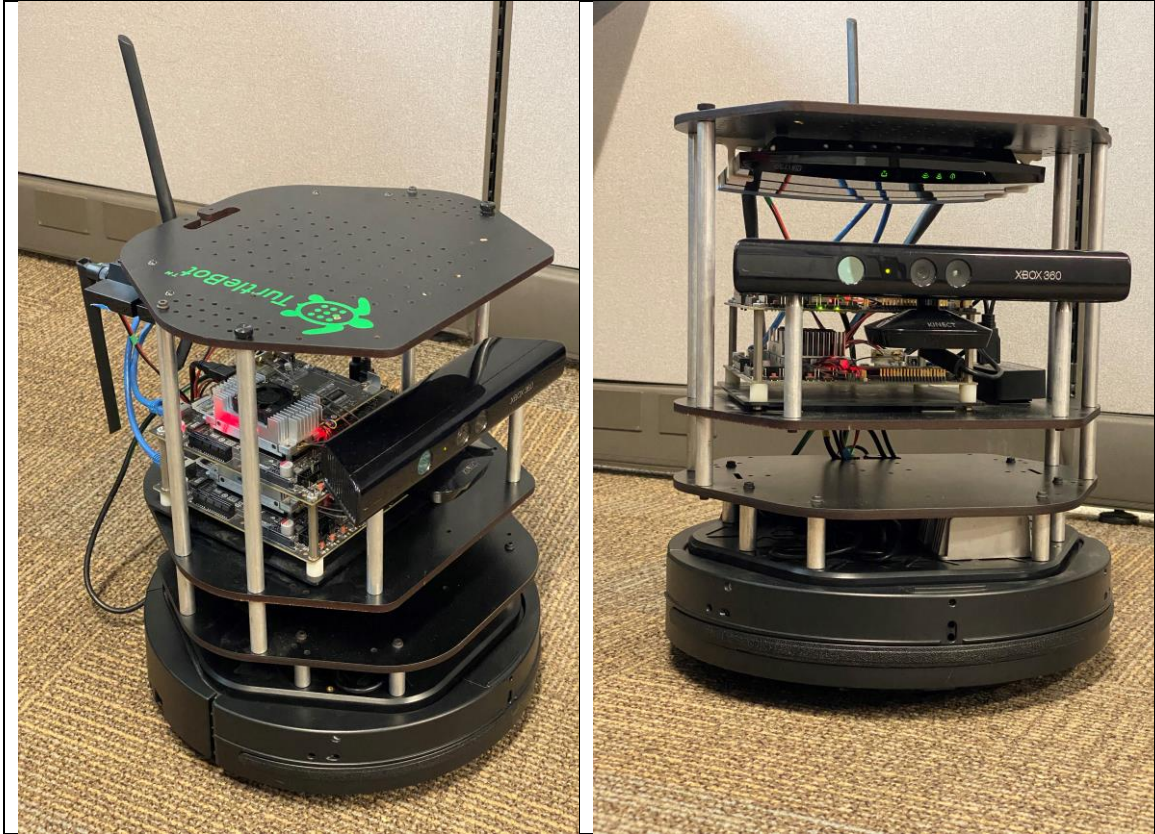


Figure 5.4: TurtleBot 2.0 with TX2 board and Xbox Kinect from the side (left) and front (right)

Code changes were made for environmental adaptation. A building environment was available, and context mappings were constructed to match. Items from the larger CNN vocabulary were also chosen to match the new conditions. Table A.4 and Table A.5 appendix section A.1 detail the room-based context maps. The code port also allowed probes used for metrics to remain the same. Data could be stored with the same method used for Gazebo simulations. This approach allowed for simple transfers after test runs.

A large open area was found to perform testing. A real robot moves faster than the simulated one. The items in the environment remained in their given location with one or two chairs removed. Attempting to replicate the Gazebo model with office equipment would have been ideal. This setup still allowed for interactions with the tables, chairs,

and benches. Several other items, like plants, were identified by the CNN but were not used in the room-based vocabulary.

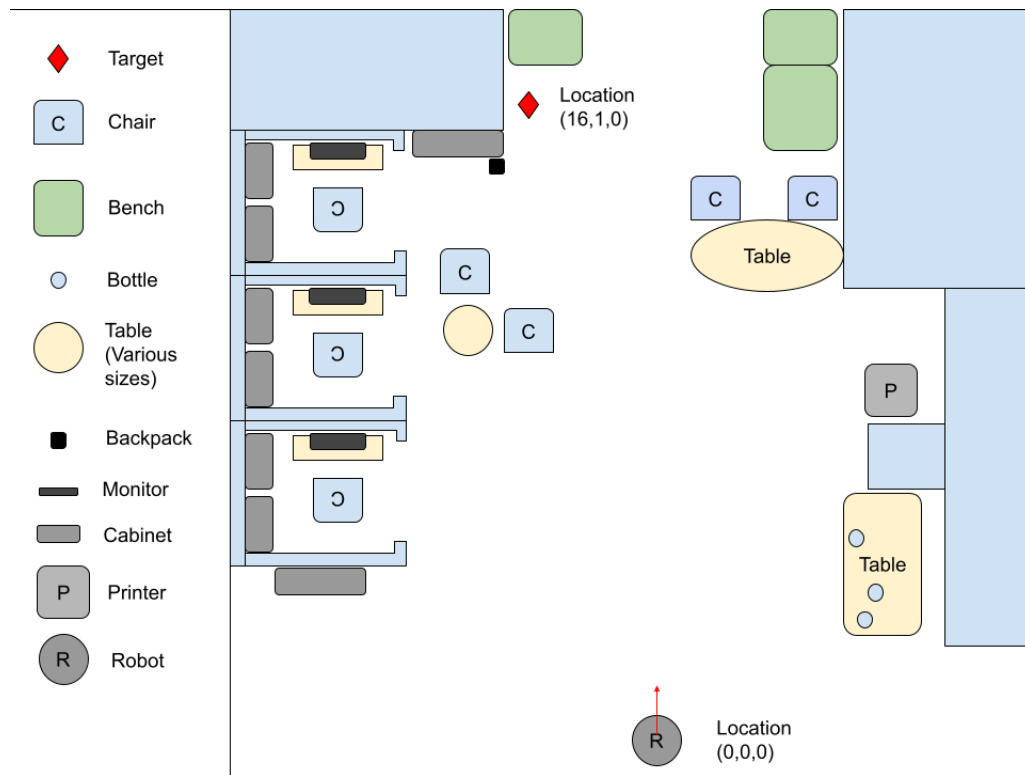


Figure 5.5: Live TurtleBot test environment with target marked with a red diamond

The location narrowed into a hallway which limited locations for a fixed target. One was decided next to an office in front of a bench. Figure 5.5 shows an overview of the area used for live testing with the target location marked with a red diamond. This location is programmed as (16,1,0) which is similar to the (16,2,0) used in the simulation. The red arrow on the TurtleBot shows its starting orientation.



Figure 5.6: View from the TurtleBot's starting location toward the target

Figure 5.6 shows the area from the TurtleBot's starting point. A window frame feature was used to identify the starting location for the robot. Two notches in the side of the TurtleBot were aligned with the color transition of the carpet to allow for easy orientation. A simple command could be executed to reset the robot's position in ROS. This simplified setup and reset for each test.

Test runs were not duplicated as they were in the Gazebo simulations. Two contexts, "Office" and "Living Room", were used in these tests. The motivations "Normal" and "Focus" were used for the Model 2 Amygdala. These were considered sufficient for ANOVA calculations.

Lock-step timing cannot be performed in the same manner in this embodiment. Methods for delaying ROS messages for synchronization exist but were not attempted.

The live embodiment is performed noting the process timing may impact performance and behavior.

6 RESULTS AND ANALYSIS

Simulated integration in Gazebo will demonstrate the interoperability between ROS, the Nengo amygdala model, and a simulated robot. The amygdala's sensory inputs are received from the CNN via ROS messages. The integrated model selects actions that drive the motors based on the amygdala's mood output. The integrated model included context and motivation circuits. This allowed the robot to respond and move about Gazebo according to system inputs.

Once simulations are validated, the software will migrate out of the virtual world and into the embodiment of a real TurtleBot. This will be executed on a Jetson TX2 board to process images from an Xbox Kinect camera. The Nengo models were run remotely on the Dell workstation. Both systems were connected via ROS maintaining the TurtleBot as the ROS master.

6.1 METRICS AND FRAMEWORK

New metrics were created for simulation and real-world embodiment. These metrics are target quality, ANOVA calculations, and timeliness. Simulation and embodiment results of the new amygdala versus the baseline design were then compared.

Target quality evaluated if the robot reached the target and how quickly. Mood responses are expected to impact the robot's trajectory. The impact is more important to observe than reaching the target. Test runs were checked for repeatability and accuracy. Nengo ensembles and modules recreate the weights and biases during each execution. Different outcomes could occur as a result. These can be set to fixed values to prevent this feature but were not for these tests.

Timeliness includes the zone of object interaction, behavioral response timing, and motor response. The Kinect camera has a limited range and viewing angle. Object placement in the simulation is designed to prevent immediate detections. The detection must also exceed the context memory threshold of 0.55 to properly activate a mood response. Timing starts at this level. The impacts of model complexity are determined by the propagation speed from object input to mood output.

6.2 EMBODIMENT: M&S AMYGDALA VS. BASELINE

Two robot models can respond to objects based upon an applied context and motivation. The impact based on these three factors is needed. Only one world was used for testing and is not considered a factor. Target quality captures the desired response data. These metrics are the nearest distance to the target and the time for the nearest distance.

A test matrix was created to determine context and motivation selection as seen in Table 6.1 which lists the test with the model, context, and motivation used for each run.

These elements were selected to observe different mood responses based on context and the impacts of motivation on those same contexts. This resulted in nine simulations for the amygdala model and three configurations for the baseline model.

Table 6.1: Simulation Test Matrix

Run	Model	Context	Motivation
Amygdala Run 1	Amygdala Model 2MC	Highway	Normal
Amygdala Run 2	Amygdala Model 2MC	Highway	Encourage
Amygdala Run 3	Amygdala Model 2MC	Highway	Focus
Amygdala Run 4	Amygdala Model 2MC	Neighborhood	Normal
Amygdala Run 5	Amygdala Model 2MC	Neighborhood	Encourage
Amygdala Run 6	Amygdala Model 2MC	Neighborhood	Focus
Amygdala Run 7	Amygdala Model 2MC	Practice	Normal
Amygdala Run 8	Amygdala Model 2MC	Practice	Encourage
Amygdala Run 9	Amygdala Model 2MC	Practice	Focus
Baseline Run 1	Baseline	Highway	Normal
Baseline Run 2	Baseline	Practice	Normal
Baseline Run 3	Baseline	Neighborhood	Normal

Four replications of each experiment combination in Table 6.1 were run to provide an understand of performance and to test repeatability. In AI, such approaches are often critical due to the implicit stochasticity of algorithms [34] , for example Nengo creates new weights and biases for neurons in each execution. The seeds for these are random unless the value is specified. Keeping the randomness of the neurons was considered desirable to allow for deviations in behavior. These deviations seemed more natural for cognition.

Statistical analysis was applied to the simulation results using JMP (JMP 16.2, SAS, Cary, NC). In JMP, Analysis of Variance (ANOVA) was computed for model,

context, and motivation to determine if any provide a significant impact on the performance of the nearest distance and nearest time. The factors were measured individually, and each shows statistical significance based on p-values associated with the ANOVA hypothesis test, i.e. at least one group is different in response than the others. Single factor ANOVA results are presented in Table 6.2 for nearest distance and each factor from Table 6.1. Model, Context, and Motivation all had statistically significant impact on nearest distance at the 5% level. However, Context without including “Practice” run results does not show a statistically meaningful influence on distance. Additionally, while not apparent in these results though context appears to have the greatest impact it doesn’t necessary improve performance but alters the way the robot behaves. This impact will be considered to provide alteration and not improvement. Additionally, the results for context include the “Practice” context environment, which was expected to impact the behavior of the robot. Re-evaluating without “Practice” suggests context is not as significant. It is important to keep this environmental context included in the results because the concept being applied is complex and should not be normalized.

Table 6.2: Oneway ANOVA applied to the nearest distance

Single Factor ANOVA Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	1	141.7734	141.773	6.7386	0.0126
Error	46	967.7874	21.039		
Total	47	1109.5608			
Context	2	432.3959	216.198	14.3671	<.0001
Error	45	677.1648	15.048		
Total	47	1109.5608			
Motivation	2	251.6678	125.834	6.6005	0.0031
Error	45	857.8930	19.064		
Total	47	1109.5608			
Context (without Practice)	1	4.59218	4.59218	1.6558	0.208
Error	30	83.201439	2.77338		
Total	31	87.793618			

Comparing Model and Context

Evaluating interactions between factors required multiple ANOVA. The Baseline model doesn't support the motivation feature, thus no baseline with motivation data was available. This difference created a problem with the ANOVA calculations and forced separated analyses. First, the model and context would be compared with the default motivation, "Normal". All other motivations are excluded. For this, the dependent variables of nearest distance and nearest time were both used, with ANOVA performed twice, with the independent variables of Model and Context.

The models were expected to have a different response due to results seen in Table 6.2. From this ANOVA, Figure 6.1, provides the effects of the model and context on nearest distance. The results in the ANOVA table show that the model itself is statistically significant (at $p < 5\%$) for the hypothesis test that at least one combination yields a different result than the others [35]. This model further explains more than 90% of the variance explained (R^2) and the adjusted R^2 presents a similar result indicating that the multiple factors are not biasing the raw R^2 [35]. The effects test table lists the

constituent parts of the ANOVA Model with further considerations for null hypotheses on the Model, Context, and Model by Context [36]. These results show that the Model variable has a reasonable impact on the performance (the hypothesis is rejected at 10%), however, Context is shown to provide a significant impact at $p < 5\%$. The combined response is less significant than the individual factors.

Whole Model					
Summary of Fit					
RSquare			0.940547		
RSquare Adj			0.924032		
Root Mean Square Error			1.639652		
Mean of Response			4.906717		
Observations (or Sum Wgts)			24		
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Ratio	
Model	5	765.56151	153.112	56.9517	
Error	18	48.39223	2.688		Prob > F
C. Total	23	813.95374			<.0001*
Effect Tests					
Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Model	1	1	11.67430	4.3424	0.0517
Context	2	2	741.69121	137.9399	<.0001*
Model*Context	2	2	12.19600	2.2682	0.1323

Figure 6.1: Multiple ANOVA of model and context for simulation results on nearest distance

Figure 6.2 was computed which shows significant differences between models in the “Highway” context but not in the others. However, both the “Neighborhood” and “Practice” contexts have confidence intervals of 95% which overlap the group means for both Baseline and Amygdala Model 2MC, indicating no statistical difference in results.

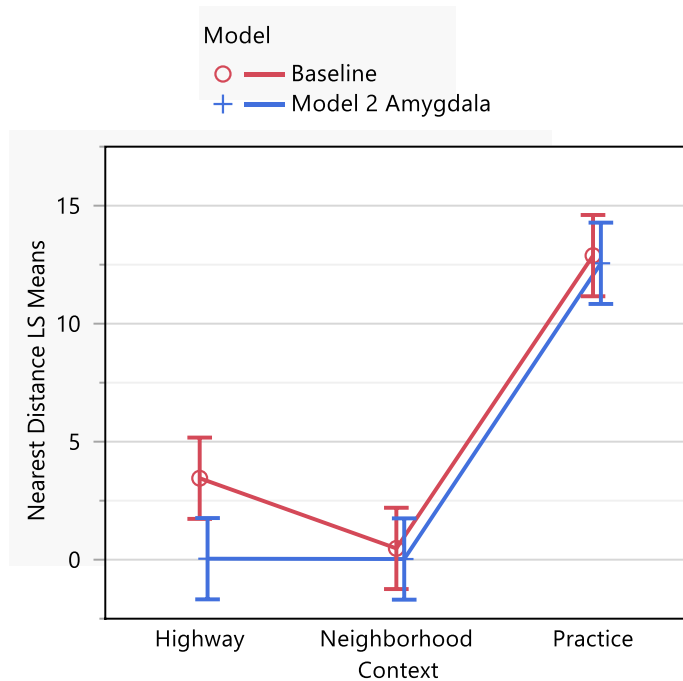


Figure 6.2 Model performance given context showing least squares mean for the nearest distance

Table 6.3 shows the least squared means for both nearest distance and nearest time. The results are paired by Baseline and Amygdala respective of context for easier one-to-one comparison. The results for nearest distance reveal the Amygdala Model 2MC always performed better than the Baseline. The Baseline was always outperformed on time except in the “Highway” context.

Table 6.3: Least squares mean given model and context given model and context

Level	LSM Distance	Std Error	LSM Time	Std Error
Baseline, Highway	3.45	0.82	2.29	0.26
Amygdala Model 2MC, Highway	0.04	0.82	2.43	0.26
Baseline, Neighborhood	0.48	0.82	3.46	0.26
Amygdala Model 2MC, Neighborhood	0.03	0.82	2.82	0.26
Baseline, Practice	12.88	0.82	0.81	0.26
Amygdala Model 2MC, Practice	12.56	0.82	0.80	0.26

Figure 6.3 provides the effects of the model and context on nearest time. The ANOVA show the whole model has a significant impact at $p < 5\%$. Additionally, in the Effects Test, context is shown to provide a significant impact, but model and the combined response are not significant. From the Summary of Fit in Figure 6.3, the model explains 83% of the variance as seen in the R^2 . The expanded components of the ANOVA table are shown in the Effects Tests table, seen in Figure 6.3. Here, the constituent parts of the Model are listed. This section displays the individual sum of squares, which sum to the model sum of squares in the ANOVA table, and significance of each feature.

Whole Model					
Summary of Fit					
RSquare		0.828329			
RSquare Adj		0.780642			
Root Mean Square Error		0.51933			
Mean of Response		2.101833			
Observations (or Sum Wgts)		24			
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Ratio	
Model	5	23.424167	4.68483	17.3703	
Error	18	4.854668	0.26970		Prob > F
C. Total	23	28.278835			<.0001*
Effect Tests					
Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Model	1	1	0.178883	0.6633	0.4261
Context	2	2	22.538847	41.7844	<.0001*
Model*Context	2	2	0.706438	1.3097	0.2944

Figure 6.3: Multiple ANOVA of model and context for simulation results on nearest time

Figure 6.4 shows the largest differences the “Neighborhood” context. The Baseline and Model 2MC 95% confidence intervals overlap by nearly 50%. The “Practice” context with a “Normal” motivation yields a high “FEAR” response. The

robot is expected to flee in either model, which creates a high distance to the target. This behavior causes the nearest distance to occur early in the simulation, appearing as a fast speed in reaching the target.

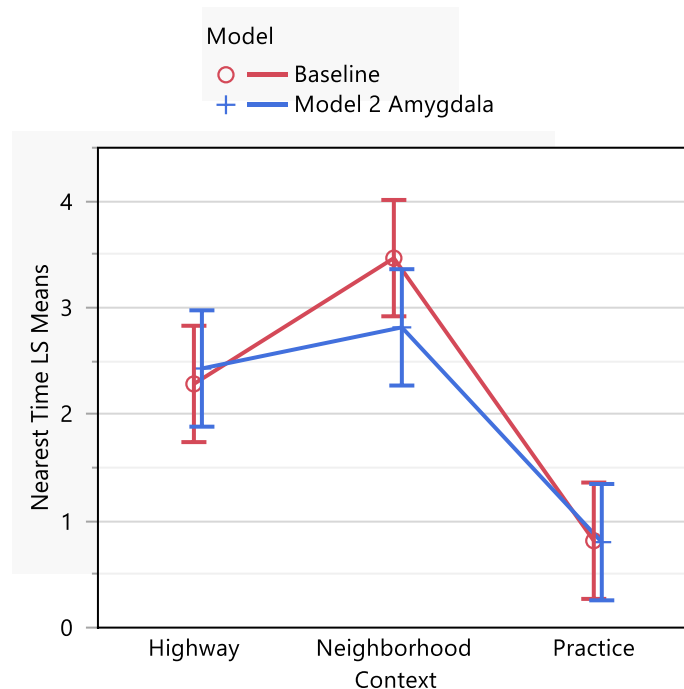


Figure 6.4: Model performance given context showing least squares mean for the nearest time

Amygdala Model 2MC always performed better than the baseline with one exception. Since the goal was reaching a target, then the nearest distance is considered more valuable for rejecting the null hypotheses. Increased performance and value are proven between the Amygdala Model 2MC and the Baseline based upon the rejection of H_{0m} and impacts of altered behavior based upon the rejection of H_{0c} .

6.2.1 COMPARING CONTEXT AND MOTIVATION

For a test comparing the effects of context and motivation a null hypothesis stating the context, motivation, or context and motivation don't make a difference. For this application, the typical ANOVA alternative hypothesis stating that a single factor or combination thereof does make a difference, was used [36]. For this, the dependent

variables of nearest distance and nearest time were both used, and the ANOVA was performed twice.

Figure 6.5 provides the effects of the model and context on nearest time. The ANOVA show the whole model has a significant impact at $p < 5\%$. According to the Effects Test, context, motivation, and context-cross-motivation are all shown to provide a statistically significant impact on nearest time. From the Summary of Fit in Figure 6.5, the model further explains 94% of the variance as seen in the R^2 . The expanded components of the ANOVA table are shown in the Effects Tests table, seen in Figure 6.5. Here, the constituent parts of the Model are listed.

Whole Model					
Summary of Fit					
RSquare			0.940366		
RSquare Adj			0.922697		
Root Mean Square Error			1.13571		
Mean of Response			1.635207		
Observations (or Sum Wgts)			36		
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Ratio	
Model	8	549.16332	68.6454	53.2203	
Error	27	34.82558	1.2898		Prob > F
C. Total	35	583.98890			<.0001*
Effect Tests					
Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Context	2	2	184.64451	71.5767	<.0001*
Motivation	2	2	121.56869	47.1256	<.0001*
Context*Motivation	4	4	242.95013	47.0893	<.0001*

Figure 6.5: Multiple ANOVA for context and motivation on simulation data for nearest distance.

Figure 6.6 was computed and shows significant differences given the context “Practice” and all motivations. The “Normal” motivation is shown as vastly different from “Encourage” and “Focus”. However, “Encourage” and “Focus” are significantly

differently with minor overlaps in their 95% confidence intervals. The “Highway” and “Neighborhood” contexts indicate no statistical difference in results.

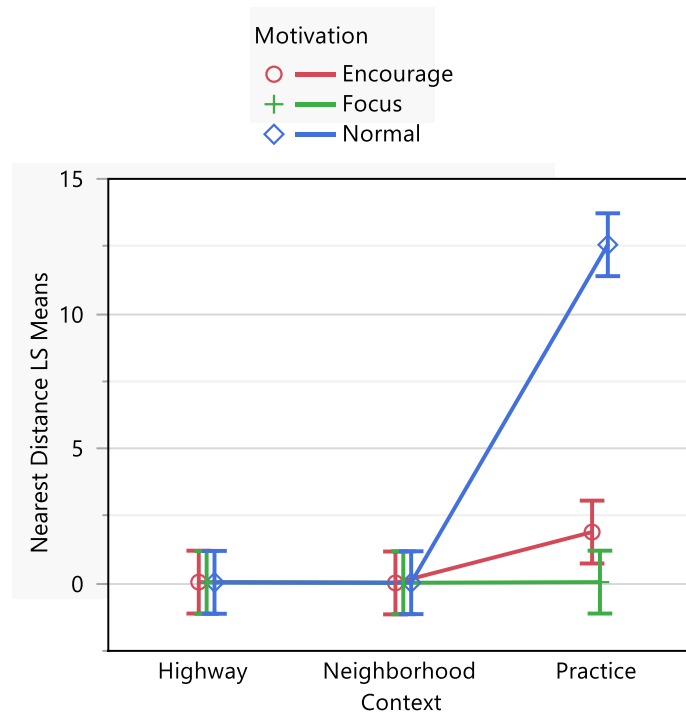


Figure 6.6: Model performance given context and motivation on nearest distance.

Figure 6.7 provides a graphic observation on the impacts of motivation. Robot movement is captured from the sets of different runs. All use the same context of “Practice” but change the motivation. Runs A7 and B3 have “Normal” motivation and don’t reach the target. Some cases of “Encourage” reach the goal while others only get close. Using “Focus” as a motivation always reaches the target. Paths are shown to exceed past the target’s location but are not considered important for these tests.

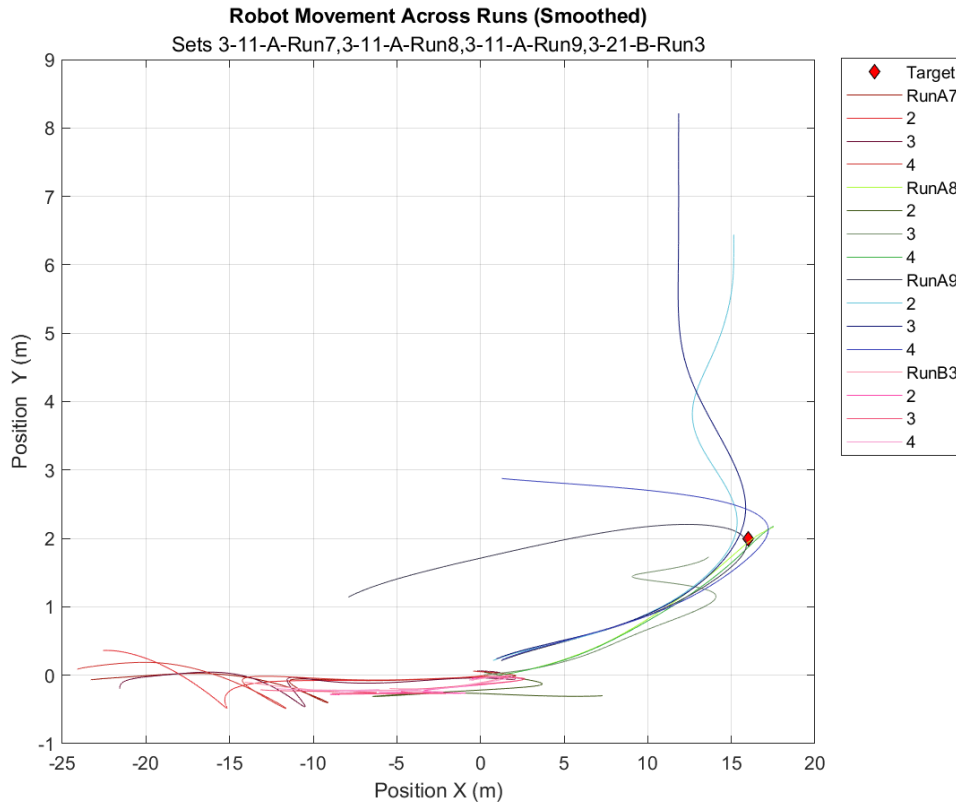


Figure 6.7: Overlay of robot movements given "Practice" context

Table 6.4: Least squares mean given model and context given context and motivation

Level	LSM Distance	Std Error	LSM Time	Std Error
Highway, Encourage	0.05	0.57	2.31	0.15
Highway, Focus	0.04	0.57	1.58	0.15
Highway, Normal	0.04	0.57	2.43	0.15
Neighborhood, Encourage	0.02	0.57	2.82	0.15
Neighborhood, Focus	0.03	0.57	1.67	0.15
Neighborhood, Normal	0.03	0.57	2.82	0.15
Practice, Encourage	1.91	0.57	2.86	0.15
Practice, Focus	0.05	0.57	1.63	0.15
Practice, Normal	12.56	0.57	0.80	0.15

Figure 6.8 provides the effects of the model and context on nearest time. The ANOVA show the whole model has a significant impact. According to the Effects Test, context, motivation, and context-cross-motivation are all shown to provide a statistically

significant impact. From the Summary of Fit in Figure 6.8, the model explains 87% of the variance as seen in the R^2 . The expanded components of the ANOVA table are shown in the Effects Tests table, seen in Figure 6.8. Here, the constituent parts of the Model are listed. This section displays the individual sum of squares, which sum to the model sum of squares in the ANOVA table, and significance of each feature.

Whole Model					
Summary of Fit					
RSquare		0.873952			
RSquare Adj		0.836604			
Root Mean Square Error		0.296821			
Mean of Response		2.101111			
Observations (or Sum Wgts)		36			
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Ratio	
Model	8	16.493145	2.06164	23.4005	
Error	27	2.378767	0.08810		Prob > F
C. Total	35	18.871912			<.0001*
Effect Tests					
Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Context	2	2	2.7188194	15.4299	<.0001*
Motivation	2	2	6.6007434	37.4606	<.0001*
Context*Motivation	4	4	7.1735818	20.3558	<.0001*

Figure 6.8: Multiple ANOVA for context and motivation on simulation data for nearest time.

Figure 6.9 was computed and shows the motivations “Normal” and “Encourage” are nearly identical in “Highway” and “Neighborhood” contexts but are significantly different in the “Practice” context. The “Focus” motivation is shown to be significantly different than the other motivations. This motivation also indicates the robot reached the target faster and validates the speed increase as an expected response.

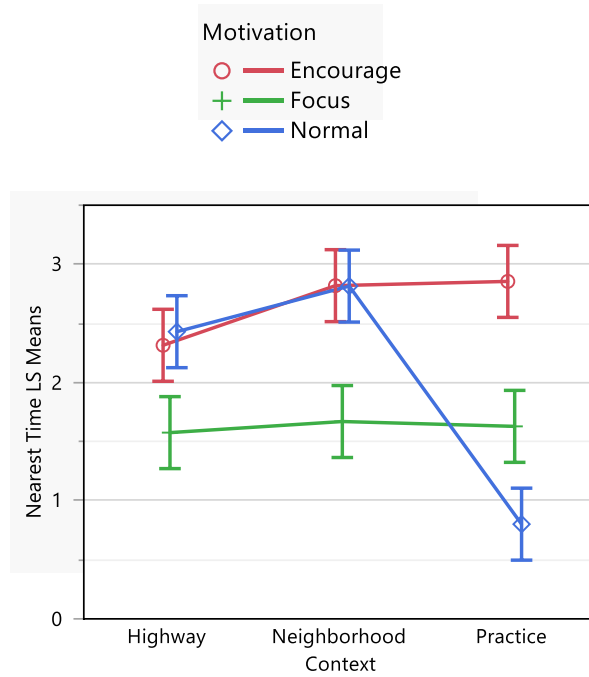


Figure 6.9: Model performance given context and motivation on nearest time.

The ANOVA results validate the expected changes to the robot's behavior due to motivation given a specific context. All null hypotheses, H_{0c} , H_{0m} and H_{0mc} , are rejected. Mimicking biological features from the amygdala is proven to be useful.

6.2.2 SIMULATION ERRORS

Some simulations revealed some erroneous positional information. The robot was observed colliding into an object, whether moving forward or in reverse, yet the data suggests the robot traveled much further in that direction. Figure 6.7 shows an example of this across different runs. The reason behind this is not understood. Positional information is sent by Gazebo and was expected to line up with the model's location. None of these outcomes impacted the overall results.

6.3 EMBODIMENT: LIVE SYSTEM DEMONSTRATION

The Amygdala Model 2MC has proved itself in simulation. Performance comparison is achieved on a real robot. The same factors and responses are considered in this application. A reduced test matrix was created. The contexts changed due to the location, and only two were used for testing. Test runs were not iterated. Table 6.5 outlines the tests performed.

Table 6.5: Live embodiment test matrix

Run	Model	Context	Motivation
Amygdala Run 1	Model 2 Amygdala	Office	Normal
Amygdala Run 2	Amygdala Model 2MC	Office	Encourage
Amygdala Run 3	Amygdala Model 2MC	Office	Focus
Amygdala Run 4	Amygdala Model 2MC	Living Room	Normal
Amygdala Run 5	Amygdala Model 2MC	Living Room	Encourage
Amygdala Run 6	Amygdala Model 2MC	Living Room	Focus
Baseline Run 1	Baseline	Office	Normal
Baseline Run 2	Baseline	Living Room	Normal

Statistical analysis was performed on the simulation results using JMP (JMP 16.2, SAS, Cary, NC). Analysis of Variance was computed for model, context, and motivation to determine if any provide a significant impact on the performance of the nearest distance and nearest time. The factors were measured individually, but only model shows some significance within the 10% band based on p-values associated with the hypothesis test under consideration. The Oneway ANOVA results in Table 6.6 show that Model provides a statistically significant impact on nearest distance at $p < 10\%$ while the other variables do not provide a statistically meaningful impact on nearest distance results. The Oneway ANOVA results in Table 6.7 show that only motivation provided a statistically

meaningful impact on nearest time with a p-value near 10%. The other variables do not provide a statistically meaningful impact. Overall, the probabilities suggest testing on the live embodiment fails to reflect the positive changes found in the simulations.

Table 6.6: YX Fit on single factors applied to nearest distance

Single Factor ANOVA Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	1	68.73061	68.7306	5.2427	0.062
Error	6	78.65847	13.1097		
Total	7	147.38908			
Context	1	34.11235	34.1124	1.8069	0.2275
Error	6	113.27673	18.8795		
Total	7	147.38908			
Motivation	2	63.48954	31.7448	1.8918	0.2445
Error	5	83.89954	16.779		
Total	7	147.38908			

Table 6.7: YX Fit on single factors applied to nearest time

Single Factor ANOVA Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	1	0.225816	0.22582	0.0492	0.8318
Error	6	27.544968	4.59083		
Total	7	27.770784			
Context	1	4.461085	4.46108	1.1483	0.3251
Error	6	113.27673	18.8795		
Total	7	147.38908			
Motivation	2	16.393266	8.19663	3.6021	0.1074
Error	5	11.377518	2.27550		
Total	7	27.770784			

Figure 6.10 shows robot movement for the Amygdala Model 2MC with the “Living Room” context applied. Run A4 is shown to move toward the target and then retreat and retry when using the “Normal” motivation. Runs A6 and A6 use “Encourage” and “Focus” respectively and do not exhibit the “Flee” behavior. They never reach the target, as shown in the plot, because they both collided with a chair and were stuck. The

tests were kept showing the responses did change but the rotational control was incorrect. Visually this can help validate changes that aren't revealed by the statistics.

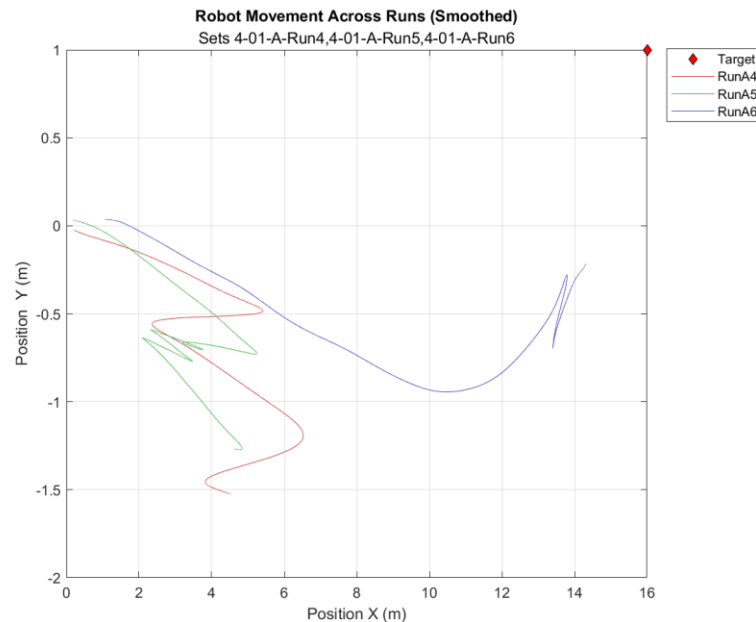


Figure 6.10: Robot movement of Amygdala Model 2MC with “Living Room” context, all motivations

Excluding the robot, numerous differences exist between the simulation and the real world. An ideal situation would allow identical setups. This duplication wasn't possible. The concept of contexts and motivations can be applied in diverse ways. These features were intended to mimic cognitive abilities for behavioral responses. It is not unreasonable to expect someone to act in identical ways in different environments. Statistically, it would suggest insignificance until presented with a situation that could reject the hypothesis. The differences may not prove valuable in the given real-world testing. However, stating the Amygdala Model 2MC cannot show value or differences in the world is believed to be erroneous and require further testing.

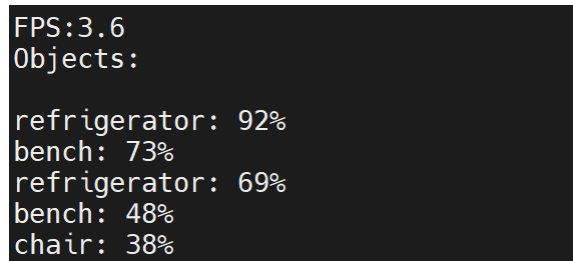
6.3.1 IMPLEMENTATION CHALLENGES

A variety of implementation challenges were discovered and addressed in this work as well. These included timing, CNN identification issues, and movement issues. Each was handled in a logical and straightforward way to ensure that the amygdala concepts under test could be effectively studied.

6.3.2 IDENTIFICATION CHALLENGES

A further implementation and integration issue involved the Kinect-based images and detections, per the Figure 5.1 process flow chart. The challenge with these components of the process were identified early in live testing. These involved real-world environments contained a higher number of recognizable objects than expected which was inconsistent with the more limited amount used in simulation.

The practical aspect of this effect was that the CNN recognized more objects than the model could process. As shown in Figure 6.11 five objects are detected; however, the Nengo robot model is only capable of receiving three inputs. Not resolving this adds an element of randomness based on the certainty of the detected object. Objects with lower certainties didn't register. Anything outside of the vocabulary subset was also ignored.



```
FPS:3.6
Objects:
refrigerator: 92%
bench: 73%
refrigerator: 69%
bench: 48%
chair: 38%
```

Figure 6.11: Console showing the number of detections in the real world exceeded the limit of three

Another challenge was the number of objects that could be misidentified. The bench object was considered a particular furniture item. Sometimes the bench would register as a chair or sofa. For example, a file cabinet would frequently be identified as a

refrigerator. Although the word was not part of the vocabulary subset and would be an issue if the detected objects were overloaded.

6.3.3 TIMING ANALYSIS CHALLENGES

Critical to implementation is the timing and routing of signals for decision making.

Measuring the timing between object input to mood output is valuable to determine propagation delays due to added neuron count and complexity. These measurements were attempted for both simulation and live embodiment. The methods didn't provide useful results. Signals were difficult to correlate and appropriate timing between events could not be calculated.

Figure 6.12 shows object detections in the first row. The horizontal line at 0.55 indicates the threshold at which the associative memory will acknowledge the signal. Mood responses based on the inputs are shown in the second row. Vertical marker lines were added to find the moods matching input signals. This method was cumbersome and tedious.

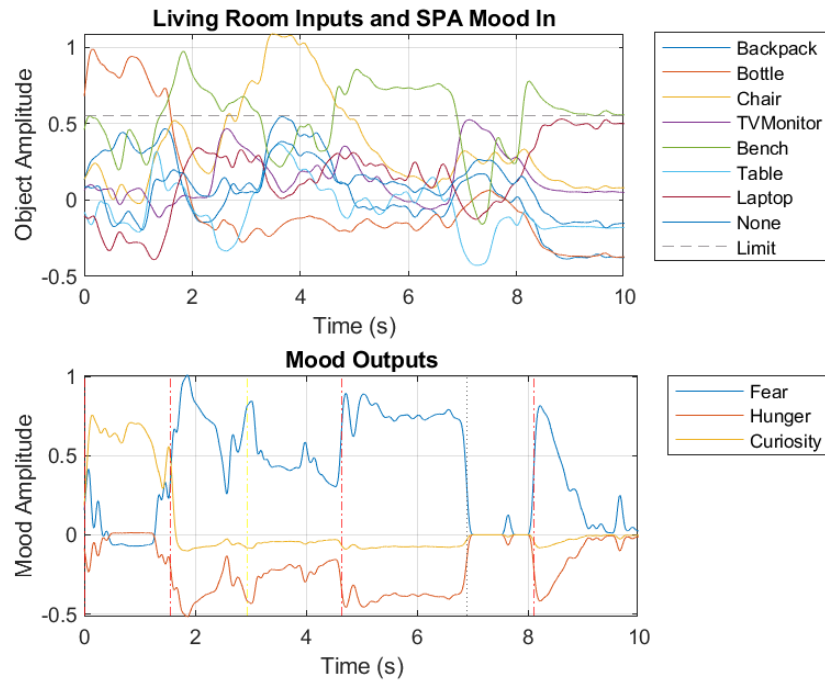


Figure 6.12: Object and mood timing information from live embodiment Run A6

A script was created in MATLAB to find the maximum signal and the peak times. Maximum signals became pulses with heights equal to their vector position. Converting the signals into discrete pulses allowed for quick peak detection. Figure 6.13 is an example of the pulse-formed data. This method allowed a cleaner approach to finding timing responses. Peak calculations performed on both data sets attempted to eliminate the need for manual searches.

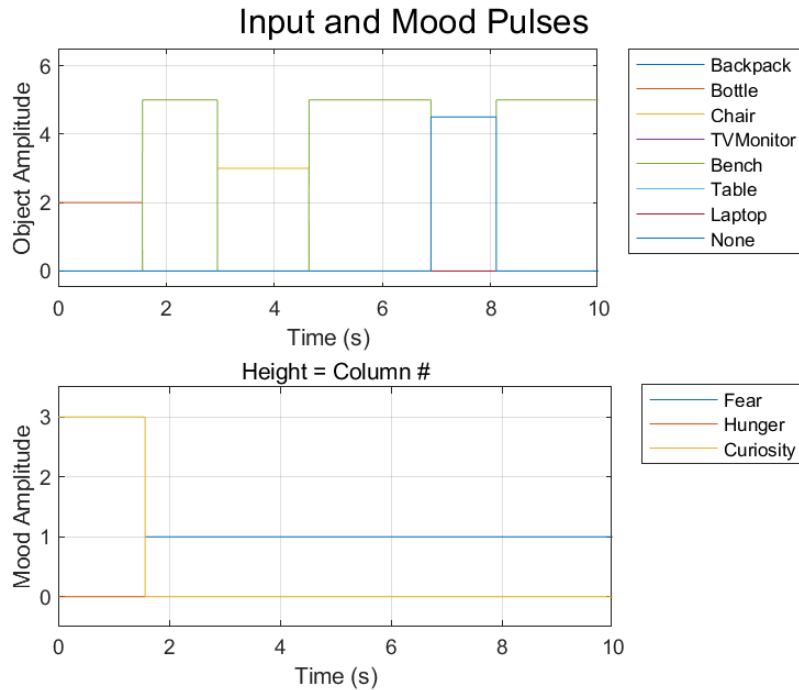


Figure 6.13: Object and mood pulses calculated from the live embodiment Run A6

Overall, the attempt proved problematic and was abandoned. No results were captured from this, which leaves a gap in understanding how the model impacted timing. A likely cause of this failure is the oversight of the type of associative memory used in context unbinding circuits. of context associative memory blocks. The calculations kept considering a WTA style and not thresholding, so the effects of multiple inputs weren't properly considered.

6.3.4 ROBOT MOVEMENT CHALLENGES

Orientation was not changed, and the radius remained at 4. This seemed sufficient for values of 2π or $\pm \pi$. The robot had difficulties rotating toward values more than 75° . The math and neuron connections were verified as correct. This forced robot to narrow and forward only pathways.

6.3.5 IMPACTS ON RUN TIMES

The amygdala module is larger and is expected to impact simulation times and response times from object detection to mood output and motor drive. These times can provide valuable performance metrics. The baseline model is the original model and does not include additional elements for motivation or context and established a reference for comparison and a minimum expected response time. The averaged results are found in Table 6.8.

Table 6.8: Summary of simulation timing				
Model	Embodiment	Neuron Count	Avg Time (s)	Standard Deviation (s)
Baseline	Gazebo	14200	224.61	8.66
Amygdala Model 2MC	Gazebo	104600	296.72	9.54
Baseline	Live	14200	33.65	1.99
Amygdala Model 2MC	Live	104600	114.87	3.12

The neuron count in the amygdala model increased by 630%. Simulation run times increased by 30%, and live embodiment run times by 241.37%. The positive impacts of the Amygdala Model 2MC suggest the value provided is greater than the increase in simulation time. Other challenges inhibit proper evaluation on features and performance versus run time.

Capturing probe data in Nengo, see Figure 4.1 , was a consideration for a slower simulation time. The Amygdala Model 2MC captures 38 different points whereas the Baseline model captures 24. A simple test removing all probes from Model 2 proved this had no impact on the simulation run times.

An adverse effect of the simulation run times must be noted. The baseline model always operated quicker than the amygdala model. Distance to target was always higher on the baseline models. Since the robot motors were configured to be the same, the only

difference is operation time. As a result, the faster execution prevented the robot from having enough time to reach the desired targets.

Synchronization between ROS, Nengo, and the TurtleBot is a known issue.

Providing a lock-step timing solution as used in simulation may yield better results.

Another possible solution is neuromorphic hardware, like an Intel Loihi, to allow rapid execution of amygdala models.

7 CONCLUSIONS AND FUTURE RESEARCH

7.1 SUMMARY OF MODEL RESULTS

This research developed computational models of the biological amygdala and applied them to a robotics decision and control problem. With this amygdala, the use of motivation, mood, and context was studied in a robotics implementation. First, this work created a foundational understanding of how to mimic the behavior of a simple biological amygdala. Next, this work developed successive models of the amygdala and implemented them in code and simulation to find a proof of concept for a robotics implementation. Finally, this was demonstrated on a live robot in a real-world environment. Both the simulation and live robot testing were conducted using an experimental design framework to understand the statistical implications of the results. Results demonstrated that the proposed amygdala model provided a simulation improvement of 70.8% ($p < 0.006$, t-test, $\text{dof}=1$) in nearest distance, 4% improvement in nearest time ($p < 0.387$, t-test, $\text{dof}=1$) and real applications improvements of 62.4% in nearest distance ($p < 0.031$, t-test, $\text{dof}=1$) over the baseline.

In practice, this thesis developed a baseline understanding of the amygdala in code; this model “Model 0” proved to be complex, slow, and inaccurate and it also proved simulation timing is not directly correlated to the number of neurons. Next, a more refined version, “Model 1” was developed as a simpler implementation; but it required five times more neurons to operate as expected. Additionally, the additional neural mechanics made Model 1 cumbersome, slow to operate, and improperly performed the weighted actions. This model was deemed invalid, and the work was discontinued. Finally, leveraging the understanding gained from the earlier

implementations, “Model 2” was developed to provide a method to emphasize inputs. Model 2’s simulation was the second-fastest but contained some accuracy issues. Model 2MC added motivation and context into Model 2 to model amygdala inputs. Noise from context unbinding created issues but resolved by changing threshold values. This improved the accuracy to 100%.

The addition of motivation proved to influence the overall behavior of Model 2. The development of context offered a new way to interact with the environment. The multiplexing nature was proven to filter information unrelated to the applied context. This method demonstrated the amygdala model produces different moods from objects in different contexts. These features mimic an amygdala and accomplish the first objectives.

Integrated simulations with Gazebo, Nengo, and ROS revealed the new Model 2 Amygdala performed better than the baseline. The new features of context and motivation proved valuable additions. The increase in simulation time was minor relative to the massive increase in neuron count. The model also allows for further research into expanded or new features.

Testing the live embodiment showed differences in simulation versus real-world application. Some of the challenges were due to conditions that were not ideal and other unexpected control-related behaviors. Statistically, the results only suggest the model differences are relevant. The context and motivation factors were not. Environments are dynamic and may not always yield significantly different results, as shown with “Highway” and “Neighborhood” contexts. Due to the broad application being introduced by the amygdala structures, the results are considered inconclusive with further research and testing required.

7.2 FUTURE RESEARCH

Brains respond to multiple sensory inputs at one time. Adding mixed inputs such as sound and touch create layers of complication while offering enhanced responses and interactions. A great example of this would be the sound of a fire alarm to trigger a new context and a new way of interacting with objects. This could be implemented within the robot by adding new sensors but was not feasible in the scope of this project.

Additional features of a biological amygdala could be added to this model. Constructing circuits to handle fear and anxiety is one portion. Valence controls the capability for reaction. This circuit could inhibit responses in particular situations. Existing features could be further enhanced.

A single motivator was applied during the simulation. Studies can be created to measure the impacts of changing motivation due to different internal or external factors. Motivation can change the psychology and response to actions. These influences could be used to train or retrain a given response to a stimulus. For example, an object created a fear response but if encouraged the robot learned not to be afraid. Motivation input is shown to come from the substantia nigra pars compacta/ventral tegmental area and raphe nucleus [13]. There might also be a consideration to separate this into a separate model.

Only a single context was applied during the simulation. Changing contexts during operation would allow for the demonstration of changing environments. A robot changing from a neighborhood onto a highway could detect key features and then choose to interact with a different context.

Context hierarchy provides greater descriptions and understanding of objects and environments. Finding methods to implement hierarchies would prove very useful for

robotic applications. The implementation here involves a single layer. In addition, using environmental clues to determine the present context. Objects around a room might indicate what type of room it is and therefore interact differently with those objects as a result. For example, recognizing the refrigerator or stove among the table and chairs can detect the environmental context of a kitchen.

The amygdala model is not assumed to be the best implementation of the most efficient. Improvements in these areas could enhance performance and create a more practical reusable model. In this research, multiple vocabularies were used. It is uncertain if this is better than a single, large vocabulary broken into their categorical subsets. This method might reduce noise when unbinding symbols. Other memory cleanup methods may also exist. Reduced noise could allow for faster identification and response.

Research on cognitive emotions could be considered for diverse mood states [37]. Research exists using semantic pointers for emotions [38]. This research was not known until late in the thesis. Exploring how this could be applied or expand the model is worthwhile.

The performance metrics of nearest distance and nearest time provided measurements to provide insights into robot movement and speed. While these values were useful for this research, a better method would be ideal for tracking behavior. One approach might be to compare an expected versus a measured movement vector.

Finer motor controls can be developed for this platform. Simple proportional controls for speed and orientation were implemented. Problems arose when selecting targets behind the TurtleBot. Creating controls for linear and angular positions and

velocities would provide more desirable behaviors and allow for greater focus on other cognitive enhancements, such as the amygdala.

8 REFERENCES

- [1] Aristotle, Aristotle's Politics, Oxford :Clarendon Press, 1905.
- [2] Wikipedia, "Robotics - Wikipedia," [Online]. Available:
<https://en.wikipedia.org/wiki/Robotics#History>. [Accessed 14 4 2022].
- [3] L. Nocks, The Robot: The Life and Story of Technology, Greenwood Publishing, 2006, p. 55.
- [4] T. Taulli, Artificial Intelligence Basics, Apress, 2019.
- [5] I. Kotseruba and J. K. Tsotsos, "Abilities, 40 Years of Cognitive Architectures: Core Cognitive," *Artificial Intelligence Review*, vol. 53, pp. 17-94, 28 July 2018.
- [6] T. J. Bihl, W. A. Young II, S. Frimel and A. Moyer, "Artificial Neural Networks and Data Science," *Encyclopedia of Data Science and Machine Learning*, 2022.
- [7] A. Kaplan, Artificial Intelligence, Business and Civilization, Routledge, 2022.
- [8] P. Langley, "Interactive Cognitive Systems and Social Intelligence," *IEEE Intelligent Systems*, vol. 32, no. 4, pp. 22-30, 2017.
- [9] B. Girard, J. Lienard, C. E. Gutierrez, B. Delord and K. Doya, "A Biologically Constrained Spiking Neural Network Model of the Primate Basal Ganglia with Overlapping Pathways Exhibits Action Selection," *European Journal of Neuroscience*, vol. 53, no. 7, pp. 2254-2277, 2021.
- [10] T. C. Stewart, X. Choo and C. Eliasmith, "Dynamic Behaviour of a Spiking Model of Action Selection in the Basal Ganglia," in *Proceedings of the 10th International Conference on Cognitive Modeling*, Philadelphia, PA, 2010.

- [11] O. Trujillo and C. Eliasmith, "A Spiking-neuron Model of Memory Encoding and Replay in Hippocampus," *BMC Neuroscience*, vol. 15, p. 166, 21 07 2014.
- [12] K. Iwadate, I. Suzuki, M. Watanabe, M. Yamamoto and M. Furukawa, "An Artificial Neural Network Based on the Architecture of the Cerebellum for Behavior Learning," *Soft Computing in Artificial Intelligence. Advances in Intelligent Systems and Computing*, vol. 270, pp. 143-151, 2014.
- [13] J. P. Fadok, M. Markovic, P. Tovote and A. Lüthi, "New perspectives on central amygdala function," *Current Opinion in Neurobiology*, vol. 49, pp. 141-147, 2018.
- [14] K. D. Fischl, "Neuromorphic Models of the Amygdala and Applications to Spike Based Computing and Robotics," Batlimore, MD USA, 2019.
- [15] T. Stewart, "AmyRobot," 2018 July 2018. [Online]. Available: <https://github.com/Neuromorphs18/AmyRobot>. [Accessed 2021].
- [16] T. Stewart, "Nengo Amygdala," 5 Septembr 2018. [Online]. Available: https://github.com/tcstewar/nengo_amygdala. [Accessed 2021].
- [17] K. D. Fischl, A. B. Cellon, T. C. Stewart, T. K. Horiuchi and A. G. Andreou, "Socio-Emotional Robot with Distributed Multi-Platform Neuromorphic Processing," in *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, 2019.
- [18] T. Bihl, T. Jenkins, C. Cox, A. DeMange, K. Hill and E. Zelnio, "From Lab to Internship and Back Again: Learning Autonomous Systems through Creating a Research and Development Ecosystem," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9635-9643, 2019.

- [19] M. Hampo, D. Fan, T. Jenkins, A. DeMange, S. Westberg and T. T. Trevor Bihl, "Associative Memory in Spiking Neural Network Form Implemented on Neuromorphic Hardware," in *International Conference on Neuromorphic Systems 2020*, Oak Ridge TN USA, 2020.
- [20] T. J. Prescott, F. M. M. González, K. Gurney, M. D. Humphries and P. Redgrave, "A Robot Model of the Basal Ganglia: Behavior and Intrinsic Processing," *Neural Networks*, vol. 19, no. 1, pp. 31-61, 2006.
- [21] D. Vernon, G. Metta and G. Sandini, "A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 151-180, April 2007.
- [22] "Cognition," 8 May 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Cognition>. [Accessed 15 July 2021].
- [23] Z. Bing, C. Meschede, F. Röhrbein, K. Huang and A. C. Knoll, "A Survey of Robotics Control Based on Learning-Inspired Spiking Neural Networks," vol. 12, 2018.
- [24] C. Eliasmith, *How to Build a Brain A Neural Architecture for Biological Cognition*, Oxford University Press, 2007.
- [25] J. C. V. Tieck, K. Secker, J. Kaiser, A. Roennau and R. Dillmann, "Soft-Grasping With an Anthropomorphic Robotic Hand Using Spiking Neurons," *IEEE Robotics and Automation Letters*, vol. 6, pp. 2894-2901, April 2021.

- [26] J. C. V. Tieck, J. Rutschke, J. Kaiser, M. Schulze, T. Buettner, D. Reichard, A. Roennau and R. Dillmann, "Combining spiking motor primitives with a behaviour-based architecture to model locomotion for six-legged robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4161-4168, 2019.
- [27] T. S. Clawson, S. Ferrari, S. B. Fuller and R. J. Wood, "Spiking neural network (SNN) control of a flapping insect-scale robot," *IEEE 55th Conference on Decision and Control (CDC)*, pp. 3381-3388, 2016.
- [28] M. Sarim, T. Schultz, M. Kumar and R. Jha, "An Artificial Brain Mechanism to Develop a Learning Paradigm for Robot Navigation," in *ASME 2016 Dynamic Systems and Control Conference*, Minneapolis, 2016.
- [29] M. D. Humphries and K. Gurney, "Making decisions in the dark basement of the brain: A look back at the GPR model of action selection and the basal ganglia," 2021.
- [30] V. Azimirad and M. Fattahi Sani, "Experimental Study of Reinforcement Learning in Mobile Robots Through Spiking Architecture of Thalamo-Cortico-Thalamic Circuitry of Mammalian Brain," *Robotica*, pp. 1558-1575, 2020.
- [31] P. Lanillos, J. Pages and G. Cheng, "Robot self/other distinction: active inference meets neural networks learning in a mirror," 2020.
- [32] C. A and T. F., "Spatial Concept Learning: A Spiking Neural Network Implementation in Virtual and Physical Robots," *Computer Intelligence and Neuroscience in Neurorobotics*, 1 April 2019.

- [33] TurtleBot, "TurtleBot," [Online]. Available: <https://www.turtlebot.com/turtlebot2/>. [Accessed 2021].
- [34] T. Bihl, J. Schoenbeck, D. Steeneck and J. Jordan, "Easy and Efficient Hyperparameter Optimization to Address Some Artificial Intelligence “ilities”," in *53rd Hawaii International Conference on System Sciences*, 2020.
- [35] T. J. Bihl, *Biostatistics Using JMP: A Practical Guide*, Cary, NC: SAS Press, 2017.
- [36] D. C. Montgomery, G. C. Runger and N. F. Hubele, *Engineering Statistics*, 5th ed., Wiley, 2010.
- [37] L. Rodríguez, F. Ramos and Y. Wang, "Cognitive Computational Models of Emotions," in *IEEE 10th International Conference on Cognitive Informatics and Cognitive Computing (ICCI-CC'11)*, Banff, AB, Canada, 2011.
- [38] I. Kajić, T. Schröder, T. C. Stewart and P. Thagard, "The Semantic Pointer Theory of Emotion: Integrating Physiology, Appraisal, and Construction," *Cognitive Systems Research*, vol. 58, pp. 35-53, 2019.
- [39] T. C. Stewart, "A Technical Overview of the Neural Engineering Framework," Centre for Theoretical Neuroscience, 2012.
- [40] S. Thorpe, A. Delorme and R. Van Rullen, "Spike-based Strategies for Rapid Processing," *Neural Networks*, vol. 14, no. 6-7, pp. 715-725, 29 January 2001.
- [41] J. Hu, H. Niu, J. Carrasco, B. Lennox and F. Arvin, "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning".

- [42] J. C. V. Tieck, S. Weber, T. C. Stewart, J. Kaiser, A. Roennau and R. Dillmann, "A spiking network classifies human sEMG signals and triggers finger reflexes on a robotic hand," *Robotics and Autonomous Systems*, vol. 131, September 2020.
- [43] Intel Corporation, "Neuromorphic Computing," Intel Corporation, [Online]. Available: <https://www.intel.com/content/www/us/en/research/neuromorphic-computing.html>. [Accessed 13 March 2021].
- [44] J. Ueda and Y. Kurita, Human Modeling for Bio-Inspired Robotics: Mechanical Engineering in Assistive Technologies, . Ueda, Ed., Elsevier Science, 2016.
- [45] K. Kumarasinghe, N. Kasabov and D. Taylor, "Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements," *Scientific Reports*, vol. 11, no. 1, 28 1 2021.
- [46] Wikipedia, "Hero of Alexandria - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Hero_of_Alexandria. [Accessed 14 4 2022].
- [47] C. Xu, S. Krabbe, J. Gründemann, P. Botta, J. P. Fadok, F. Osakada, D. Saur, B. F. Grewe, M. J. Schnitzer, E. M. Callaway and A. Lüthi, "Distinct Hippocampal Pathways Mediate Dissociable Roles of Context in Memory Retrieval," *Cell*, vol. 167, no. 4, pp. 961-972, 2016.

A ADDITIONAL INFORMATION

A.1 TURTLEBOT REFERENCE TABLES

Table A.1: Highway context mapping used for simulation embodiment

Input	Mapping
Bus	Fear
Car	Hunger
Fire Hydrant	Fear
Stop Sign	Curiosity
Traffic Light	Hunger + 0.25*Fear
Truck	Fear – 0.5*Curiosity

Table A.2: Neighborhood context mapping used in simulation embodiment

Input	Mapping
Bus	Hunger
Car	Curiosity
Fire Hydrant	None
Stop Sign	Hunger – 0.25*Fear
Traffic Light	Curiosity – 0.5*Fear
Truck	0.25*Hunger + Curiosity

Table A.3: Practice context mapping used in simulation embodiment

Input	Mapping
Bus	Curiosity
Car	Fear
Fire Hydrant	Curiosity
Stop Sign	Hunger – 0.25*Fear
Traffic Light	Curiosity – 0.5*Fear
Truck	0.25*Hunger + Curiosity

Table A.4: Office context

Input	Mapping
Backpack	$0.25 * \text{Fear}$
Bench	$0.5 * \text{Fear}$
Chair	$0.33 * \text{Hunger}$
Bottle	$\text{Hunger} + 0.25 * \text{Fear}$
TV Monitor	Curiosity
Laptop	Hunger
Dining Table	NA

Table A.5: Living room context

Input	Mapping
Backpack	Hunger
Bench	Fear
Chair	$0.4 * \text{Fear}$
Bottle	$0.75 * \text{Curiosity}$
TV Monitor	NA
Laptop	Fear
Dining Table	Hunger

Table A.6: Relating motivation to mood

Motivation	Mood Output
Discourage	$0.2 * \text{FEAR} - 0.1 * \text{HUNGER} - 0.1 * \text{CURIOSITY}$
Encourage	$0.25 * \text{CURIOSITY} - 0.3 * \text{FEAR}$
Focus	$0.5 * \text{HUNGER} - 0.25 * \text{CURIOSITY} - 0.25 * \text{FEAR}$
Normal	$0.15 * \text{CURIOSITY}$

Table A.7: Action to motor effects

Action	Linear X Effect	Angular Z Effect	Behavior
Flee	-2	1.2	Reverse
Feed	2	1	Accelerate
Explore	1	2	Rotate (Look)

Table A.8: Simulation Test Matrix

Run	Model	Context	Motivation
Amygdala Run 1	Amygdala Model 2MC	Highway	Normal
Amygdala Run 2	Amygdala Model 2MC	Highway	Encourage
Amygdala Run 3	Amygdala Model 2MC	Highway	Focus
Amygdala Run 4	Amygdala Model 2MC	Neighborhood	Normal
Amygdala Run 5	Amygdala Model 2MC	Neighborhood	Encourage
Amygdala Run 6	Amygdala Model 2MC	Neighborhood	Focus
Amygdala Run 7	Amygdala Model 2MC	Practice	Normal
Amygdala Run 8	Amygdala Model 2MC	Practice	Encourage
Amygdala Run 9	Amygdala Model 2MC	Practice	Focus
Baseline Run 1	Baseline	Highway	Normal
Baseline Run 2	Baseline	Practice	Normal
Baseline Run 3	Baseline	Neighborhood	Normal

Table A.9: Live embodiment test matrix

Run	Model	Context	Motivation
Amygdala Run 1	Amygdala Model 2MC	Office	Normal
Amygdala Run 2	Amygdala Model 2MC	Office	Encourage
Amygdala Run 3	Amygdala Model 2MC	Office	Focus
Amygdala Run 4	Amygdala Model 2MC	Living Room	Normal
Amygdala Run 5	Amygdala Model 2MC	Living Room	Encourage
Amygdala Run 6	Amygdala Model 2MC	Living Room	Focus
Baseline Run 1	Baseline	Office	Normal
Baseline Run 2	Baseline	Living Room	Normal

Table A.10: Simple model evaluation points [16]

Eyes	Mouth	Teeth	Known	Mood X	Mood Y
-1	1	-1	1	1	-1
1	-1	-1	-1	-1	0.3
1	-1	1	1	-1	1
1	-1	1	0	-1	0.6
1	0	-1	-1	-0.5	-0.5
1	0	-1	1	-1	-1
-1	1	-1	-1	0.5	0.5
-1	1	-1	1	1	1
-1	0	-1	-1	0.5	-0.5
-1	0	-1	1	1	-1

Table A.11: Simple model mood values [16]

Mood X	Mood Y	Meaning
1	1	Happy
-1	1	Angry
-1	-1	Sad
1	-1	Calm

B ROBOT MOVEMENT PLOTS

The following figures provide graphical representations of the paths followed by the robot during different simulation scenarios. Data presented in the movement plots has been smoothed due to the noisy nature of these SNNs. Increased smoothing causes the robot to appear on the right side of the target when it was on the left. The plots are intended for general information but are not considered to be accurate.

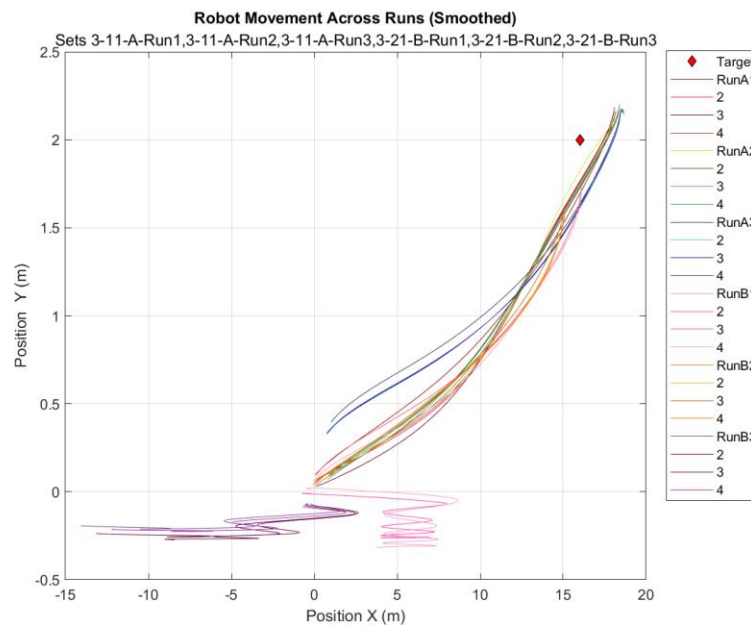


Figure B.1: Robot movement from simulations, both models, “Normal” motivation, all contexts

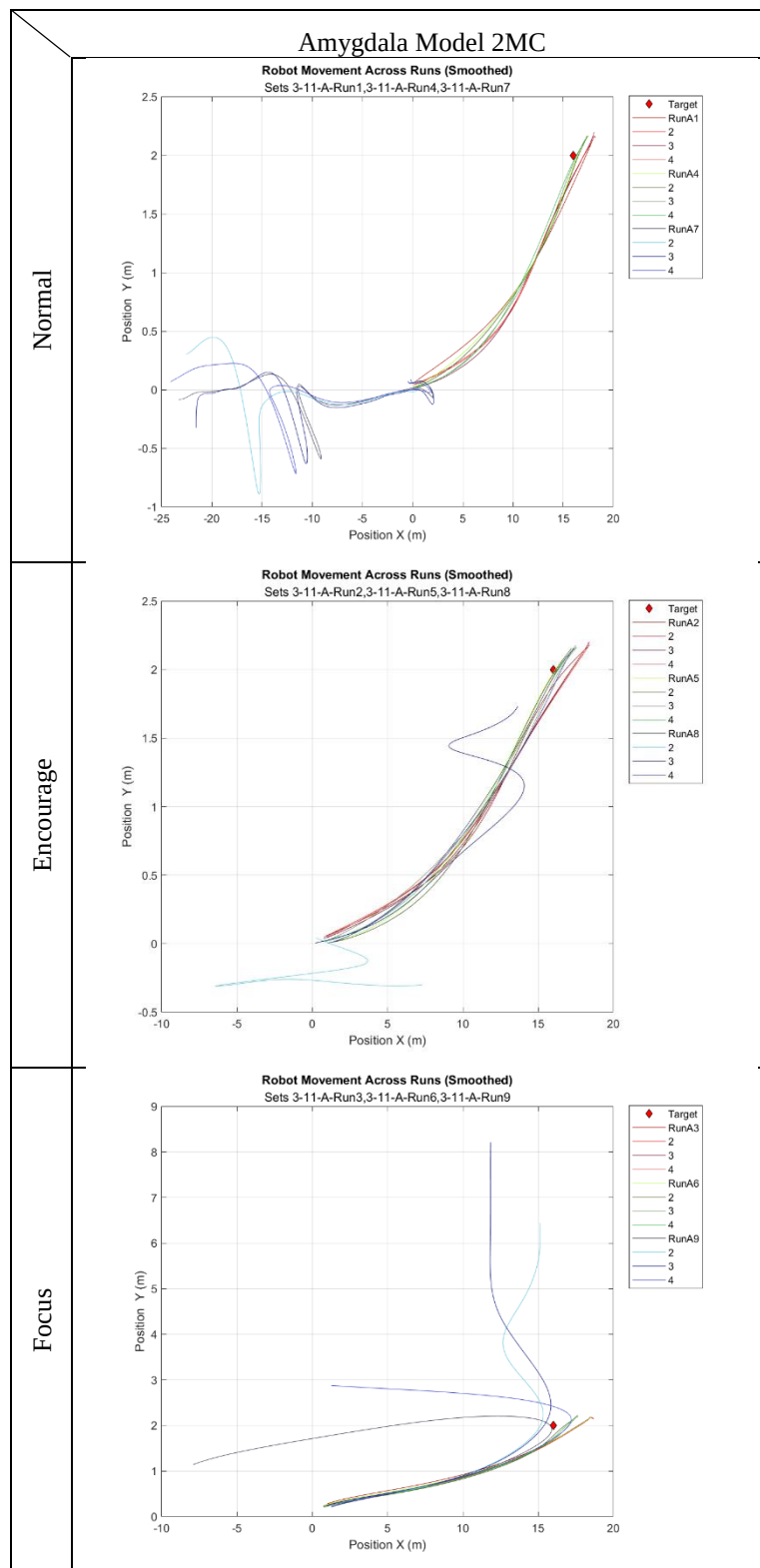


Figure B.2: Robot movement from simulations, arranged by motivation

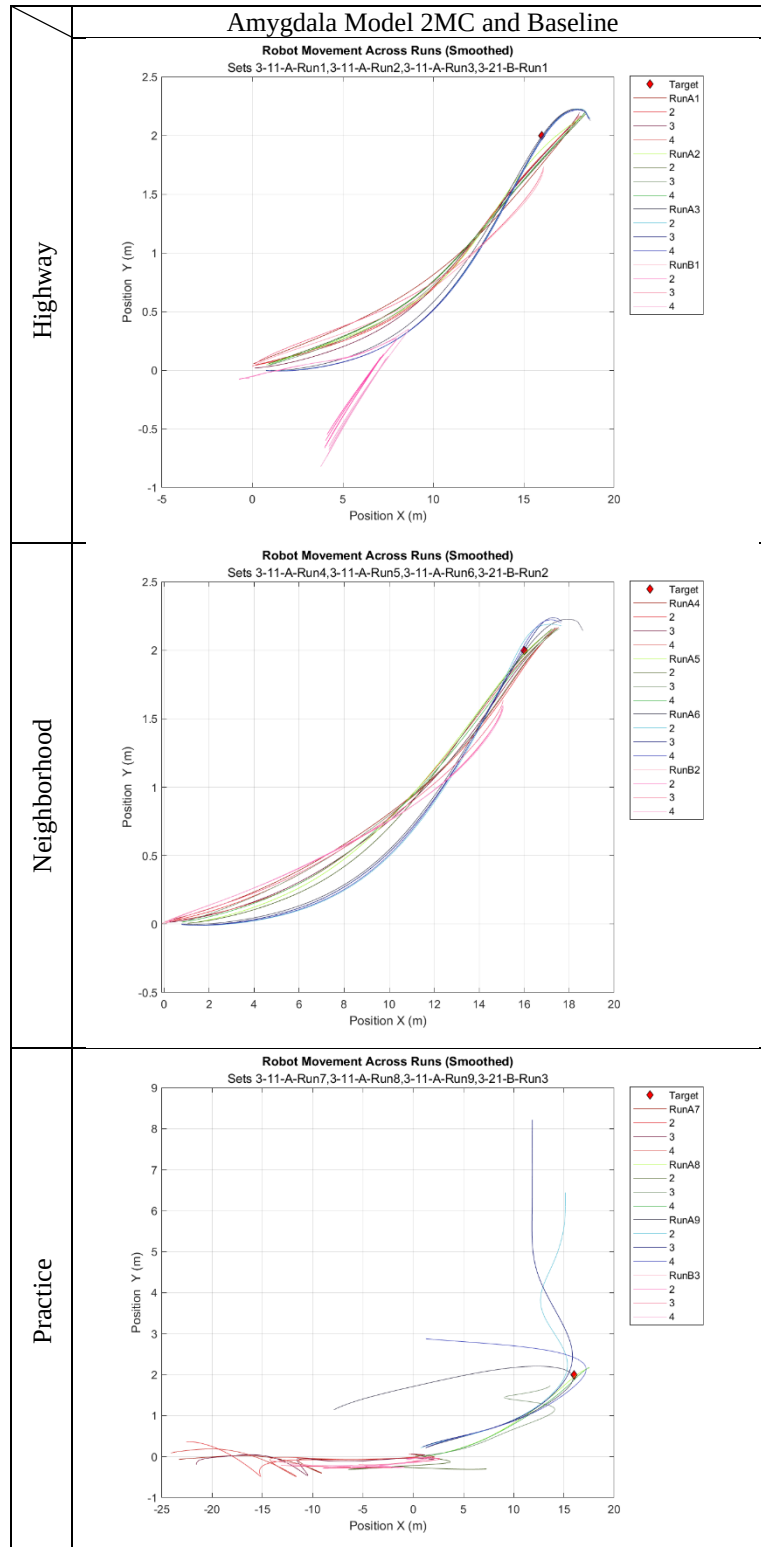


Figure B.3: Robot movement from simulations, arranged by context

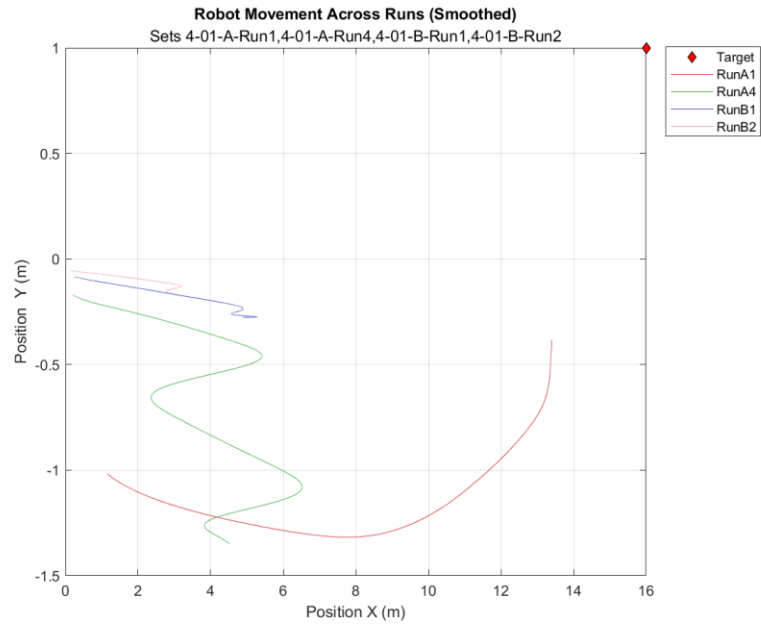


Figure B.4: Robot movement from live embodiment, both models, “Normal” motivation, all contexts

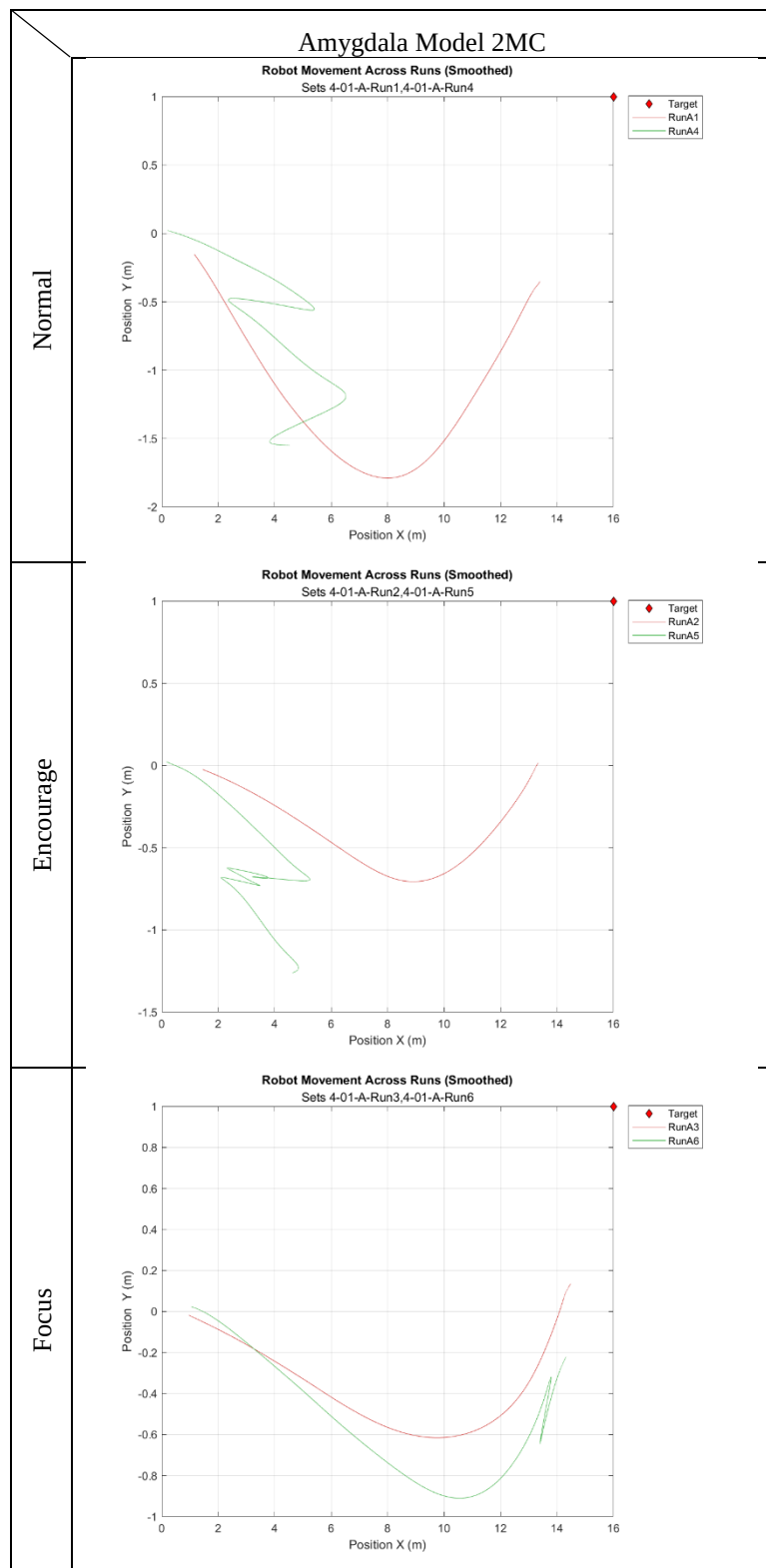


Figure B.5: Robot movement from live embodiment, arranged by motivation

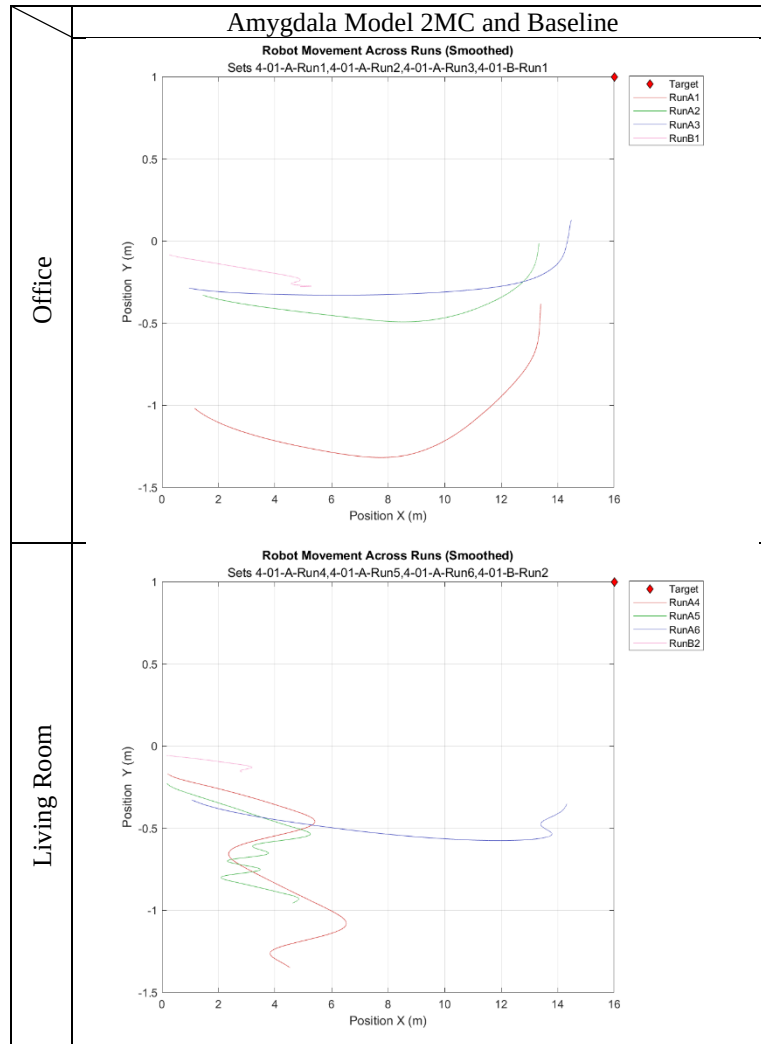


Figure B.6: Robot movement from live embodiment, arranged by context

B.1 TIMING ANALYSIS PLOTS

Timing analysis was attempted to determine the impacts of neuron count and complexity on the robot. Attempts at this were difficult and needed better analysis methods. Several figures are included here to show CNN input to mood detections.

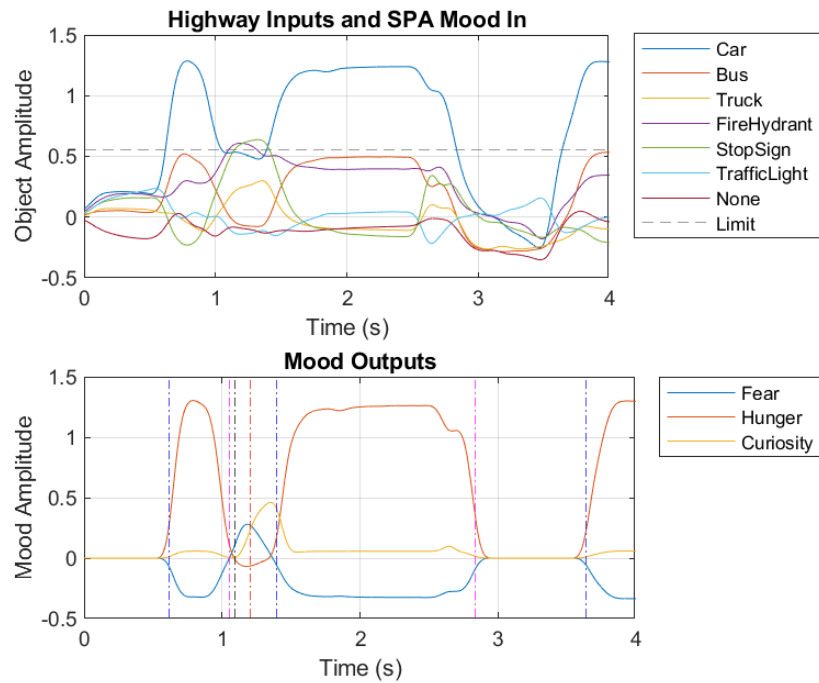


Figure B.7: Object and mood timing information from simulation Run A1

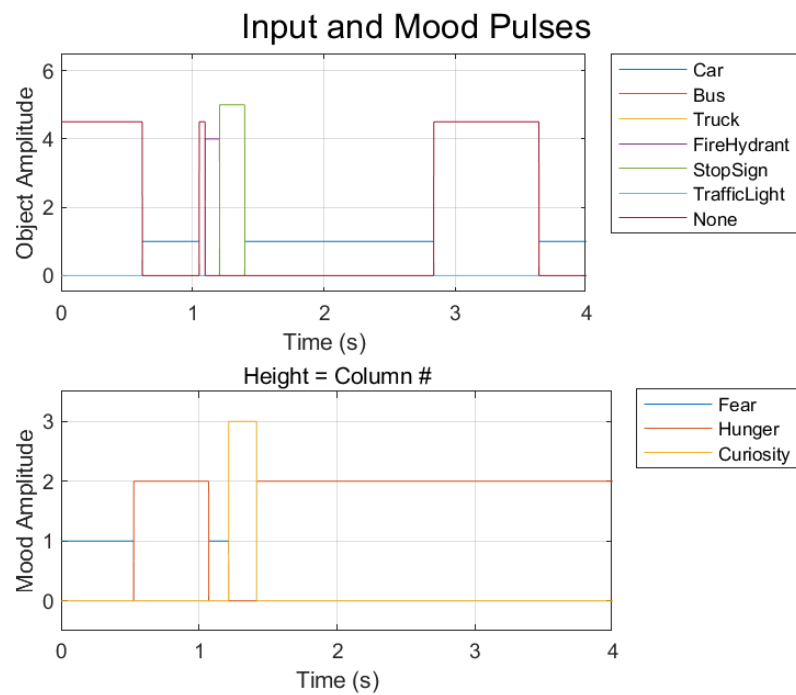


Figure B.8: Pulse form of object and mood data from simulation Run A1

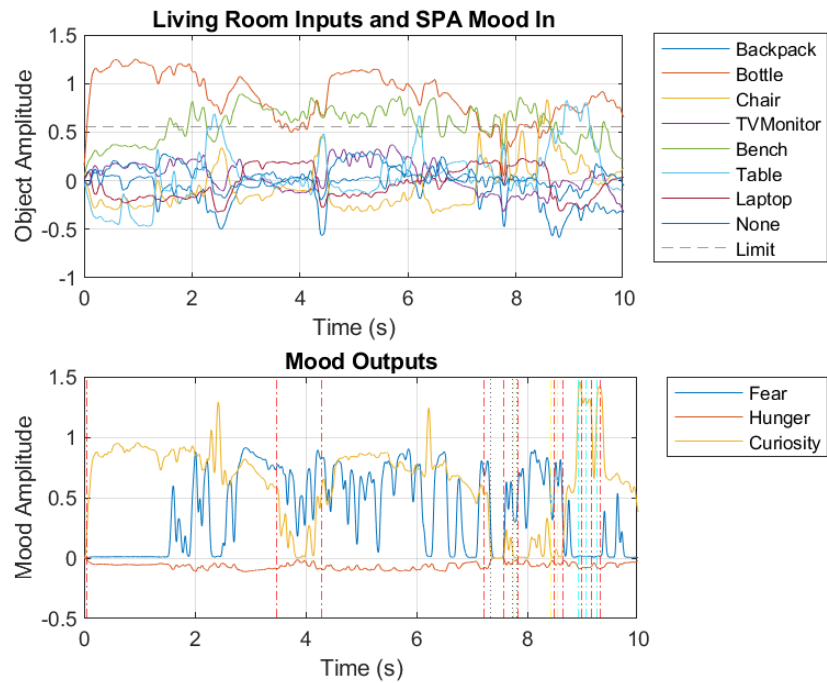


Figure B.9: Object and mood timing information from live embodiment Run A4

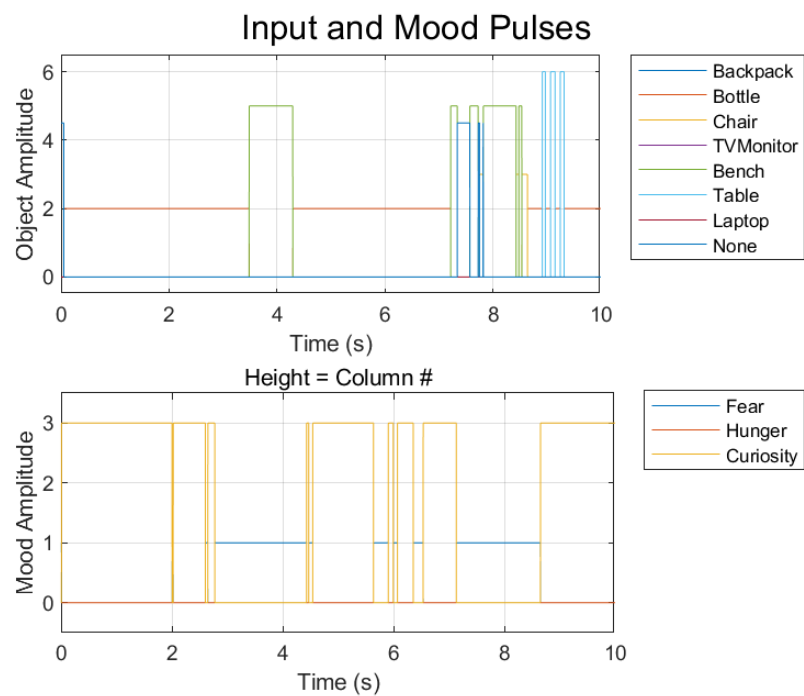


Figure B.10: Pulse form of object and mood data from live embodiment Run A4

The connectivity for robot actuator controls are shown in Figure B.11. The block diagram shows how the action states and target distance influence the drivers. The action state is the input of interest because this is controlled by the amygdala output, or mood. These controls drive the linear x and angular z motors. The scalar influence of these actions can be found in Table A.7.

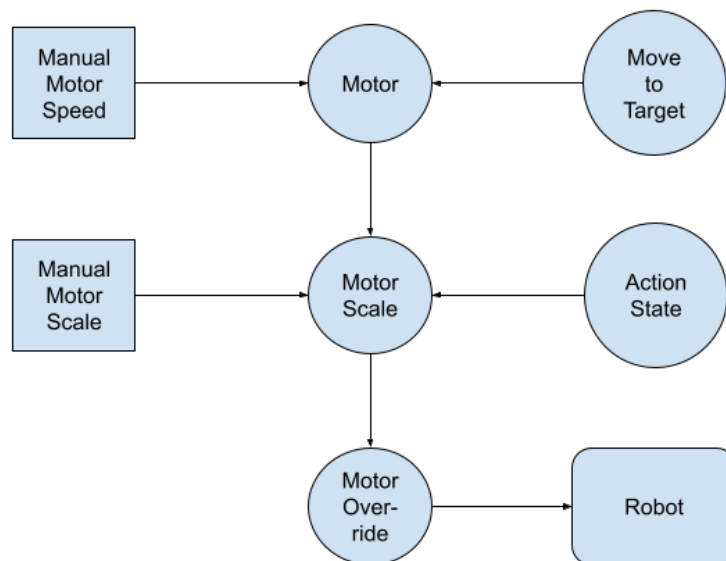


Figure B.11: Block diagram of motor control connectivity

Figure B.12 shows how mood relates to the action state, which influences the motor controls. The actions states are shown in dashed lines and scaled to allow an overlay with the mood responses. Transitions from curiosity (yellow) to fear (blue) are reflected in the explore (dashed blue) and flee (dashed purple) actions.

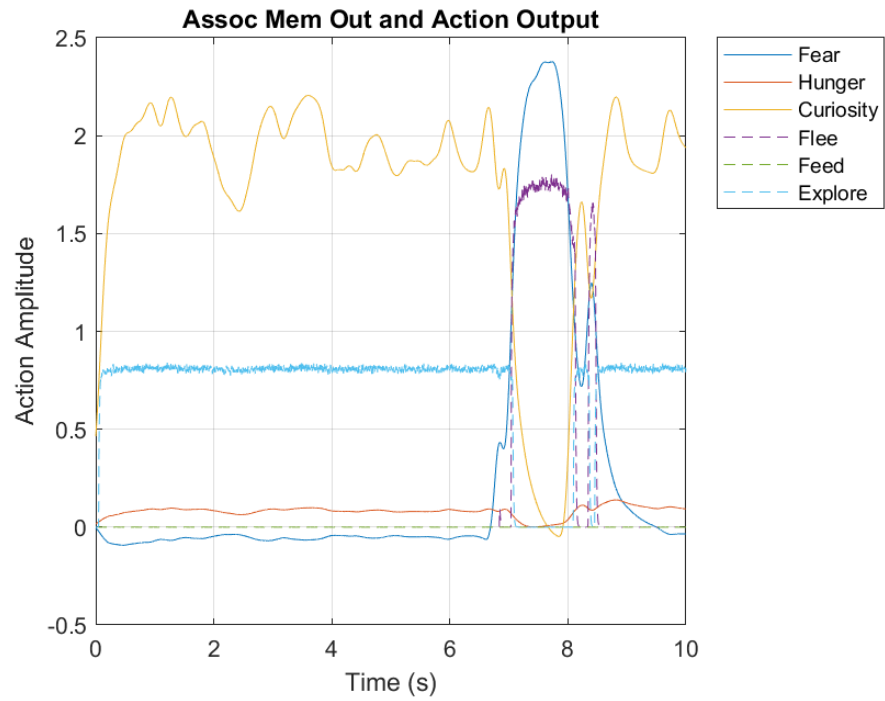


Figure B.12: Overlaying action (scaled, dashed) with mood responses

Figure B.13 shows the linear x motor response. Vertical dashed lines denote the action state transitions for timing calculations. This response matches the mood and action states shown in Figure B.12.

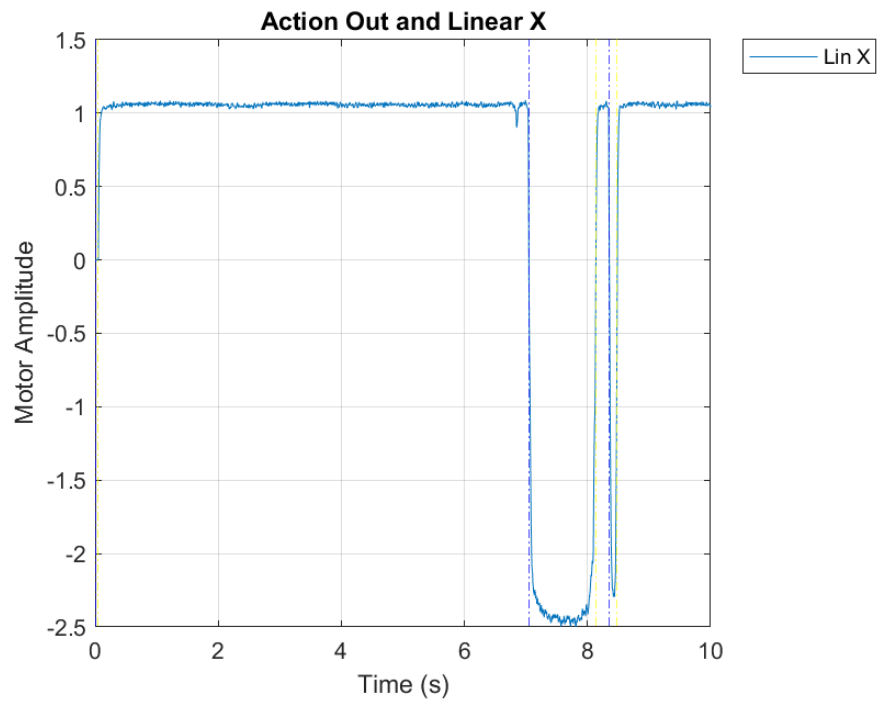


Figure B.13: Linear X motor response with vertical markers for action states

C EQUIPMENT AND SETUP

C.1 COMPUTER EQUIPMENT

Two computers were used during research, development, and testing. Initial development was performed on an HP Spectre x360 computer with the following hardware: Intel Core i7-10510U CPU at 1.8 GHz, 16.0 GB of RAM, NVIDIA GeForce MX250. This computer was the property of the researcher and readily available. Gazebo simulations were executed on a Dell 3650 with the following hardware: Intel Core i9-11900K CPU at 3.5 GHz with 8 cores, 32 GB of RAM, NVIDIA RTX 3090. A high-end GPU was required for the training of the Darknet image recognition CNN. The software is also tied with CUDA software which is proprietary to NVIDIA. This research worked with a pre-trained network. Possible methods exist to utilize the CPU but were not explored.

Initially, Nengo and ROS were loaded onto a Raspberry Pi 4. There was an early assumption of a simple installation. The Pi device is ARM-based and cannot support an x86 version of Linux. The original intent was abandoned for a desktop PC.

C.2 SOFTWARE

The HP Spectre x360 computer uses a Windows 10 OS. The following software was installed for operation: Windows 10, Docker, VcXsrv, and supporting packages for software dependencies or development. A Docker image was created to provide a development and testing environment consistent with the previous research. The image contains Ubuntu 20.4 OS, Gazebo, Nengo, ROS Noetic, Python 3.8, and other supporting packages for software dependencies.

The Dell workstation uses Ubuntu 20.4 OS, Gazebo, Nengo, ROS Noetic, Python 3.7, and other supporting packages for software dependencies. The configuration was set up to match the Docker image. This software setup allowed for easier network connections when connecting with multiple devices and the TurtleBot.

Gazebo provides the simulation environment. Nengo is the SNN development tool. The cross-platform communications are provided by ROS. Python is a software dependency for Gazebo, Nengo, and ROS.

Several challenges were encountered during this research project. Many of them were related to the equipment and software installation around the simulation work. Programs and simulation environments were new, and the learning curve was steeper than anticipated.

The original workstation provided was an older machine from 2007. Due to the lower demands of Linux and the Python-based Nengo environment, the given specs were adequate at the time. The starting platform was based on previous work. The source code was missing important elements which prevented usability. Software version requirements caused another set of challenges. Python 2.7 was required for Gazebo and ROS, but Nengo required Python 3.7. A split environment was created but caused other problems. Additionally, the graphics card was unable to run the CNN. The intended use of the CNN was used to process images in the virtual environment.

A possible solution was to operate the CNN on another machine. A server was recruited to perform this function. While plausible, the solution was less practical than having it available on the same computer. Server setup, resource scheduling, and cross-computer integration posed challenges to the timeline.

A new workstation was requested with university funding and a quote was put together. Supply shortages added to the challenge. The Dell workstation was acquired and delivered in early January 2022.

C.3 TURTLEBOT 2.0

The TurtleBot 2.0 is an open source, wheeled, robotic development kit. Platforms stacked vertically above the mobile base allowed for easy additions of sensors and other hardware [33]. Vision for the robot was established using an Xbox 360 Kinect camera and a standard NVIDIA Jetson TX2 was used for image processing. The TurtleBot used in this research is the exact one used in the previous one. Darknet code for CNN processing existed on the robot.

Access to a physical TurtleBot was not available late in 2021. Setup challenges were presented due to the inability to connect to the device through Docker. An UpBoard with previous work was used to provide the necessary connectivity. The new computer was used in place of the UpBoard once received.

Gradual steps were taken to prove connectivity and control over the TurtleBot using the ROS teleoperation commands and Nengo. Integrating the intended code presented a challenge with target control and motion. The robot commonly wanted to spin in circles instead of moving toward a target. Actions were then taken to understand and debug the reasons behind this behavior.

Debugging revealed a connection between the position and target ensemble that needed to be removed. The math, functions, and transformations were correct. An odd behavior was observed where a target behind the robot would create a spinning response or failure to reach the target. It was noted that targets not in front of the robot were not

reachable despite the proper math. The reasons for this were not researched due to it being outside of the scope of the project. Target locations were always placed in front of the robot.

D APPENDIX - LINUX-ROS-NENGO SETUP

D.1 WINDOWS SOFTWARE

1. Windows 10
2. Docker 4.2.0
3. VcXsrv
4. Other supporting packages

D.2 LINUX AND DOCKER IMAGE

1. Ubuntu 20.4
2. ROS Noetic
3. Gazebo
4. Anaconda
5. Python 3.8
6. Nengo
7. Other supporting packages

D.3 DEVELOPED TOOLS

1. Amygdala models for proof of concept and robot integration
2. FakeNet for fabricating CNN messages
3. Data export/import for calculation and plotting
4. Gazebo worlds for maneuvering and exploration

D.4 WINDOWS SETUP

1. Download VDL Package
 - a. <https://community.vdl.afrl.af.mil/neuromorphic-computing/demos-and-publications/turtlebot-reasoner>
2. Copy fakenet_ros files into the noetic workspace
 - a. These files are options when the CNN is not available
3. Update docker.yml file to map local ports from Docker to Windows

```
ports:
  - 8080:8080
  - 8888:8888
```

4. Build docker environment
 - a. In the noetic directory:

```
docker-compose up --build
```

5. Build ROS Workspace

- a. From the noetic directory, launch the docker environment

```
docker-compose run --service-ports --rm -e  
DISPLAY=host.docker.internal:0.0 -e LIBGL_ALWAYS_INDIRECT=0 tb_ros bash
```

- b. In the bash prompt, build the workspace

```
catkin_make
```

6. Verify Setup using Fakenet

- a. From docker start roscore

```
roscore &
```

- b. Launch Gazebo

```
roslaunch turtlebot_gazebo turtlebot_world.launch
```

- c. New terminal
- d. Start the fakenet publisher

```
roslaunch fakenet_ros fakenet_publisher.py
```

- e. New terminal
- f. Start Nengo in Docker (GUI)

```
nengo --no-browser --unsecure --listen '*' --password pwd  
src/turtlebot_nengo/scripts/turtlebot.py
```

- g. In Windows by open a browser to localhost:8080 for Nengo GUI
- h. Enter the password pwd
- i. Click play in the Nengo GUI
- j. OR run script

```
python src/turtlebot_nengo/scripts/turtlebot.py
```

Be sure to run the proper source commands in each terminal window.

```
source devel/setup.bash
```

If issues require an update to the Darknet package, updates to the CMakeList.txt file are required to match CUDA to the proper graphics card code.

```
vi ~/src/cnn/darknet_ros/darknet_ros/CMakeList.txt

if (CUDA_FOUND)
  find_package(CUDA REQUIRED)
  message(STATUS "CUDA Version: ${CUDA_VERSION_STRINGS}")
  message(STATUS "CUDA Libraries: ${CUDA_LIBRARIES}")
  set(
    CUDA_NVCC_FLAGS
    ${CUDA_NVCC_FLAGS};
    -O3
    #-gencode arch=compute_30,code=sm_30
    -gencode arch=compute_35,code=sm_35
    -gencode arch=compute_50,code=[sm_50,compute_50]
    -gencode arch=compute_52,code=[sm_52,compute_52]
    -gencode arch=compute_61,code=sm_61
    -gencode arch=compute_62,code=sm_62
    -gencode arch=compute_75,code=sm_75
    -gencode arch=compute_86,code=sm_86
  )
  add_definitions(-DGPU)
```

New network settings are also wrong... copy from the original folder set not from the git repo

```
Darknet_ros/darknet_ros/yolo_network_config
```

Configuration information for the TurtleBot is not included to retain network and login privacy. Information for this can be found in the Git repo.

D.5 MODELS AND SUPPORTING SCRIPTS

A separate file was created for each amygdala model. Several supporting Jupyter notebooks and MATLAB scripts were created to allow for data collection, calculations, and plotting for each of the files. These are made available in the Git repo.

D.6 FAKENET SOFTWARE

The original environment provides a CNN for image processing and recognition. The Darknet software is computationally intensive during the training operation. Running this

code via CUDA cores on a GPU provides the best performance. Hardware and software challenges prevented some of the original use of this feature, which would otherwise render the program useless. Published messages from the ROS node were fabricated so the system could operate without knowing the CNN wasn't present. Modes of operation included continuous, randomly generated messages or the ability to playback from a file. The fixed output file provided consistent data for testing models.

Integration testing involved the merging of the amygdala code into the existing Nengo TurtleBot model. Initial testing could be exercised without a Gazebo environment to prove connectivity. Proving full functionality and performance required Gazebo and ROS. The robotic system uses a CNN for image classification. Equipment limitations in earlier testing prevented the use of processed images as inputs. Fakenet was used to mimic CNN messages. The messages alongside ROS interactions worked as expected but there were synchronization issues between Gazebo and the Nengo script. This issue made repeatable responses nearly impossible to create. Figure D.14 shows how ROS message inputs were tracked, context filtered, and the resulting moods.

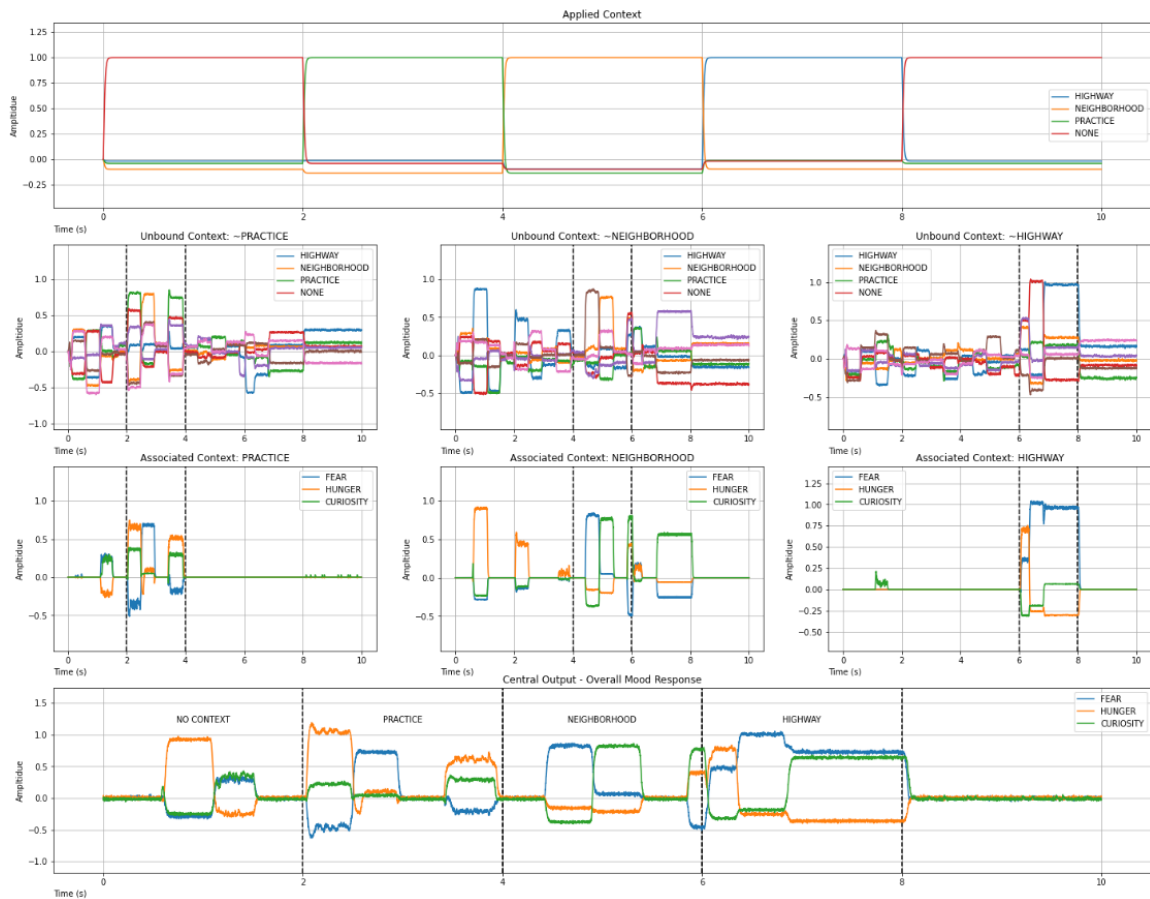


Figure D.14: Initial integration with ROS and Fakenet without motivation

Fakenet was helpful to prove out system inputs without having access to the CNN. The tool was abandoned because inputs and timing could not be repeated. Internally generated inputs were used as a short-term solution to provide movement and interaction in the simulated world. Using these different test input approaches in a simulated environment proved that integration does not affect the amygdala pathway. Further testing could be accomplished once new equipment was received.