

# IDENTIFYING AND PREDICTING RAT BEHAVIOR USING NEURAL NETWORKS

A Thesis  
presented to  
the Faculty of California Polytechnic State University,  
San Luis Obispo

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in Biomedical Engineering

by  
Jonathan Gettner  
December 2015

© 2015  
Jonathan Gettner  
ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: Identifying and Predicting Rat Behavior Using Neural Networks

AUTHOR: Jonathan Gettner

DATE SUBMITTED: December 2015

COMMITTEE CHAIR: Robert Szlavik, Ph.D.  
Professor of Biomedical Engineering

COMMITTEE MEMBER: David Clague, Ph.D.  
Professor of Biomedical Engineering

COMMITTEE MEMBER: Scott Hazelwood, Ph.D.  
Professor of Biomedical Engineering

## ABSTRACT

### Identifying and Predicting Rat Behavior Using Neural Networks

Jonathan Gettner

The hippocampus is known to play a critical role in episodic memory function. Understanding the relation between electrophysiological activity in a rat hippocampus and rat behavior may be helpful in studying pathological diseases that corrupt electrical signaling in the hippocampus, such as Parkinson's and Alzheimer's. Additionally, having a method to interpret rat behaviors from neural activity may help in understanding the dynamics of rat neural activity that are associated with certain identified behaviors.

In this thesis, neural networks are used as a black-box model to map electrophysiological data, representative of an ensemble of neurons in the hippocampus, to a T-maze, wheel running or open exploration behavior. The velocity and spatial coordinates of the identified behavior are then predicted using the same neurological input data that was used for behavior identification. Results show that a nonlinear autoregressive process with exogenous inputs (NARX) neural network can partially identify between different behaviors and can generally determine the velocity and spatial position attributes of the identified behavior inside and outside of the trained interval.

# TABLE OF CONTENTS

	Page
LIST OF TABLES .....	vii
LIST OF FIGURES .....	viii
Chapter 1: Introduction .....	1
1.1 Statement of Problem .....	2
1.2 Research Questions .....	3
1.3 Issues and Assumptions .....	4
1.4 General Approach .....	5
1.5 Organization of Thesis .....	6
Chapter 2: Literature Review .....	8
2.1 Background .....	8
2.2 Traditional Methods .....	9
2.3 Neural Networks .....	9
Chapter 3: System Design.....	11
3.1 Three Stage Design .....	11
3.2 Pre-Processing Stage .....	12
3.2.1 Raw Data Gathering.....	12
3.2.2 Spike Sorting.....	14
3.2.3 Forming Temporal Component of Input Data, Illustrative Example Using Simplified NARX Network.....	17
3.3 Identification Stage .....	21
3.3.1 Identification Network Parameters .....	21
3.3.2 Identification Training Methods .....	22
3.4 Behavior Attribute Prediction Stage .....	24
3.4.1 Behavior Attribute Network Parameters.....	24
3.4.2 Behavior Attribute Training.....	26
3.5 System Implementation.....	27
Chapter 4: Testing and Evaluation.....	29
4.1 Testing Identification Network .....	29
4.1.1 Self-Testing Datasets For Identification Network .....	30
4.1.2 Cross-Testing Datasets For Identification Network.....	36
4.2 Testing of Behavior Attribute Network .....	39

4.2.1 T-Maze Behavior Attribute Network.....	40
4.2.2 Wheel Running Behavior Attribute Network .....	42
4.2.3 Open Exploration Behavior Attribute Network.....	43
4.3 Results Discussion.....	45
Chapter 5: Summary and Conclusions.....	48
5.1 Contributions.....	49
5.2 Future Work .....	49
REFERENCES .....	50
APPENDICES	
Appendix A: Network Learning Process .....	53
Appendix B: Program.....	57
Identification Network Code.....	57
Behavior Attribute Network Code for T-Maze.....	68
Appendix C: Self-Testing Identification Network with Datasets 2, 3 and 4.....	75
Appendix D: Total Error Calculation.....	79

## LIST OF TABLES

Table	Page
Table 1: General location of rat hippocampal neural activity associated with shank ID number .....	13
Table 2: Four time steps of input data with four additional steps of no input .....	19
Table 3: Parameters of the NARX identification network.....	21
Table 4: Parameters for every behavioral network.....	25
Table 5: Possible datasets that can be used to train or test the identification network .....	29
Table 6: The number of training sample points used for each behavior from dataset 1 to train the identification network.....	30
Table 7: The time interval of the training data, time gap between training and testing and testing data.....	32
Table 8: Time intervals of the training set, time gap between training and testing sets and 2 <sup>nd</sup> testing set .....	35
Table 9: Datasets used for behavior attribute network training and testing .....	39

## LIST OF FIGURES

Figure	Page
Figure 1: (a) Anatomy of the rat hippocampus, illustrating the transfer of electrical signals between regions in the hippocampus [2]. The electrical signal will start at the EC region go through the CA3 and CA1 regions, then loop back to EC region where the information carried by the signal is projected into long-term memory. (b) A computer simulation illustrating the possible firing of an ensemble of neurons in a rat hippocampus, demonstrating the spatial aspect of neurological activity as an ensemble of neurons consisting of individual neurons firing in relation to one another [3].	1
Figure 2: The overall design consists of three stages: pre-processing stage, identification stage and behavior attribute prediction stage. The pre-processing stage decomposes filtered LFP spike waveforms into cluster ID data. The identification stage uses the cluster ID and shank ID data as input for the identification network, which categorizes the input into an output of 1, 2 or 3 representing T-maze, open exploration or wheel running behavior, respectively. The behavior attribute prediction stage uses the same input used to identify the general behavior to determine the velocity and spatial coordinate attributes of the behavior identified in the identification stage.	12
Figure 3: (a) An example of a shank with six recording sites on it. Illustrating how one shank may gather multiple signals at once [23]. (b) A sample LFP of a rat performing wheel running task for a 1000ms interval. Every shank is represented by a unique color.	14
Figure 4: Example of aligning spike waves and finding the resulting principal components [25].	16
Figure 5: An example of creating boundaries around clusters to identify the unique neurons in a system [25].	16
Figure 6: Cluster data for a rat performing wheel running behavior. (a) The local cluster data illustrating how many neurons an individual shank recognizes. (b) The cluster ID# arranged from 1-128, ordering cluster ID sequentially from shank ID 1 to shank ID 14.	17
Figure 7: A simplified NARX network used to illustrate the temporal component. The input $x_1$ and $x_2$ represents shank ID and cluster ID, respectively. Weights ( $w$ ) are used to factor the importance of a particular input or previous output and are optimally determined by training the network with a representative dataset. For this example, input weights ( $w_1$ and $w_2$ ) are set to unity, value of 1, and the weight of the previous output ( $w_0$ ) is 0.5. This network calculates the next output by linearly summing the weighted current input and weighted previous output. The nonlinear hidden layer of the network is removed to simplify the example.	18



Figure 8: The resulting output calculated by the network using eight time steps. This plot illustrates that the calculated output is the sum of the input and half the previous output at any given time point. However, when there is no further input, the network still retains half the value of the previous time point. ....	20
Figure 9: A diagram of the NARX network used for identification of T-maze, wheel running or open exploration behavior. Both of the inputs have a delay of 30, where “D” represents a time-series delay, which implies the other 29 previous values of input and their weights not illustrated. The output has a single delay ( $z^{-1}$ ) that feeds back the previous output as input to the hidden layer. All of the inputs are fully connected to the hidden layer.....	22
Figure 10: A general NARX network in series-parallel mode. A series parallel mode uses the previous value of the desired output, rather than the previous value of the network output, to avoid using incorrect network output as input for training.....	23
Figure 11: The NARX model with the parameters of the behavior attributes network implemented. Input consists of the cluster ID and shank ID at the current time-point and the previous 50 inputs along with the previous output. The input is fully connected to the 30 neurons in the hidden layer where the input is nonlinearly transformed, then linearly summed at the output. ....	25
Figure 12: An open mode NARX network illustrated in MATLAB, where the previous desired output is used, rather than the calculated output of the network. Using the desired output as input to the network allows the network to make mistakes during training, but still converge to a solution.....	27
Figure 13: A closed-loop mode NARX network with the same parameters used in the identifier network. This model is used for testing the network with new data that it has not been previously trained for. ....	28
Figure 14: A plot of all 15000 time points of input data used to train the identifier NARX network. The input data consists of cluster ID as a function of shank ID for each behavior. Input data of the open exploration is well distinguished from T-maze and wheel running data, while the T-maze and wheel running input closely overlap. ....	31
Figure 15: Identification results for the input used to train the network. The desired identification for the corresponding input data (left) includes 5000 time points for each behavior, where the first 5000 time points is T-maze, time points between 5001-10000 is open exploration and the last 5000 time points is wheel running. The network identification (right) demonstrates that the network can effectively identify input data from within the training interval. ....	33

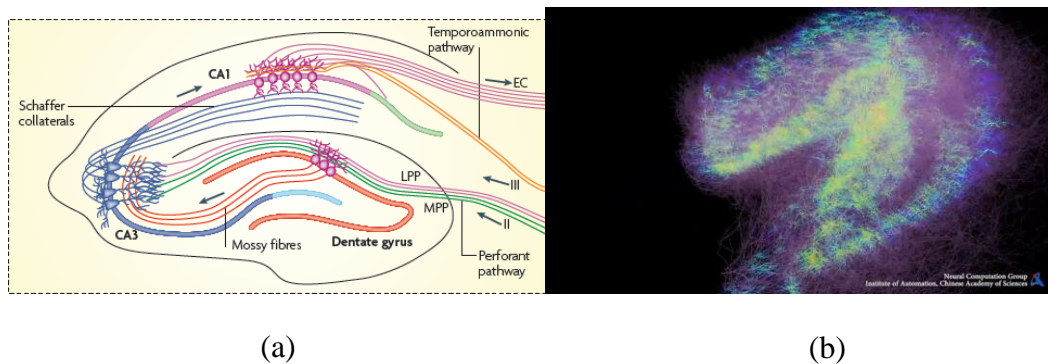
Figure 16: Results when identifying new input data the network was not trained with. The network combines wheel running and T-maze behavior into one behavior and identifies the behavior as 1 (T-maze behavior). The T-maze and open exploration behavior is correctly identified as 1 and 2, respectively, outside of the training interval. ....	34
Figure 17: Identification results with the 2 <sup>nd</sup> set of testing data for dataset 1. The desired identification results (left) and the network identification (right) when there is a significant time gap between the training and testing data. The network correctly identifies between open exploration and T-maze behavior and still confuses the wheel running behavior as T-maze behavior. ....	35
Figure 18: Identification results when identification network is trained by dataset 1 and tested with dataset 2. The top row involves the network being tested with early time points of dataset 2 and the bottom row involves the network being tested with later time points of dataset 2. ....	37
Figure 19: Identification results when identification network is trained by dataset 1 and tested with dataset 3. The top row involves the network being tested with early time points of dataset 3 and the bottom row involves the network being tested with later time points of dataset 3. ....	37
Figure 20: Identification results when identification network is trained by dataset 1 and tested with dataset 4. The top row involves the network being tested with early time points of dataset 4 and the bottom row involves the network being tested with later time points of dataset 4. ....	38
Figure 21: A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the T-maze behavior attribute network. The network prediction resulted in an average velocity when compared to the actual velocity and the predicted spatial coordinates did not completely traverse the actual path of the rat. The short-comings in both the velocity and spatial position of the rat is what makes this network more generalized, or applicable to other neurological data that the network has not been trained with. ....	41
Figure 22: A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the T-maze behavior attribute network tested with a dataset that it was not trained for. The network was trained with T-maze data from dataset 1 and tested with T-maze data from dataset 2. The neurological data given to the network was completely different from the data that the network was trained with in Figure 21, however, the network was able to converge to a new average velocity and path for the rat. The form of the predicted spatial position is similar to the desired spatial position, but the predicted direction differed from the actual path, which demonstrates that the network was able to partially	

	adapt to a new event that it was not trained for. ....	41
Figure 23:	A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the wheel running behavior attribute network. For this testing input, the prediction of the velocity generally stayed constant at 600 mm/s, demonstrating the tendency of the network to converge to an output value that generally modeled the desired velocity. The spatial coordinates were well predicted for most of the wheel running data, due to lack of spatial movement during wheel running behavior.....	42
Figure 24:	A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the wheel running behavior attribute network for a dataset the network had not been trained for. The network was trained with dataset 1 and tested with dataset 2. For this testing data, the network prediction of velocity generally models the desired velocity. The spatial coordinates were well predicted, due to lack in spatial movement during wheel running behavior.....	43
Figure 25:	A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the open exploration behavior attribute network. The predicted velocity remained at approximately 100 pixel/s, demonstrating the tendency of the network to converge to an output value that reasonably fits the desired velocity. The network's prediction of the spatial position poorly reflects the desired spatial positioning of the rat, due to the networks nature to stop training when either of the outputs (i.e. velocity or spatial position) converges to a solution. Thus, the network's velocity prediction converged to a solution before the spatial position prediction was able to completely converge to a solution, providing poor results for the predicted path in this case.....	44
Figure 26:	A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the open exploration behavior attribute network for a dataset the network had not been trained for. The network was trained with dataset 1 and tested with dataset 3. The predicted velocity converged to an output value of approximately 100 pixel/s, which is faster than most of the desired velocity data and the predicted waveforms do not fit the desired velocity data well. The lack in adaptability demonstrates how modeling the training data well can work against the network when new data is presented. The predicted spatial position of the rat started in the same area the network predicted for the trained data, however, it adapted to the new	

neurological data and converged to a region similar to that of the desired spatial position.....	44
Figure A-1: A series-parallel mode of the NARX network used for training. Training utilizes the previous desired output rather than the network output to minimize performance error of the network during training. ....	53
Figure C-1: 15000 time points of input data for dataset 2 used to train the identification network. The T-maze and wheel running behavior data are closely overlapped, while most of the open exploration behavior data is distinguishable from T-maze and wheel running behavior. ....	75
Figure C-2: Identification network results when data from dataset 2 was used. The network was able to identify all three behaviors within the training interval (top-right), but confused T-maze behavior as wheel running behavior outside of the training interval (bottom-right). ....	76
Figure C-3: 15000 time points of input data for dataset 3 used to train the identification network. The T-maze and wheel running behavior data are closely overlapped, while most of the open exploration behavior data is distinguishable from T-maze and wheel running behavior. ....	76
Figure C-4: Identification network results when data from dataset 3 was used. The network was able to identify all three behaviors within the training interval (top-right), but confused T-maze behavior as wheel running behavior outside of the training interval (bottom-right). ....	77
Figure C-5: 15000 time points of input data for dataset 4 used to train the identification network. The T-maze and wheel running behavior data are closely overlapped, while all of the open exploration behavior data is distinguishable from T-maze and wheel running behavior. ....	77
Figure C-6: Identification network results when data from dataset 4 was used. The network was partially able to identify all three behaviors within the training interval (top-right), but confused T-maze behavior as wheel running behavior outside of the training interval (bottom-right). The identification results were more difficult for this dataset due to a larger variation of input data.....	78

## Chapter 1: Introduction

The hippocampus is known to be essential for the creation of declarative, or fact-based short-term memories. Information is stored by transferring processed spatio-temporal neurological activity from the entorhinal cortex (EC) region to the CA1 region [1], where the information is projected into long-term memory. The anatomical location of the CA1, CA3 and EC region is shown in Figure 1, along with a graphic illustration of what neurological spatial-temporal activity may look like in a rat hippocampus [2], [3].



**Figure 1:** (a) Anatomy of the rat hippocampus, illustrating the transfer of electrical signals between regions in the hippocampus [2]. The electrical signal will start at the EC region go through the CA3 and CA1 regions, then loop back to EC region where the information carried by the signal is projected into long-term memory. (b) A computer simulation illustrating the possible firing of an ensemble of neurons in a rat hippocampus, demonstrating the spatial aspect of neurological activity as an ensemble of neurons consisting of individual neurons firing in relation to one another [3].

The process of information propagating through the rat hippocampus can be described in terms of its spatial and temporal properties when considering the neurological activity of an ensemble of neurons. The spatial component of neural activity can be described by an ensemble of neurons that consist of individual neurons firing in relation with one another, while the temporal component considers the influence that previous neural activity has on the current predicted output at a point in time.

When determining a time series prediction of rat behavior from neural activity in the hippocampus, there are many factors that have to be taken into consideration, such as the effect an animal's location and direction of movement has on the spatio-temporal firing patterns of hippocampal neurological activity [4]. By altering the environment, certain neural properties may change, such as firing location, peak firing rate and field overlap of neural activity due to the neurological network adjusting to task conditions [5]. However, there are aspects of cognition that depend on the brain to self-generate sequential (temporal) neural activity [6], which is independent of where the animal is spatially located in the environment. Thus, factors that influence the dynamic properties of neural activity is a combination of both the external environment and internal generation of electrical signals in a rat hippocampus. Dynamic properties that hippocampal neurons may exhibit, include axonal conduction speed, synaptic strength of neural connections, firing rate and inhibition affecting post synaptic spike patterns [5]. Additionally, the mechanisms of electrophysiological activity is inherently nonlinear, where the spike trains of neurons are not additive and are shown to have varying amplitudes depending on the temporal pattern of the previous input [7]. This demonstrates the need of a nonlinear and dynamic system to predict the behavior of a rat from recorded neurological activity.

### **1.1 Statement of Problem**

Damage to the hippocampal region from events, such as a stroke or head trauma, can impair electrophysiological transfer of information and disrupt the spatio-temporal patterns between hippocampal regions possibly causing amnesia or impairing long-term memory formation [8]. Alzheimer's disease and other forms of dementia are typically

associated with loss of neurons in the EC region, the region that provides input to the hippocampus [9]

Understanding the electrophysiological signal encoding of hippocampal neurons from the EC to CA1 region in a rat hippocampus may provide insight concerning the relation of electrophysiological activity between the EC and CA1 hippocampal regions and rat behavior, which has not been as extensively studied as electrophysiological activity between the CA3 to CA1 region [1], [8]. This thesis aims to use neurological patterns simultaneously recorded from the EC and CA1 region in a rat to identify whether the neurological activity is associated with wheel running, T-maze or open exploration behavior. The spatial coordinates and velocity attributes of the identified behavior are then predicted. Possible applications of this thesis, include improving deep brain stimulation therapy for neural implant devices that stimulate based on the pattern of neurological activity [10], advancing research in brain-to-machine-interfaces [11], and providing additional insight in how neurological activity might be related to behavior.

Review of literature suggests that research interests focus on identifying rat behavior or predicting neurological activity in certain regions of the hippocampus. However, by applying a model that can both identify and predict the behavior of the rat given a spatio-temporal representation of the EC and CA1 region, the range of possible applications for this research increases.

## **1.2 Research Questions**

Questions that this thesis addresses and raises include the following: Can T-maze, wheel running and open exploration behavior of a rat in laboratory conditions be identified with neural networks using input data that consists of cluster ID and shank ID

to represent the spatial component of hippocampal neural activity? Additionally, can neural networks predict velocity and spatial coordinate attributes of T-maze, wheel running and open exploration behavior? Answering these questions may give insight on possibilities for recovering functions of damaged brain regions, controlling behavior through external electrical stimulation, and enhancing the performance of artificial intelligence.

### **1.3 Issues and Assumptions**

The data was obtained online from an external source [12]. Assumptions the researchers made during the gathering of this data is unknown and may not be coincident with the assumptions that were made in this thesis. Results shown in this thesis assume that the shanks, during the live gathering of neural activity, did not drift between sessions and the rat's state of mental and physical health were constant throughout all of the sessions. Additionally, it was assumed that one behavior would not have an influence on another behavior's neurological activity, if multiple behaviors occurred during the same session. The behaviors that can occur in one session include T-maze and wheel running, which could possibly cause the neural activity of the rat to be different from a rat that only experiences either T-maze or wheel running behavior during a session.

Recurrent neural networks are utilized in this thesis to map the dynamical process between rat neurological activity and T-maze, wheel running and open exploration behavior. Although neural networks are non-parametric, requiring the user to know nothing about the generation of the signal, there are still some difficulties that present themselves while pre-processing the input data and determining the parameters of the network structure. Pre-processing the input data is needed, in most cases, for good



prediction to occur, such as removing linear and nonlinear trends and periodic patterns [13]. The only pre-processing done in this thesis was normalization, which scaled all of the data between the values of -1 and 1. Difficulty with the network structure involves choosing an appropriate input delay, which represents the amount of previous input time points that have a direct impact on the next output, and number of hidden neurons of the network. Choosing the parameters pose difficulties due to a lack in deterministic processes to calculate the parameters of the network. Parameters of the network are commonly found through heuristics that change depending on the application of the network and trial-and-error approaches [14].

Modifying network parameters can enhance or degrade the performance of the network in various ways. By increasing the number of hidden neurons, the network's ability to solve complex problems increases, however, too many hidden neurons can cause the network to model the noise in the input and provide poor results for data outside of the training interval. Increasing the number of input delays can give the network the ability to recognize patterns over a longer period of time, however, this can be a limitation if the network recognizes patterns that are nonexistent between input and output pairs, generally resulting in poor prediction.

#### **1.4 General Approach**

The general approach entails three stages, the pre-processing stage, general behavior identification stage and behavioral attribute stage. Stage 1 (pre-processing stage) decomposes the rat's raw neurological activity into clusters that represent neurons. Stage 1 was done by other researchers [12], but is a necessary step in obtaining the input used in stage 2 and 3. In Stage 2 (general behavior identification stage), the resultant

cluster ID and shank ID input data obtained from stage 1 are used as input for the identification network to determine whether the general behavior associated with the input data is a T-maze, open exploration or wheel running task. In Stage 3 (behavior attribute prediction stage), velocity and spatial coordinate attributes of the identified behavior, from stage 2, are determined using the same cluster ID and shank ID input data that was used to identify the behavior.

This approach uses four Nonlinear autoregressive processes with exogenous inputs (NARX) networks, one for general behavior identification and three additional networks for predicting the velocity and spatial coordinate attributes for the identified behavior. Each network is trained with data that is specific to the networks application. Neural networks are used as the general approach in this thesis because a neural network has properties that represent a biophysically-realistic spiking network and employ continuous variables that may be used to represent the temporal component of neural activity [15].

## **1.5 Organization of Thesis**

Chapter 2 considers other works that have used similar methods for other applications or alternative methods to analyze the dynamics of neural activity. Additionally, advantages and limitations of using a neural network approach for applications involving neural dynamic systems are discussed. In chapter 3, the overall system design is explained with each stage discussed in more detail. There is also a section that considers implementation of the model into MATLAB. Chapter 4 describes testing and evaluation of networks with respect to hippocampal data and discusses overall finding and results. This thesis concludes with Chapter 5 which includes a summary of

the methods, results and contributions of this work. Additionally, there is a future works section, which describes how the results might be improved and what further applications this work might inspire.

## **Chapter 2: Literature Review**

### **2.1 Background**

Understanding the mechanisms underlying neural activity and transfer of information in the hippocampus has been a research topic of interest for neural prostheses applications involving sensory input, motor control or cognitive function [8]. Many aspects of neurological activity have been extensively researched, including mechanisms underlying synaptic transmission and generation of electrical activity for individual neurons [8], transfer of information based on spatio-temporal neural activity of neuron ensembles in the CA3 to CA1 region in rats [1], [5], [8], [16], and transfer of information in the prefrontal cortex in nonhuman primates [15]. These studies contribute to modeling dynamic properties of neural activity and behavior associated with neural activity. However, a review of the literature suggests there is little research focused on identifying multiple behaviors from the same brain region and determining specific attributes of those behaviors (i.e. velocity and spatial coordinates).

Previous research has shown that spatio-temporal patterns observed in hippocampal neural activity is a result of the inner mechanisms associated with intrinsic networks and the external environment of a rat [6], [16]. Additionally, the parameters of electrical activity that are dynamic or stay constant in response to environmental perturbations has been previously determined [5]. The parameters of neurons shown to be dynamic in response to an environmental stimulus include the firing location and firing rate, thus providing further evidence that neurological activity can be effectively represented by using spatial and temporal components of electrophysiological signals.

## **2.2 Traditional Methods**

Due to a high dimension of nonlinearity in neural activity, alternative methods typically consist of orthogonality techniques to decrease the dimensionality and number of parameters needed to interpret, analyze or predict neurological activity [17]. The most extensively used method includes variations of Volterra integral methods, which provide a general mathematical representation of the nonlinear dynamics involved in signal transformation and coding by class of neural systems [18]. This method has an orthogonalization scheme to reduce the dimensionality of the problem, thus reducing the parameters needed to model the given input. However, implementing orthogonality techniques for nonlinear biological systems may be difficult for researchers who are unfamiliar in using Volterra kernels for modeling nonlinear systems, which may require the use of higher-order complex terms.

Using a neural network approach allows for a non-parametric method for mapping a time-series input to an output, where no information concerning how the signal was generated is necessary, thus, providing a more accessible way to interpret and analyze nonlinear biological systems.

## **2.3 Neural Networks**

A variety of neural networks have been used for neurological applications. Radial basis function (RBF) recurrent networks have been used to successfully predict epilepsy up to 60s before occurrence [19] and have also been used to transform rat brain signals into robotic movement [20]. Three layer artificial networks have been shown to solve simple spatial tasks similar to those that are handled by the hippocampus, which allowed the network to determine an optimum path using relative landmarks [21]. And Hopfield

neural networks have been utilized to model auto-associative and hetero-associative memory of the hippocampus [22]. Thus, networks have been used to interpret and analyze hippocampal neurological activity, mimic hippocampal functions and form brain-to-machine interfaces (BMI).

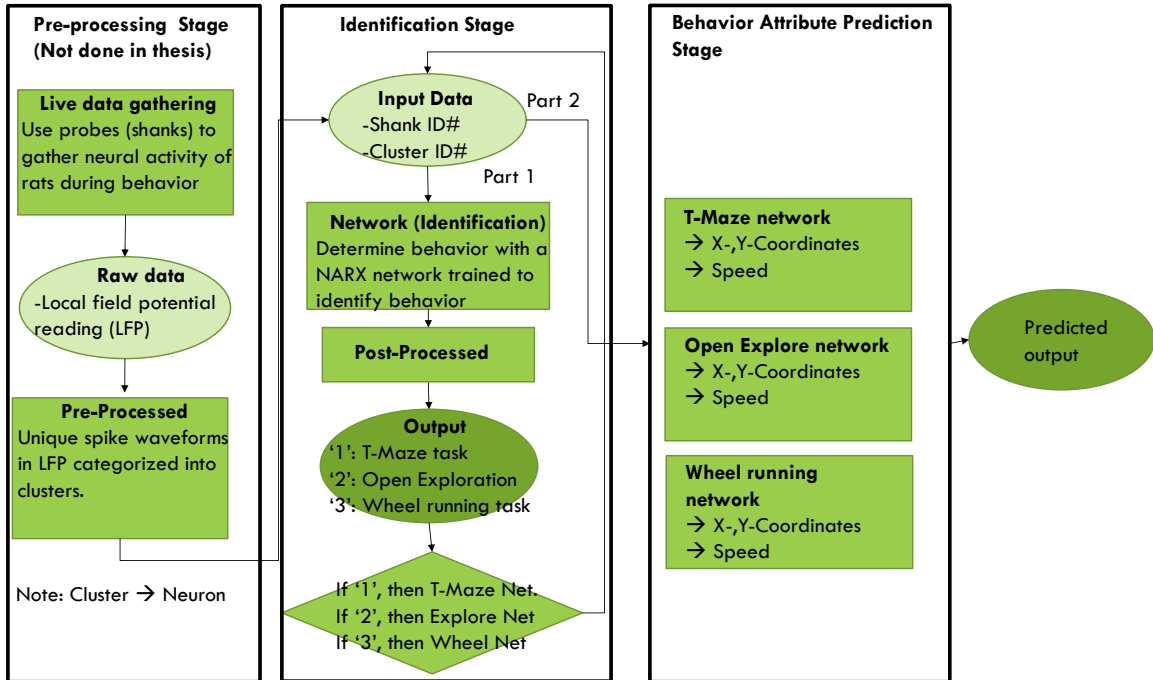
The model proposed in this thesis is best compared to modeling a chaotic time series using a NARX network [13]. A NARX network is essentially a feedforward network with input delay and output delay, or recurrence. This network has been shown to have potential in modeling nonlinear dynamic systems and is typically faster in converging to a solution for a complex dynamic and nonlinear problem [13].

## **Chapter 3: System Design**

This chapter discusses the overall design through illustration and explanation of the three stages: pre-processing, identification and attribute prediction, in detail. Each stage has its own section regarding the contents and processes the respective stage entails.

### **3.1 Three Stage Design**

The three stage design consists of importing cluster ID and shank ID data, that represents the neurological activity of neuron ensembles, as input to a NARX network that identifies the general behavior associated with the input data as a 1, 2 or 3 representing T-maze, open exploration, or wheel running behavior, respectively. After identification, the input data is sent to another NARX network that predicts the velocity and spatial coordinate attributes of the identified general behavior. Thus, a total of four NARX networks are used in this system to account for all the possible variations of input data. This thesis only considers predicting behavior attributes for one identified behavior, rather than multiple identified behaviors in a testing session. The diagram in Figure 2 illustrates the overall design and process of this thesis.



**Figure 2:** The overall design consists of three stages: pre-processing stage, identification stage and behavior attribute prediction stage. The pre-processing stage decomposes filtered LFP spike waveforms into cluster ID data. The identification stage uses the cluster ID and shank ID data as input for the identification network, which categorizes the input into an output of 1, 2 or 3 representing T-maze, open exploration or wheel running behavior, respectively. The behavior attribute prediction stage uses the same input used to identify the general behavior to determine the velocity and spatial coordinate attributes of the behavior identified in the identification stage.

### 3.2 Pre-Processing Stage

The pre-processing stage takes the recorded local field potential and decomposes it into clusters that represent individual firing of neurons. This stage was done by other researchers [12], but is a crucial step in obtaining data that may be representative of neuronal spatial firing in the hippocampus. Methods on how to spike sort LFP signals to obtain cluster ID data is briefly discussed in this thesis as the first stage.

#### 3.2.1 Raw Data Gathering

Local field potential (LFP) data was gathered using 16 probes (shanks) that simultaneously recorded neural activity of the CA1 region, in the left and right

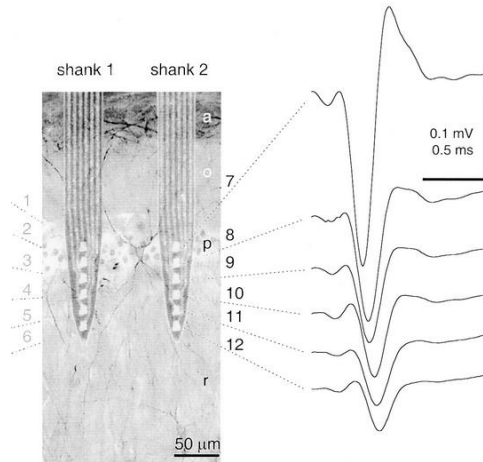


hemisphere, and EC region in the right hemisphere of the rat hippocampus, while the rat performed T-maze, wheel running or open exploration behavior. Table 1 outlines the relative location of a shank with the corresponding shank ID number. Each of the 16 shanks have 8 recording sites, allowing for a total of 128 channels to gather neural activity from the CA1 and EC region. The LFP is a representation of the raw data that was recorded using the shanks, where each shank ID is associated with a specific color.

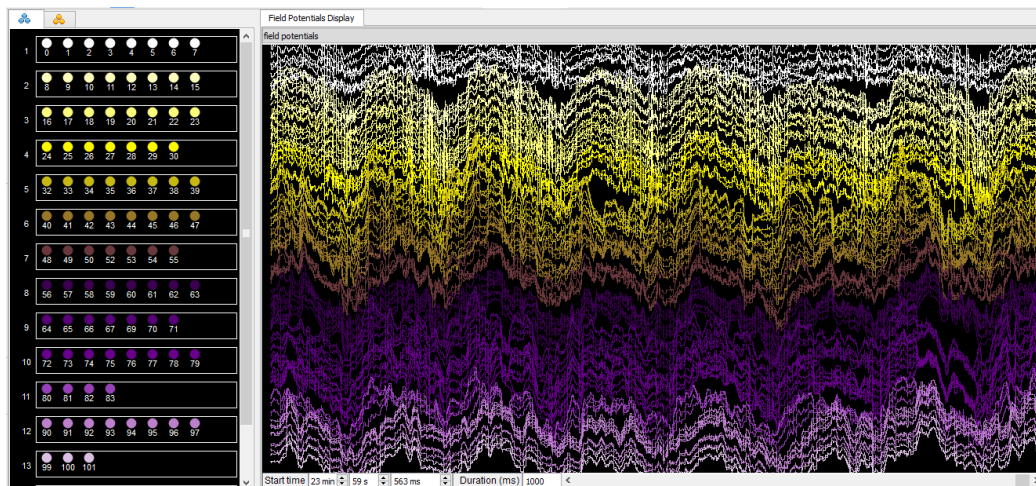
**Table 1:** General location of rat hippocampal neural activity associated with shank ID number

<b>Shank ID#</b>	<b>Location in rat hippocampus</b>
1-4	Right EC
5-8	Right CA1
9-16	Left CA1

Figure 3 is an example of an LFP recorded from a rat hippocampus during wheel running activity for a 1000ms time interval. There are eight distinct lines of one specific color representing the eight recording sites of a particular shank.



(a)



(b)

**Figure 3:** (a) An example of a shank with six recording sites on it. Illustrating how one shank may gather multiple signals at once [23]. (b) A sample LFP of a rat performing wheel running task for a 1000ms interval. Every shank is represented by a unique color.

### 3.2.2 Spike Sorting

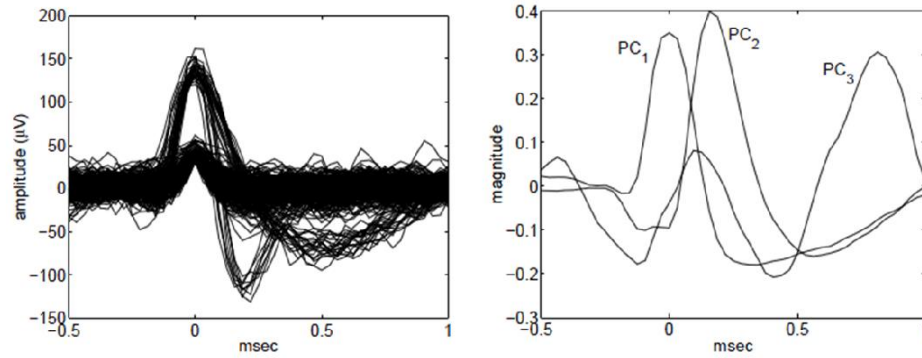
Spike sorting is the general process of relating a unique spike waveform to an individual neuron, the neurons are represented by clusters which are optimally identified. The primary spike sorting technique used to decompose the LFP spike waveforms into clusters is called principal component analysis (PCA). PCA reduces the dimensionality of a spike waveform into three components, which represent the features that exhibit the

greatest variability between spike waveforms, then by plotting the principal components against one another formations of specific clusters are observed. When spike sorting, the following assumptions are typically made:

1. Every unique spike waveform is associated with the firing of a specific neuron.
2. The spike waveform associated with a neuron is not changing over time.
3. Neurons are firing at a constant frequency.

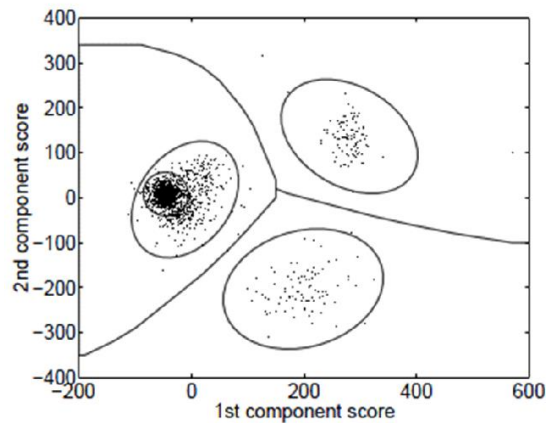
Realistically, the assumptions made are not necessarily true, but since the spike waveforms and frequency of any given neuron are dynamic and change with time, the above assumptions are typically used for consistent identification of individual neurons or clusters.

Amplitude thresholds were used to filter out extracellular waveforms that might represent noise or smaller spikes from neurons in the local region, this step is used to separate the neuron of interest from background noise and weak neural signals from neurons not in the local region, before the use of PCA [24]. For PCA, features are automatically chosen by using a program to find an ordered set of orthogonal basis vectors that capture the largest variations in data. Figure 4 demonstrates the separation of spike waves into principal components.



**Figure 4:** Example of aligning spike waves and finding the resulting principal components [25].

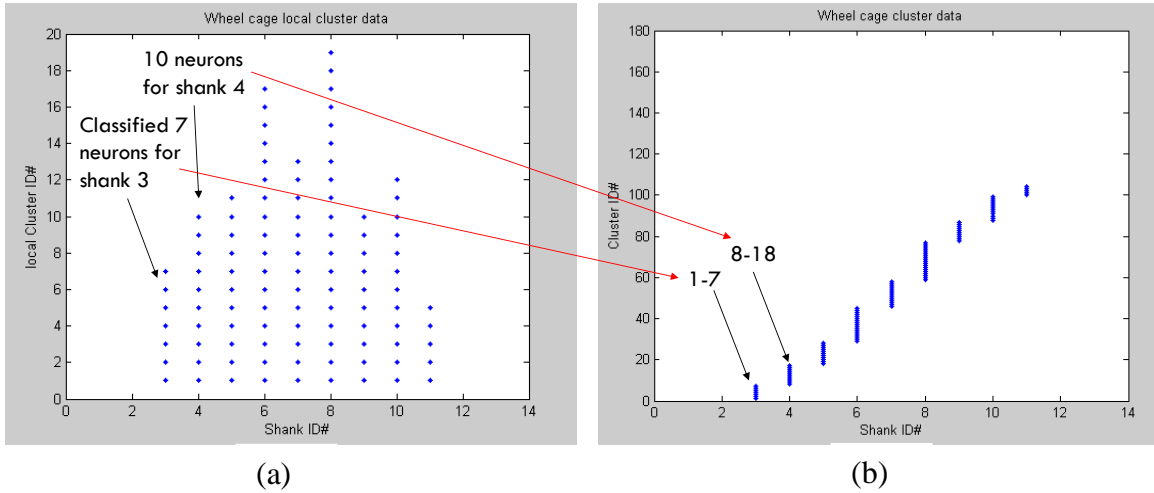
Individual neurons for the spike waveforms shown in Figure 4, can be found by plotting the principal components against one another and forming boundaries around clusters that might represent the firing of a specific neuron, which is illustrated in Figure 5.



**Figure 5:** An example of creating boundaries around clusters to identify the unique neurons in a system [25].

Using the spike sorting process shown above, cluster ID data was obtained for each shank. The spatial component for neurological activity will be represented by the

global cluster ID as a function of the shank ID for the remainder of this thesis. The relation between local cluster data and global cluster data is shown in Figure 6.



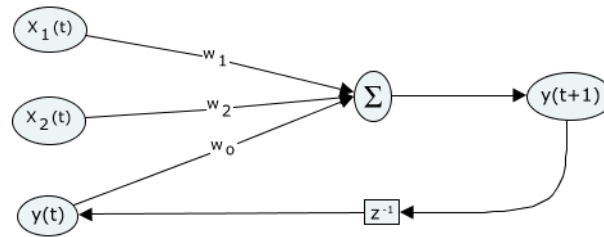
**Figure 6:** Cluster data for a rat performing wheel running behavior. (a) The local cluster data illustrating how many neurons an individual shank recognizes. (b) The cluster ID# arranged from 1-128, ordering cluster ID sequentially from shank ID 1 to shank ID 14.

### 3.2.3 Forming Temporal Component of Input Data, Illustrative Example Using Simplified NARX Network

Information of specific neurons (cluster ID) recorded by a shank at a specific point in time forms the spatial component of the input data. The spatial component of the system, represented by the input data, is given temporal properties by considering the influence of input data (cluster ID and shank ID) and output (general behavior) at one point in time has on the output at another point in time. Input and output was arranged into a time-series, where an input and corresponding output was defined for every time point. Thus, the temporal component is represented through the process of feeding input to the neural network, such that the previous input and output data has an effect on the next output being predicted. The rest of this section demonstrates how the temporal

component is formed through the network with a simplified example of sequentially introducing four cluster ID and shank ID data points into a NARX network.

Consider the nonlinear autoregressive process with exogenous inputs (NARX) network in Figure 7, where the next output depends on the current input data and the previous output. Note, the nonlinear hidden layer was removed for demonstration purposes, however, any properties exhibited by the simplified network apply to the three-layer structure networks used for identification and behavior attribute prediction as well.



**Figure 7:** A simplified NARX network used to illustrate the temporal component. The input  $x_1$  and  $x_2$  represents shank ID and cluster ID, respectively. Weights ( $w$ ) are used to factor the importance of a particular input or previous output and are optimally determined by training the network with a representative dataset. For this example, input weights ( $w_1$  and  $w_2$ ) are set to unity, value of 1, and the weight of the previous output ( $w_0$ ) is 0.5. This network calculates the next output by linearly summing the weighted current input and weighted previous output. The nonlinear hidden layer of the network is removed to simplify the example.

The output for the simplified NARX network shown above can be described by the following equation:

$$y(t + 1) = x_1(t) * w_1 + x_2(t) * w_2 + y(t) * w_0 \quad [\text{Eq. 1}]$$

For this example the weights are:

$$w_1 = w_2 = 1$$

$$w_o = 0.5$$

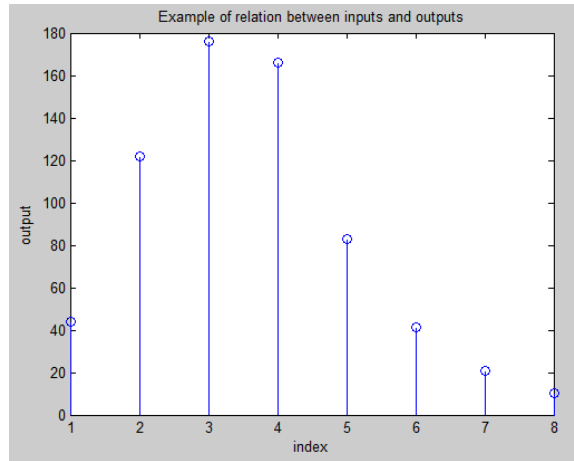
The weights ( $w$ ) scale a particular input based on the significance the input has on the next output. For this example, the cluster ID data and shank ID data are weighted equally and the previous output is weighted half as much as the input.

Using 4 time steps from the input data in Figure 6 gives the example input data shown in Table 2. Calculation of the output is carried out for four additional steps with no input data to demonstrate that the network still holds memory of previous data, which the network uses to predict the next value.

**Table 2:** Four time steps of input data with four additional steps of no input

$X_1$ (Shank ID)	4	9	11	7	0	0	0	0
$X_2$ (Cluster ID)	40	91	104	71	0	0	0	0

The example data in Table 2 is sequentially presented to the simplified network shown in Figure 7 to produce the results seen in Figure 8.



**Figure 8:** The resulting output calculated by the network using eight time steps. This plot illustrates that the calculated output is the sum of the input and half the previous output at any given time point. However, when there is no further input, the network still retains half the value of the previous time point.

The simplified NARX network still calculates the next output value when there is no input and retains half the value of the previous time point. The NARX network's ability to store memory of previous output data was essential to form the temporal component of neural activity. The associative memory properties of the NARX network demonstrate that it may have the capability to map complex dynamic and nonlinear neurological activity to a behavior. All of the properties demonstrated for a simplified NARX network apply for a NARX network with a nonlinear hidden layer as well.

In summary, the cluster ID and shank ID input data presented to the network form the spatial component of neurological activity, while the temporal component of neural activity is represented by the sequential loading of input data into a NARX network. In other words, the input data is the spatial component and how the input data is processed in the network forms the temporal component.



### 3.3 Identification Stage

The identification stage uses the cluster ID and shank ID obtained from the pre-processing stage to determine whether the general behavior is a T-maze, wheel running or open exploration behavior represented by a value of 1, 2 or 3, respectively. The result of the identification network is post-processed to determine the most likely value that is predicted by the NARX network.

#### 3.3.1 Identification Network Parameters

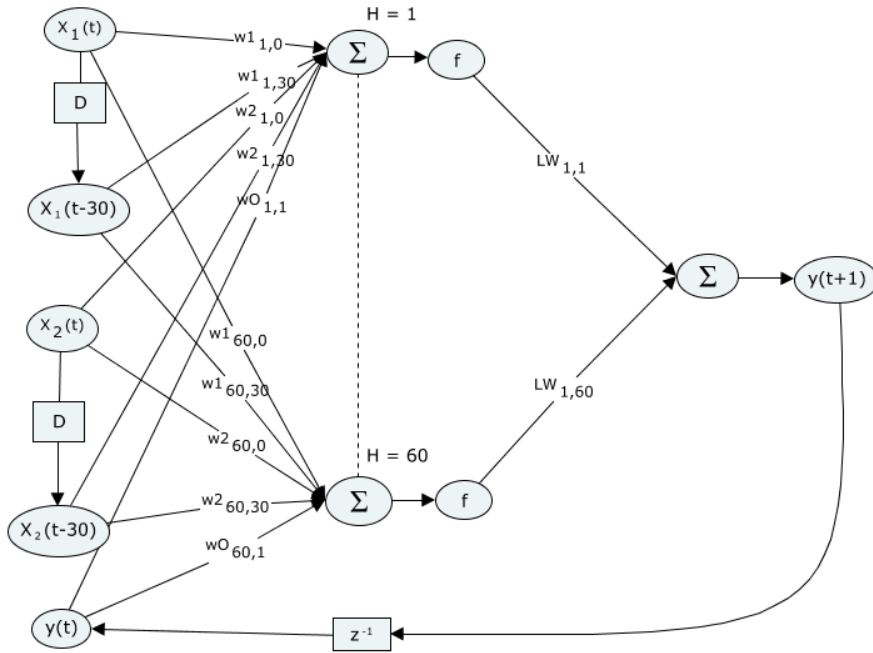
Possible modifiable NARX network parameters include the input delay, output delay, number of layers, number of hidden neurons and the nonlinear activation function of the hidden layer. The network for identifying behaviors had best results when a ratio of 1:2 for input delay and number of hidden neurons was used. However, below an input delay of 25, the network was generally not able to identify behaviors within the training interval.

The activation function for the hidden layer is chosen based on the application of the network. Only nominal values of 1, 2 and 3 representing T-maze, open exploration and wheel running behavior, respectively were desired, thus a hard-limiting nonlinear activation function was used for identification. The parameters that yielded the best results with the data used in this thesis are shown in Table 3.

**Table 3:** Parameters of the NARX identification network

<b>Number of layers</b>	3 (input-hidden-output)
<b>Number of hidden layer neurons, H</b>	60
<b>Input delay</b>	30
<b>Output delay</b>	1
<b>Hidden layer activation function, f</b>	Hard limiting

Figure 9 illustrates a NARX network with the parameters outlined in Table 3.

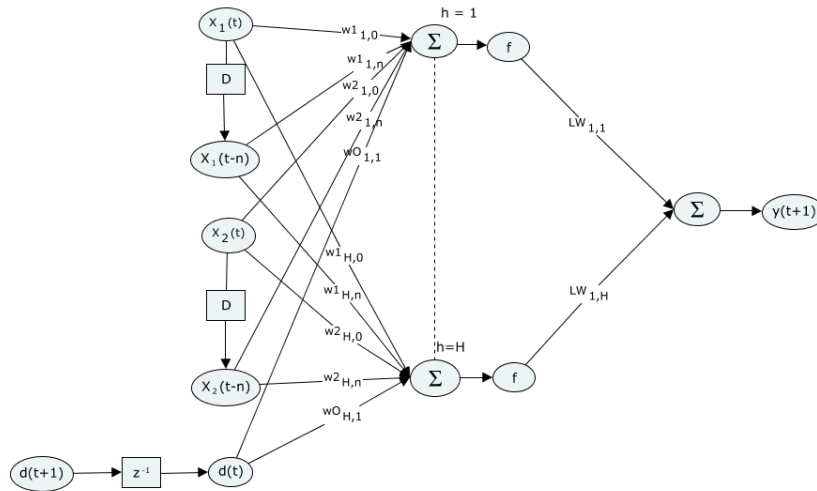


**Figure 9:** A diagram of the NARX network used for identification of T-maze, wheel running or open exploration behavior. Both of the inputs have a delay of 30, where “D” represents a time-series delay, which implies the other 29 previous values of input and their weights not illustrated. The output has a single delay ( $z^{-1}$ ) that feeds back the previous output as input to the hidden layer. All of the inputs are fully connected to the hidden layer.

### 3.3.2 Identification Training Methods

The identification NARX network was trained with 5000 time points of samples consisting of input (cluster ID and shank ID) and desired output (identification number) data for T-maze, open exploration and wheel running behavior data from an early portion of each behaviors respective dataset. Taking time points from early portions of a dataset was done for practical purposes, to simulate a situation where researchers only had to train the network with the first few seconds of gathered data, then move on to testing. .

A series-parallel model of the NARX network was used for training. A series-parallel NARX network uses the previous desired output value as input, rather than feeding back the output value the network calculates. Training with this network model allows for errors in network training to occur and still provide satisfactory training results, a diagram of the training process is shown in Figure 10.



**Figure 10:** A general NARX network in series-parallel mode. A series parallel mode uses the previous value of the desired output, rather than the previous value of the network output, to avoid using incorrect network output as input for training.

After training, the identification network was tested with input data outside of the training interval to determine how well the network can generalize to other time points that it has not been trained for. A parallel model of the NARX network was used for testing, where the calculated output of the network feeds back as input to the hidden layer, model previously shown in Figure 9. See Appendix A for more information regarding the theory of the learning process.

Note that the input data is split between training, validation and testing within the training procedure. After the network reaches a minimum error value goal validation will occur and if the network gets a wrong value for the input sample, then it will automatically go back to training. Because the sample data is randomly split between training, validation and testing, there are a large variety of results that can occur even when the network is tested for a time interval that it has been trained for.

### **3.4 Behavior Attribute Prediction Stage**

This stage determines the spatial coordinates and velocity attributes for the behavior that was identified in the previous, identification, stage. T-maze, wheel running and open exploration behavior has its own respective NARX network for behavior attribute prediction. The use of multiple networks within the behavioral attribute stage was necessary to predict multivariate output for the identified behavior. Each network has the same parameters and structure, but trained with data representative of a specific networks behavior. The behavior attribute prediction network uses the same input data as the identification network during testing.

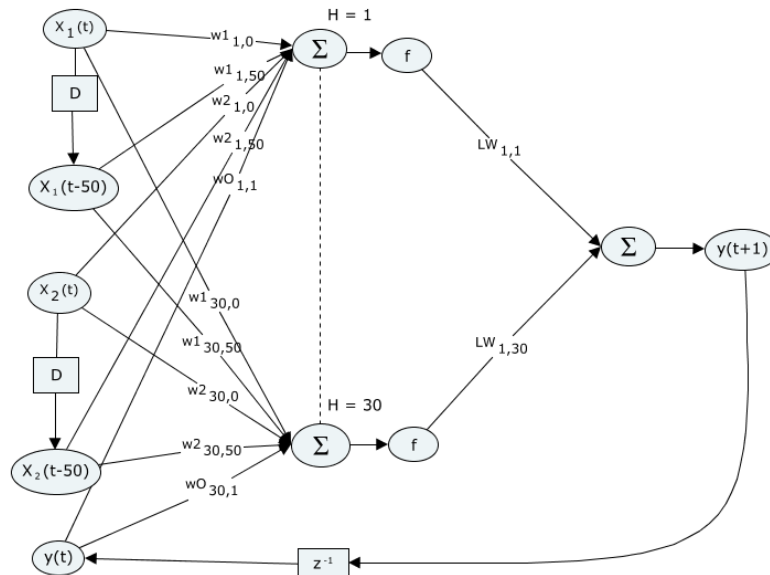
#### **3.4.1 Behavior Attribute Network Parameters**

Possible modifiable behavior attribute NARX network parameters include the input delay, output delay, number of layers, number of hidden neurons and the nonlinear activation function of the hidden layer. Best results were obtained using 40-60 input delays and 30-50 number of hidden neurons, Table 4 outlines the parameters of the behavioral attribute network.

**Table 4:** Parameters for every behavioral network

<b>Number of layers</b>	3 (input-hidden-output)
<b>Number of hidden layer neurons</b>	30
<b>Input delay, dx</b>	50
<b>Output delay, dy</b>	1
<b>Hidden layer activation function, f</b>	Sigmoidal

Parameters of the behavioral attribute network that differ from the identification network include the nonlinear activation function,  $f$ , of the hidden layer, which was changed to sigmoidal for attribute prediction. By changing the nonlinear function to sigmoidal, a range of values can be predicted, rather than nominal values. Additionally, the number of hidden neurons and the input delay was changed and chosen through trial-and-error. The structure of the model, shown in Figure 11, looks much like the identifier stage NARX network.



**Figure 11:** The NARX model with the parameters of the behavior attributes network implemented. Input consists of the cluster ID and shank ID at the current time-point and the previous 50 inputs along with the previous output. The input is fully connected to the

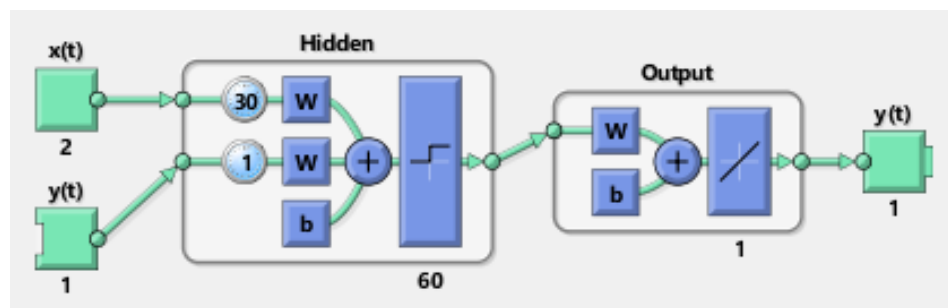
30 neurons in the hidden layer where the input is nonlinearly transformed, then linearly summed at the output.

### **3.4.2 Behavior Attribute Training**

The behavior attribute stage NARX network was trained and tested similarly to the identification stage network. The network was trained in series-parallel (open) mode, which allowed the next output value to use the previous desired output value as input, rather than the output value calculated by the network. Training data consisted of a large range of possible velocity and spatial coordinates the rat could exhibit during a session. After training, the network was converted to parallel (closed) mode where the output calculated by the network was used, rather than the desired output. Closing the network allowed for a multi-step ahead prediction of the spatial coordinates and velocity. Testing was done with new input data outside of the training interval, consisting of cluster ID and shank ID input data, the network was not previously trained for. The output of the network is compared with the desired output to determine whether the methods in this thesis are applicable.

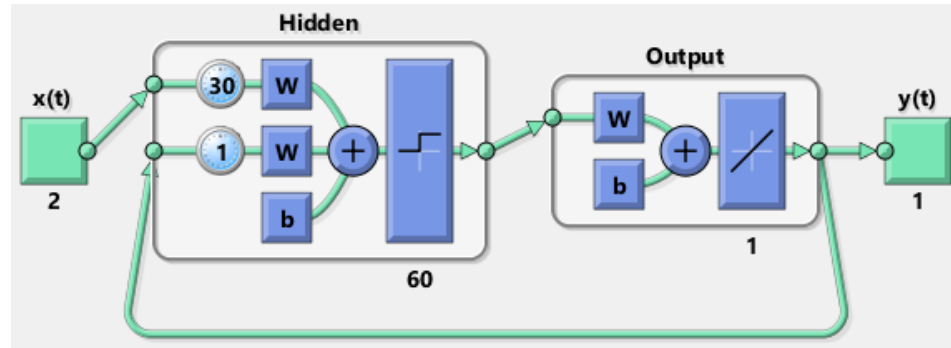
### 3.5 System Implementation

The overall design proposed in this work is implemented in MATLAB. This program was used due to availability and convenience of implementation. The identifying and behavior attribute NARX networks were designed using the neural networking toolbox (NNT) in MATLAB, Appendix B contains the code used when implementing the neural networking toolbox. The NNT has two different modes for a NARX network, consisting of open and closed. When the network is in open mode, it will delay and use the desired output as input to calculate the next output, a MATLAB illustration of the open network structure is shown in Figure 12. This is a one-step ahead prediction model suitable for training the network.



**Figure 12:** An open mode NARX network illustrated in MATLAB, where the previous desired output is used, rather than the calculated output of the network. Using the desired output as input to the network allows the network to make mistakes during training, but still converge to a solution.

After training, the network was closed and the previous calculated output of the network was fed back as part of the input, Figure 13 illustrates the MATLAB model of the closed loop structure. This NARX network mode was used for testing and has the ability to predict multiple steps ahead, whereas open loop mode can only predict one-step ahead.



**Figure 13:** A closed-loop mode NARX network with the same parameters used in the identifier network. This model is used for testing the network with new data that it has not been previously trained for.



## Chapter 4: Testing and Evaluation

Testing consists of two parts: self-testing with new input data from the same dataset the network was trained with and cross-testing with new input data from a different dataset that the network was not trained with. Datasets include two sessions of T-maze and wheel running behavior and two sessions of open exploration behavior. T-maze and wheel running behaviors are combined due to both behaviors taking place in the same cage, making it possible for both behaviors to occur in one session. Different combinations of the four sessions creates four different datasets for testing identification and two different datasets for testing attribute behavior networks. A summary of the combinations of sessions used to form each dataset is shown in Table 5.

**Table 5:** Possible datasets that can be used to train or test the identification network

<b>Open Exploration Session</b> <b>T-Maze and Wheel Run Session</b>	<b>Session 1:</b> Large 2D platform exploration, 60 min (252,925 time points)	<b>Session 2:</b> Large 2D platform exploration, 60 min (3,166,564 time points)
<b>Session 1:</b> T-maze: 35 min Wheel runs: 20sec intervals (998,725 time points)	<b>Dataset 1</b>	<b>Dataset 3</b>
<b>Session 2:</b> T-maze: 23 min Wheel runs: 20sec intervals (897,776 time points)	<b>Dataset 2</b>	<b>Dataset 4</b>

### 4.1 Testing Identification Network

Effectiveness of the identification network was tested using two methods, self-testing datasets and cross- testing datasets. For self-testing datasets, the network was

trained with early time points of a dataset, then tested with later time points from the same dataset used to train the network, this tested how far outside the training interval the identification network remains effective when identifying behavior of the same dataset. Cross-testing datasets involved using one dataset to train the network and another dataset to test the network, this tested for generalization of the identification network across sessions. For both testing methods, the identification network is always trained using one dataset, rather than multiple datasets at once.

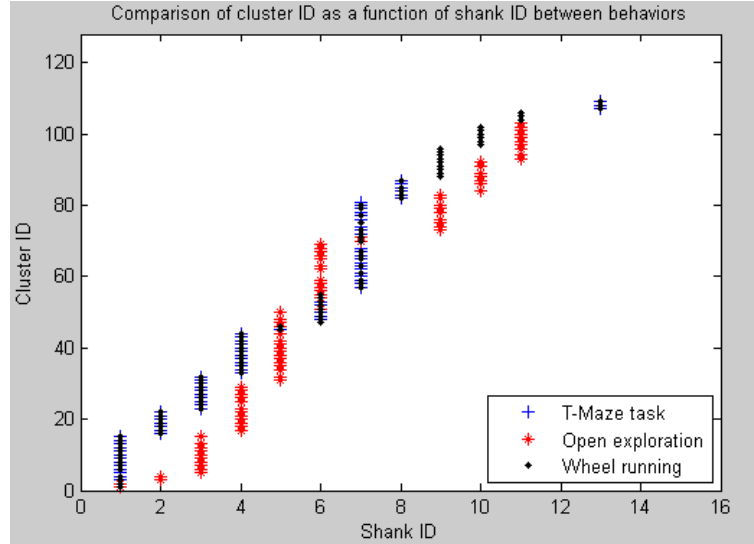
#### 4.1.1 Self-Testing Datasets For Identification Network

Self-testing datasets involved using one dataset for both training and testing. Training data consisted of 5000 early time points of cluster ID and shank ID input data and desired output for T-maze task, wheel running and open exploration behaviors, amounting to a total 15000 input-output pairs used for training. Table 6 summarizes what identification number is associated with each behavior.

**Table 6:** The number of training sample points used for each behavior from dataset 1 to train the identification network

<b>Behavior</b>	<b>Identification</b>	<b>Training Sample points</b>
T-Maze	1	5000
Open Exploration	2	5000
Wheel Run	3	5000

Plotting all the time points of the input data from Table 6 demonstrates that cluster ID as a function of shank ID between behaviors can closely overlap or be easily distinguished, see Figure 14.



**Figure 14:** A plot of all 15000 time points of input data used to train the identifier NARX network. The input data consists of cluster ID as a function of shank ID for each behavior. Input data of the open exploration is well distinguished from T-maze and wheel running data, while the T-maze and wheel running input closely overlap.

Although the amount of time points for each behavior during training and testing were the same, the actual time period between behaviors would differ. A time point refers to a recognizable spike of an individual neuron at a point in time, while the time period refers to a time span of the behavior in seconds. Table 7 describes the time period of the training and testing interval of the input data shown previously in Figure 14. Where training interval is the time between the first and last time point of the training set, time gap is the wait time between the last time point of the training set and the first time point of the testing set and testing interval is the time between the first and last time point of the testing sample. Note, there is a larger training interval period for open exploration behavior compared to T-maze or wheel running behavior when the same number of

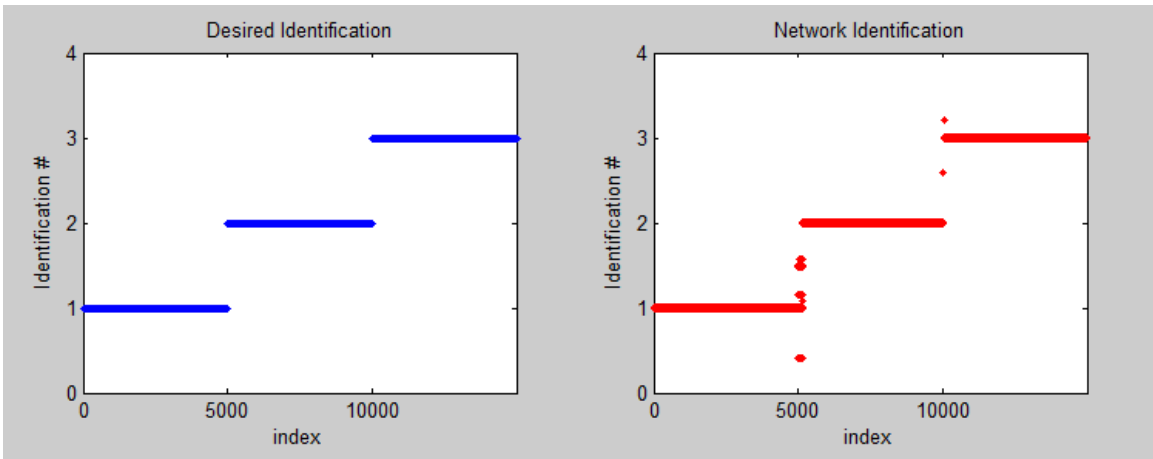
training time points were used for all three behaviors. This implies that there was a lower frequency of spikes during open exploration behavior than there were for T-maze or wheel running.

**Table 7:** The time interval of the training data, time gap between training and testing and testing data

<b>Dataset 1- Test 1</b>	<b>T-Maze</b>	<b>Open Exploration</b>	<b>Wheel Run</b>
<b>Training Interval (s)</b>	8.40	12.4	7.35
<b>Time gap (s)</b>	30.2	41.8	50.2
<b>Testing Interval (s)</b>	5.10	7.10	6.10

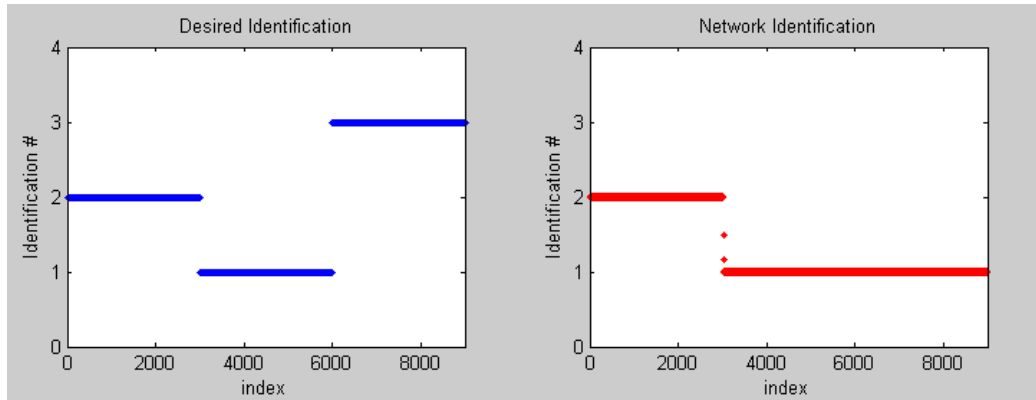
All of the training time points were taken early on in the dataset to try and replicate a real event, where only the first few seconds of behavior would be needed for training, then succeeding time points can be identified without knowing the actual behavior of the rat. The next output is dependent on the previous values of input and output, thus, it is necessary to use sequential training samples consisting of successive time points. Training with sequential time points, rather than random time points, makes it possible for the network to find a pattern in the input that can be associated with the desired output.

After the network was trained using a series-parallel structure, which allowed the network to use previous desired output, rather than the previous network output. Figure 15 illustrates the training results of the network, while the closed network structure was tested using the interval of data that it was trained with, to determine if the network was trained well. A good training session would result in the network output closely matching the desired output, but not completely. A small deviation from the desired output shows that the network can be generalized to other data.



**Figure 15:** Identification results for the input used to train the network. The desired identification for the corresponding input data (left) includes 5000 time points for each behavior, where the first 5000 time points is T-maze, time points between 5001-10000 is open exploration and the last 5000 time points is wheel running. The network identification (right) demonstrates that the network can effectively identify input data from within the training interval.

A successfully trained identification network with good generalization would be able to distinguish between all three behaviors inside and outside the training interval, however, networks that showed appropriate distinction between all three behaviors within the training interval would typically confuse two behaviors as one outside of the training interval, thus, only two behaviors outside of the training interval were correctly identified. The results from using behavior data outside of the network training interval are shown in Figure 16.



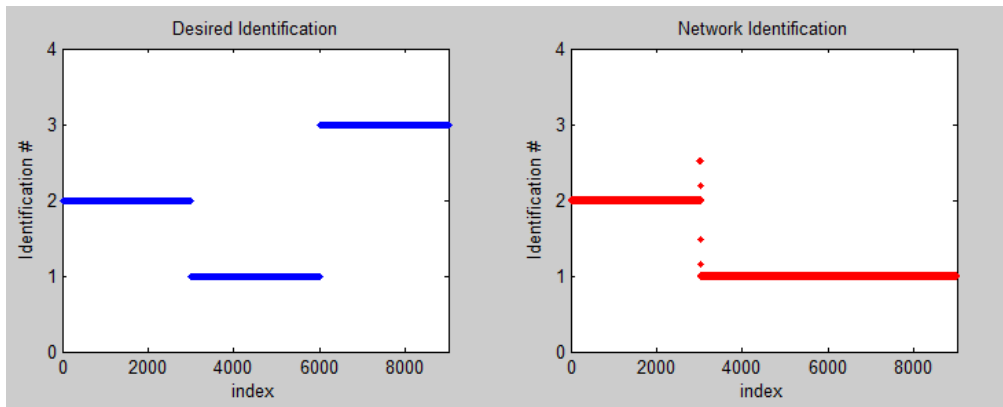
**Figure 16:** Results when identifying new input data the network was not trained with. The network combines wheel running and T-maze behavior into one behavior and identifies the behavior as 1 (T-maze behavior). The T-maze and open exploration behavior is correctly identified as 1 and 2, respectively, outside of the training interval.

Different training iterations of the network can yield different results. However, identification results generally confused the T-maze and wheel running behavior outside of the training interval. This is not surprising, as both behaviors take place in a single session and the neurological activity between T-maze and wheel running behaviors are closely overlapped, as seen in Figure 14. Although, the network was only able to distinguish between open exploration and T-maze behaviors outside of the training interval, the identification network was able to consistently identify between open exploration and T-maze behaviors throughout the entire dataset that the network was trained with. Consistency of identification between open exploration and T-maze behaviors outside of the training interval is shown using testing time points from a later portion of dataset 1, demonstrating that the network can still distinguish between T-maze and open exploration behavior with a much larger time gap between the training and testing time points. A summary of the time interval for specific time frames is given in Table 8.

**Table 8:** Time intervals of the training set, time gap between training and testing sets and 2<sup>nd</sup> testing set

<b>Dataset 1: Test 2</b>	<b>T-Maze</b>	<b>Open Exploration</b>	<b>Wheel Run</b>
<b>Training Interval (s)</b>	8.40	12.4	7.35
<b>Time gap (s)</b>	155	203	213
<b>Testing Interval (s)</b>	10.0	9.36	6.50

A comparison of the desired identification output and network identification output for the testing interval shown in Table 8, demonstrates that the network can still identify between open exploration and T-maze behavior, but not wheel running behavior, see Figure 17.



**Figure 17:** Identification results with the 2<sup>nd</sup> set of testing data for dataset 1. The desired identification results (left) and the network identification (right) when there is a significant time gap between the training and testing data. The network correctly identifies between open exploration and T-maze behavior and still confuses the wheel running behavior as T-maze behavior.

General observations when self-testing with the identification network include the network being unable to distinguish one behavior outside of its training interval, thus, if three behaviors were correctly identified inside the training interval, then two behaviors were typically identified outside the training interval. When the network was only able to distinguish between two behaviors inside the training interval, then only one behavior

was identified outside the training interval. However, once the network scaled down one behavior outside the training interval, it would identify consistently throughout the rest of the dataset that was used for training, assuming that the network was trained correctly and accurate initial results were obtained during training.

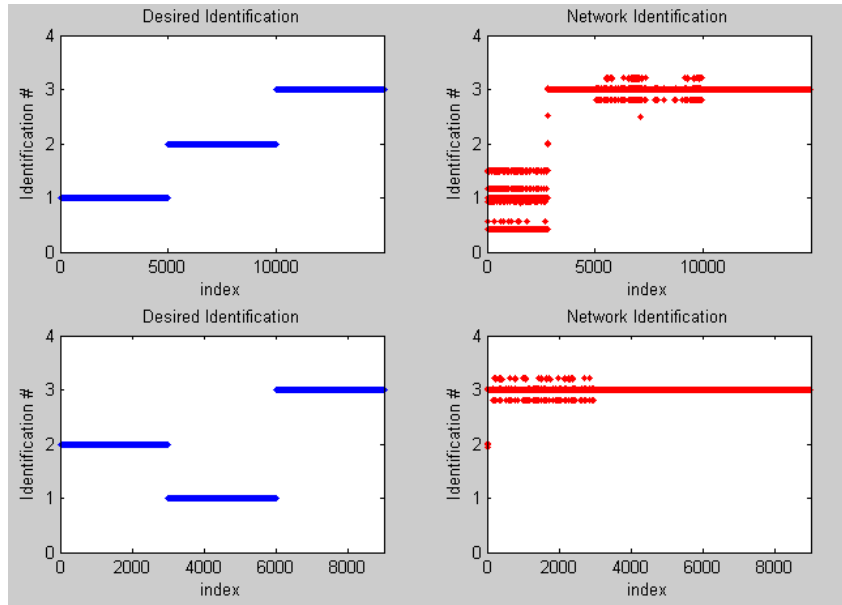
The property of the identification network to scale down one behavior outside of the training interval based on the input data was observed with self-testing datasets 2, 3 and 4 as well, thus, to reduce redundancy with the thesis, results for the additional datasets are provided in appendix C.

#### **4.1.2 Cross-Testing Datasets For Identification Network**

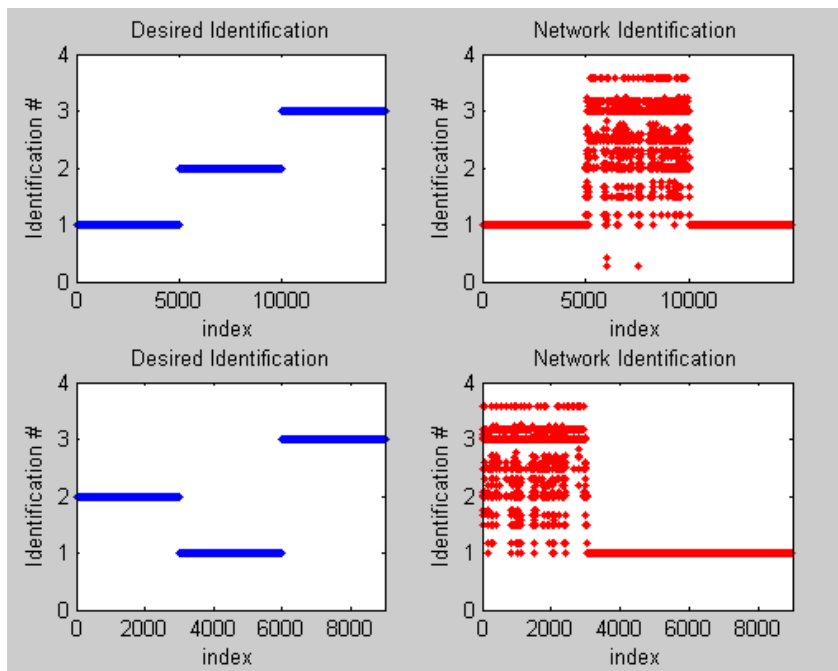
Cross-testing consists of training the identification network with one dataset and using another dataset to test it. The same amount of time points used for self-test training are used to train the identification network for cross-testing, previously shown in Table 6. Testing input data was then taken from a random interval within another dataset the network was not trained with.

Dataset 1 was used to train the network with 5000 time points of T-maze, wheel running and open exploration behavior. Figure 18-Figure 20 demonstrates the prediction results of the network, trained with dataset 1, when testing with input data taken from another dataset, specifically the early and latter portion of datasets 2, 3 and 4.

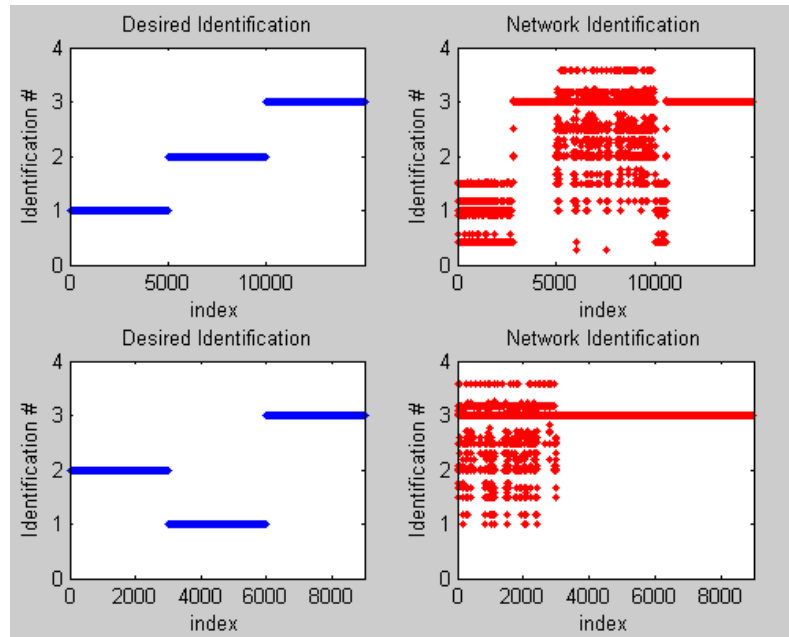




**Figure 18:** Identification results when identification network is trained by dataset 1 and tested with dataset 2. The top row involves the network being tested with early time points of dataset 2 and the bottom row involves the network being tested with later time points of dataset 2.



**Figure 19:** Identification results when identification network is trained by dataset 1 and tested with dataset 3. The top row involves the network being tested with early time points of dataset 3 and the bottom row involves the network being tested with later time points of dataset 3.



**Figure 20:** Identification results when identification network is trained by dataset 1 and tested with dataset 4. The top row involves the network being tested with early time points of dataset 4 and the bottom row involves the network being tested with later time points of dataset 4.

General observations of cross-testing between datasets include the identification network being unable to effectively identify behaviors from early or latter time points of a dataset that the network is not trained with. The results of cross-testing demonstrate that a larger difference in datasets would exhibit poorer identification. Dataset 1 has one session changed between datasets 2 and 3, so the network may identify one or two behaviors correctly, however, both sessions are changed between datasets 1 and 4 resulting in completely incorrect identification for any time point in dataset 4 with a network that is trained with dataset 1. Cross-testing between the other datasets were unsatisfactory as well and were not included in this thesis to reduce redundancy.

## 4.2 Testing of Behavior Attribute Network

The behavior attribute network was trained and tested similarly to the identification network, however, a behavioral attribute network only takes input data, consisting of cluster ID and shank ID, for one behavior. Behavioral attribute networks are trained with input-output data representative of possible velocity and spatial coordinates the rat can undergo during a session. The behavioral attribute network is more compliant when training, compared to the identification network, due to the cluster ID and shank ID data coming from one behavior, rather than all three behaviors at once, as in the identification network. This network is a multi input-multi output NARX network that maps the cluster ID and shank ID (two inputs) from one behavior to the x- and y-coordinate positions and velocity (three outputs) of the rat during the identified behavior. The network was tested with the same time points as the identification network due to the nature of the model shown in the overall design.

The spatial position and velocity of the rat predicted by the network was generally satisfactory, but may vary between training sessions. Two datasets containing different sessions of a specific behavior are used for training and testing the behavior's respective behavioral attribute network. A summary of the datasets used for each BAN is shown in Table 9.

**Table 9:** Datasets used for behavior attribute network training and testing

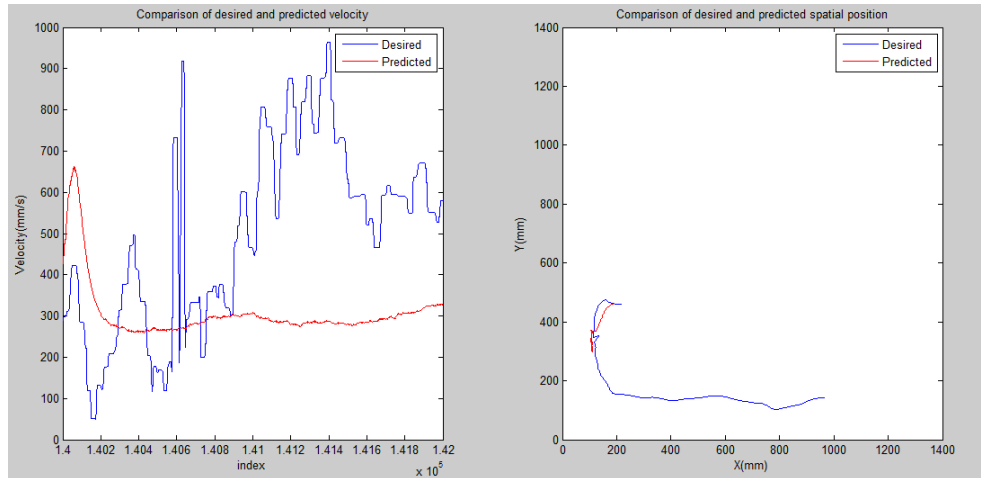
<b>T-Maze BAN</b>	<b>Open Exploration BAN</b>	<b>Wheel Running BAN</b>
Dataset 1	Dataset 1	Dataset 1
Dataset 2	Dataset 3	Dataset 2

The network was tested in the training interval to validate if training was successful, then tested outside of the training interval to determine generalization of the

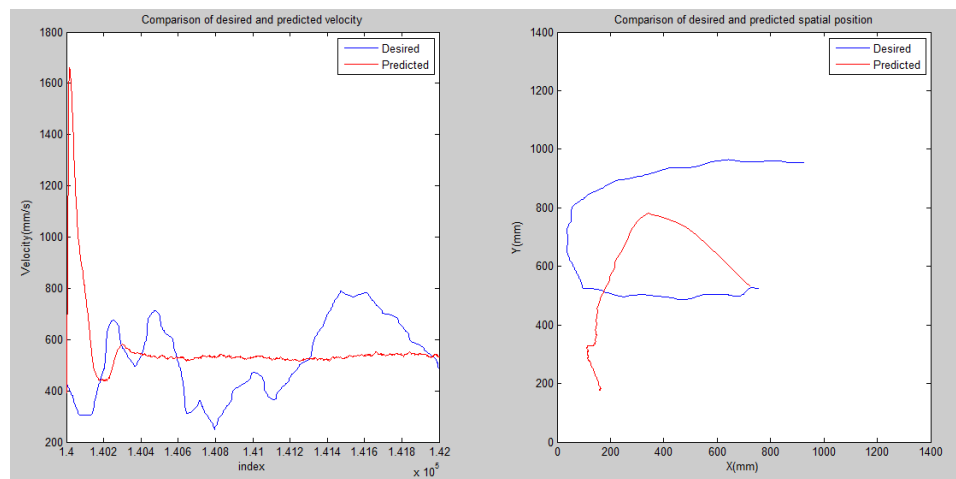
network. Additionally, the network was tested across datasets, where the network is trained with one dataset and tested with another. For practical purposes, it would be advantageous for an attribute network to be trained with one dataset and tested with another due to the range of values that a behavior attribute network needs to be trained with, unlike the identification network, which only has an output of 1, 2 or 3.

#### **4.2.1 T-Maze Behavior Attribute Network**

Training for the T-maze behavior attribute network was done using approximately  $1 \times 10^5$  time points of T-maze behavior from one session that covered the entire range of possible velocity magnitude and trajectories the rat could undergo during one session. The network was then tested with the same data used for training the network to test how well the network modeled the training data. A successful test of training data would result in the network generally modeling the training data but not exactly. If satisfactory generalization was observed, then the network was self-tested and cross-tested. Figure 21 illustrates the results of a network trained and tested with dataset 1, while Figure 22 cross-tests the network with dataset 2.



**Figure 21:** A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the T-maze behavior attribute network. The network prediction resulted in an average velocity when compared to the actual velocity and the predicted spatial coordinates did not completely traverse the actual path of the rat. The short-comings in both the velocity and spatial position of the rat is what makes this network more generalized, or applicable to other neurological data that the network has not been trained with.

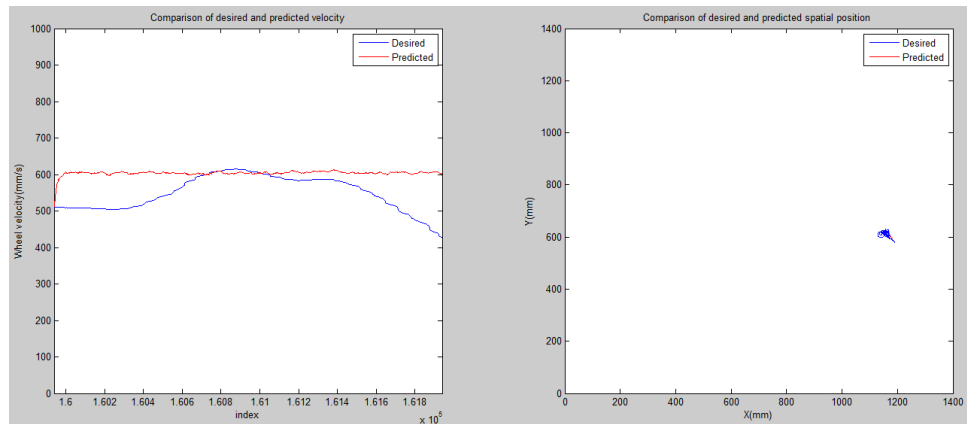


**Figure 22:** A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the T-maze behavior attribute network tested with a dataset that it was not trained for. The network was trained with T-maze data from dataset 1 and tested with T-maze data from dataset 2. The neurological data given to the network was completely different from the data that the network was trained with in Figure 21, however, the network was able to converge to a new average velocity and path for the rat. The form of the predicted spatial position is similar to the desired spatial position, but the predicted direction differed from the actual path, which demonstrates that the network was able to partially adapt to a new event that it was not trained for.

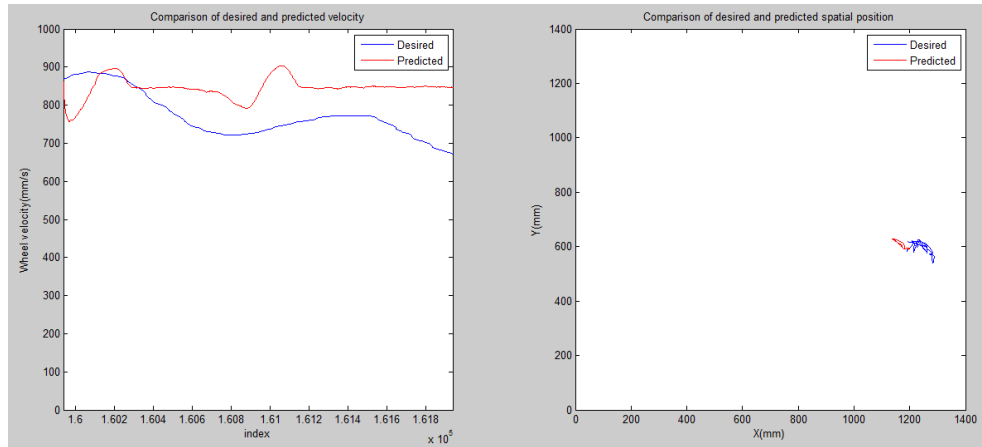
## 4.2.2 Wheel Running Behavior Attribute Network

The datasets used for training and testing the T-maze behavior attribute network are used for the wheel running behavior attribute network as well, due to wheel running and T-maze behavior taking place within the same session. However, only time points representative of wheel running behavior are taken from dataset 1 and 2 for training and testing. The training data is representative of a range of velocities and trajectories that the rat may undergo during a session.

The wheel running behavior attribute network was self- and cross-tested to determine generalization of the trained network prediction outside of the training interval for a dataset the network was trained with and across datasets, see Figure 23 and Figure 24.



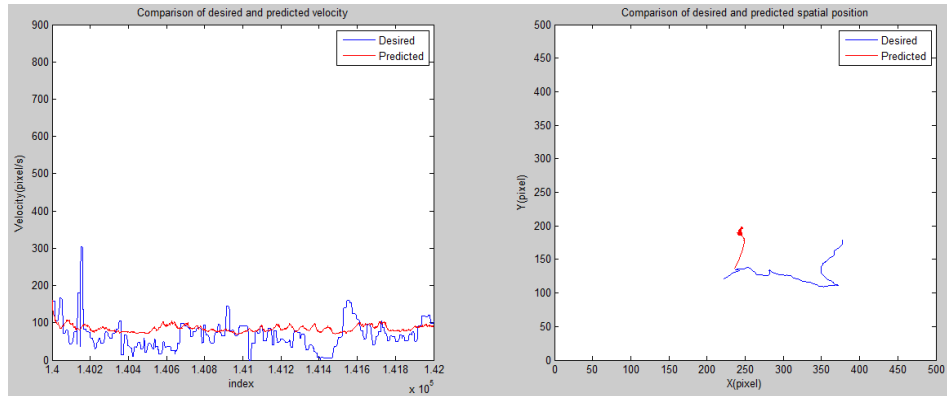
**Figure 23:** A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the wheel running behavior attribute network. For this testing input, the prediction of the velocity generally stayed constant at 600 mm/s, demonstrating the tendency of the network to converge to an output value that generally modeled the desired velocity. The spatial coordinates were well predicted for most of the wheel running data, due to lack of spatial movement during wheel running behavior.



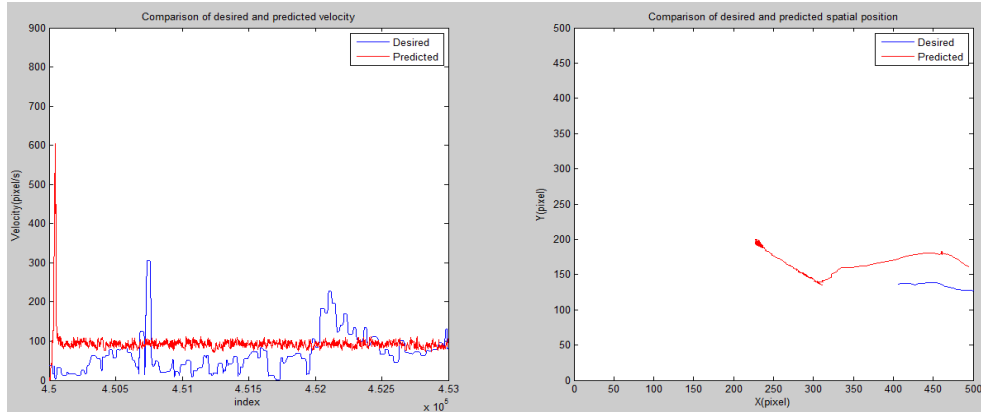
**Figure 24:** A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the wheel running behavior attribute network for a dataset the network had not been trained for. The network was trained with dataset 1 and tested with dataset 2. For this testing data, the network prediction of velocity generally models the desired velocity. The spatial coordinates were well predicted, due to lack in spatial movement during wheel running behavior.

### 4.2.3 Open Exploration Behavior Attribute Network

Open exploration behavior datasets contain entire sessions of just open exploration behavior, unlike the T-maze and wheel running behavior datasets, training and testing data for open exploration can be chosen from any interval in the dataset. Open exploration behavior involves placing a rat in a cage with no inner boundaries giving the rat freedom to explore and imposing no specific task onto the rat. The open exploration behavior attribute was trained with a possible range of velocity and spatial position data. The network was then self-tested with new data outside of the training interval, from the same dataset used for training, and cross-tested with new data from a dataset the network was not trained with, which is illustrated in Figure 25 and Figure 26.



**Figure 25:** A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the open exploration behavior attribute network. The predicted velocity remained at approximately 100 pixel/s, demonstrating the tendency of the network to converge to an output value that reasonably fits the desired velocity. The network’s prediction of the spatial position poorly reflects the desired spatial positioning of the rat, due to the networks nature to stop training when either of the outputs (i.e. velocity or spatial position) converges to a solution. Thus, the network’s velocity prediction converged to a solution before the spatial position prediction was able to completely converge to a solution, providing poor results for the predicted path in this case.



**Figure 26:** A comparison of the desired rat velocity (left-blue) and spatial coordinates (Right-blue) to the predicted velocity (left-red) and spatial coordinates (right-red) of the open exploration behavior attribute network for a dataset the network had not been trained for. The network was trained with dataset 1 and tested with dataset 3. The predicted velocity converged to an output value of approximately 100 pixel/s, which is faster than most of the desired velocity data and the predicted waveforms do not fit the desired velocity data well. The lack in adaptability demonstrates how modeling the training data well can work against the network when new data is presented. The predicted spatial position of the rat started in the same area the network predicted for the trained data, however, it adapted to the new neurological data and converged to a region similar to that of the desired spatial position.



### 4.3 Results Discussion

The various results found in this thesis raise the following questions: What possible factors caused the identification network to confuse the T-maze and wheel running behavior outside of the training interval? Why does cross-testing work for behavior attribute networks and not identification? How close does the comparison between the desired and predicted behavior have to be to consider the results of the network satisfactory? And what modifications can be made to the design to improve the predicted outcome? Each of these questions are addressed, but still remain open ended.

A time series spatial representation of neural activity in the form of cluster ID and shank ID for T-maze, wheel running and open exploration behavior was loaded into an identification NARX network resulting in identification of two out of three behaviors outside of the training interval. There are many factors that could possibly be affecting the performance of the identification network. Assumptions that were made in this thesis, such as the mental and physical state of the rat remaining the same and electrode drift not occurring between sessions, may not be coincident with the assumptions made by the researchers that gathered the live data. Additionally, the network parameters may not be optimally sufficient for a problem of this complexity. Distinguishing between three different time-series spatial representations of behavior may be too dynamic and nonlinear for a NARX network with a three layer structure consisting of an input, hidden and output layer.

The content of the input data may have been another contributing cause for the confusion between T-maze and wheel running behaviors. The cluster ID and shank ID

data used to train the network may not have appropriately represented the entire dataset. However, the most probable explanation for the subpar performance in identification results may be due to the similarity of the wheel running and T-maze behavior input data. There is a clear overlap of input data between T-maze and wheel running behavior, previously shown in Figure 14, which would most likely cause the network to generalize the two behaviors into one. Moreover, the chosen parameters for the input, cluster ID and shank ID, may not have appropriately represented the spatial component of neurological activity in the hippocampus. Additional parameters, such as theta phase or time between firing of neurons, may be needed to completely represent the spatio-temporal firing of neuron ensembles in a hippocampus.

Multiple methods were used to extract input data from datasets, such as random sampling and uniformly spaced time points, to train the NARX network with. However, best results were observed when the network was trained with sequential data. This finding supports the initial assumption of an existing pattern between the current and previous input data and behavior.

Cross-testing the identification network demonstrated that the identification network trained with one dataset would not be practical to use for another dataset. The identification network was typically not able to distinguish between behaviors in another dataset that it was not trained with. However, there were training sessions where the identification network distinguished two behaviors in another dataset. Thus, the identification network may be suitable for cross-testing with further modification of the datasets, network parameters and/or input parameters.

The behavior attribute network for T-maze, wheel running and open exploration were generally successful in self- and cross-testing. This demonstrates that the NARX network is much more effective recognizing patterns within one behavior, than between different behaviors. However, the behavior attribute network required much more training to familiarize the network with the general range of velocities and spatial positions that the rat might undergo. Due to the amount of training the behavior attribute network may require, the network becomes more practical when trained with a range of input and output values from one dataset and tested with another dataset.

Due to the mentioned limitations and advantages of the NARX network, the following training process is suggested to enhance practicality of the overall design. The identification NARX network provides the best results when trained with data from the same session as the test data. The most suitable method of training the identification network would involve gathering neural activity from the rat as all three behaviors are performed for a few seconds. Whereas, the behavior attribute NARX networks for T-maze, wheel running and open exploration behaviors can be trained with input from another session for the networks respective behavior. Training the network with a variety of spatial positions and velocities from another session representative of all the possible time points is suggested for the network to generalize well and appropriately predict data outside of the training interval.

## Chapter 5: Summary and Conclusions

In summary, the three stage design was utilized to create cluster ID and shank ID data representative of neuronal firing in the CA1 and EC hippocampal regions during the pre-processing stage by other researchers [12]. The cluster ID and shank ID data was used as input for the identification network, which identified whether the data was T-maze, wheel running or open exploration behavior during the identification stage. The velocity and spatial coordinate attributes were predicted during the behavior attribute stage for the behavior previously identified.

The overall design proposed in this thesis showed partial success during the identification stage when the same dataset was used for both training and testing. Two of the three behaviors were generally identified correctly outside of the training interval, which demonstrates that the network may have the ability to correctly identify all three behaviors with further modification of the network structure or input data. One possible modification includes implementing a quantitative assessment of the results, such that a reasonable total error, or mean squared error (MSE), can be produced, which would help the user determine the reliability and consistency of the network output. Refer to Appendix D, for an explanation about calculating the total error.

The NARX behavioral attribute network was observed to generally provide satisfactory prediction of the velocity and spatial coordinates for a rat when the same dataset was used for both training and testing or when a dataset, the network was not trained with, was used for testing. General success in self- and cross-testing of behavioral attributes provide support that data representative of an ensemble of neurons in a time-series formation may be appropriate to describe behavior attributes.

## **5.1 Contributions**

By providing a design that can identify T-maze, wheel running and open exploration behavior and predict velocity and spatial coordinate attributes of the identified behavior, it may be possible to determine what a rat is dreaming of during rapid eye movement (REM) of a sleep cycle. This situation is analogous to the theoretical black box model considered in this thesis, where the neurological activity of the rat can be seen, but the behavior and behavior attributes are unknown. Further implications of the proposed design include possible enhancements for brain-to-machine-interfaces (BMI), by taking neurological activity from the hippocampus, rather than the motor or parietal cortex of the brain. The three stage design could potentially be used to recognize multiple cognitive diseases, such as Parkinson's and epilepsy, with one device.

## **5.2 Future Work**

There are many modifications and advancements possible for the design presented in this thesis. Future work includes gathering the neurological data personally, rather than acquiring the information through an external source. Testing of the three stage design with more than one rat or subject to determine if the results found in this thesis are conclusive. Modifying the NARX network structure to have more than three-layers and determining if partial success in this thesis was due to the structure of the network. And creating a robotic interface that responds to output of the three stage design.

## REFERENCES

- [1] Berger et al., "A Hippocampal Cognitive Prosthesis: Multi-Input, Multi-Output Nonlinear Modeling and VLSI Implementation," *IEEE Trans Neural Syst Rehabil Eng*, vol. 20, no. 2, p. 198–211, 2012.
- [2] W. Deng, J. Aimone and F. Gage, "New neurons and new memories: how does adult hippocampal neurogenesis affect learning and memory?," *Neuroscience*, vol. 11, pp. 339-350, 2010.
- [3] Zeng et al., "CASIA Brain : A Multi-scale Simulator of the Cognitive Brain," 4 November 2014. [Online]. Available: <http://brain-eng.ia.ac.cn/CASIA-Brain.htm>. [Accessed 16 May 2015].
- [4] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat.," *Brain Res*, vol. 34, pp. 171-175, 1971.
- [5] K. Diba and G. Buzsáki, "Hippocampal Network Dynamics Constrain the Time Lag between Pyramidal Cells across Modified Environments," *The Journal of Neuroscience*, vol. 28, no. 50, pp. 13448-13456, 2008.
- [6] Pastalkova et al., "Internally Generated Cell Assembly Sequences in the Rat Hippocampus," *Science*, vol. 321, pp. 1322-1327, 2008.
- [7] D. Song, V. Marmarelis and T. Berger, "Parametric and non-parametric modeling of short-term synaptic plasticity. Part I: computational study," *Journal of computational Neuroscience*, vol. 26, pp. 1-19, 2009.
- [8] Song et al., "Nonlinear Dynamic Modeling of Spike Train Transformations for Hippocampal-Cortical Prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 6, pp. 1053-1066, 2007.
- [9] G. Simic, . I. Kostovic, B. Winblad and N. Bogdanov, "Volume and number of neurons of the human hippocampal formation in normal aging and Alzheimer's disease," *The Journal of Comparative Neurology* , vol. 379, pp. 482-494, 1997.
- [10] Halpern et al., "Deep Brain Stimulation for Epilepsy," *Neurotherapeutics*, vol. 5, no. 1, pp. 59-67, 2008.
- [11] J. Carmena and J. Millan, "Brain-machine interfaces: your brain in action," *Neuroscience*, vol. 1, no. 7, pp. 1-3, 2013.
- [12] Mizuseki et al., "Multiple single unit recordings from different rat hippocampal and entorhinal regions while the animals were performing multiple behavioral tasks.

- CRCNS.org., "CRCNS - Collaborative Research in Computational Neuroscience - Data sharing, 2013. [Online]. Available: <http://dx.doi.org/10.6080/K09G5JRZ>. [Accessed 2015].
- [13] E. Diaconescu, "The use of NARX Neural Networks to predict Chaotic Time Series," *WSEAS Transactions on Computer Research*, vol. 3, no. 3, pp. 182-191, 2008.
- [14] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., New Jersey: Pearson Education inc., 2014.
- [15] Marmarelis et al., "On parsing the neural code in the prefrontal cortex of primates using principal dynamic modes," *The Journal of Computational Neuroscience*, vol. 36, pp. 321-337, 2014.
- [16] Csicsvari et al., "Ensemble Patterns of Hippocampal CA3-CA1 Neurons during Sharp Wave-Associated Population Events," *Neuron*, vol. 28, pp. 585-594, 2000.
- [17] M. Korenberg and I. Hunter, "The Identification of Nonlinear Biological Systems: Wiener Kernel Approaches," *Annals of Biomedical Engineering*, vol. 18, pp. 629-654, 1990.
- [18] V. Marmarelis and M. Orme, "Modeling of Neural Systems by Use of Neuronal Modes," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 11, pp. 1149-1158, 1993.
- [19] Chiu et al., "Prediction of Seizure Onset in an In-Vitro Hippocampal Slice Model of Epilepsy Using Gaussian-Based and Wavelet-Based Artificial Neural Networks," *Annals of Biomedical Engineering*, vol. 33, no. 6, pp. 798-810, 2005.
- [20] Mano et al., "An Artificial Neural Network Based Robot Controller that Uses Rat's Brain Signals," *Robotics*, vol. 2, pp. 54-65, 2013.
- [21] E. Kelemen, "Artificial Neural Networks and Artificial Evolution as Tools to Study Organization of Spatial Behavior in Rats," in *Advances in Artificial Life*, Pittsburgh, Springer-Verlag Berlin Heidelberg New York, 2001, pp. 236-241.
- [22] Z. Wang-Xiong, Q. Qiao and D. Wang, "Hippocampus Function Modeling With Neural Network," *IEEE*, vol. 1, pp. 1128-1131, 2009.
- [23] Henze et al., "Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo," *Journal of Neurophysiology*, vol. 84, no. 1, pp. 390-400, 2000.
- [24] Mizuseki, "hc3 processing flowchart," *crcns.org*, pp. 1-3, 2014.

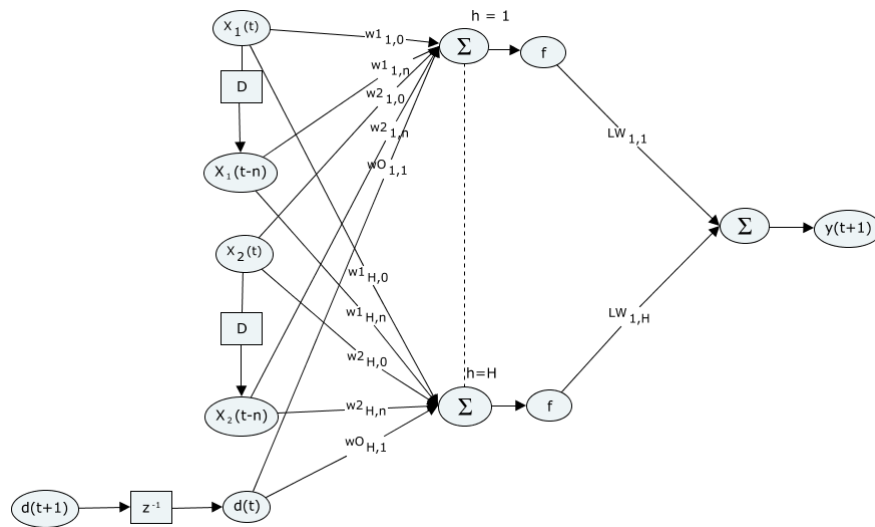
- [25] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computational Neural System*, vol. 9, pp. R53-R78, 1998.



## APPENDICES

### Appendix A: Network Learning Process

This appendix section is used as supplementary material on the Levenberg-Marquardt learning method used to train the identification and attribute behavior NARX networks. A series parallel mode is used for training the network, see Figure A-1



**Figure A-1:** A series-parallel mode of the NARX network used for training. Training utilizes the previous desired output rather than the network output to minimize performance error of the network during training.

The equation that describes the series-parallel mode of the NARX network is provided in Eq. A-1.

$$\begin{aligned}
 y(t + 1) = & \sum_{h=1}^H w_{oh} \\
 & * f \left( \sum_{i1=0}^n x_1(t - i1) * w_{1hi} + \sum_{i2=0}^n x_2(t - i2) * w_{2hi} \right. \\
 & \left. + \sum_{j=1}^n d((t + 1) - j) * w_{ohj} \right) \quad [\text{Eq. A - 1}]
 \end{aligned}$$

Eq.A-1 has the same form as the generalized NARX network equation, however, the only difference is the replacement of the network output,  $y$ , with the desired output,  $d$ . Levenberg-Marquardt Back-propagation learning algorithm was used to optimize the weights by using the following equation:

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \quad [\text{Eq. A - 2}]$$

Where:  $e_k$ , is the error between the desired and network output, calculated by,

$$e_k = d_k - y_k$$

$w_k$ , are the current weights that need to be updated

I, is the identity matrix

$\mu$ , is the learning constant that gradually gets smaller as the iterations increase

J, is the first order Jacobian matrix, calculated by,

$$J = \begin{pmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \cdots & \frac{\partial e_{1,1}}{\partial w_N} \\ \frac{\partial e_{1,M}}{\partial w_1} & \cdots & \frac{\partial e_{1,M}}{\partial w_N} \\ \cdots & \cdots & \cdots \\ \frac{\partial e_{p,M}}{\partial w_1} & \cdots & \frac{\partial e_{p,M}}{\partial w_N} \end{pmatrix}$$

Equation A-2 describes how the weights are updated after every iteration. J, represents the first order Jacobian matrix used to approximate the second order Hessian matrix without the need of actually computing the Hessian matrix. This decreases the computation necessary, thus decreasing the amount of time needed to solve the problem. The learning rate parameter,  $\mu$ , is constantly changed throughout the process of training, which contributes to fast and stable training.

The Levenberg-Marquardt algorithm starts  $\mu$  off at a large value at the beginning of training, which would reduce Eq. A-2 into the gradient descent method equation seen below.

$$w_{k+1} = w_k - \alpha g_k \quad [\text{Eq. A - 3}]$$

Where:  $\alpha$  is the learning rate parameter

$g$  is the gradient of the error performance, calculated by,

$$g = \left( \frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \cdots \quad \frac{\partial E}{\partial w_N} \right)^T$$

$$\text{where, } E = \frac{1}{2} \sum_{p=1}^P \sum_{m=1}^M e_{p,m}^2$$

The gradient descent method has the advantage of quickly decreasing the error performance at high errors, but has the disadvantage of taking a relatively long time to find the minimum error. This disadvantage is avoided by decreasing the learning rate after every training step, which eventually reduces eq.A-2 to the learning newton method seen in the equation below.

$$w_{k+1} = H_k^{-1} g_k \quad [\text{Eq. A - 4}]$$

Where,  $H$  is the approximate Hessian matrix calculated by the Jacobians and  $g$  is the error gradient. The Newton method is much better at finding an error minimum than the gradient descent method, but has the disadvantage of getting caught in a local minimum.

## Appendix B: Program

The code used to produce the previously shown results is provided in this appendix. The complete code is separated into four different programs representing the four different NARX networks, which consist of identification, T-maze behavior attribute prediction, wheel running behavior attribute prediction and open exploration behavior attribute prediction. However, since all of the behavior attribute networks are essentially the same, but trained and prepared with different data, only the T-maze behavior attribute network is provided for example, which can be modified with little effort, to produce attribute results shown for other behaviors.

Results obtained using the provided code in this appendix may or may not be similar to the results provided in this thesis, due to random partitioning of training data and network training method. Additionally, use of neural networks may require some understanding of the input-output data being used and how fast the network might converge to a solution, in some cases, early stopping may be needed to stop the network from modeling noise and obtain generalized training results

### Identification Network Code

```
%% Identification NARX network
% This network identifies whether the general behavior is a
T-maze, open exploration or wheel running behavior by
providing an output of 1, 2 or 3, respectively, using shank
ID and cluster ID as input.
```

```
% DISCLAIMER: The results obtained with this code may or
may not be similar

% to the results provided in the thesis. Possible
conditions that may

% provide poor results, include using data that poorly
represents the

% entire dataset, training the network for too long or
stopping training too early.

%% Clear MATLAB and load input data

close all

clc

clear

%% Load one of the possible datasets

% Currently, dataset 1 is selected

load('Dataset1')

% load('Dataset2')

% load('Dataset3')

% load('Dataset4')

%% Set index period of interest within data set

% Here are three possible indices used for training or
testing
```

```

t_whlrun = 97195:102159;           % 127565:128565; %
158940:159940;

t_maze_explore = 117607:122607;   % 138804:139804; %
172000:173000;

% Calculate time periods

% The time period of the training interval is calculated
WhlTime_Interval = (Sorted.Time(t_whlrun(end))-
...Sorted.Time(t_whlrun(1)))/1250;
TMazeTime_Interval = (Sorted.Time(t_maze_explore(end))-
...Sorted.Time(t_maze_explore(1)))/1250;
OpenTimeInterval = (SortedM.Time(t_maze_explore(end))-
...SortedM.Time(t_maze_explore(1)))/1250;

% Create Input Data Matrix

% Cluster ID and Shank ID input data for T-maze behavior
sorted by time

T_maze_input =
...[Sorted.Shank(t_maze_explore)';Sorted.Clu(t_maze_explore
...)]';

% Cluster ID and Shank ID input data for open exploration
behavior sorted by time

```

```

Explore_input =
...[SortedM.Shank(t_maze_explore)';SortedM.Clu(t_maze_explo
...re)'];

% Cluster ID and Shank ID input data for wheel running
behavior sorted by time

WheelRun_input =
...[Sorted.Shank(t_whlrun)';Sorted.Clu(t_whlrun)'];

% Plot the input data of cluster ID vs shank ID for all
behaviors

figure(1);

hold on

plot(T_maze_input(1,:),T_maze_input(2,:),'b.')
plot(Explore_input(1,:),Explore_input(2,:),'r.')
plot(WheelRun_input(1,:),WheelRun_input(2,:),'k.')

title('Comparison of cluster ID as a function of shank ID
between behaviors')

xlabel('Shank ID'),ylabel('Cluster ID')

legend('T-Maze task','Open exploration','Wheel
running','Location','SouthEast')

axis([0 16 0 128])

box on

hold off

```



```

%% Arrange data suitable for network input

x1 =
...[T_maze_input(1,:),Explore_input(1,:),WheelRun_input(1,:
...)]];

x2 =
...[T_maze_input(2,:),Explore_input(2,:),WheelRun_input(2,:
...)]];

% Combine the shank ID and cluster ID for all behaviors
into one input matrix

X = [x1;x2];

% Change the concurrent input into sequential input for a
time-series

% representation of the data

XCseq = con2seq(X);

% Create the output identification for the input

y1 = ones(1,length(T_maze_input(1,:)));           % Value for
T-maze task

y2 = 2.*ones(1,length(Explore_input(1,:)));       % Value for
Open exploring

```

```

y3 = 3.*ones(1,length(WheelRun_input(1,:)));    % Value for
Wheel running

% Join output values, such that they correspond with the
input data
Y = [y1,y2,y3];

% Change concurrent output to sequential output
YCseq = con2seq(Y);

%% Define structure parameters of network
du = 30;    % Delay of shank and cluster input
dy = 1;    % Delay of the output
L = 60;    % Number of hidden layers

%% Create the identification NARX network
NARX_ID = narxnet(du,dy,L);

% Initialize the parameters and transfer function of the
network and input
NARX_ID.trainParam.epochs = 150;    % Max epoches of
network
NARX_ID.layers{1}.transferFcn = 'hardlim'; % Hard limiting
activation f'n

```

```

%% initialize the delay

% This function will set aside the first 30 input values
for the initial
% delay. Output does not need to be initialize due to there
being one delay

% , the first delayed output = 0, previously shown in
simplified example.

[p,Pi,Ai,t] = preparets(NARX_ID,XCseq,{},YCseq);

%% Train the network

% Extra training with the same training data is optional.
However, it is
% recommended to not over train with same data, otherwise
poor
% generalization results occur.

for i = 1:2
NARX_ID = train(NARX_ID,p,t,Pi);
end

%% Close the network

NARX_ID_Close = closeloop(NARX_ID);

```

```

% Optional to have MATLAB illustrate closed network
% view(ClassnetClose)

%% Test the closed network with training data
uclose = XCseq;          % Define closed input same as
training input
yclose = YCseq;          % Define closed output same as
training output

% Initialize the input and output data
[p1,Pi1,Ai1,tfl] =
...preparets(NARX_ID_Close,uclose,{},yclose);
% Output of the NARX network in time-series formation
yp1 = NARX_ID_Close(p1,Pi1,Ai1);
% Convert time-series output to array output for
illustration purposes
yp1M = cell2mat(yp1);

%% Create new input-output data to test network
Whl_test = 206999:209999; % 128000:131000;
Explore_TM_test = 200000:203000; % 139000:142000;

% Calculate time period information of new data
% Calculation for time gap between training interval and
testing interval

```

```

% for each behavior

WhlIntervalBetweenTrainAndTest = (Sorted.Time(Whl_test(1))-
...Sorted.Time(t_whlrun(end)))/1250;

TMazeIntervalBetweenTrainAndTest =
...(Sorted.Time(Explore_TM_test(1))-
...Sorted.Time(t_maze_explore(end)))/1250;

OpenIntervalBetweenTrainAndTest =
...(SortedM.Time(Explore_TM_test(1))-
...SortedM.Time(t_maze_explore(end)))/1250;

% Calculation of testing interval for each behavior

WhlTime_Interval_test = (Sorted.Time(Whl_test(end))-
    Sorted.Time(Whl_test(1)))/1250;

TMazeTime_Interval_test =
...(Sorted.Time(Explore_TM_test(end))-
...Sorted.Time(Explore_TM_test(1)))/1250;

OpenTimeInterval_test =
... (SortedM.Time(Explore_TM_test(end))-
...SortedM.Time(Explore_TM_test(1)))/1250;

% Table for time periods

% This code provides convenient display of time periods for
data calculated earlier

Intervals = {'Interval', 'Wheel Run', 'T-Maze', 'Open';

```

```

... 'Train(s)', WhlTime_Interval,
... TMazeTime_Interval, OpenTimeInterval;
... 'Between(s)', WhlIntervalBetweenTrainAndTest,
... TMazeIntervalBetweenTrainAndTest, OpenIntervalBetweenTrai
... nAndTest; 'Test(s)', WhlTime_Interval_test, TMazeTime_Inter
... val_test, OpenTimeInterval_test}

%% Arrange test input

xt1 =
... [SortedM.Shank(Explore_TM_test)', Sorted.Shank(Explore_TM
... _test)', Sorted.Shank(Whl_test)'];
xt2 =
... [SortedM.Clu(Explore_TM_test)', Sorted.Clu(Explore_TM_tes
... t)', Sorted.Clu(Whl_test)'];
Xt = [xt1;xt2];
XCseq2 = con2seq(Xt);

%% Arrange test output

Yt_TMaze = ones(1, length(Sorted.Shank(Explore_TM_test)));
Yt_Explore =
... 2*ones(1, length(SortedM.Shank(Explore_TM_test)));
Yt_WhlRun = 3*ones(1, length(Sorted.Shank(Whl_test)));

Yt = [Yt_Explore, Yt_TMaze, Yt_WhlRun];

```

```

YCseq2 = con2seq(Yt);

%% Initialize the testing data
[p2,Pi2,Ai2,t2] =
...preparets(NARX_ID_Close,XCseq2,{},YCseq2);

%% Use the closed network
% Output is in time-series form
a2 = NARX_ID_Close(p2,Pi2,Ai2);
% Convert to array form for illustration purposes
Ad2 = cell2mat(a2);

%% Plot the results
figure(2);
% Plot of desired identification
subplot(2,2,1)
plot(Y,'b.')
title('Desired Identification')
xlabel('index'),ylabel('Identification #')
axis([0 2*length(t_maze_explore)+length(t_whlrun) 0 4])

% Plot of network identification
subplot(2,2,2)
plot(yp1M,'r.')

```

```

title('Network Identification')
xlabel('index'),ylabel('Identification #')
axis([0 2*length(t_maze_explore)+length(t_whlrun) 0 4])

% Plot of desired identification
subplot(2,2,3)
plot(Yt,'b.')
title('Desired Identification')
xlabel('index'), ylabel('Identification #')
axis([0 2*length(Explore_TM_test)+length(Whl_test) 0 4])

% Plot of network identification
subplot(2,2,4)
plot(Ad2,'r.')
title('Network Identification')
xlabel('index'), ylabel('Identification #')
axis([0 2*length(Explore_TM_test)+length(Whl_test) 0 4])

```

### **Behavior Attribute Network Code for T-Maze**

```

% The network predicts the velocity and spatial coordinate
attributes given T-maze input-output data.

% This program follows the NARX_Identifier program if the
output is identified as a '1.'

```



```

%% Clear

close all

clc

clear

%% Load dataset

% Currently, dataset 1 is selected

load('dataset1')

% load('dataset2')

%% Initialize the training interval

% Here are possible intervals that were used for
training/testing

t_train = 107000:114000;    %180000:183500;  %
140000:142000; %

%% Initialize the inputs that get fed into the network

x1 = Sorted.Shank(t_train)';
x2 = Sorted.Clu(t_train)';

% Combine the inputs into one matrix

Xm1p = [x1;x2];

% Change from concurrent to sequential data type

```

```

Xseq = con2seq(Xmlp);

%% Initialize the outputs for training
y1 = Sorted.X(t_train)';
y2 = Sorted.Y(t_train)';
y3 = Sorted.speed(t_train)';

% Combine into one matrix
Ymlp = [y1;y2;y3];

% Convert into time-series form
Yseq = con2seq(Ymlp);

%% Define network structure parameters
delay1 = 50;
delay2 = 1;
layer = 30;

%% Create the open loop NARX network
TMazeNARX = narxnet(delay1,delay2,layer);

% Max number of training iterations before network stops
% Generally, early stopping would be necessary before max
% epoch is reached.

```

```

TMazeNARX.trainParam.epochs = 1000;

%% Prepare the input and output parameters
[p,Pi,Ai,t] = preparets(TMazeNARX,Xseq,{},Yseq);

%% Train the open network
TMazeNARX = train(TMazeNARX,p,t,Pi,Ai);

%% Close the network
TMazeClose = closeloop(TMazeNARX);

% Optional to view MATLAB illustration of closed
%   view(WhlRunNARX)

%% Prepare testing points for the network
tClose_test = t_train;

% Testing input data
x1_test = Sorted.Shank(tClose_test)';
x2_test = Sorted.Clu(tClose_test)';

% Combine the inputs into one matrix
Xmlp_test = [x1_test;x2_test];
Xseq_test = con2seq(Xmlp_test);

```

```

%% Prepare desired outputs to compare with network output
y1_test = Sorted.X(tClose_test)';
y2_test = Sorted.Y(tClose_test)';
y3_test = Sorted.speed(tClose_test)';

% Combine outputs into a matrix
Ymlp_test = [y1_test;y2_test;y3_test];
Yseq_test = con2seq(Ymlp_test);

%% Test closed loop network with data used to train
% Note: that for testing, the training section is commented
out and another sample of input data is chosen

uclose = Xseq_test;
yclose = Yseq_test;

[p1,Pi1,Ai1,tf1] =
preparets(TMazeClose,uclose, {},yclose);

yp1 = TMazeClose(p1,Pi1,Ai1);
yp1M = cell2mat(yp1);
p1M = cell2mat(p1);

% view(WhlRunClose)

```

```

% Plot the desired and predicted velocity for comparison
figure(1)
subplot(1,2,1)
hold on
% Plot of desired output
plot(t_train,y3')
% Plot of networks output
plot((linspace(t_train(1),t_train(end),length(yp1M(3,:)))),
yp1M(3,:),'r')

title('Comparison of desired and predicted velocity')
xlabel('index'), ylabel('Velocity(mm/s)')
axis([t_train(1) t_train(end) 0 1000])
legend('Desired','Predicted')
box on
hold off

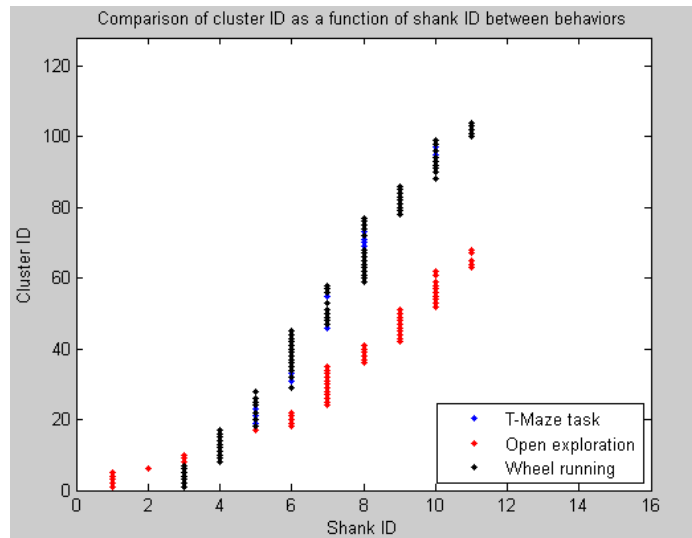
% Plot the desired and predicted spatial coord. For
% comparison
subplot(1,2,2)
hold on
% Plot desired output
plot(y1,y2)

```

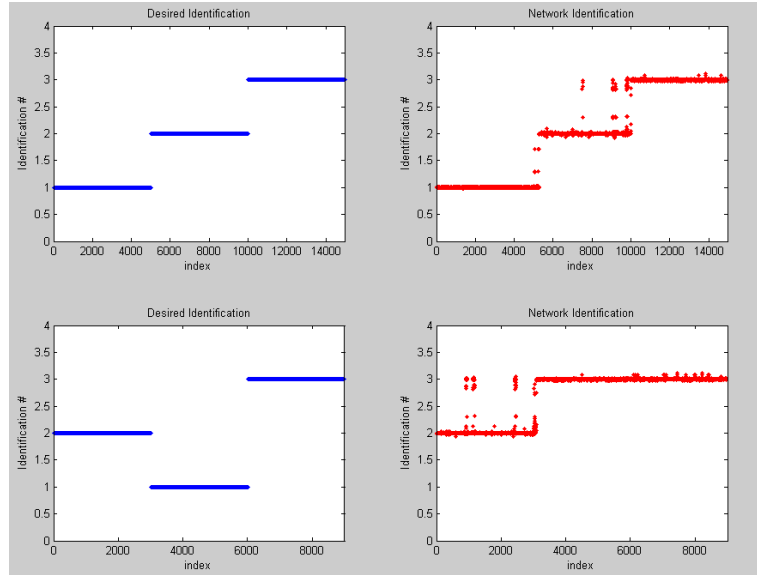
```
% Plot predicted output
plot(yp1M(1,:),yp1M(2,:), 'r')
title('Comparison of desired and predicted spatial
position')
xlabel('X(mm) '), ylabel('Y(mm) ')
axis([0 1400 0 1400])
legend('Desired', 'Predicted')
box on
hold off
```

## Appendix C: Self-Testing Identification Network with Datasets 2, 3 and 4

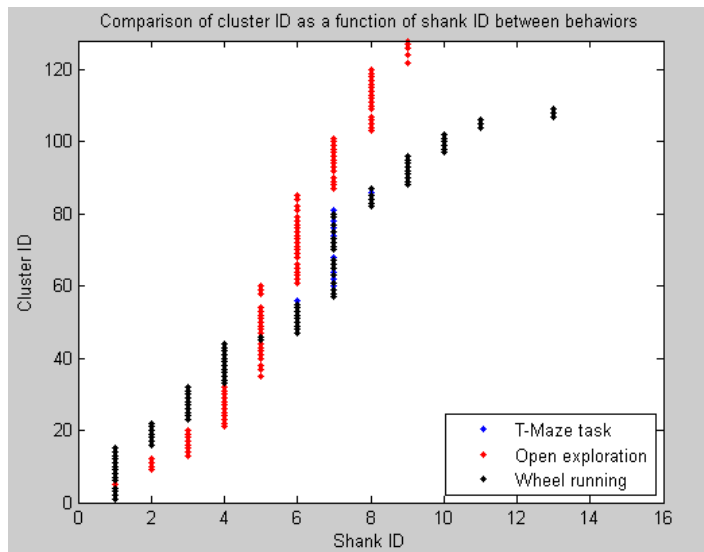
Self-testing results, which uses the same dataset for both training and testing, of the identification network with datasets 2, 3 and 4 are shown in this appendix. The results of identification with datasets 2, 3 and 4 were similar to the training and testing done with dataset 1. An identification network that was able to distinguish between T-maze, wheel running and open exploration behavior inside the training interval would typically confuse one behavior outside of the training interval. For datasets 2, 3 and 4, the input data is plotted to illustrate the difference of input data between datasets. Additionally, network identification of training data and new data outside of the training interval from the same dataset used to train the network are presented here, see figures C-1-C-6.



**Figure C-1:** 15000 time points of input data for dataset 2 used to train the identification network. The T-maze and wheel running behavior data are closely overlapped, while most of the open exploration behavior data is distinguishable from T-maze and wheel running behavior.

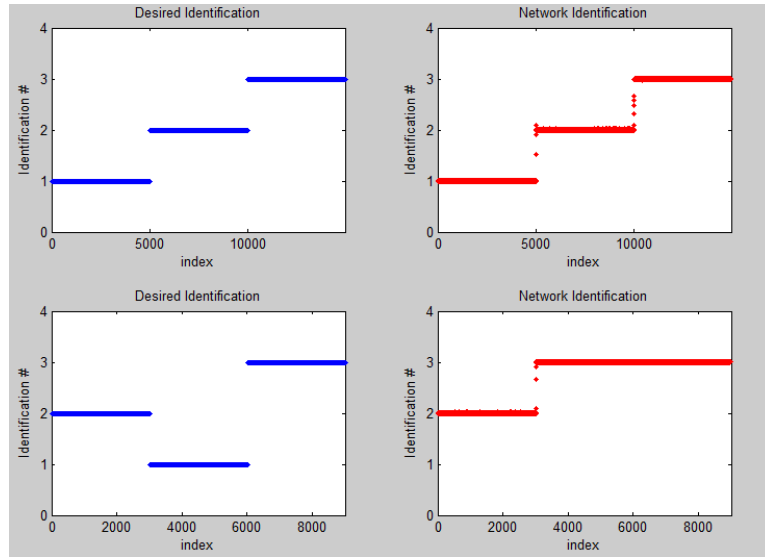


**Figure C-2:** Identification network results when data from dataset 2 was used. The network was able to identify all three behaviors within the training interval (top-right), but confused T-maze behavior as wheel running behavior outside of the training interval (bottom-right).

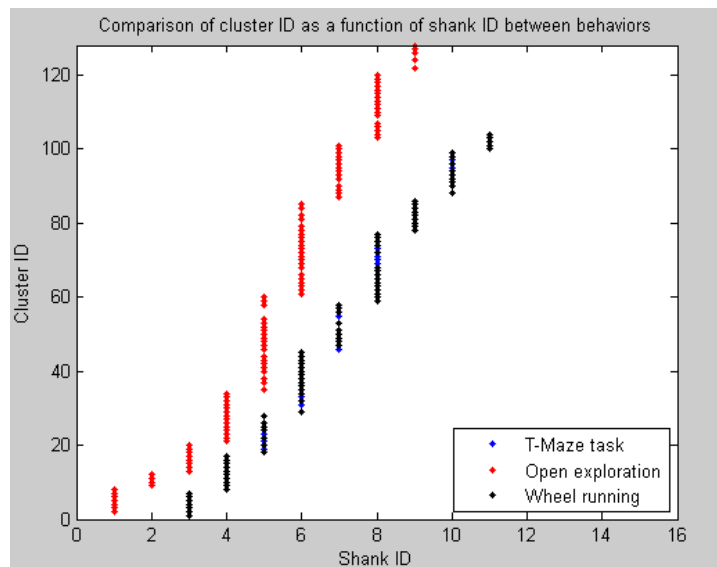


**Figure C-3:** 15000 time points of input data for dataset 3 used to train the identification network. The T-maze and wheel running behavior data are closely overlapped, while most of the open exploration behavior data is distinguishable from T-maze and wheel running behavior.

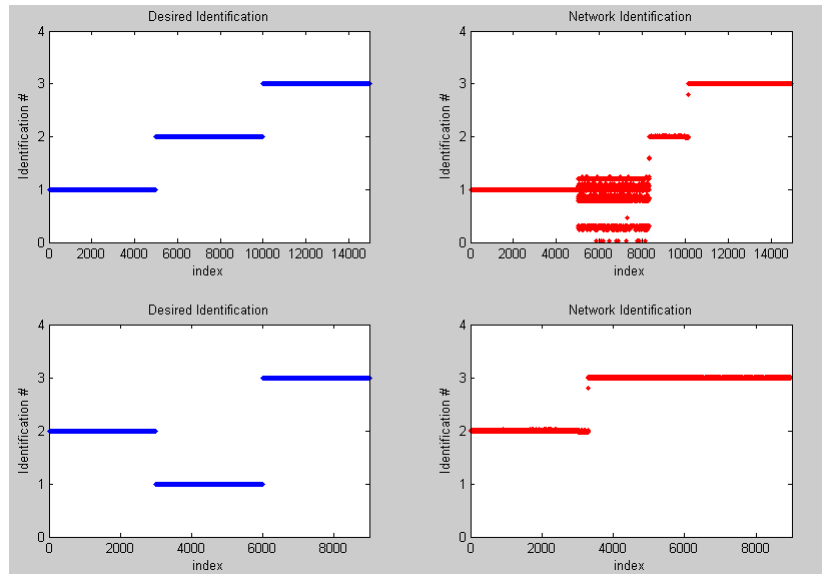




**Figure C-4:** Identification network results when data from dataset 3 was used. The network was able to identify all three behaviors within the training interval (top-right), but confused T-maze behavior as wheel running behavior outside of the training interval (bottom-right).



**Figure C-5:** 15000 time points of input data for dataset 4 used to train the identification network. The T-maze and wheel running behavior data are closely overlapped, while all of the open exploration behavior data is distinguishable from T-maze and wheel running behavior.



**Figure C-6:** Identification network results when data from dataset 4 was used. The network was partially able to identify all three behaviors within the training interval (top-right), but confused T-maze behavior as wheel running behavior outside of the training interval (bottom-right). The identification results were more difficult for this dataset due to a larger variation of input data.

## Appendix D: Total Error Calculation

A statistical and quantitative analysis of the network results will be needed in future modifications to determine and improve the consistency and reliability of the network output. Thus, this appendix provides a possible method to determine the total error of the network output, which can be used to develop a standard of efficiency for behavior prediction from neurological activity.

The total error of the network output would be helpful in understanding how reliable the output data is. The mean-squared error (MSE) is commonly used when determining the overall error of the network output and can be calculated using equation D-1,

$$MSE = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2 \quad [Eq. D - 1]$$

Where  $d$  is the desired output and  $y$  is the network output. By comparing the MSE of the network output for training data with new testing data, it may be possible to determine a quantitative value that represents the adaptability of the network.