MACHINE LEARNING APPLICATIONS IN CONTROL SYSTEMS

DORUK EREN

JANUARY 2023

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

M.Sc. THESIS IN

ELECTRICAL AND ELECTRONICS ENGINEERING

MACHINE LEARNING APPLICATIONS IN CONTROL SYSTEMS

DORUK EREN

JANUARY 2023

ABSTRACT

Machine Learning Applications in Control Systems

EREN, Doruk

M.Sc. in Electrical and Electronics Engineering

Supervisor: Asst. Prof. Dr. Oğuzhan ÇİFDALÖZ

Co-Supervisor: Assoc. Prof. Dr. Erdem AKAGÜNDÜZ

January 2023, 57 pages

In this thesis, the use of Long Short-Term Memory (LSTM) networks as a potential alternative to Kalman filters in the task of state estimation was investigated. State estimation involves inferring the true state of a system from noisy or incomplete observations. Kalman filters are a widely used technique for this purpose, but they have certain limitations, such as the assumption of linearity.

LSTMs are a type of recurrent neural network (RNN) that have demonstrated effectiveness in modeling complex, nonlinear systems in tasks such as language translation and image captioning. It was proposed that LSTMs may also be capable of accurately estimating the state of dynamic systems.

To test this hypothesis, the performance of LSTMs of differing learning parameters and a Kalman filter was compared on a non-linear system benchmark dataset. The training of LSTM models required a considerable amount of processing power and time, which posed a challenge on consumer grade home computing devices. Efforts were made to optimize the training process, but the results were not entirely satisfactory.

It is important to note that the scope of this investigation was limited to consumer grade computing devices, and the findings about LSTMs should not be taken as definitive. Further research using more powerful computing resources may be necessary to fully assess the capabilities of LSTMs in the task of state estimation.

Keywords: Long short-term memory, Kalman filters, Control systems, Machine leaning, Non-linear systems**.**

ÖZ

Kontrol Sistemlerinde Makine Öğrenim Uygulamaları

EREN, Doruk
Elektrik-Elektronik Mühendisliği Yüksek Lisans

Danışman: Dr. Öğr. Üyesi. Oğuzhan ÇİFDALÖZ
Ortak Danışman Doç. Dr. Erdem AKAGÜNDÜZ
Ocak 2023, 57 sayfa

Bu tezde Kalman filtresi ile durum tahmini yapaktansa, bu görevi Uzun Kısa Dönemli Bellek (LSTM) ağına vermeyi bir potansiyel alternatif olarak kunlanmanın etkileri araştırılmıştır. Durum tahmini, bir sistemin gereç halini gürültüle veya tam olmayan gözlemlemelerden çıkartmaktır. Kalman filterler bu amaçta yaygın olarak kunlanılan bir tekniktirler, fakat bazı kısıtlamaları vardır, örneğin çıkartma yaptıkları sistemin doğrusal olduğunu varsaymaları gibi.

Yinelemeli Sinir Ağlarının (RNN)lerin bir türü olan LSTM'ler, dil çevirisi ve görütü alt yazısı olusturmak givi karmaşık ve dorusal olamayan sistemlerin modellenmesinded etkinlik gösterebilirler. Bu yüzden LSTM'lerin dinamik sistemlerin durumlarını doru şekilde tahmin etme yeteneğine sayip olabilecekleride önerildi.

Bu hipotezi deneyebilmek için farklı örenme parametrelerine sahip LSTM'lerin performansları ile bir Kalman Filtersini performansı doğrusal olmayan bir sistemde karşılaştırıldı. LSTM modellerinin eğitimi, önemli miktarda işlem gücü ve zaman gerektiriyordu, bu da tüketici sınıfı ev işlem cihazlarında zorluk yarattı. Eğitim sürecini optimize etmek için çabalar sarf edildi, ancak sonuçlar tamamen tatmin edici değildi.

Bu araştırmanın kapsamı tüketici sınıfı ev işlem cihazlarıyla sınırlı olduğundan, bu sonuçların LSTM'ler için çıkan bulguların kesin olarak kabul

edilmemesi gerekir. Durum tahmini görevinde LSTM'lerin kabiliyetlerini tam olarak değerlendirmek için daha güçlü bilgi işleme cihazları gerekli olarbilir.

Anahtar Kelimeler: Uzun kısa dönemli bellek (LSTM), Kalman filresi, Kontrol sistemleri, Makine öğrenimi, Doğrusal olmayan

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my parents for their support and encouragement throughout my academic journey. Their belief in me and their love and sacrifice have been invaluable in my pursuit of higher education. I am deeply indebted to them and am grateful for all that they have done for me.

I am deeply grateful to my supervisor, Asst. Prof. Dr. Oğuzhan ÇİFDALÖZ, for his guidance and support throughout the entire process of completing this thesis. His patience and understanding have been invaluable, and his expert advice has helped me to navigate the often-challenging waters of academic research. I am particularly thankful for the way in which he allowed me the freedom to explore and discover new insights, while at the same time keeping me grounded and focused on the task at hand. I am forever indebted to him for his invaluable contributions, and I am extremely grateful to have had the opportunity to work with such a brilliant and dedicated mentor.

I would like to express my sincere gratitude to my co-supervisor, Assoc. Prof. Dr. Erdem AKAGÜNDÜZ, for his invaluable guidance and support throughout my research process. His trust in my abilities and the freedom he has given me to explore and create my own path have been greatly appreciated. I am deeply grateful for his invaluable contributions and am indebted to him for his invaluable guidance.

I gratefully acknowledge the assistance of my friend Ezel Yalçınkaya, who was also pursuing their own master's degree while supporting me during this study. I am thankful for their friendship and support.

My special gratitude also goes to the rest of the thesis committee for the encouragement and insightful comments

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM.....iii
ABSTRACT.....iv
ÖZ.....vi
ACKNOWLEDGMENT.....viii
TABLE OF CONTENTS.....ix
LIST OF TABLES.....xi
LIST OF FIGURES.....xii
LIST OF SYMBOLS AND ABBREVIATIONS.....xiv
CHAPTER I: INTRODUCTION AND LITERATURE REVIEW.....1
1.1 MOTIVATION AND AIM OF THE STUDY.....1
1.2 FUNDAMENTALS.....2
1.2.1 Closed Loop Control Systems.....2
1.2.2 Open Loop Control Systems.....2
1.2.3 Continuous Control Systems.....3
1.2.4 Discrete Control Systems.....3
1.2.5 Machine Learning.....4
1.2.6 Long Short Term Memory.....4
1.3 LITERATURE REVIEW.....5
1.3.1 Literature Review On Control Systems With Machine Learning.....5
1.3.2 Literature Review On Control Systems With Lstms.....6
CHAPTER II: MATHEMATICAL FORMULATION.....8
2.1 INTRODUCTION.....8
2.2 LAGRANGIAN APPROACH.....8
2.3 DERIVATION OF FIXED-BASE INVERTED PENDULUM MODEL.....9

2.4 DERIVATION OF 2-LINK INVERTED PENDULUM MODEL.....9
2.5 FIXED-BASE INVERTED PENDULUM SYSTEM.....10
2.6 DOUBLE INVERTED PENDULUM.....11
CHAPTER III: MATLAB SIMULATION.....15
3.1 INTRODUCTION.....15
3.1.1 Model Importing And Validation.....15
3.2 SIMULATION OF CONTROL SYSTEM.....19
3.3 LSTM DESIGN.....21
3.3.1 Lstm Training.....21
CHAPTER IV: RESULTS AND DISCUSSION.....26
4.1 INTRODUCTION.....26
4.2 KALMAN FILTER.....26
4.3 LSTM.....28
CHAPTER V: CONCLUSION.....31
5.1 INTRODUCTION AND SUMMARY OF RESULTS.....31
5.2 COMPARISON OF KALMAN FILTER AND LSTM.....32
5.3 LIMITATION OF THE STUDY.....34
5.4 FUTURE STUDY.....34
5.5 CONCLUSION.....35
REFERENCES.....36
APPENDICES.....38
Appendix 1: LSTM Comparison Graphs.....38

LIST OF TABLES

Table 2.1: Variables and Parameters for Single Link Inverted Pendulum.....11
Table 2.2: Variables and Parameters for i-Link Inverted Pendulum.....12
Table 3.1: Specifications of the computer used for all Simulations and Training.....22

LIST OF FIGURES

Figure 1.1: A Block Diagram of a Closed Loop Control System 2
Figure 1.2: A Block Diagram of an Open Loop Control System 3
Figure 1.3: A basic representation of a Continuous Control System 3
Figure 1.4: A basic representation of a Discrete Control System 4
Figure 1.5: A LSTM memory cell 5
Figure 2.1: A 2d Front View Representation of a Fixed Base Inverted Pendulum 11
Figure 2.2: A 2d Front View Representation of a Double Inverted Pendulum 12
Figure 3.1: Bode plot for T_{ry}. 17
Figure 3.2: Bode plot for T_{re}. 17
Figure 3.3: Bode plot for T_{diy} 18
Figure 3.4: Bode plot for T_{ru} 19
Figure 3.5: Simulink Diagram of the Simulation Process. 20
Figure 3.6: Animation Output from MATLAB Simulink Simulation. 21
Figure 3.7: Training Progress Graph 1 for Network with 400 Hidden Values 1000 Epoch, the first of four training steps. 23
Figure 3.8: Training Progress Graph 2 for Network with 400 Hidden Values 1000 Epoch, the second training step. 23
Figure 3.9: Training Progress Graph 3 for Network with 400 Hidden Values 1000 Epoch, the third training step. 24
Figure 3.10: Training Progress Graph 4 for Network with 400 Hidden Values 1000 Epoch, the last training step. 25
Figure 4.1: Difference between Simulated Data (Green), and Kalman Approximation (Yellow) of $\dot{\theta}_1$ 27
Figure 4.2: Difference between Simulated Data (Purple), and Kalman Approximation (Blue) of $\dot{\theta}_2$ 27
Figure 4.3: Simulink Diagram of the Simulation Process with added LSTM block ...28

Figure 4.4: Difference between Simulated Data (Yellow), and LSTM Approximation (Purple) of $\dot{\theta}_1$ 29
Figure 4.5: Difference between Simulated Data (Blue), and LSTM Approximation (Teal) of $\dot{\theta}_2$ 29
Figure R.1: 5 Lstms θ_1 response from simulataenously responding to the same input 38
Figure R.2: θ_1 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R1 is present) 38
Figure R.3: θ_1 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R1 or R2 is present) 39
Figure R.4: θ_1 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R1, R2 or R3 is present) 39
Figure R.5: 5 Lstms θ_2 response from simulataenously responding to the same input 40
Figure R.6: θ_2 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R5 is present) 40
Figure R.7: θ_2 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R5 or R6 is present) 41
Figure R.8: θ_2 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R5, R6 or R7 is present) 41

LIST OF SYMBOLS AND ABBREVIATIONS

Symbols

rad	: Radian
M	: Meter
L	: Lagrangian
N	: Newtons of Force
E_k	: Kinetic Energy (N)
E_p	: Potential Energy (N)
F	: Force (N)
s	: Seconds
Nm	: Newton Meters
g	: Accelaration due to Gravity ($^{m}/_{s^2}$)
c	: Friction coefficient (Nm)
cg	: Center of gravity
x	: Unknown variable
y	: Unknown output
I	: Inertia
u	: Control variable
r	: Input variable
e	: Error
di	: Input distrubance
T	: Transfer function

Greek Letters

θ	: Theta, used for angles (rad)
τ	: Torque (N)

Subscripts

1	: To do with the base link
2	: To do with the second link
i	: To do with link ‘i’
ry	: From input r to output y
re	: From input r to error e
diy	: From input distrubance di to output y
ru	: From input r to control variable u

Abreviations

LSTM	: Long Short Term Memory
RNN	: Reccurant Neural Network
RMSE	: Root Mean Square Error
SISO	: Single Input Single Output
MIMO	: Multiple Input Multiple Output

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

1.1 MOTIVATION AND AIM OF THE STUDY

Control systems are systems designed to manage devices or systems. These devices can range from the simple thermostat inside a home to the complex machines and processes used to control industrial systems, machines and processes. Kalman Filters also known as linear quadratic estimators are algorithms that uses measurements over time along with any inaccuracies in said measurements to create a more accurate result than any single measurement alone using joint probability distribution. Kalman Filters are recursive estimators and therefore do not require a history of observations nor estimates. Machine learning is an umbrella term used to describe a way to, in theory, teach computer systems how to do complex and sometimes impossible-to-code actions by training data. Long short-term memory (LSTM) is a machine learning system that is well suited to classify, process or make predictions based on data collected over time.

One of the most common issues with control systems in real world conditions is that the required information for the control system to control all states can sometimes be unavailable to access due to either limited sensors or due to certain states simply being hidden. A common method to deal with limited sensors and hidden states is to use a Kalman filter. The method in creating a Kalman filter uses the systems' state space model and therefore the accuracy and the ability of the Kalman filter is directly proportional to the accuracy of the model. Kalman filters therefore are rather computationally expensive since Kalman filters are technically simulating the system they are trying to estimate. An LSTM has the capability to replace the processing and predicting current and future data that would be the result of a Kalman Filter while using less computational power when pretrained and deployed, as well as under ideal situations, adapt to changes to the preset model, since instead of being fed an "expected

parameters" list, the LSTM learns possible results from possible inputs and estimates from that step onward.

1.2 FUNDAMENTALS

Control Systems are systems that take in inputs from the world and manipulate them mathematically so that the resultant output is as desired. To achieve this effect, control systems can operate in a variety of methods, and under different conditions. Categorization in control systems can be based on feedback type, closed loop or open loop, the type of signal used, continuous or discrete, and finally based on the number of inputs and outputs, SISO (Single Input Single Output) or MIMO (Multiple Inputs Multiple Outputs). [12]

1.2.1 Closed Loop Control Systems

Closed loop control systems are control systems, where the output is fed back into the control system as an input, thus allowing the control system to be able to monitor and react to the changes in the output as well as the input. These systems are more accurate as the reaction from the system is based on the intended results, however such systems are more difficult to design, and more expensive.

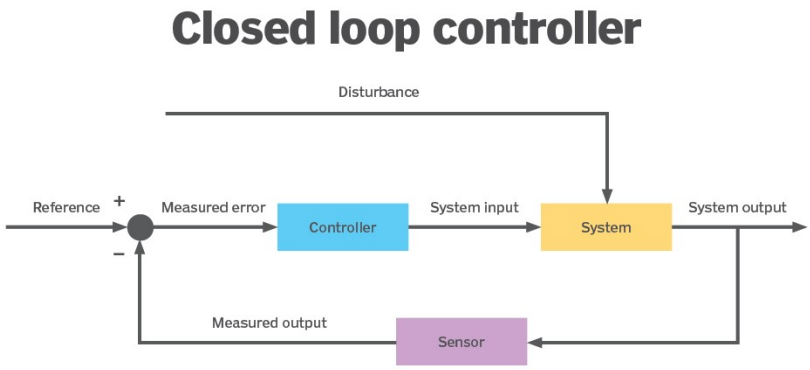

Figure 1.1: A Block Diagram of a Closed Loop Control System [13]

1.2.2 Open Loop Control Systems

Open loop control systems are control systems, where the output is not fed back into the control system as an input. Since the control system is unaware of the output,

the input is the only factor determining the actions taken by the system to achieve a desired result. These systems are technically not automatic, since they cannot adapt to changes. Due to having no feedback, they are sometimes also called non-feedback control systems.

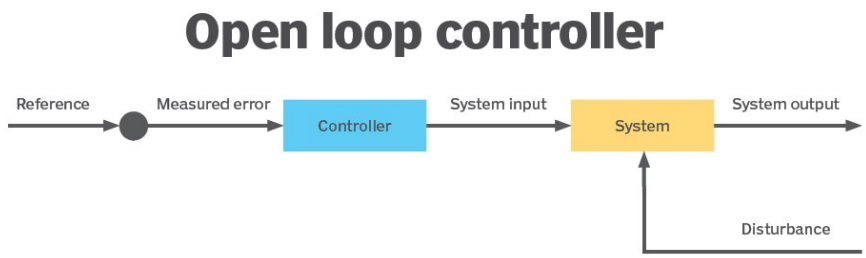

Figure 1.2: A Block Diagram of an Open Loop Control System [13]

1.2.3 Continuous Control Systems

Continuous control systems are systems that continuously monitor and control a process or physical variable. These systems are designed to keep a process within a certain range or to track a specific set point. They're used in a wide range of operations, such as temperature control, speed control, position control, and chemical process control. They're extensively used in artificial processes, robotics, and other areas where precise control is warranted.

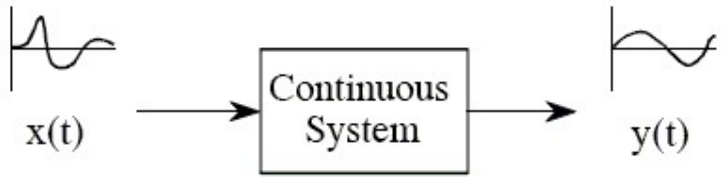

Figure 1.3: A basic representation of a Continuous Control System [9]

1.2.4 Discrete Control Systems

Discrete control systems use discrete input signals to extrapolate discrete output signals. The variables used change predominantly in a discontinuous manner. These changes occur on specific time intervals and depending on the rate at which each next discrete step is taken (the sampling rate) approaches or strays aways from a continuous control system.

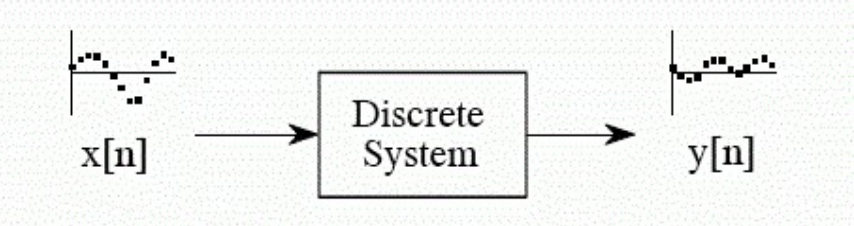

Figure 1.4: A basic representation of a Discrete Control System [9]

1.2.5 Machine Learning

Machine Learning is a topic under the umbrella of computer science. Machine Learning deals with creating algorithms that use collection of examples of prior experiences to become useful. Machine learning might therefore be defined as gathering a dataset, and then training a statistical model according to the dataset. [4] How well a statistical model can be trained depends on the method used in training the statistical model.

The different training methods can be summarized as Supervised, Unsupervised, Semisupervised and Reinforcement learning. Supervised learning is where the dataset is clearly labeled, so the training doesn't have to guess as to what the dataset contains, the main goal is to allow the training to focus on the connections between inputs and outputs. Unsupervised learning therefore is the opposite, where the dataset is not labeled, thus allowing for the statistical model to only find patterns, this form of training works well for dimensionality reduction, or outlier detection. Semisupervised learning is where a dataset that is both labeled and not labeled is used, with non-labeled data count being higher, which helps in the training process. [3] Reinforcement learning is where the statistical model is fed information from the states directly, and where the actions feedback to the states directly, a sort of trial-and-error method where the statistical model measures the responses to the changes made by itself and modifies itself so that the responses to the changes are in the parameters of the intended target.

1.2.6 Long Short Term Memory

Long Short Term Memory (LSTM) are a type of Recurant Neural Networks (RNN) first proposed by Schmidhuber and Hochreiter [8] to allow for learning

algorithms to store information over extended time intervals. LSTM's are able to be used for multitudes of issues, and therefore are the most common type of RNN in use. LSTM's main role is sustained by a 'cell state' inside a memory cell. This cell state can be interacted with, and information can be removed or added inside the memory cell.

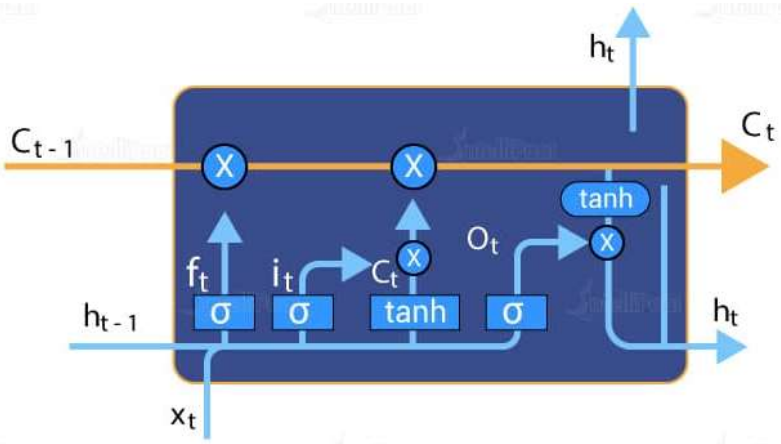

Figure 1.5: A LSTM memory cell [7]

1.3 LITERATURE REVİEW

There Machine Learning Systems and all possible applications of Machine learning systems are comparably new topics, due to the interest in the possibilities, there are multiple studies on different use cases. The studies with goals that are relevant and helped mold this study are summarized below.

1.3.1 Literature Review on Control Systems with Machine Learning

The major issue with control systems, that machine learning hoped to solve initially was identification of a nonlinear dynamic system. In a study done by B. Fernandez, A. G. Parlos and W. K. Tsai in the year 1990 a multilayer perceptron network was used to identify nonlinear dynamic systems from a black box where only inputs and outputs were available as input for the machine learning system [2]. While the goal of this study differs from the identification goal of the 1990 paper, the use of recurrent dynamic networks is a good baseline. In the 2019 paper by A. Ata, M.A. Khan, S. Abbas, G. Ahmad and A. Fatima design a system capable of predicting and

reacting before congestion occurs in a traffic control system via the use of artificial neural networks [1]. While the usage of a machine learning system is predominantly still in the identification stage, this study does highlight the ability of machine learning systems to be able to adapt to previous events and use this adaptation to predict future events.

In a paper from 2020 the team of Plasmas 27 used a machine learning algorithm, trained on data from a real-time control system, to control and predict events in a tokamak style plasma fusion reactor [14]. While the study only achieved around 90% accuracy with true positives and around 10% erroneous false positives, the possibility of further success is very possible. During a 2020 study S. Yang, M. P. Wan, W. Chen, B. F. Ng and S Dubey explored the use of a machine learning based system to control a building in real-time [10]. Using machine learning as again a way to model a changing environment. By the methodology used, a thermal energy saving of 58.5% in one case, and 36.7% in another had been achieved. While these numbers are high, these savings are only the amount saved from the initial non-machine learning based control system, and therefore are not the actual saving to energy consumption if no control system was present beforehand.

All these studies share the understanding that while machine learning systems are impressive in what they can achieve, the need for prior data, and multiple learning cycles cause machine learning systems to be only applicable in areas where datasets have been gathered, and extensively moderated.

1.3.2 Literature Review on Control Systems with LSTM's

Within the umbrella of machine learning, LSTMs are a great addition to control systems due to their high adaptability, as well as wide range of use scenarios. During a 2019 study S. M. Gharghory explored the use of a LSTM to predict the temperature and humidity inside a greenhouse. The LSTM trained with the data collected over the course of 27 days led to a root mean square error of just 0.16 and 0.62 for temperature and humidity respectively [6].

During a 2020 paper by C. Su, X. Wang, L. Shen and H. Yu studied the flight of an UAV (unmanned aerial vehicle). Noticing the missing areas in the control system for a UAV during aggressive maneuvering, the team placed a LSTM based motion

command generation method to better handle the coupling and singularity problems that occurred without the additional control systems [11].

General interest in LSTM's has only increased as this study started, and new studies were released during the duration of this study with overlapping purposes. A 2019 study by Z. Zhou, R. Zhang and Z Zhu explored the use of Kalman filtering with a LSTM for image-based visual servo control. The team added an LSTM with a robust Kalman filtering algorithm, allowing the Kalman filtering that is usually sensitive to noise to gain a strong anti-noise interference ability [16].

A 2020 study by Y. Tian, R. Lai, X Li, L Xiang, and J Tian used LSTM's alongside Kalman filters. Using an LSTM alongside a Kalman filter, this study managed to decrease the estimator complexity while allowing for the combined system to show better generalization and convergence abilities [15].

During a 2022 study F. Song, Y. Li, W. Cheng, L. Dong, M. Li and J. Li worked on improving a Kalman filter based on a LSTM for the purpose of tracking nonlinear radar targets. Using 3 separate LSTMs to calculate motion equation, motion noise and measurement noise, and then using these calculations as the input of a Kalman filter. The end results of this hybrid approach were that the end results were better than either Kalman filters or LSTMs could produce on their lonesome [5].

CHAPTER II
MATHEMATICAL FORMULATION

2.1 INTRODUCTION

This chapter defines the decision process, and the mathematical calculations done for the model that serves as the basis for all the testing during the scope of this study. All linearization done to the non-linear model, and how the different equations describe the model will be expanded upon.

2.2 LAGRANGIAN APPROACH

The first step is to find the Lagrangian Function by computing the difference between the kinetic and potential energy of a system. By letting E_p denote the potential energy associated with the system and E_k denote the kinetic energy of the system, the Lagrangian of the system is given by:

$$\boldsymbol{L = E_k - E_p}$$

Eq. 2. 1

After the Lagrangian Function is found, the dynamic equation can be expanded upon. If the dissipative forces or torques on θ_i can be defined as F_{θ_i} then the dynamical equations that denote the full system may be written as below:

$$\boldsymbol{\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\theta}_i}\right) - \frac{\delta L}{\delta \theta_i} = F_{\theta_i}}$$

Eq. 2. 2

where θ_i denotes the states.

2.3 DERIVATION OF FIXED-BASE INVERTED PENDULUM MODEL

A fixed base inverted pendulum can be located on any point along the circle of radius r_p depending on the angle at the base. This can be represented with the equation below:

$$r_p = \begin{bmatrix} l_{cg} sin\theta \\ l_{cg} cos\theta \end{bmatrix}$$

Eq. 2. 3

Then the total kinetic energy of the system can be defined as:

$$E_k = \frac{1}{2} m \dot{r}_p * \dot{r}_p + \frac{1}{2} I_{cg} \dot{\theta}^2$$

Eq. 2. 4

While the total potential energy of the system is:

$$E_p = \mathrm{m} g l_{cg} cos\theta$$

Eq. 2. 5

Therefore, the Lagrangian for the fixed base inverted pendulum:

$$L = E_k - E_p = \frac{1}{2}\left(m l_{cg}^2 + I_{cg}\right)\dot{\theta}^2 - \mathrm{m} g l_{cg} cos\theta$$

Eq. 2. 6

2.4 DERIVATION OF THE 2-LINK INVERTED PENDULUM MODEL

Let the position vector for the first link be:

$$\mathrm{r}_1 = \begin{bmatrix} l_1 \, sin\theta_1 \\ l_1 \, cos\theta_1 \end{bmatrix} \Rightarrow \dot{r}_1 = \begin{bmatrix} l_1 \dot{\theta}_1 \, cos\theta_1 \\ -l_1 \dot{\theta}_1 \, sin\theta_1 \end{bmatrix}$$

Eq. 2. 7

Then the total kinetic energy in the first link shown in equation (2.8), the total potential energy in the first link is shown in equation (2.9) and the non-conservative moment acting on the first link can be seen in equation (2.10)

$$E_{k_1} = \frac{1}{2}(m_1 l_1^2 + I_1)\dot{\theta}_1^2$$

Eq. 2. 8

$$E_{p_1} = m_1 g l_1 cos\theta_1$$

Eq. 2. 9

$$-c_1\dot{\theta}$$

Eq. 2. 10

Similarly, the second links position vector can be described via equation (2.11).

$$r_2 = \begin{bmatrix} L_1\, sin\theta_{1+l_2 sin\theta_2} \\ L_1\, cos\theta_1 + l_2 cos\theta_2 \end{bmatrix} \Rightarrow \dot{r}_2 = \begin{bmatrix} L_1\dot{\theta}_1\, cos\theta_1 - l_2\dot{\theta}_2 cos\theta_2 \\ -L_1\dot{\theta}_1\, sin\theta_1 - l_2\dot{\theta}_2 sin\theta_2 \end{bmatrix}$$

Eq. 2. 11

Using equation (2.11) the kinetic energy in the second link can be described by equation (2.12), the potential energy by equation (2.13) and the non-conservative moments by equation (2.14).

$$E_{k_2} = \frac{1}{2}m_2 L_1^2\dot{\theta}_1^2 + \frac{1}{2}(m_2 l_2^2 + I_2)\dot{\theta}_2^2 + m_2 L_1 l_2\dot{\theta}_1\dot{\theta}_2 cos(\theta_1 - \theta_2)$$

Eq. 2. 12

$$E_{p_2} = m_2 g L_1 cos\theta_1 + m_2 g l_2 \cos\theta_2$$

Eq. 2. 13

$$-c_2\dot{\theta}$$

Eq. 2. 14

Combining equations (2.7) to (2.14) the Lagrangian for the system using the formula in equation (2.15) can be seen in equation (2.16)

$$\mathrm{L} = \sum_i E_{k_i} - \sum_i E_{p_i}$$

Eq. 2. 15

$$\mathrm{L} = \frac{1}{2}(m_1 l_1^2 + m_2 L_1^2 + I_1)\dot{\theta}_1^2 + \frac{1}{2}(m_2 l_2^2 + I_2)\dot{\theta}_2^2 + m_2 L_1 l_1\dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) - m_1 g l_1 cos\theta_1 - m_2 g l_1 cos\theta_1 - m_2 g l_2 cos\theta_2$$

Eq. 2. 16

2.5 FIXED-BASE INVERTED PENDULUM SYSTEM

As can be seen below in (Figure 2.1) a single link inverted pendulum has multiple variables and parameters that must be considered before mathmatical representation can be made. These variables and parameters can be seen below in (Table 2.1).

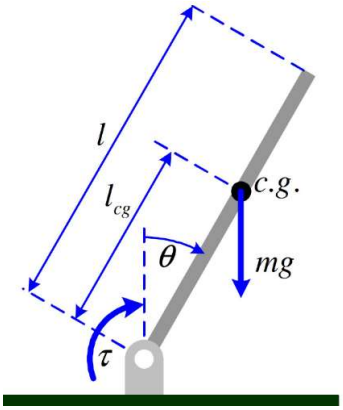

Figure 2.1: A 2d Front View Representation of a Fixed Base Inverted Pendulum.

Table 2.1: Variable and Parameters for Single Link Inverted Pendulum

Identification	Symbol	Definition	Unit
Variable	θ	Angle between vertical and pendulum	rad
Variable	τ	Torque applied to pendulum	Nm
Parameter	m	Mass of pendulum	kg
Parameter	c	Friction coefficient for motion of pendulum	kg * m/s^2
Parameter	g	Specific acceleration due to gravity	m/s^2
Parameter	l_{cg}	Location of center of gravity for the pendulum	m
Parameter	I_{cg}	Inertia acting on the center of gravity of the pendulum	Nm/rad^2/s^2

Using the information from (Figure 2.1) and (Table 2.1) dynamical equation can be calculated.

2.6 DOUBLE INVERTED PENDULUM

The system used in this project is designed to be a good analog for general nonlinear systems. Using the special characteristics of a double inverted pendulum system, it becomes possible to test the different computational costs, and accuracies of different methodology for finding sensor values without the use of sensors with either common filters or machine learning systems. The first step when dealing with this new nonlinear system is to simulate the model in MATLAB so that different methods can be tested and logged accuratly, efficienctly and consistently.

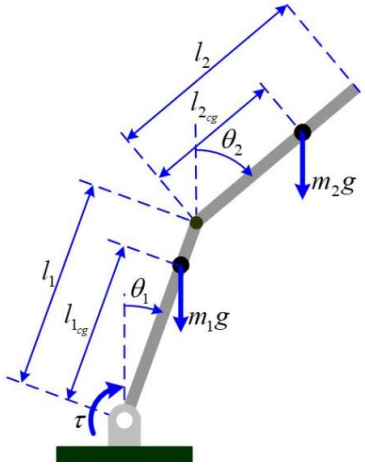

Figure 2.2: A 2d Front View Representation of a Double Inverted Pendulum.

As can be seen in the figure above (Figure 2.2), the model used during this study has a fixed joint and a moving joint. On the fixed joint, where the pendulum connects to the ground plane, a motor is fixed such that the angle of the bottom joint (fixed joint (θ_1)) can be controlled via the control system. On the moving joint (upper join (θ_2)) there is an encoder which allows for high speed, accurate logging of the angle of the upper joint. Both joints move in the same plane, thus allowing for 1 motor to control both angles, albeit with difficulty.

Compared to the single link inverted pendulum, pendulums of *i* links have slightly different parameters, which can be seen below in (Table 2.2)

Table 2.2: Variable and Parameters for *i*-Link Inverted Pendulum

Identification	Symbol	Definition	Unit
Variable	θ_1	Angle between vertical and pendulum	rad
Variable	θ_2	Angle between the *i*th link and the link before	rad
Variable	τ	Torque applied to the first link	Nm
Parameter	m_i	Mass of the *i*th link	kg
Parameter	l_i	Lenght of the *i*th link	m
Parameter	$l_{i_{cg}}$	distance between *i*th links center of gravity and the *i*th hinge	m
Parameter	$I_{i_{cg}}$	moment of inertia on the *i*th link about the center of gravity	$Nm/rad^2/s^2$
Parameter	g	Specific acceleration due to gravity	m/s^2
Parameter	c_i	Friction coefficient for motion of the *i*th link	$kg * m/s^2$

Using information from (Figure 2.2), (Table 2.2), (Equation 2.16) and (Equation 2.17) the nonlinear dynamical equation for a 2-link pendulum can be calculated as:

$$m_2 L_1 l_2 \cos(\theta_1 - \theta_2) \quad \text{Eq. 2. 17}$$

$$m_1 l_1^2 + m_2 L_1^2 + I_1 \quad \text{Eq. 2. 18}$$

$$-m_2 L_1 l_2 \sin(\theta_1 - \theta_2)\, \dot{\theta}_2^2 + (m_1 g l_1 + m_2 g L_1) \quad \text{Eq. 2. 19}$$

$$sin\theta_1 - c_1 \theta_1 + \tau \quad \text{Eq. 2. 20}$$

$$m_2 L_1 l_2 \sin(\theta_1 - \theta_2)\, \dot{\theta}_1^2 + m_2 g l_2 \sin\theta_2 - c_2 \dot{\theta}_2 \quad \text{Eq. 2. 21}$$

$$\begin{bmatrix} (Eq.2.19) & (Eq.2.18) \\ (Eq.2.18) & m_2 l_2^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} (Eq.2.20) \\ (Eq.2.21) \end{bmatrix} \quad \text{Eq. 2. 22}$$

Defining the state vector as below in (Equation 2.23) and linearizing (Equation 2.22) at (Equation 2.24) and (Equation 2.25) yields the state-space representation for the 2-link pendulum.

$$x_p = [\theta_1 \quad \dot{\theta}_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T \quad \text{Eq. 2. 23}$$

$$x_{p_e} = [0 \quad 0 \quad 0 \quad 0]^T \quad \text{Eq. 2. 24}$$

$$\tau_e = 0 \quad \text{Eq. 2. 25}$$

$$E_p \dot{x} = \widehat{A}_p x + \widehat{B}_p \tau \quad \text{Eq. 2. 26}$$

$$E_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (m_1 l_1^2 + m_2 L_1^2 + I_1) & (m_2 L_1 l_2) \\ 0 & 0 & (m_2 L_1 l_2) & m_2 l_2^2 + I_2 \end{bmatrix} \quad \text{Eq. 2. 27}$$

$$\widehat{A}_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ m_1 g l_1 + m_2 g L_1 & 0 & -c_1 & 0 \\ 0 & m_2 g l_2 & 0 & -c_2 \end{bmatrix} \quad \text{Eq. 2. 28}$$

$$\widehat{B}_p = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{Eq. 2. 29}$$

(Equation 2.26) can also be written in state-space form as seen below in (Equation 2.30).

$$\dot{x}_p = A_p x_p + B_p \tau = \left(E_p^{-1} \widehat{A}_p\right) x_p + \left(E_p^{-1} \widehat{B}_p\right) \tau$$

Eq. 2. 30

CHAPTER III
MATLAB SIMULATION

3.1 INTRODUCTION

The model and state space equations (2.28), (2.29) and (2.30) are moved into MATLAB then prepped for simulation by designing a basic control system, so that the model's base responses can be validated.

As per the focus of this study, following the simulation of the model, an LSTM neural network is designed. This network is then modified, so that the changes made to the amount of hidden values present in the network, as well as the training duration, otherwise known as max epochs of training, can be modified, to see the effects on the computational cost during training and operation, as well as changes to the results.

The system modeled in MATLAB can be seen in Figure 3.2. As can be seen the system has 2 joints. The joint connected to the ground plane with the angle designation θ_1 and is connected to a motor directly, thus allowing θ_1 to be directly controlled. θ_1 is in a plane directly perpendicular to the ground plane. θ_1 is connected to a secondary joint with angle designation θ_2. θ_1 and θ_2 move in the same plane. The distance between the two joints is L_1. L_1 is assumed to be a rod of uniform mass distribution. After joint 2 (θ_2) a secondary rod of uniform mass L_2 exists. Joints 1 and 2 have 0 extra mass, beyond the mass of the rods L_1 and L_2 in those locations.

3.1.1 Model Importing and Validation

To begin with, the equations found previously (The state space descriptor equations (2.28), (2.29) and (2.30)) are converted to a State Space Matrices where:

$$\boldsymbol{A = inv(E_p) * A_p} \qquad \textbf{Eq. 3. 1}$$

$$B = inv(E_p) * B_p$$

Eq. 3. 2

$$C = [1 \quad 0 \quad 0 \quad 0]$$

Eq. 3. 3

$$D = 0$$

Eq. 3. 4

By incorporating an integrator in equations (3.1), (3.2), (3.3) and (3.4) along with a LQ Servo control system design, the final form of the equations can be attained for farther analysis. The goal of this feedback system is to control the angle of the bottom link, while maintaining the upper link in a perpendicular position relative to the ground, which is represented by $\theta_2 = 0$.

$$C_p = [1 \quad 0 \quad 0 \quad 0]$$

Eq. 3. 5

$$C_r = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Eq. 3. 6

$$C = \begin{bmatrix} C_P \\ C_r \end{bmatrix}$$

Eq. 3. 7

$$A_{aug} = \begin{bmatrix} 0 & C_P \\ 0 & A \end{bmatrix}$$

Eq. 3. 8

$$B_{aug} = \begin{bmatrix} 0 \\ B \end{bmatrix}$$

Eq. 3. 9

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Eq. 3. 10

$$R = 0.01$$

Eq. 3. 11

Using the equations above from (3.5) to (3.11), the figure below (3.1) is generated to validate the expected frequency response of the system, both proving feasibility of the control system, and anchoring the expected results from the control system.

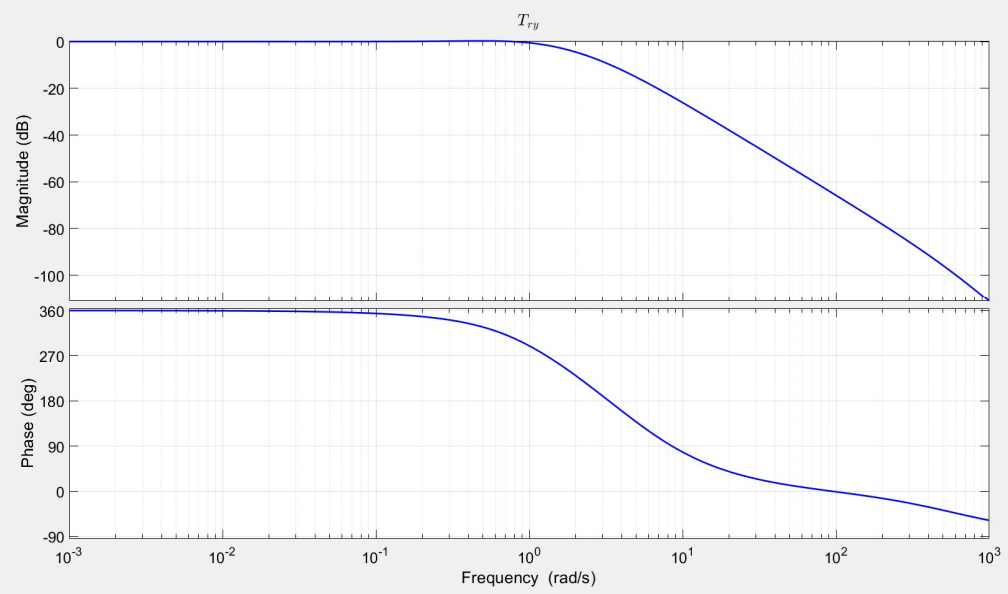

Figure 3.1: Bode plot for T_{ry}.

As Figure 3.1 illustrates at low frequencies, the magnitude of the error signal is 0 dB, indicating that the control system directly follows reference commands when the input frequency is smaller than 2 rad/ sec. This is in line with the anticipated behavior of the system. However, at high frequencies, the magnitude of the error signal is approaching 0, which implies that the control system effectively attenuates high-frequency noise. This result is also compatible with the expected behavior of the system.

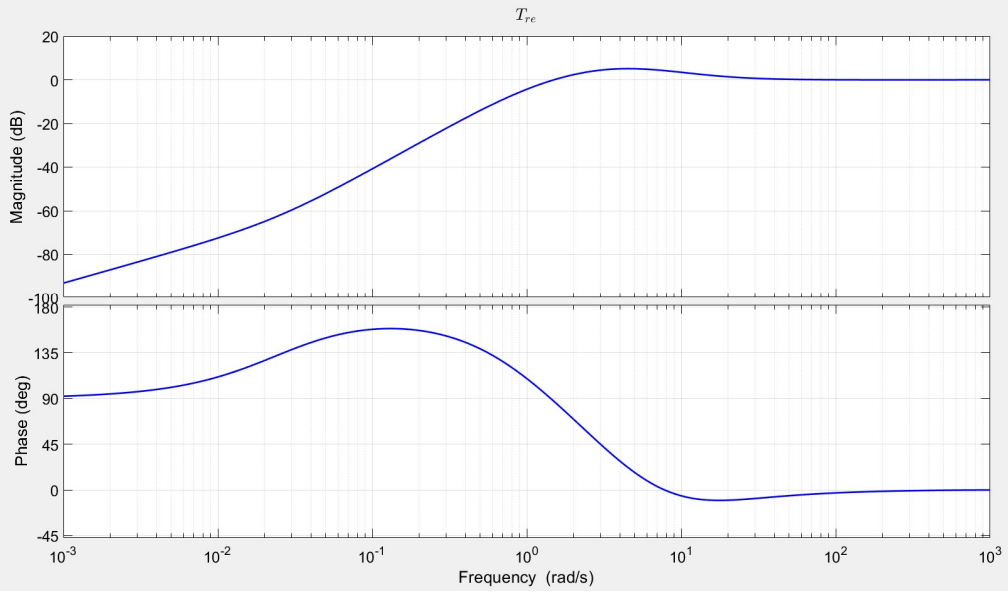

Figure 3.2: Bode plot for T_{re}.

The graph in Figure 3.2 shows the relationship between the input signal(r) and the error signal(e) of a control system. The graph indicates that the input signal has a minimum impact on the error signal. At most frequency points, the effect of the input signal on the error signal is either zero or has a negative dB magnitude, indicating a minimum impact. The graph also shows that at low frequencies, the steady- state error is small, which is in line with the frequency response of the T_{ry} system. also, the small magnitude at low frequencies means that low frequency output disturbances will be rejected. still, at high frequencies, the steady- state error is large, which is also in compliance with the T_{ry} frequency response.

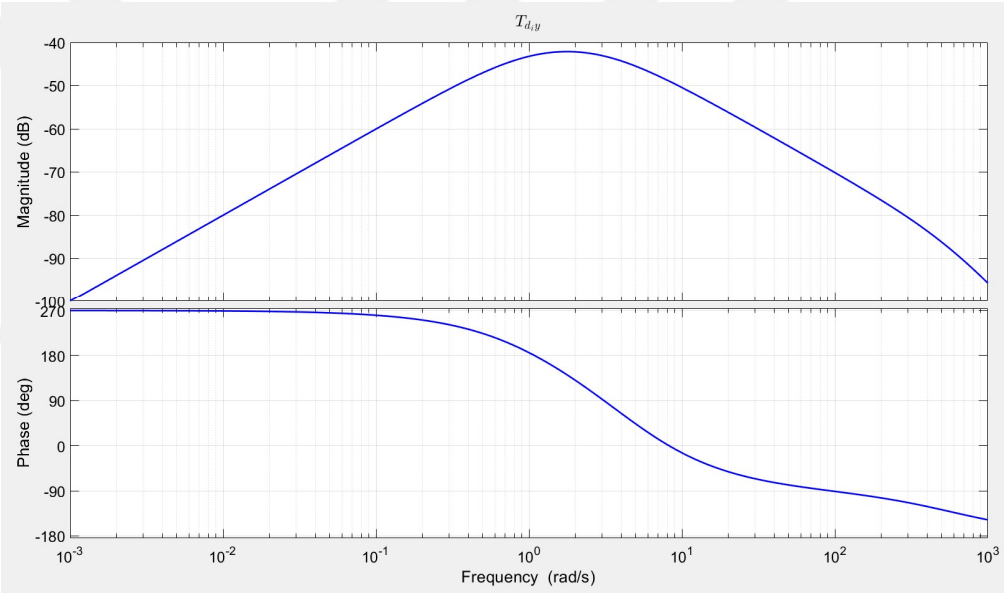

Figure 3.3: Bode plot for T_{diy}

The graph in Figure 3.3 demonstrates the effect of input disturbance *di* on the output signal *y* of a control system. As shown in the graph, the magnitude of the effect of input disturbance *di* on the output signal *y* is very low. It reaches at most -40dB at its highest point, indicating that the disturbance has a minimal impact on the output signal. This implies that the system is relatively robust to input disturbances.

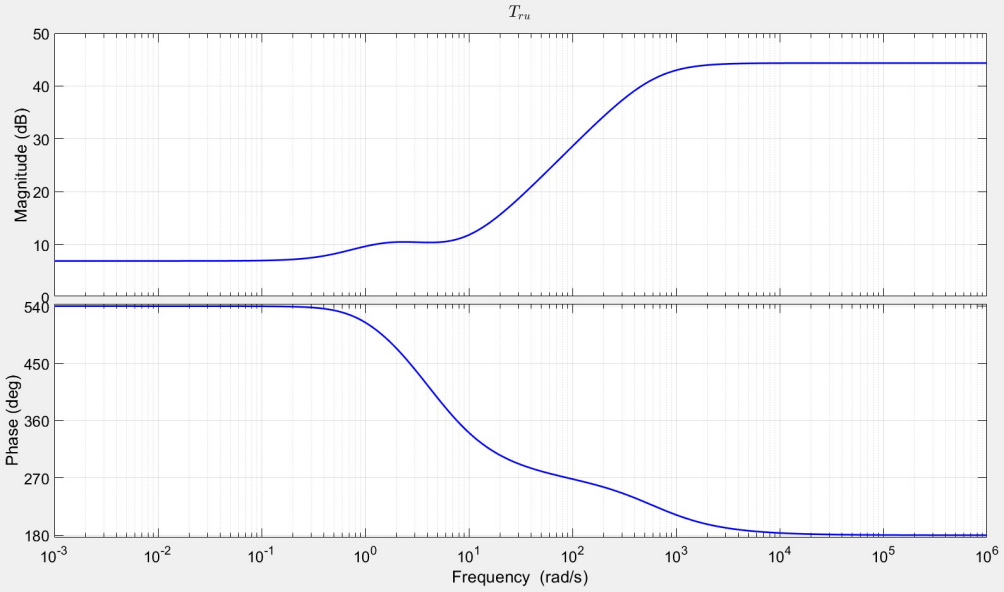

Figure 3.4: Bode plot for T_{ru}

The frequency response graph in Figure 3.4 illustrates the effect of the reference input r on the control input u in a control system. The graph provides insights into the relationship between the reference input r and the control input u and how they affect the system's performance. Variations in the effect of the input *r* on the control variable *u* at different frequencies was taken into account when designing the control system. At low frequencies the magnitude of the control input is smaller than at high frequencies. The magnitude of T_{ru} at high frequencies is large, meaning that measurement noise is going to have a large effect on the control output. In a real physical implementation controlers are augmented with a low pass filter in order to remedy the noise applification problem. In this thesis, the low pass filter is ommited in order to keep the system simple. This ommition is not going to have any impact on the study conducted during this thesis.

3.2 SIMULATION OF CONTROL SYSTEM

Using the equations above (3.1) to (3.4) a Kalman Filter is designed, with the parameters of Eq. (3.11) and

$$Q = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \tag{3.12}$$

This Kalman Filter is Discretized and placed into a MATLAB SIMULINK along with the initial model in a nonlinearized fashion and simulated to control for all unexpected errors that may occur when dealing with a nonlinear system.

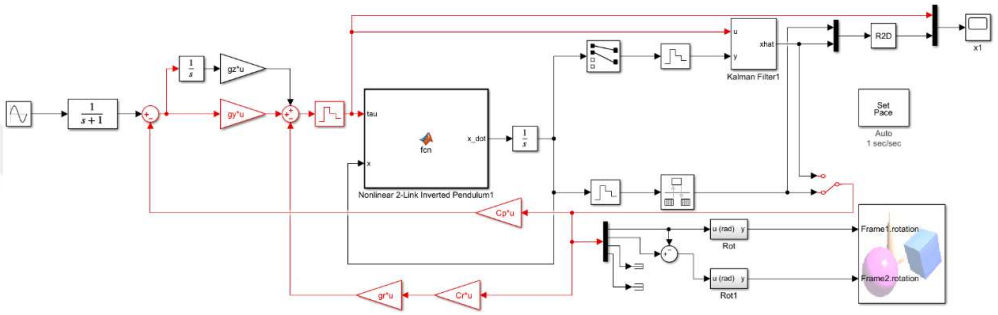

Figure 3.5: Simulink Diagram of the Simulation Process.

As can be seen on Figure 3.5 the simulation has multiple paths that lead to the same direction, this is done to allow for both simulated and real world scenarios to be tested simultaneously, and with minimal downtime. The Kalman Filter in Figure 3.5 is assuming real world scenario where angle measurements for θ_1 and θ_2 are avaliable, however angular velocities ($\dot{\theta}_1$ and $\dot{\theta}_2$) are not avaliable. The simulation also has a 3d animation, in which the motion of the double inverted pendulum can be tracked and observed. A still from this animation can be observed below, with pink being the top link, yellow being where the top link is attached to the bottom link, blue being the bottom link, and red being the motor attached to the bottom link, controlling the bottom links angle.

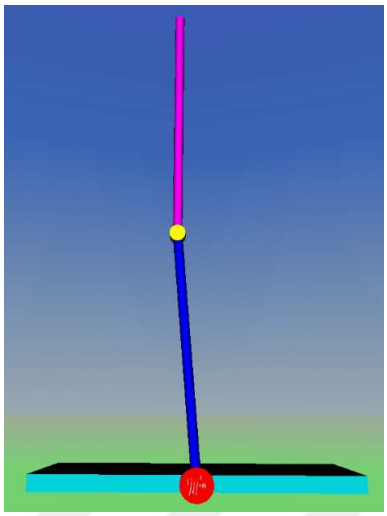

Figure 3.6: Animation Output from MATLAB Simulink Simulation.

3.3 LSTM DESIGN

The start of the design process was to decide what kind of neural network would work best for replacing the Kalman Filter. Given that Kalman Filter's are estimators, an estimator network being a replacement made direct logical sense, and therefore a LSTM was picked. LSTM's ability to predict values with comparibly small network sizes was the lead reasoning. The LSTM design used during the study always had a total of four layers. The first layer, a Sequence Input Layer, with three inputs, allows the LSTM to take in values as a sequence, allowing the LSTM to continue predicting during operation. The second layer, an LSTM Layer, where the inputs from the input layer are operated on using hidden values that will be trained. The third layer, the one and only Fully Connected Layer, allowing each value inside the previous layer to affect the other, and congregating the values down to two outputs. The fourth and last layer, consisting of a Regression Layer, helping establish connections between dependent and independent values

3.3.1 LSTM Training

The LSTM design was modified, so that the hidden values in layer 2 can be changed, to allow for multiple different hidden value counts to be tested. These different LSTM models were then trained four times, using separate datasets each time, for different max epochs, to test for over and under training, as well as memorization of the data. Of all networks trained, networks trained with one thousand max epochs

performed the best, and therefore were prioritized. Each network was trained on a personal computer, with the specs in the table below:

Table 3.1: Specifications of the computer used for all Simulations and Training.

Part Name/Designation	Part Size/Core Count	Rated/Running Speed
CPU-AMD Ryzen 7 3800X Processor	8-cores	3.89 GHz
Corsair Vengeance Pro DDR4 Ram	4 x 8 GB sticks for a total of 32 Gigabytes	3200 MHz
MSI Nvidia GeForce RTX 2080 SUPER	8 GB of GDDR6 Memory	1650 MHz
Samsung 970 Evo Nvme M.2 storage	500GB	3400 MB Read 2300 MB Write

Each training of a 1000 max epoch network took roughly 240 minutes, and each network was trained 4 times, leading to each fully trained network being 960 minutes or 16 hours of training. Training was left overnight in most cases, since the personal computer would be locked up to calculate the training. Despite the intensive training however, all training was completed successfully, and both the Loss function and the RMSE of the models were seen to drop significantly after 4 training sessions.

To train each network datasets from the simulation were used, however for the training datasets, the data acquired was from the original data, and not the Kalman Filter approximations. The training was done with initial conditions for the system (the input in terms of sin wave) having been altered. These alterations were to both the amplitude and the frequency of the sin wave. The highest amplitude input used was $9\pi/180$ while the lowest was $3\pi/180$. The fastest frequency used was 2rad/sec while the slowest frequency was 0.5rad/sec. The changes to frequency and amplitude were not linked.

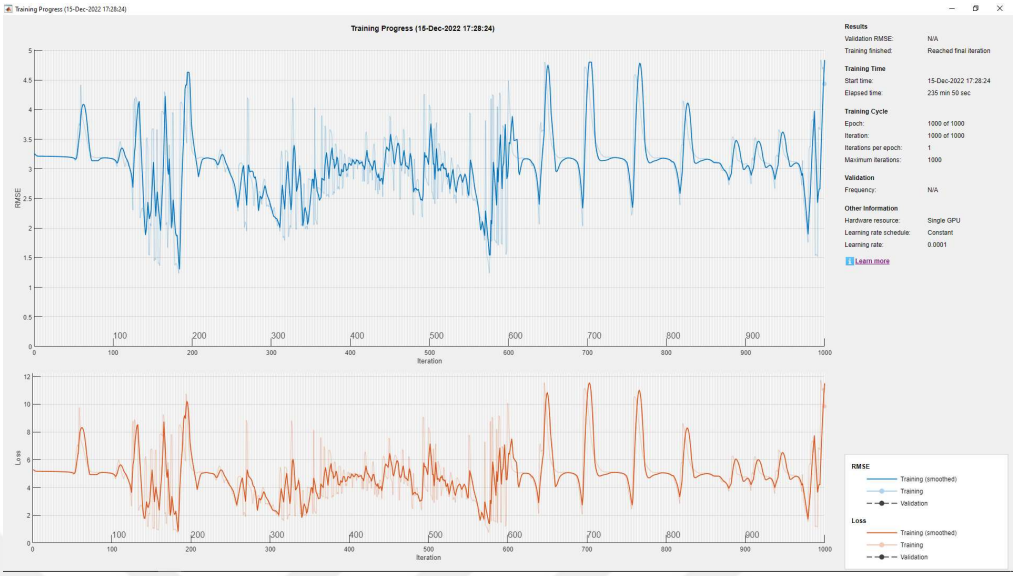

Figure 3.7: Training Progress Graph 1 for Network with 400 Hidden Values 1000 Epoch, the first of four training steps.

In Figure 3.7 we can see the network fail to learn how to deal with the changing sinus values from the training data, and the irregularities in the nonlinear system. At the end of the training the loss function sits at a peak of almost 12 meaning a high number of predictions made by the LSTM were wrong. The RMSE at the same point goes as high at 4.8, meaning that those errors are rather seperate. This training step was completed in 235 minutes and 50 seconds.

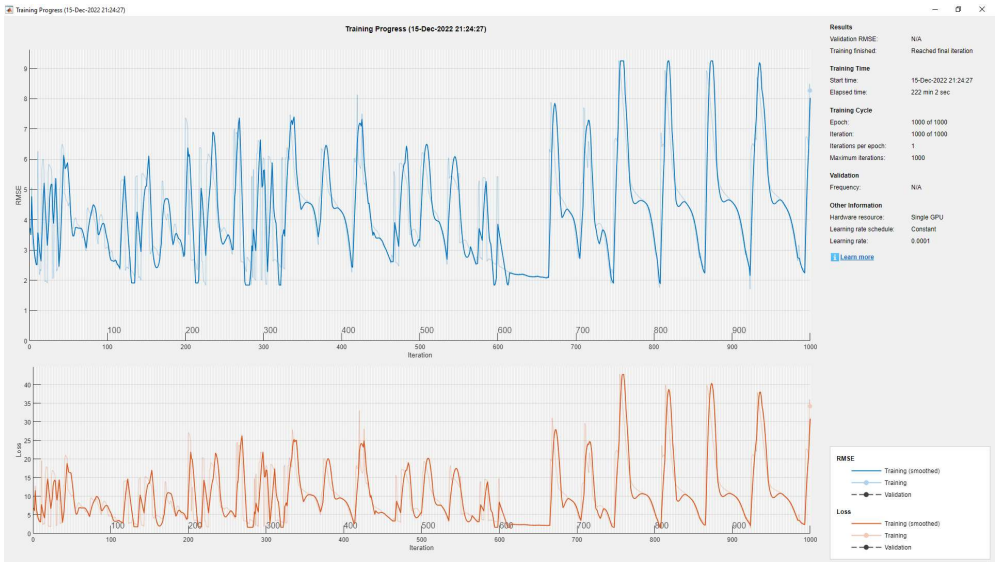

Figure 3.8: Training Progress Graph 2 for Network with 400 Hidden Values 1000 Epoch, the second training step.

In Figure 3.8 a repeat of Figure 3.7 is occuring, where the priviously learned values have changed due to the new dataset, and the previously learned values are clashing with the new amplitude and frequency, causing large erros. This behaviour is most noticed towards the end, where the new learnt values are still weighted wrong and thus cause peaks in both the loss function and the RMSE function. These error peaks go as high as 44 on the loss function, and above 9 on the RMSE function, meaning not only is the LSTM wrong more often, but the margin by which it is wrong has also increased. This training step was completed in 222 minutes and 2 seconds.

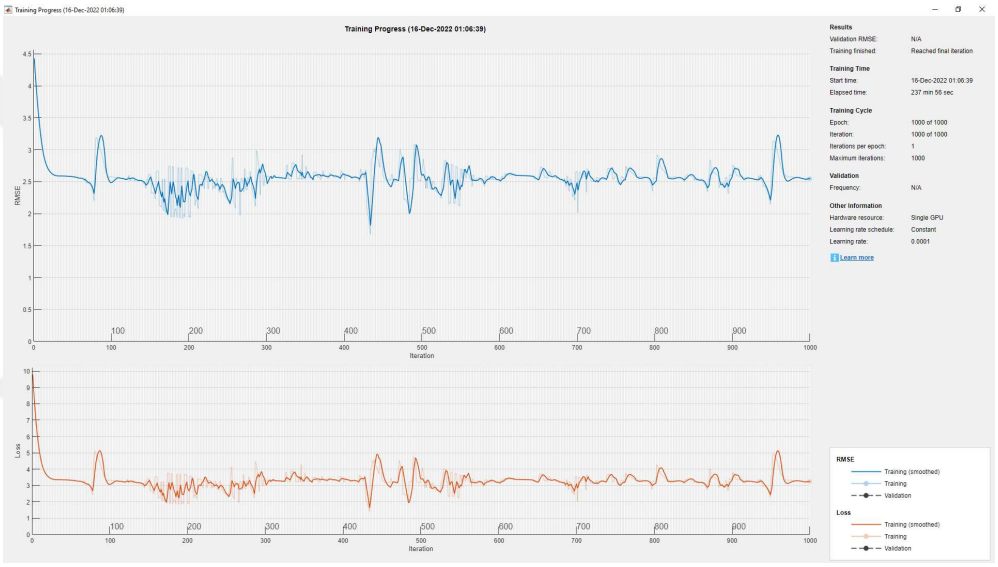

Figure 3.9: Training Progress Graph 3 for Network with 400 Hidden Values 1000 Epoch, the third training step.

In Figure 3.9 the amplitude of inputs has increased yet again, however the frequency has decreased beyond the initial value even beyond the original in Figure 3.7. This decrease in frequency allows the LSTM to adapt to the new values much quicker, and thus for the first time the highest peak is at the start, and after the first hundred epochs have passed, no peak surpasses 5.5 in the loss function or 3.5 in the RMSE function. This training step was completed in 237 minutes and 56 seconds.

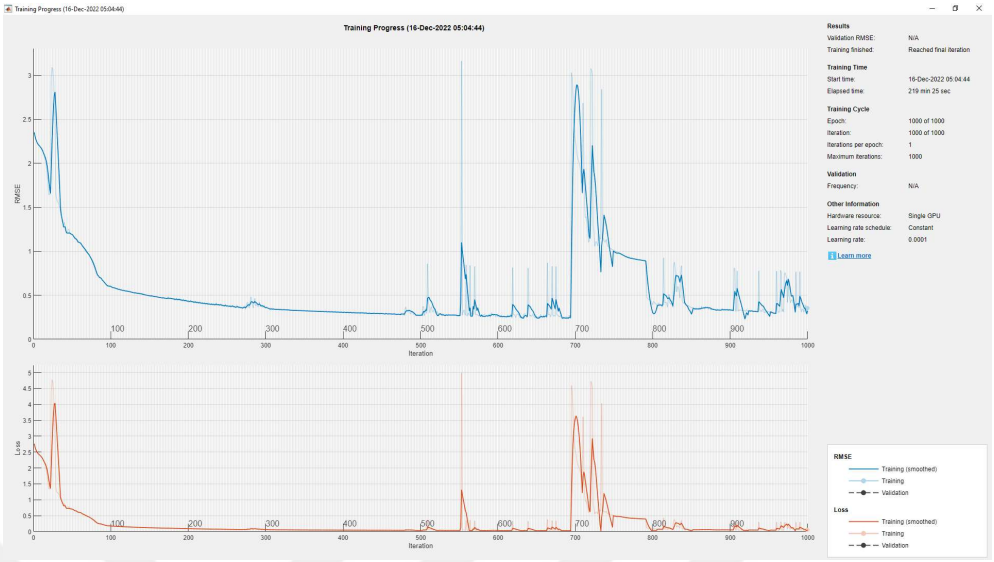

Figure 3.10: Training Progress Graph 4 for Network with 400 Hidden Values 1000 Epoch, the last training step.

In Figure 3.10 the frequency of the input is raised again, to its highest value, while the amplitude is decreased to its lowest value, this combination allows for the LSTM to have less peaks in errors, and the peaks it does have to be in the minimums of Figure 3.7, Figure 3.8, Figure 3.9. This shows that not only has the LSTM learned to adapt to different frequencies and amplitude, but that a high frequency which was initially an issue in Figure 3.8 is no longer a weakness of the LSTM. This training step was completed in 219 minutes and 25 seconds.

As can be seen from the change between Figure 3.7 and Figure 3.10 the many stages of training led to massive improvements in the final loss, and RMSE values. While the first three stages look erratic, since each stage is a mere step, the final training allows for the congregation of all training to achieve significantly more accurate, and much more reliable results. The total training time for this LSTM was 15 hours, 15 minutes and 13 seconds.

CHAPTER IV
RESULTS AND DISCUSSION

4.1 INTRODUCTION

During the scope of this study, a total of 20 LSTM models at different hidden values and epoch counts were modeled and trained. These trained LSTMs were then placed alongside the Kalman Filter from figure Figure 3.5 to test whether the resultant LSTMs were trained sufficiently, or if any patterns on training mistakes surfaced.

Kalman Filter had an expectant result throughout its use, despite not being an extended Kalman Filter, as that would be even more computationally intensive. A common theme in the Kalman Filter was a slight delay and slight errors, but due to the tolerances calculated via the frequency response graphs in Figure 3.1 to Figure 3.4 the delay and errors were not enough to compromise the system's stability.

LSTMs suffered throughout the course of this study, mostly due to an inability to adapt to the zero conditions of the system. This inability to start at zero coupled with a rushed frequency caused the LSTM to cause massive issues to the system stability. Despite this, LSTM did show extremely similar graphs with each other and almost all different LSTM models fell into one of 3 categories.

4.2 KALMAN FILTER

The Kalman Filter worked almost as well as the original data from the simulation in Figure 3.5. While during the simulation a delay of 0.15seconds and a gain error of 0.3 was recorded from the Kalman Filter, these values were well within the phase margin of 125 degrees, time margin of 2.78seconds and gain margin of 9.69db's. These errors can be noticed in the figures below:

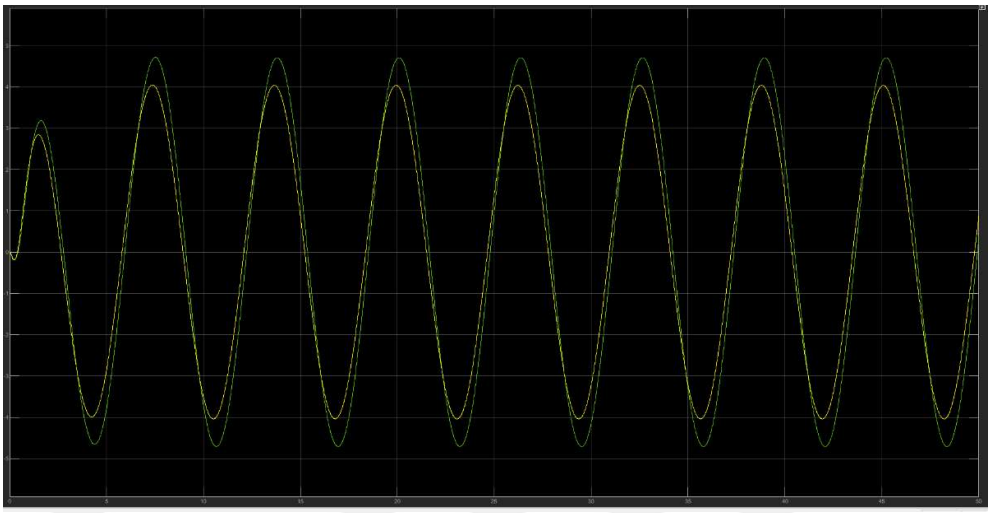

Figure 4.1: Difference between Simulated Data (Green), and Kalman Approximation (Yellow) of $\dot{\boldsymbol{\theta}}_{\mathbf{1}}$

As can be seen in Figure 4.1 while Kalman Filters estimates of $\dot{\theta}_1$ are close, they are subject to errors in both the phase, and the amplitude.

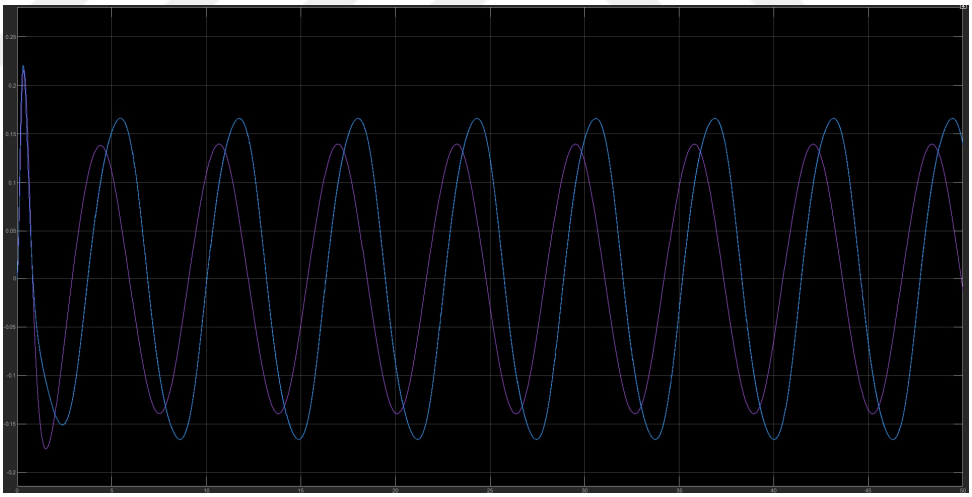

Figure 4.2: Difference between Simulated Data (Purple), and Kalman Approximation (Blue) of $\dot{\boldsymbol{\theta}}_{\mathbf{2}}$

Unlike the findings in Figure 4.1, it can be seen that $\dot{\theta}_2$ calculations done for Figure 4.2 cause Kalman values to drift in the opposite way as they do with calculations for $\dot{\theta}_1$.

These values, and their drift could further be improved upon by changing the Kalman Filter into an Extended Kalman Filter, however for the scope of this study, the Kalman Filter working was adequate.

4.3 LSTM

To simulate the LSTM successfully, the MATLAB Simulink file show in Figure 3.5 had to be modified, to allow for the addition of the "Predict" block, allowing the LSTM to be implanted into the simulation model. The resultant simulation model can be seen below:

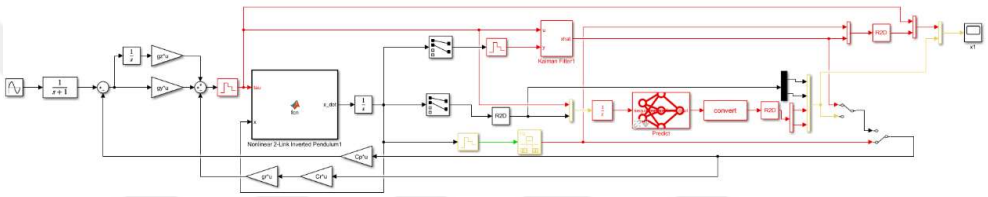

Figure 4.3: Simulink Diagram of the Simulation Process with added LSTM block.

As can be seen in Figure 4.23 the addition of the LSTMs Predict block increased the complexity of the simulation. Due to training being done with radian to degrees converted data, the data received from the simulation has to be converted prior to insertion into the LSTM block, this combined with the added delay to discretize the data, as well as make it arrive in manigeable chunks however increase the delays, and errors in the predictions made by the LSTM. The results are as follows:

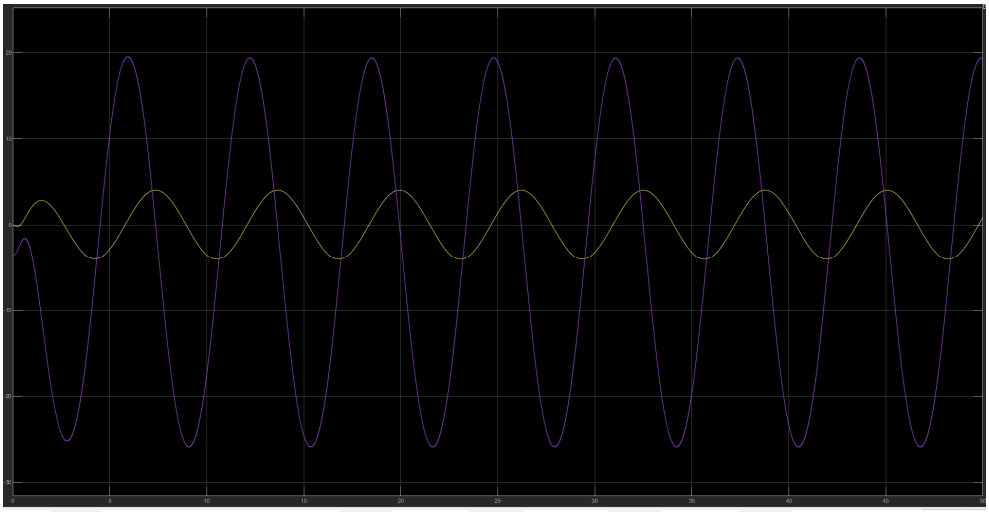

Figure 4.4: Difference between Simulated Data (Yellow), and LSTM Approximation (Purple) of $\ddot{\theta}_1$

As can be seen in Figure 4.2 while the frequency and time delays are within the margins barely, the amplitude is many times over the margins. This error only gets exasperated when the system's control loop becomes dependent on the information from the LSTM, and it causes the nonlinear system to crash.

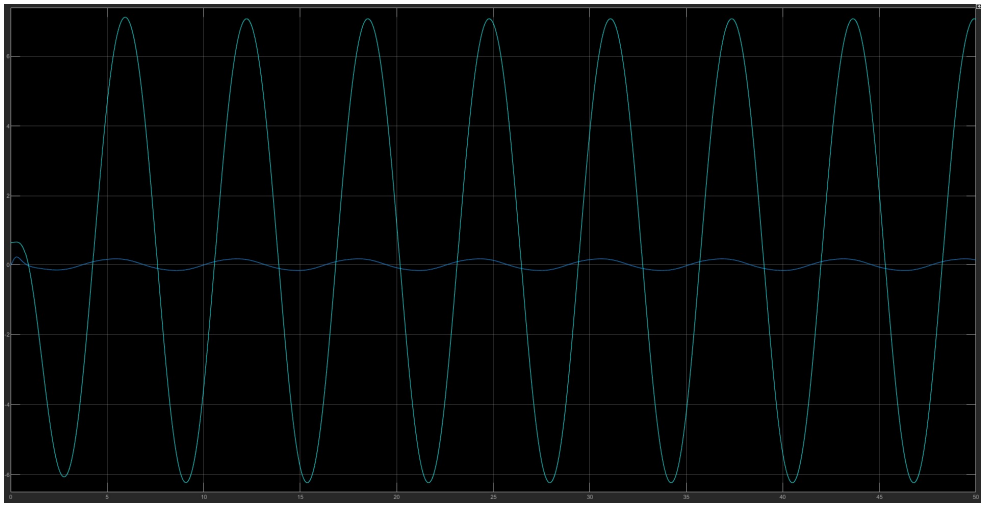

Figure 4.5: Difference between Simulated Data (Blue), and LSTM Approximation (Teal) of $\ddot{\theta}_2$

The issue prevalent in Figure 4.24 is even more prevalent in Figure 4.25, while the frequency and time response are accurate, due to the amplitude being multiple times what the real value is, it is not possible to call the data acquired a success.

While the data in Figure 4.24 and Figure 4.25 were only from a single LSTM trained at 400 hidden values and 1000 max epochs, it is a good representation of the general results from all twenty different LSTMs trained during the duration of this study.

CHAPTER V
CONCLUSION

5.1 INTRODUCTION AND SUMMARY OF RESULTS

While the initial goal of replacing a Kalman Filter with a LSTM was not achieved, during this study, many different parameters of LSTMs were explored, and their effects to results obtained has been documented. The multitudes of trained LSTMs have also been compared with each other, allowing for direct, visual comparisons to be made to different parameters, and training processes. MATLAB Simulink has been used extensively, beyond the initial assumed scope of the program especially as it relates to the simulation and simultaneous usage of multiple LSTM models predicting the same system using different parameters.

In this study, a comparison was made between the performance of Kalman filters and LSTMs in the task of state estimation. The results show that the Kalman filter performed well, with a slight delay and slight errors that were still within the tolerances calculated via the frequency response graphs. The Kalman filter's capability to handle direct systems made it a suitable choice for the task at hand. Still, it should be noted that the use of an Extended Kalman filter may have further enhanced the results. On the other hand, the LSTMs struggled with acclimating to the zero conditions of the system. The training of LSTM models necessitated a considerable amount of processing power and time, and the results weren't entirely satisfactory. The LSTMs caused delays, errors, and stability issues when compared to the Kalman filter. The study also found that LSTMs showed extremely comparable graphs with each other and practically all different LSTM models fell into one of 3 families.

As this study ends, two things have become clear, first, while neural networks hold incredible possibilities, it is still up to the users to be able to bring out all the possibilities to bear fruit and flourish. Second, a neural network requires multiple tries, and each try can be many hours of training, having dedicated hardware, or using cloud

solutions when dealing with neural network dedicated projects should always be an option that is considered and explored.

5.2 COMPARISON OF KALMAN FILTER AND LSTM

In terms of accuracy, the Kalman filter performed well, with only a slight delay and slight errors in comparison to the simulated data. The errors were still within the tolerances calculated via the frequency response graphs, which indicates that the Kalman filter was able to provide a reliable estimate of the true state of the system. However, the use of an Extended Kalman filter may have further improved the results.

In contrast, the LSTMs struggled with accuracy, showing significant errors in both the phase and amplitude of the estimates. The LSTMs were unable to adapt to the zero conditions of the system, causing delays and errors in their predictions. These errors were further exacerbated when the system's control loop became dependent on the information from the LSTM, leading to issues with the stability of the system.

The findings of the study showed that the oscillation frequency of the natural system, when measured with a sinusoidal input of 1 rad/ sec, was 6.284 seconds for both $\ddot{\theta}_1$ and $\ddot{\theta}_2$. This frequency is identical across both the Kalman filter and the LSTM models as well. While the frequency for both models were identical, the phase shift was different. The Kalman filter trails ahead of the natural system in the frequency of $\ddot{\theta}_1$ by 0.297seconds. This difference is minimal and therefore the Kalman filter can keep the system within the requested parameters. The LSTM results trail ahead of the natural system in the frequency of $\ddot{\theta}_1$ by 1.570seconds, or exactly 1/4 the frequency of oscillation. This difference is a major reason that the system becomes unstable under the control of the LSTM system. The Kalman filter trails behind of the natural system in the frequency of $\ddot{\theta}_2$ by 0.967seconds. This difference is comparably large, but the Kalman filter can keep the system within the requested control parameters and therefore this delay is still within excepted margins. The LSTM results trail behind of the natural system in the frequency of $\ddot{\theta}_2$ by 1.570seconds, or exactly 1/4 the frequency of oscillation again. While this delay is large, it is more uniform with the system in comparison with the Kalman filter. However, the LSTM system can't keep the system controlled, and therefore this 'uniform' delay is not within acceptable parameters.

On top of frequency, magnitude was another point of issue. For $\ddot{\theta}_1$ the systems natural peak sits at 4.707 radians per second squared. The Kalman filter calculated the same peak at 4.046 radians per second squared. The 0.661 radians per second squared difference was within the acceptable margins of error due to the nature of the control system. The LSTM calculated the same peak to be 19.520 radians per second squared. This difference of 14.813 radians per second squared, or in other words, a over 4 times multiple of the original value, causes the results from the LSTM to cause the system to fall into chaos. This issue with magnitude is amplified for $\ddot{\theta}_2$. The systems natural peak of 1.394 radians per second squared was calculated by Kalman filter to be 1.661 radians per second squared. The difference of 0.267 radians per second squared is within operating margins and therefore the system works without issues. The LSTM calculated the same peak as 7.054 radians per second squared. This difference of 5.660 radians per second squared is the largest error calculated during the entire test, and when paired with the error in the calculation for $\ddot{\theta}_1$ explains why the LSTM can't keep the system functioning.

In terms of stability, the Kalman filter performed well, with only a slight delay and slight errors in comparison to the simulated data. The Kalman filter was able to provide a stable estimate of the true state of the system, even in the presence of noise and disturbances. On the other hand, the LSTMs caused significant issues with the stability of the system, leading to crashes of the nonlinear system.

In terms of computational requirements, the Kalman filter was computationally efficient, requiring minimal resources to perform the state estimation task. However, it should be noted that the use of an Extended Kalman filter would have been more computationally intensive. In contrast, the LSTMs required a considerable amount of processing power and time to train, and the results were not entirely satisfactory. While the embedded version of the LSTM will require significantly less processing power, during this study the LSTM was only tested within MATLAB. Therefore, the LSTMs also required additional resources to perform the state estimation task, such as the "Predict" block in MATLAB Simulink.

5.3 LIMITATIONS OF THE STUDY

The study performed on the use of Kalman filter and LSTMs for state estimation was conducted using consumer- grade computing devices. These devices may not have had the computational power and time to completely appraise the performance of LSTMs in this task. As a result, the study's findings may not be thoroughly representative of the true capabilities of LSTMs for state estimation. It's possible that with more important computing devices, the results from the LSTMs could have been enhanced and potentially have been suitable to match or surpass the performance of the Kalman filter. However, this wasn't explored in the current study.

The study performed on the use of Kalman filter and LSTMs for state estimation assumed that the system being estimated was linearizable. This assumption may not always hold true in real- world applications and could have influenced the accuracy of the results attained. As a result, the conclusions drawn from the study may not be representative of the true performance of Kalman filter and LSTMs for state estimation in non-linear systems. The effectiveness of these techniques in non-linear systems should be evaluated in unborn studies to supply a further comprehensive understanding of their capabilities.

The study that was conducted on the performance of Kalman filter and LSTMs for state estimation was limited in scope as it only concentrated on a particular type of non-linear system. This narrow focus may have impacted the validity of the results and the conclusions drawn from the study. The findings may not be applicable to other types of non-linear systems, and consequently, its generalizability to other real- world scenarios are limited. It's imperative to undertake fresh research that encompasses a broader spectrum of non-linear systems to gain a comprehensive understanding of the efficacity of Kalman filters and LSTMs in the task of state estimation.

5.4 FUTURE STUDY

The findings of this study have important implications for the use of LSTMs in state estimation tasks. While LSTMs showed promise as a potential alternative to Kalman filters, the results of this study indicate that more powerful computing resources and farther optimization of the training process may be necessary to completely assess the capabilities of LSTMs in this environment. Given time,

multitudes more LSTM models can be produced, trained and tested. While this study was unable to find the right model parameters and training parameters to generate an LSTM qualified of superseding a Kalman Filter, since this study began there has been successful homogenizing between Kalman Filter and LSTMs to produce results faster and with better precision than any system by itself. Further more, the supposition of a linearized system in this study limits the generalizability of the findings, and future research could explore the use of LSTMs in nonlinear systems that haven't been linearized, or in ranges of linearized systems where the motion approaches more non-linear actions. Other types of recurrent neural networks could also be researched as potential alternatives to Kalman filters. Overall, the limitations of this study punctuate the need for farther exploration to completely understand the capabilities and limitations of LSTMs in state estimation tasks.

5.5 CONCLUSION

The study found that the Kalman filter outperformed LSTMs in terms of accuracy, stability, and control of the system in state estimation tasks. The Kalman filter showed minimum errors and was within the calculated tolerances, while LSTMs struggled with accuracy, stability, and adapting to zero conditions of the system. The study also highlighted the differences in the phase shift and magnitude errors between the Kalman filter and LSTMs, with the Kalman filter having errors within respectable margins and LSTMs having significant errors. The study concluded that LSTMs necessitate a considerable amount of processing power and time for training, but the significance of this study lies in the finding that multiple training sessions can increase accuracy. This information is useful for those considering using LSTMs for state estimation tasks, as it sheds light on the significance of multiple training sessions for increased accuracy.

REFERENCES

[1] Ata Ayesha, Abbas Sagheer, Fatima Areej, Khan Muhammad Adnan, Ahmad Gulzar (2019), "Modelling Smart Road Traffic Congestion Control System Using Machine Learning Techniques", *Neural Network World 2*, Vol. 2019, No. 2, pp.99-110.

[2] Fernandez Benito R., Parlos Alexander G., Tsai Wie Kang (1990), "Nonlinear dynamic system identification using artificial neural networks", *IJCNN International Joint Conference on Neural Networks*, Vol. 2, pp. 133-141, San Diego.

[3] Burkov Andriy (2019), *The Hundred-Page Machine Learning Book*, Self-Published, Quebec City, Canada, pp. 26.

[4] Burkov Andriy (2020), *Machine Learning Engineering*, True Positive Inc., Quebec City, Canada, pp. 32.

[5] Song Fei, Li Yong, Cheng Wei, Dong Limeng, Li Minqi, Li Junfang (2022), "An Improved Kalman Filter Based on Long Short-Memory Recurrent Neural Network for Nonlinear Radar Target Tracking", *Wireless Communications and Mobile Computing,* Vol. 2022, No. 8280428, pp. 1-10.

[6] Gharghory Sawsan Morkos (2020), "Deep Network based on Long Short-Term Memory for Time Series Prediction of Microclimate Data inside the Greenhouse", *International Journal of Computational Intelligence and Applications*, Vol. 19, No. 02, pp. 2050013.

[7] Intellipaat (2021), *What is LSTM?,* Bangalore, https://intellipaat.com/blog/what-is-lstm/, DoA. 26.05.2021.

[8] Hochreiter Sepp and Schmidhuber Jürgen (1997), "Long Short-term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780.

[9] Javatpoint (2020), *Continuous Systems vs. Discrete System*, Noida, https://www.javatpoint.com/continuous-systems-vs-discrete-systems, DoA. 26.05.2021.

[10] Yang Shiyu, Wan Man Pun, Chen Wanyu, Ng Bing Feng, Dubey Swapnil (2020), "Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization", *Applied Energy,* Vol. 271, pp. 115147.

[11] Su Cao, Wang Xiangke, Shen Lincheng, Yu Huangchao, (2020), "Adaptive UAV maneuvering control system based on dynamic inversion and long-short-term memory network", *Chinese Automation Congress (CAC),* Shanghai, China, pp. 6880-6885.

[12] Tutorials Point (2019), *Control Systems – Introduction*, Hyderabad, https://www.tutorialspoint.com/control_systems/control_systems_introduction.htm, DoA. 26.05.2021.

[13] Wright Gavin (2022), "Closed Loop Control System", https://www.techtarget.com/whatis/definition/closed-loop-control-system, DoA. 26.05.2021.

[14] Fu Yichen, Eldon David, Erickson Keith, Kleijwegt Kornee, Lupin-Jimenez Leonard, Boyer Mark D, Eidietis Nick, Barbour Nathaniel, Izacard Olivier, Kolemen Egemen (2020), "Machine learning control for disruption and tearing mode avoidance", *Special Collection: Invited Papers from the 2nd International Conference on Data-Driven Plasma Science*, Vol. 27, No. 2, pp. 22501, Marseille, France, https://aip.scitation.org/doi/10.1063/1.5125581 , DOI: 10.1063/1.5125581, DoA 26.05.2021.

[15] Tian Yong, Lai Rucong, Li Xiaoyu, Xiang Lijuan, Tian Jindong (2020), "A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter", *Applied Energy,* Vol. 265, pp. 114789.

[16] Zhou Zhiyu, Zhang Ruoxi, Zhu Zefei (2019), "Robust Kalman Filtering With Long Short-Term Memory For Image-Based Visual Servo Control", *Multimedia Tools and Applications,* Vol. 78, pp. 26341-26371.

APPENDICES

Appendix 1: LSTM Comparison Graphs

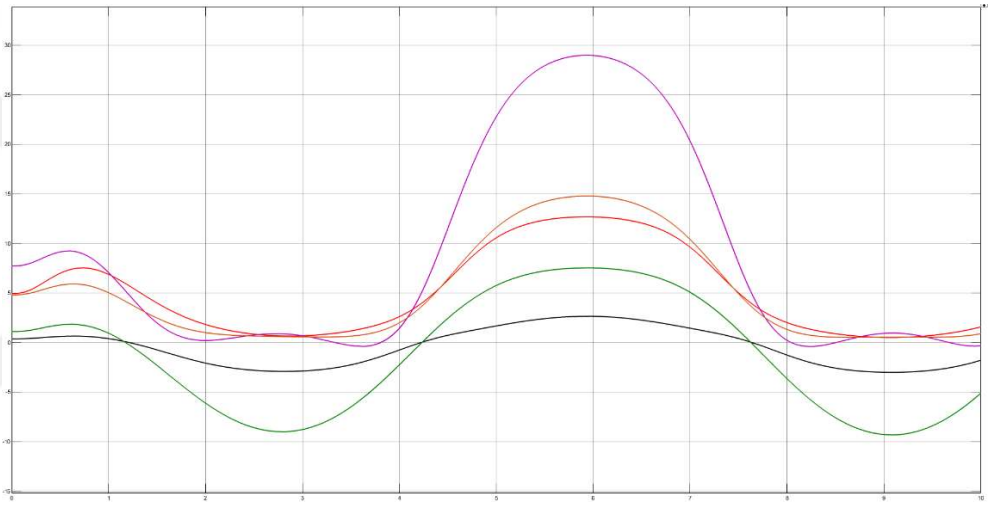

Figure R.1: 5 Lstms θ_1 response from simulataenously responding to the same input

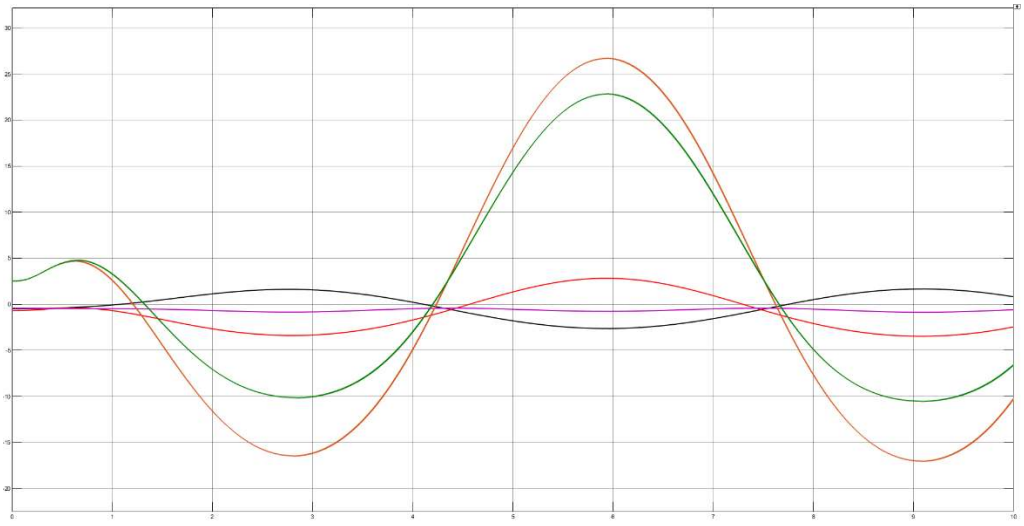

Figure R.2: θ_1 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R1 is present)

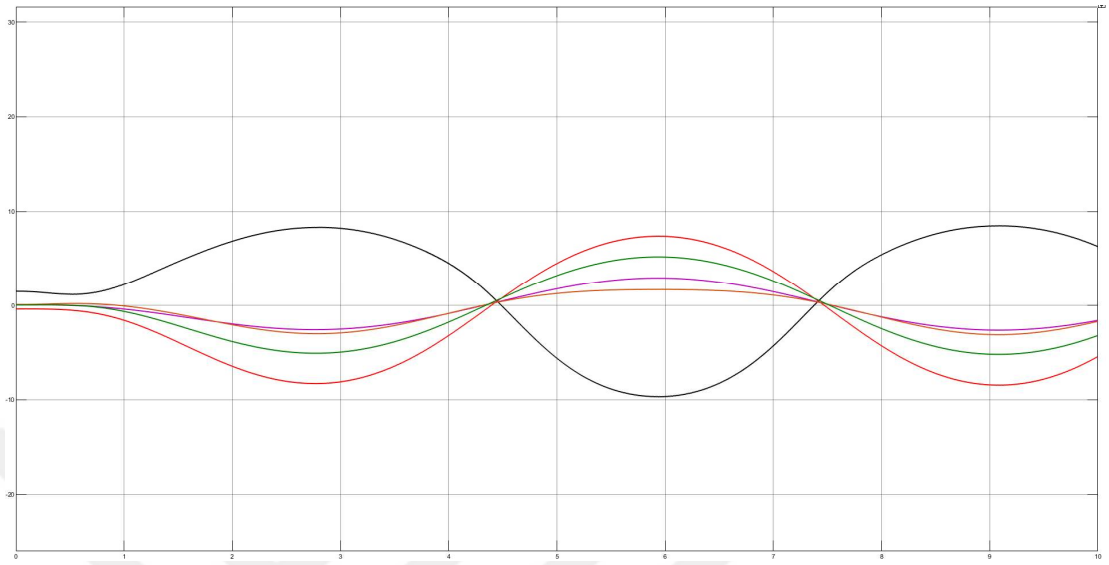

Figure R.3: θ_1 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R1 or R2 is present)

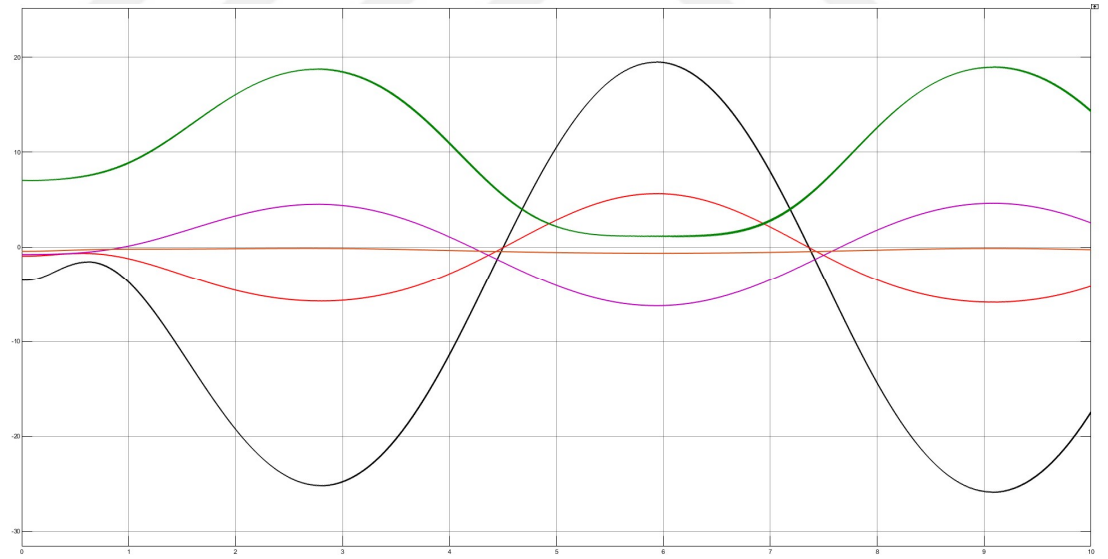

Figure R.4: θ_1 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R1, R2 or R3 is present)

Figure R.5: 5 Lstms θ_2 response from simulataenously responding to the same input

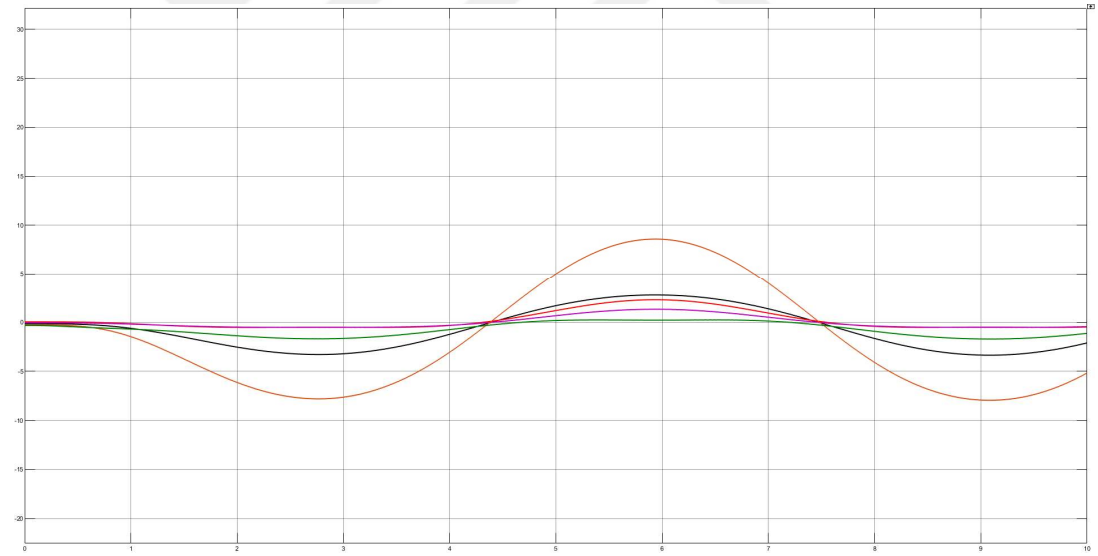

Figure R.6: θ_2 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R5 is present)

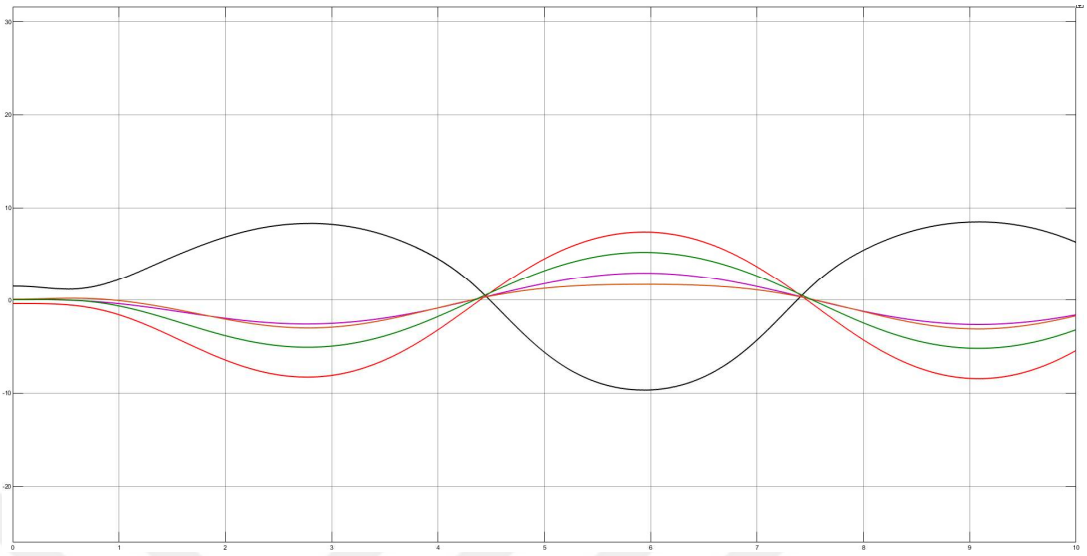

Figure R.7: θ_2 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R5 or R6 is present)

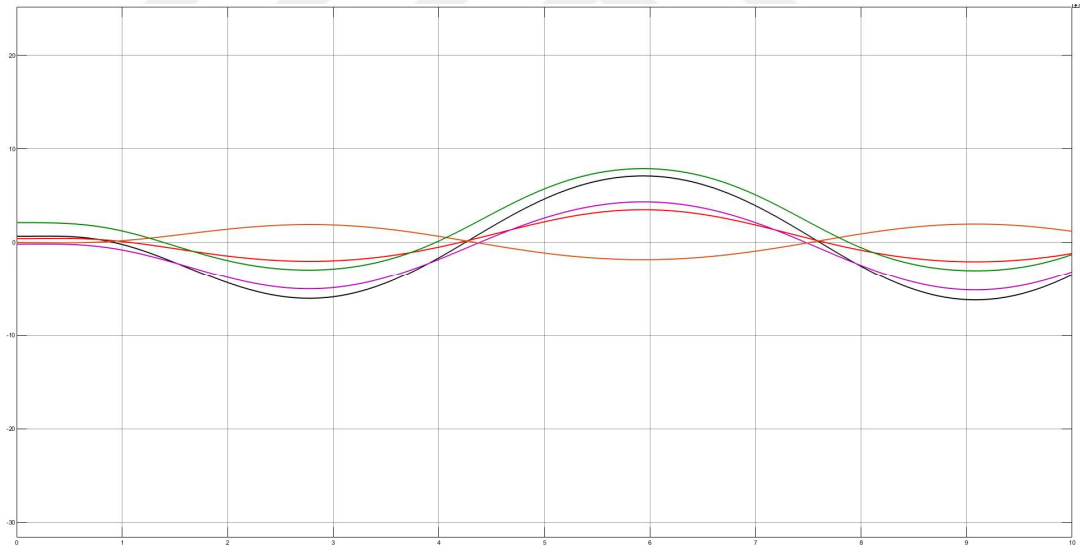

Figure R.8: θ_2 response from 5 more Lstms simulataenously responding to the same input as R1 (no LSTM from R5, R6 or R7 is present)