CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Neural State Space Model-based Predictive Control for Varying Initial Conditions in Inverted Pendulum System

by

Aqsa Hussain Raja

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

in the

Faculty of Engineering
Department of Electrical Enginering

2024

I would like to dedicate this dissertation to my loving parents.

CERTIFICATE OF APPROVAL

Neural State Space Model-based Predictive Control for Varying Initial Conditions in Inverted Pendulum System

by

Aqsa Hussain Raja

(MEE213002)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Iftikhar Ahmad Rana	NUST, Islamabad
(b)	Internal Examiner	Dr. Muhammad Ashraf	CUST, Islamabad
(c)	Supervisor	Dr. Imtiaz Ahmad Taj	CUST, Islamabad

Dr. Imtiaz Ahmad Taj
Thesis Supervisor
May, 2024

Dr. Noor Muhammad Khan
Head
Dept. of Electrical Engineering
May , 2024

Dr. Imtiaz Ahmad Taj
Dean
Faculty of Engineering
May , 2024

Author's Declaration

I, **Aqsa Hussain Raja** hereby state that my MS thesis titled "**Neural State Space Model-based Predictive Control for Varying Initial Conditions in Inverted Pendulum System**" is my work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Aqsa Hussain Raja)

Registration No: MEE213002

Plagiarism Undertaking

I solemnly declare that the research work presented in this thesis titled "**Neural State Space Model-based Predictive Control for Varying Initial Conditions in Inverted Pendulum System**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above-titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred to/cited.

I undertake that if I am found guilty of any formal plagiarism in the above-titled thesis even after the award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Aqsa Hussain Raja)

Registration No: MEE213002

Acknowledgement

At the forefront, immense gratitude to Allah (SWT) for endowing strength and guidance, His divine blessings igniting my journey's inspiration and unwavering determination.

Working alongside my esteemed former supervisor, Dr. Aamer Iqbal Bhatti has been both an honor and a privilege. I extend my heartfelt gratitude to him, specifically for his invaluable guidance and unwavering support. I am also thankful to my supervisor, Dr. Imtiaz Ahmad Taj for his remarkable guidance throughout the process. I extend my sincere gratitude to Sir Mohsin Ullah, Ms. Memoona Zulfiqar and Ms. Sidra Ghayour Bhatti for their invaluable insights and innovative ideas, which significantly enhanced the clarity of various concepts.

Last but not least, I would like to extend a heartfelt appreciation to my parents and my sisters Hifsa and Samavia, whose boundless love and unwavering support have been the foundation of my academic journey.

(Aqsa Hussain Raja)
Registration No: MEE213002

Abstract

In recent years, various studies have extensively explored the integration of neural networks, particularly to tackle challenges associated with varying initial conditions. These investigations focus on the learning capabilities of neural networks to enhance system stability and performance, especially in scenarios where the initial conditions may vary dynamically. Addressing stability challenges in nonlinear systems, this study explores a solution that can dynamically adapt to varying initial conditions, enhancing stability and robustness.

Introducing a novel Model Predictive Control (MPC) system based on the Neural State Space Model (NSSM), this study systematically identifies system dynamics and trains the MPC using a data-driven approach. The methodology involves utilizing Neural Networks to understand the functioning of the Inverted Pendulum system, leading to the creation of an MPC policy based on the acquired knowledge of system dynamics. The proposed control scheme is designed to impart robustness and adaptability to nonlinear systems. The novelty of this approach is validated by implementing it in a system with uncertain dynamics, utilizing data obtained solely during the training of the neural state-space model.

Simulation results currently demonstrate the remarkable performance of the proposed approach. The effectiveness and adaptability of the proposed approach are compared with the Feedback Linearization technique and a Neural Networks-based control trained using a single initial condition. Assessing generalizability and performance across diverse scenarios, this control approach proved flexible and efficient in controlling an inverted pendulum system, supported by evaluating the performance metrics and convergence time.

Contents

List of Figures

List of Tables

Abbreviations

ANN	Artificial Neural Network
BIBO	Bounded Input and Bounded Output
BP	Back Propagation
CNN	Convolutional Neural Network
FFNN	Feedforward Neural Network
FBL	Feedback Linearization
GRU	Gated Recurrent Units
LQR	Linear Quadratic Regulator
LSTM	Long Short-Term Memory Networks
MPC	Model Predictive Control
MRC	Model Reference Control
NN	Neural Network
NSSM	Neural State Space Model
RC	Robust Control
RBF	Radial Basis Function
RPNN	Recurrent Paraconsistent Neural Network
RNN	Recurrent Neural Networks
SMC	Sliding Mode Control

Chapter 1

Introduction

1.1 Background

A nonlinear system encompasses a collection of nonlinear equations employed to characterize a physical device or process that cannot be precisely delineated by a set of linear equations in any form [1]. The techniques of analysis and design of linear systems have been well developed, and they are used not only for perfectly linear systems but also for almost linear systems. However, when the system is nonlinear the general identification procedure is much more complicated, and detailed models are often impossible to obtain [2].

States may undergo sudden shifts not just when they cease to exist, but also when they become unstable when a parameter changes gradually. As a result of instability, a tiny change may have a great influence in the sense that a minor disturbance at a particular instant may grow and become important such that after a long period, the behavior of the system depends largely on the nature of the disturbance, even though it was small [3].The response of nonlinear systems is dependent on the input-signal size and initial conditions as well as the system parameters. A stable response for one input signal may be unstable for another [4]. The simplicity of the control algorithm as well as guaranteeing the stability and robustness in the closed-loop system is a challenging task in real situations. The theory of nonlinear dynamical systems, or non-linear control systems if control inputs are involved, has been greatly advanced since the nineteenth century.

In the literature, system identification has been widely explored. Artificial neural networks have been used by many authors to identify systems because of their ability to learn and their good performance for the approximation of nonlinear functions [4]. It is demonstrated that neural networks can be used effectively for the identification and control of nonlinear dynamical systems [5].

They come from the need to explain and comprehend real-world occurrences that display nonlinear behaviors and linkages. In a nonlinear system, the starting conditions are crucial since they have a significant impact on the behavior of the system later on [3]. Nonlinear systems, in contrast to linear systems, show sensitivity to the initial state, resulting in a variety of frequently unanticipated trajectories. To effectively forecast and manage the dynamic evolution of nonlinear systems, which affects stability, convergence, and overall system performance, it is imperative to comprehend and control the impact of initial conditions [6].

Neural networks have significance for modeling and controlling nonlinear systems because they can learn complex relationships and adapt to changing behaviors [4]. Moreover, Neural networks are used in control applications to create adaptive controllers that can adapt to a system's nonlinear dynamics and provide increased performance and robustness [7, 8]. Furthermore, neural networks are effective tools for modeling because they can capture complicated nonlinearities, making it possible to create adaptable and accurate representations of complex system dynamics that may challenge conventional analytical methods [8]. Multi-layered neural networks offer an exciting alternative for modeling complex non-linear systems [9].

The motivation for using neural networks for modeling and controlling nonlinear systems comes from the difficulties that come with the complex dynamics and unpredictable nature of real-world applications [9–13]. When it comes to managing non-linearities and adjusting to dynamic changes, traditional control techniques often underperform Neural networks present an appealing option because of their exceptional capacity to learn from data and recognize intricate patterns [12, 14, 15]. They offer adaptive techniques for control that can dynamically adapt to a nonlinear system's changing environment, improving overall performance and stability [16].

1.2 Non-Linear Systems: An Overview

Linear control has a well-established background, featuring numerous effective techniques and a lengthy track record of successful implementations in industrial settings [17]. Thus, it is natural to wonder why so many researchers and designers from such diverse fields as aircraft [18] spacecraft control [19], robotics [20], and biomedical engineering [21], [22], have recently expressed a strong interest in the development and application of non-linear control methodologies [18–21]. There are several reasons to be interested in Figure 1.1 [23].

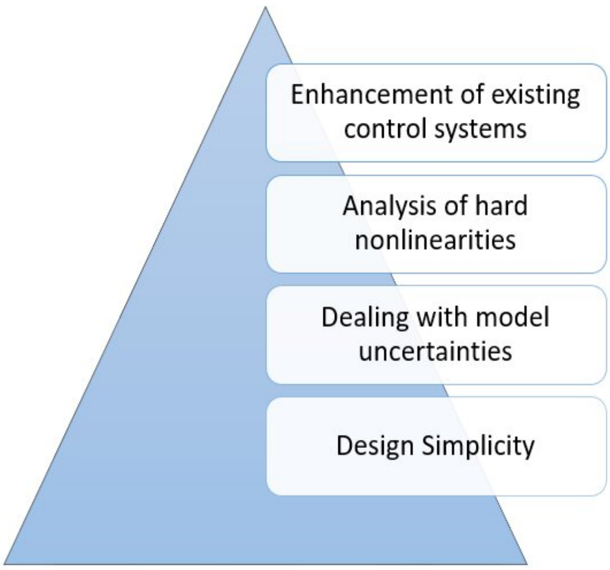

FIGURE 1.1: Rising Interests in Nonlinear Control

1.2.1 Behavior of Non-Linear System

Nonlinear systems, by definition, are employed to characterize physical systems, indicating that most systems possess some degree of nonlinearity. Nonstationary control systems are typically represented by nonlinear differential equations. Conversely, if a control system operates within a limited range and exhibits smooth nonlinearity, it can be effectively approximated as a linearized system. In such cases, the system's dynamics can be described using a set of linear differential equations [1], [24], [25].

The behavior of nonlinear systems, however, is much more complex. Due to the lack of linearity and the associated superposition property, nonlinear systems respond to external inputs quite differently from linear systems [2].

1.2.2 Sources of Non-Linearity

Sources of non-linearity in dynamic systems can stem from various physical and operational characteristics. Here are some common sources: Sources are given below:

1.2.2.1 Intentional Nonlinearities

Designers deliberately incorporate nonlinearities into control systems, utilizing advanced tools such as bang-bang optimal control laws and adaptive control laws [26, 27].

1.2.2.2 Saturation

Within a limited range of input signals, the system's output exhibits proportionality to the input. However, beyond this range, the output tends to stabilize and remain nearly constant. Various physical variables, such as a vehicle's velocity or electrical impulses, often have upper limits, and once these limits are reached, linearity diminishes [28, 29]. Figure 1.2 illustrates the saturation non-linearity phenomenon.This behavior is known as saturation non-linearity, a common characteristic in many physical systems. When the input signal remains within a specific range, the output maintains a direct, linear relationship with the input, facilitating predictable and controllable responses. However, as the input signal exceeds this predefined range, the system's ability to respond proportionally diminishes. Instead, the output levels off and approaches a near-constant value, indicating that the system has reached its operational limit.

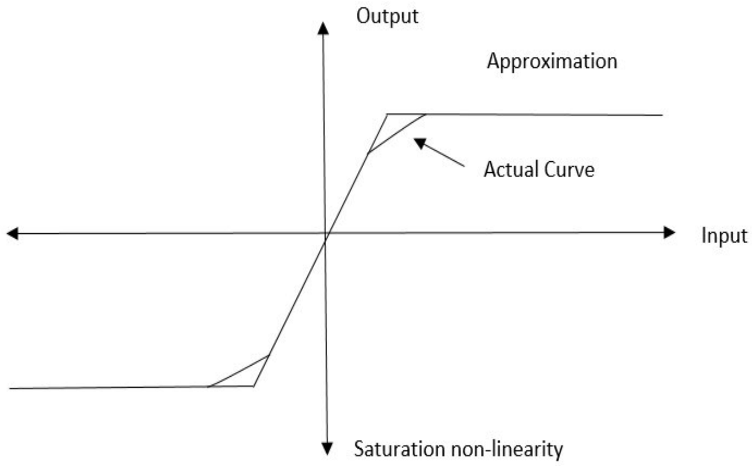

FIGURE 1.2: Saturation Non-Linearity

1.2.2.3 Backlash

Backlash refers to the difference between tooth spacing and tooth width in a mechanical system, crucial for the efficient transmission of gears during rapid operation. Consequently, minimal input signals may not produce any output. Backlash is a common occurrence in numerous mechanical and hydraulic systems [30]. Figure 1.3 depicts the backlash effect, showcasing the hysteresis or delayed response in a system due to mechanical clearance.

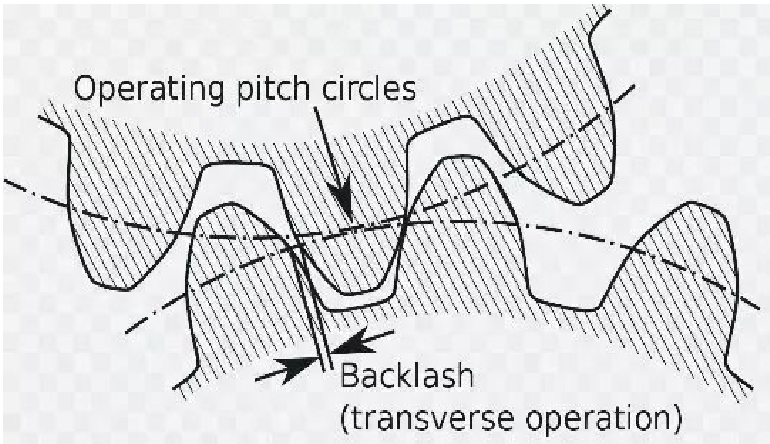

FIGURE 1.3: Backlash [31]

1.2.2.4 Dead Zone Nonlinearity

This form of nonlinearity is defined by the system's lack of response to the provided input until it surpasses a particular threshold. Conversely, once the input value exceeds a specified limit, the output abruptly diminishes to zero [27]. Figure 1.4 demonstrates the dead zone non-linearity, depicting a range of input values where there is no output response, leading to a nonlinear relationship between input and output signals.

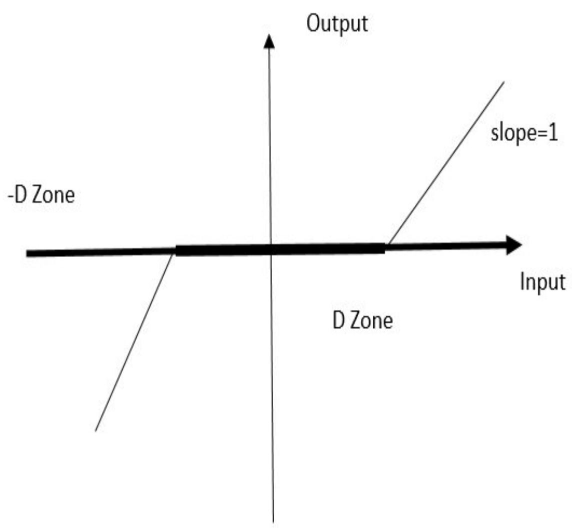

FIGURE 1.4: Dead zone Non-Linearity

1.2.2.5 Hysteresis

Hysteresis represents the relationship between the current state and the preceding state of a system. The behavior of ferromagnetic cores in electrical machines and transformers is often described using nonlinear magnetization curves and equations [32]. Figure 1.5 illustrates hysteresis, showcasing the phenomenon where a system's output depends not only on its current input but also on its history, often represented by a looped curve indicating lag or memory effects.

In addition to ferromagnetic materials, hysteresis can also be found in other systems, such as mechanical systems with friction, electronic circuits with components like Schmitt triggers, and thermal systems with phase changes. Recognizing and modeling hysteresis is essential for accurately predicting and controlling the behavior of such systems.

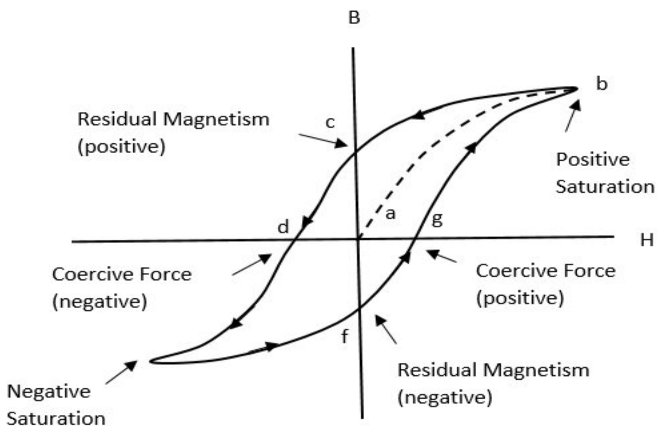

FIGURE 1.5: Hysteresis Non-Linearity

1.2.3 Challenges in Nonlinear System Control

Nonlinear systems present numerous challenges in contrast to linear systems, introducing complexities in their analysis, modeling, and control. Here are some of the primary challenges associated with nonlinear systems: [23]

1. **Complexity of Dynamics:**

   Nonlinear systems often exhibit complex and intricate dynamics, including bifurcations, chaos, and multiple stable states. Understanding and predicting the behavior of such systems can be challenging due to the absence of simple mathematical relationships.The complexity of dynamics in nonlinear systems demands advanced mathematical and computational approaches to analyze and predict system behavior. Techniques such as bifurcation theory, chaos theory, and numerical simulations are essential tools for understanding the rich and often surprising dynamics that nonlinear systems can exhibit. [33].

2. **Lack of Superposition:**

   Linear systems follow the concept of superposition, which states that the reaction to a sum of several inputs is equal to the total of the responses to each input. Nonlinear systems do not always follow this concept, making analysis and synthesis more difficult [1].

3. **Limited Generalization:**

   Nonlinear system behavior can be highly specific to the particular system under consideration. Generalizing results and methodologies across different nonlinear systems can be challenging due to the diversity of nonlinear behaviors [25].

1.2.4 Importance in Real-world Applications

Acquiring proficiency in the fundamental techniques of nonlinear control analysis and design is paramount for control engineers, significantly enhancing their effectiveness in addressing practical control problems. Mastery of these methods not only equips engineers with a diverse skill set but also fosters a deeper understanding of the real world, characterized by inherent nonlinearity. This heightened awareness enables control engineers to navigate and harness the intricacies of diverse systems, anticipating and mitigating challenges associated with nonlinear dynamics. Historically, computational difficulties limited the application of nonlinear control methods; however, recent advances in computer technology have alleviated these challenges, facilitating more extensive and practical implementation in real-world scenarios. The convergence of improved computational power and the demand for sophisticated control solutions underscores the vital role of nonlinear control in promoting innovation and ensuring the resilience of control systems in the face of complex, nonlinear dynamics [23].

1.3 Motivation

The motivation behind exploring data-driven control systems stems from the recognition that traditional models may fall short when dealing with nonlinear systems due to a consistent lack of guaranteed convergence and stability, posing significant challenges to the advancement of Neural Networks-based approaches. In this context, motivation can be seen as the drive to develop control solutions that can dynamically adapt to different starting points and navigate the inherent complexities of nonlinear systems. The reliance on a data-driven approach is crucial, as it allows the control system to learn from experimental data obtained directly from the plant. Consequently, there is a growing need for enhanced research efforts aimed at refining data-driven techniques for the optimal control of nonlinear and uncertain systems. Emphasis should be placed on addressing variable initial conditions, optimizing performance, ensuring resilience, achieving convergence, enhancing generalizability, and fulfilling data requirements. This challenge serves as a significant source of inspiration for our ongoing research endeavors.

1.4 Research Objectives

Considering the limitations and motivations mentioned in the previous part, the research establishes the following goals and objectives:

1. Develop a precise neural network model for the Inverted pendulum system, ensuring an accurate representation of its dynamic behavior.
2. Combine the Neural State Space Model (NSSM) with a Model Predictive Controller (MPC) to improve predictive capabilities and control performance.
3. Implement an adaptive control strategy based on NSSM to determine optimal control actions, ensuring effective system stabilization.
4. Enhance the developed adaptive control to be robust in handling varying initial conditions, providing stability across a range of system states.

1.5 Thesis Organization

In Chapter 2, a comprehensive literature survey is conducted, focusing on various control techniques proposed by different researchers. The survey explores a range of methodologies and strategies employed in the control of Non-Linear Systems.

This chapter aims to provide a thorough understanding of the evolution of control techniques, shedding light on the advancements and challenges faced by researchers in the field. By reviewing the existing literature, the chapter sets the stage for the subsequent chapters by identifying gaps and areas that require further exploration in the context of controlling non-linear systems.

Chapter 3 is dedicated to establishing a mathematical model for the Inverted Pendulum System. The mathematical model is crucial for comprehending the dynamics of the system. It includes equations and formulations that describe the physical behavior of the inverted pendulum. To enhance the understanding of the system's dynamics, simulations are conducted using MATLAB and Simulink. This chapter serves as the foundation for the practical aspects of the research, providing insights into how the system behaves under different conditions and laying the groundwork for subsequent analysis and implementation.

Chapter 4 outlines the detailed methodology employed in implementing the proposed solutions or models derived from the mathematical analysis in Chapter 3. This includes a step-by-step description of the procedures, algorithms, and tools used in the implementation process. The chapter provides transparency in the technical aspects of translating theoretical concepts into practical solutions. It discusses the specific techniques and technologies employed, ensuring replicability and understanding of the implementation process by other researchers or practitioners in the field.

Chapter 5 includes the results and findings derived from the research are presented. It includes the outcomes of experiments, the performance of the proposed models, and any insights gained during the implementation phase. The findings are thoroughly analyzed and discussed to draw meaningful conclusions.Overall, the findings provide a comprehensive understanding of the performance and effectiveness of the proposed nonlinear control strategies, demonstrating significant improvements in stability, robustness, control accuracy, and energy efficiency. These advancements contribute to the broader field of nonlinear control theory and offer valuable insights for future research and practical applications. The research successfully addressed key challenges associated with nonlinear control of the inverted pendulum and highlights the potential for integrating new technologies to further enhance nonlinear control systems.

The final chapter, Chapter 6, summarizes the research, drawing conclusions based on the major findings presented in Chapter 5. It also discusses the broader implications of the research within the field of non-linear systems and control. Additionally, this chapter provides insights into potential directions for future work, suggesting areas that could benefit from further exploration or improvement. It serves as a guide for researchers who may build upon the current study, addressing limitations and advancing the state of knowledge in the domain of non-linear system control.

1.6 Chapter Summary

The introduction delves into the complexities of nonlinear systems, emphasizing their sensitivity to initial conditions. It positions neural networks as pivotal for modeling and controlling these systems, driven by their adaptability and ability to capture intricate nonlinear dynamics. Motivated by the shortcomings of traditional control methods, the research aims to enhance data-driven approaches for optimal control in nonlinear and uncertain systems.

Chapter 2

Literature Review

2.1 Introduction

Stability theory holds great significance in system engineering, particularly within the realms of control systems and automation. This theory plays a crucial role in addressing both the dynamics and control aspects of these systems [34]. The stability of a dynamical system, whether subjected to control or disturbance inputs, is an essential prerequisite for its practical usefulness, particularly in the context of most real-world applications [35].

Stability signifies that the system's outputs and internal signals are confined within acceptable limits, a condition termed Bounded Input and Bounded Output (BIBO) stability [36]. On the other hand, the system outputs converge towards a specific equilibrium state of interest, referred to as asymptotic stability [37].

In the realm of nonlinear dynamics and control systems, there exist several conceptual types of stability, with three being particularly significant: stability concerning a system's equilibria, orbital stability involving a system's output trajectory, and the structural stability of the system. The foundational idea of stability originated from the examination of the equilibrium state of a mechanical system, dating back to 1644 when E. Torricelli investigated the equilibrium of a rigid body subjected to gravitational force [38].

G. Lagrange's classical stability theorem, articulated in 1788, stands as one of the most renowned outcomes regarding the stability of conservative mechanical systems. According to this theorem, if a conservative system, situated at an isolated equilibrium and potentially subjected to straightforward constraints, possesses a minimum potential energy, then the equilibrium position of the system is deemed stable [17]. The fundamental concepts of system and trajectory stabilities have undergone evolutionary progress over time, marked by numerous successful advancements and developments. These concepts are crucial in understanding and controlling the behavior of dynamic systems, particularly in the context of nonlinear systems where stability analysis is inherently more complex than in linear systems.

The concepts of system and trajectory stability have evolved significantly, driven by theoretical advancements and practical needs. From the foundational work of Lyapunov to modern computational and data-driven methods, stability theory continues to be a vital area of research in control theory. Each development has contributed to a deeper understanding and more robust methods for ensuring the stability of dynamic systems, particularly in the challenging realm of nonlinear control. [39].

Since the 1970s, the inverted pendulum has served as the predominant standard in nonlinear control theory [37].The objective of this study is to enhance this robotic benchmark and present a comprehensive overview of past and present advancements in nonlinear control theory, all centered around the intricate nonlinearity and uncomplicated structure of the model. This robotic system is employed to address and confirm an exploration of novel concepts and challenging issues within nonlinear control theory.The inverted pendulum, with its simple yet profoundly nonlinear dynamics, offers a rich platform for studying and testing various control strategies. Its behavior exemplifies many of the complexities associated with nonlinear systems, including instability, sensitivity to initial conditions, and the need for robust control methods. By focusing on this model, researchers can explore fundamental questions and develop innovative solutions that are broadly applicable to a wide range of nonlinear control problems.The inverted pendulum remains a quintessential model in nonlinear control theory, serving as a bridge between theoretical advancements and practical implementations. By continually refining and testing control strategies on this benchmark, researchers can address fundamental issues and push the boundaries of what is possible in nonlinear control. This study aims to contribute to this ongoing journey, offering new perspectives and solutions to enhance our understanding and control of nonlinear systems [40].

2.2 Conventional Control Methods: A Review

Traditional control techniques encompass conventional methodologies like PID (Proportional-Integral-Derivative) controllers, state-space methods, and frequency-domain analysis. These methods rely on linear system assumptions and mathematical models for controller development, but they often encounter difficulties when confronted with complex nonlinear systems.

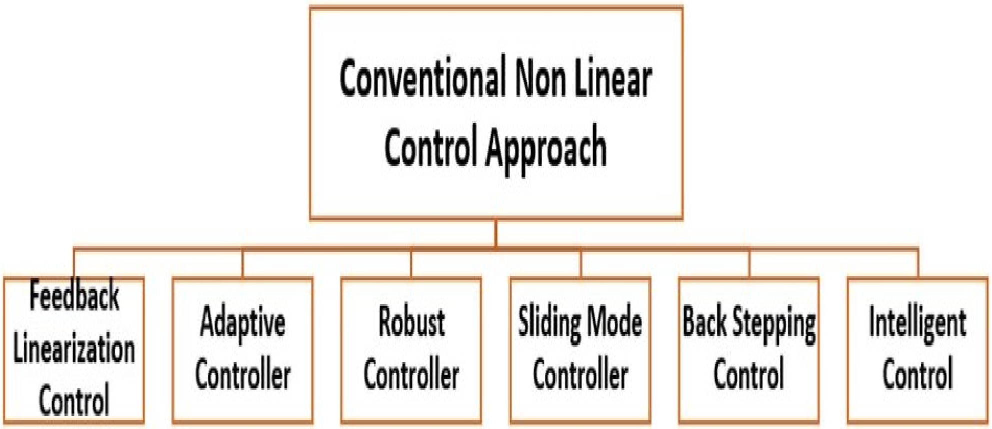

FIGURE 2.1: Conventional Nonlinear Control Approach

2.2.1 Feedback Linearization Control

Feedback Linearization Control (FLC) serves as a widely adopted strategy for stabilizing nonlinear systems. This technique transforms nonlinear systems into linear ones by modifying variables and applying suitable control inputs. FLC utilizes linear control methodologies to mitigate the nonlinear characteristics of a system [41]. Figure 2.1 illustrates the conventional nonlinear control approach, highlighting the system's response using established control methodologies distinct from linear control techniques, such as PID or LQR. However, a notable drawback of FLC lies in its elimination of crucial nonlinear elements during the design phase of control [42]. Despite not guaranteeing robust performance against uncertainties, FLC can substantially mitigate the impact of uncertainties [43].

2.2.2 PID-based Control

In recent years, advancements in control theory have extended the applicability of the PID controller to nonlinear systems and challenging operating environments. Research efforts have focused on refining tuning methods, incorporating adaptive and robust control techniques, and integrating machine learning algorithms to enhance the PID controller's capabilities. These developments aim to address the evolving needs of modern industrial processes, where flexibility, adaptability, and performance are paramount.

The Proportional Integral Derivative (PID) controller is reliable and offers a practical control solution for numerous industrial processes. One of its key advantages is the straightforward implementation and customization it allows, striking a suitable balance between ease and cost of implementation. Ongoing research affirms satisfactory stability margins and overall performance across a diverse array of processes. The analysis of closed-loop stability for a nonlinear PID control system involves supervisory control and a modified adaptation rule with projection [44].

2.2.3 Energy-based Control

Energy-based control represents an engineering framework that employs energy as a primary variable for system analysis and design. This approach centers on the manipulation and regulation of system energy to achieve the desired performance and stability.By emphasizing the energetic components of a system, energy-based control offers a robust framework for comprehending and optimizing dynamic system behavior [45].The utilization of energy-based control methods is applied to investigate the dynamics of an inverted pendulum on a cart model. Practical considerations such as friction and quantization are taken into account, ensuring that the suggested control approaches are adaptable and resilient in real-world operations [46].

2.2.4 Robust Control

Robust control (RC) is a controller design methodology explicitly tailored to address uncertainty, aiming to constrain uncertainty rather than characterize it in terms of distribution. The primary objective of resilient design is to ensure system performance despite model flaws and modifications. RCs are designed to handle significant uncertainties within size constraints, utilizing either pre-defined temporal and state functions or constant regulation of limited uncertainties based on established bounding functions [47]. Employing robust methodologies is crucial in creating reliable embedded systems, allowing the exploration of design alternatives that remain insensitive to system changes while upholding stability and performance. The robustness of RCs has garnered significant attention, making them widely applicable in various domains [48].

2.2.5 Sliding Mode Control

Sliding mode control (SMC) represents an emerging nonlinear control approach that modifies the dynamics of a nonlinear system by employing a discontinuous control signal, causing the system to slide across a cross-section of its typical behavior [49]. The design of an SMC controller is typically divided into two phases: the first involves creating a sliding surface that fulfills the design criteria, and the second entails selecting a control rule that attracts the system state to the switching surface.

In the initial phase, the emphasis is on establishing a sliding surface that meets the design specifications, exhibiting robust resistance to external disruptions and uncertainty [50]. It's crucial to highlight that the employment of discontinuous signals in Sliding Mode Control (SMC) can result in chattering. This phenomenon, in turn, has adverse effects such as diminishing system performance, increasing energy consumption, contributing to actuator degradation due to heightened mechanical wear, and posing risks to overall system stability.

2.2.6 Intelligent Control

Controllers endowed with intelligence are particularly well-suited for handling nonlinear systems, even in the presence of disturbances. The term "intelligent controller" encompasses a category of control systems that leverage various artificial intelligence computer technologies. These technologies include fuzzy logic, neural networks, machine learning, reinforcement learning, and evolutionary computation, from which the controllers derive their designation [51].

A neural network (NN) is a type of machine learning model in computer science designed to emulate the organization of the human brain. NNs are constructed using a network of mathematical equations as their foundational components. In each successive step, these networks of neurons predict the system's output based on both the system's prior output and input data. Training neural network controllers require a significant amount of time. The term neural control is employed to mitigate systematic deviations and counteract external disturbances when the system is distant from its equilibrium state [14].

Reinforcement learning stands as a subfield within machine learning, concentrating on the training of models to make sequential decisions in response to input data. In this process, an agent actively engages with its environment, acquiring knowledge through experiences of both mistakes and rewards [40].

Reinforcement learning stands as a subfield within machine learning, concentrating on the training of models to make sequential decisions in response to input data. In this process, an agent actively engages with its environment, acquiring knowledge through experiences of both mistakes and rewards. Through trial and error, the agent learns to optimize its decision-making strategy, aiming to maximize cumulative rewards over time. This paradigm has found applications in various domains, including robotics, gaming, finance, and autonomous systems, where complex decision-making tasks can benefit from adaptive and data-driven approaches.Reinforcement learning is a subdivision of machine learning that includes aspects of both supervised and unsupervised learning, among other techniques. Nonetheless, the teaching process for models in reinforcement learning differs fundamentally from other machine learning methods. Its key advantage lies in the ability to learn intricate behaviors without relying on labeled training data. Furthermore, it excels at making immediate decisions while optimizing for long-term objectives [52].

2.3 Neural Network-based Models - An Overview

The literature survey on models based on neural networks presents an in-depth examination of diverse models, their applications, and progress within the neural network field. This thorough survey is organized to offer insights into the historical development, fundamental concepts, and current applications of models based on neural networks.

2.3.1 Historical Perceptive

2.3.1.1 Perceptron and Early Models

The historical evolution of models based on neural networks can be traced back to the foundational perceptron model, a concept introduced by Warren McCulloch and Walter Pitts in the 1940s. The perceptron was conceptualized as a simplified model of a biological neuron, to emulate its fundamental functioning [53]. In this model, inputs are fed into the network, and through a series of interconnected layers, information is processed to generate outputs. The learning process entails fine-tuning the weights associated with each connection in the network, and optimizing them to minimize the difference between predicted and actual outputs for a given set of inputs. Figure 2.2 provides a visual representation of this iterative learning process, demonstrating the network's ability to adapt and improve its performance over time.

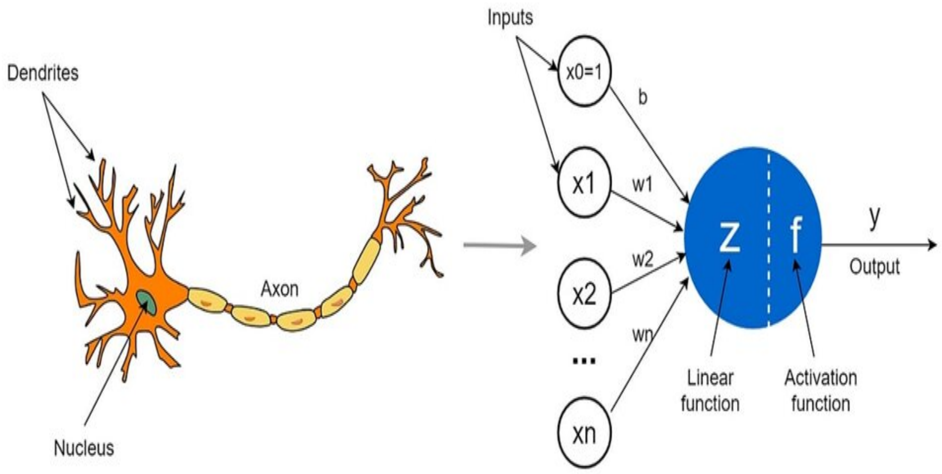

FIGURE 2.2: Perceptron Model [54]

Nevertheless, perceptrons faced limitations in addressing complex problems due to their linear nature and incapability to learn non-linear relationships. The advancement of neural network architectures gained momentum with the introduction of the multilayer perceptron (MLP) in the 1960s, as depicted in figure 2.3.

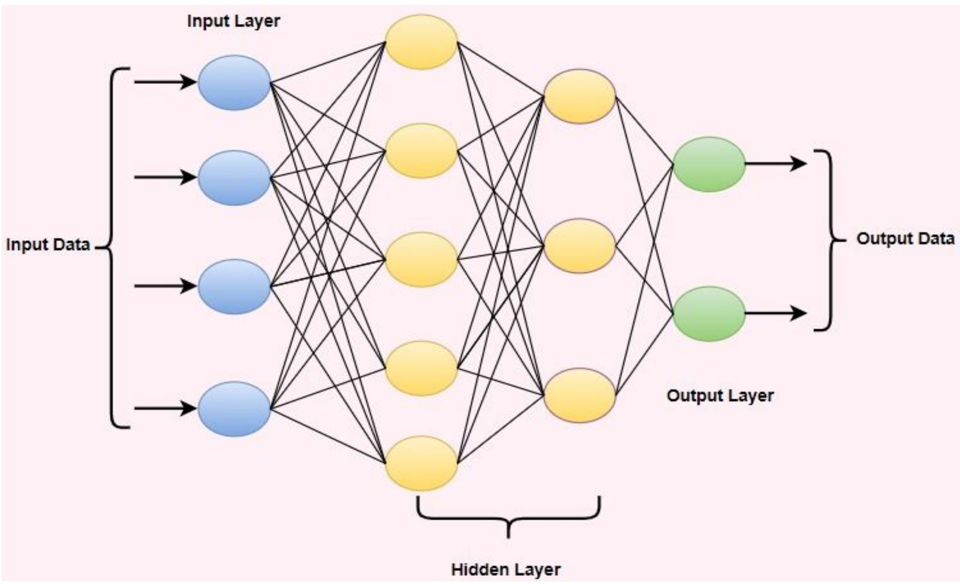

FIGURE 2.3: Multi-layer perceptron

This extension facilitated the development of more sophisticated models by integrating multiple layers of interconnected neurons. This enhancement empowered the network to capture intricate patterns and relationships within data. However, despite this progress, early neural network models encountered challenges, including difficulties in training deep networks attributed to issues like the vanishing or exploding gradient problem.

2.3.1.2 Backpropagation and Training Algorithms

The subsequent resurgence of interest in neural networks, especially with the advent of backpropagation by Geoffrey Hinton and others in the 1980s, marked a pivotal moment. Backpropagation enhances neural network training by iteratively adjusting weights through the use of gradient descent citer61. The figure visually illustrates this process, demonstrating how errors are minimized through both forward and backward passes. Figure 2.4 illustrates the step-by-step process of updating weights in a neural network using the backpropagation algorithm, facilitating iterative learning by minimizing the difference between predicted and actual outputs.

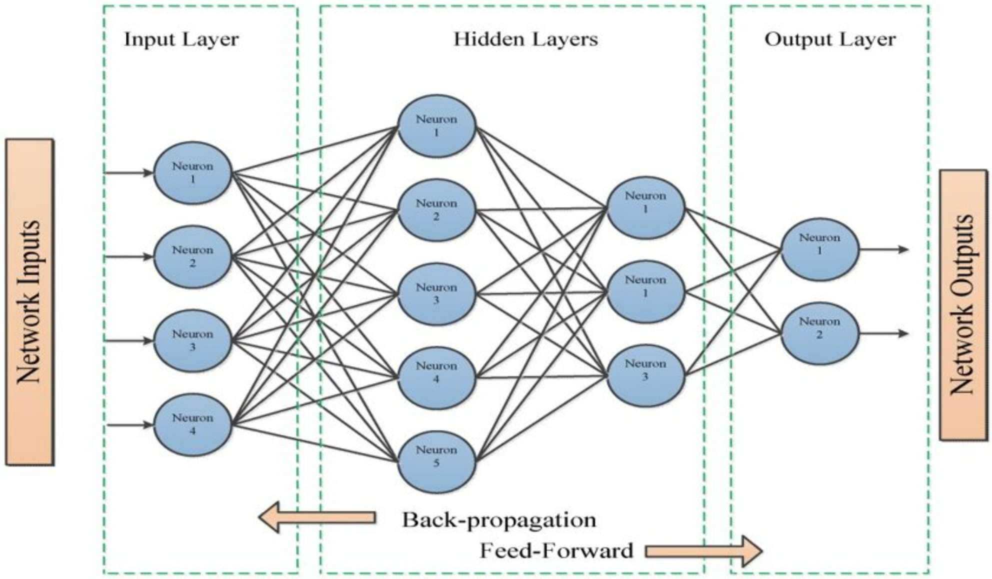

FIGURE 2.4: Backpropagation algorithm [55]

This breakthrough laid the foundation for the contemporary deep learning era, giving rise to diverse architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and advanced techniques like transfer learning and attention mechanisms. In summary, the historical journey from the foundational perceptron to modern neural network architectures underscores the iterative process of overcoming challenges and refining models to attain their current level of sophistication and applicability.

2.4 Applications of Neural Network-Based Models in Control Systems

2.4.1 System Identification

Neural network-based models are employed for system identification, where they learn and represent the dynamic behavior of complex systems [56]. This facilitates accurate modeling without explicit knowledge of the system dynamics.

2.4.2 Nonlinear Control

Neural networks excel in effectively managing nonlinearities present in control systems. Their proficiency lies in approximating intricate nonlinear mappings, providing a powerful solution for controlling systems with complex dynamics that may pose challenges for traditional linear controllers. This inherent capability allows neural networks to navigate and comprehend the nuanced behaviors associated with nonlinear complexities, ultimately enhancing the adaptability and overall performance of the control system.

In practical applications, the aptitude of neural networks to handle nonlinearities becomes particularly advantageous when dealing with real-world scenarios where system behaviors deviate from linear patterns. By capturing and representing these nonlinear relationships, these models contribute to more precise predictions and enable responsive control actions. This broadens the utility of control systems across diverse and demanding environments, ensuring a resilient and efficient response to the intricate dynamics associated with nonlinear system behaviors.

The capacity of neural networks to handle nonlinearities proves particularly valuable in real-world applications where system behaviors may deviate from linear patterns. By capturing and representing nonlinear relationships within the control framework, these models contribute to more accurate predictions and responsive control actions. This not only broadens the applicability of control systems to diverse and challenging environments but also ensures a robust and efficient response to the complexities inherent in nonlinear system dynamics [9].

2.4.3 Model Predictive Control (MPC)

Neural network-based models enhance the predictive capabilities of MPC by providing accurate models for predicting future system behavior. This improves the decision-making process in real-time control applications [57]. In the realm of control systems, neural network-based models contribute to various aspects, including system identification, adaptive control, handling nonlinearities, fault detection, optimization, reinforcement learning, robotics control, and enhancing the capabilities of Model Predictive Control (MPC). These applications showcase the versatility of neural networks in addressing complex control challenges and optimizing system performance.

2.5 Literature Review

In [58], a novel approach using Artificial Neural Networks (ANN), specifically a Feed-Forward Network (FFN), is introduced to tackle the nonlinearity of an inverted pendulum (IP). The ANN is trained with the "Trainlm" network function, employing the Levenberg Marquardt (LM) back-propagation algorithm. The main focus is on addressing the nonlinear behavior associated with parameters like the angle and position of the IP. The goal is to enhance accuracy and response time in comparison to conventional controllers. Through MATLAB simulations, the paper evaluates the neural network controller's performance and compares it with traditional controllers. A notable limitation is the training of neural networks using a single initial condition. The conclusion highlights the advantages of ANN, emphasizing its speed, accuracy, and stability, suggesting its potential for real-time IP system control.

In [11], explained how Segway is controlled. It compared conventional PID and a Neural network-based control– to manage the Segway's position and handlebar angle. The study used MATLAB simulations and assessed Segway performance through a simplified inverted pendulum model. The results showed that the NN controller is better than the traditional PID one, offering improved stability and response. It suggests that using an NN controller instead of PID can boost the Segway's performance in industrial applications. However, it's worth noting a limitation of the NN PID training process, relying on a single random initial condition, which might cause oscillation and instability due to noise and measurement errors. It suggests that using an NN controller instead of PID can boost the Segway's performance in industrial applications.

In [59], discussed the challenge of stabilizing the Rotary Inverted Pendulum (RIP) system. It combined a linear quadratic regulator (LQR) for basic control and a Neural Network compensator to handle deviations and disturbances. Stability is ensured through analyses. The effectiveness was shown in simulations and real experiments, proving its ability to handle uncertainties. However, it's worth noting a limitation: the control approach is based on a linear method and may miss some system dynamics. Future research could explore tweaks to enhance performance, like adjusting LQR gains and refining the neural network's learning.

In [60], enhanced stable tracking for the inverted pendulum system, proposing a novel PID controller based on Radial Basis Function (RBF) neural network supervision. An advanced version, IPID-RBF, incorporated both feed-forward and feedback supervision, improved the system response speed, and reduced tracking signal overshoot. Simulations highlight the effectiveness of these controllers under various input signals, with IPID-RBF outperforming traditional PID and demonstrating superior tracking precision. Simulations highlight the effectiveness of these controllers under various input signals, with IPID-RBF outperforming traditional PID and demonstrating superior tracking precision. While showcasing simplicity, robustness, and high accuracy, the study acknowledges the need for further validation across diverse systems to ensure generalizability.

Table 2.1: Literature Review Table

S.N.	Ref	Year	Contributions	Limitations
1	[56]	2020	The paper presented a method for employing ANN to regulate a non-linear inverted pendulum. The 'Trainml' function was used for training. This method focuses on the nonlinearity of the position and velocity of an inverted pendulum	Neural Networks were trained using a single initial condition.
2	[61]	2015	It presents a unique method of stabilizing a cart-inverted pendulum system using an artificial neural network (ANN) and a PID controller based on LQR. Improved performance and disturbance rejection were suggested by this method.	This technique's primary drawback is that it requires a thorough understanding of the problem's current condition, which is not always achievable.
3	[15]	2013	BP neural network combined with Levenberg-Marquardt to address the instability issue. This study built up the FFNN and used the LQR control mechanism to gather the data for Neural Networks and ability to predict and stabilize system behavior effectively.	If the starting guess is far from the minimum or if the matrix is not functioning properly, the LM method may converge very slowly.

4	[16]	2020	The use of traditional PID control is the primary focus of this study, which also evaluates an alternative control based on neural networks to replace the PID controller. Based on the simulation, NN control outperformed PID control.	The neural network PID was trained using a single random reference signal. Noise and measurement errors might create oscillation and instability in the input signal.
5	[14]	2022	To stabilize the system in this study, an intelligent control was created. The controller was built using an online radial basis function (RBF) neural network compensator and a linear quadratic regulator (LQR). Lyapunov studies are used to show that the closed-loop system is stable.	This control was set up using a linear approach alone, which does not account for all of the system dynamics.
6	[62]	2021	This study suggests a new PID controller called PID-RBF, which is based on the RBF neural network supervisory control approach. This technique enables the system's reliable tracking signal to be adjusted adaptively. Both the system's reaction time and the tracking signal's overshoot were decreased.	Validation mainly examines a system, raising concerns about its generalizability to other systems.

7	[63]	2022	The primary component of the model reference control (MRC) in this work is the application of recurrent paraconsistent neural networks (RPNNs). The paraconsistent annotated logic by 2-value annotations (PAL2v) rules serves as the activation function for the RPNN's hidden neurons. In recurrent neural networks, the PAL2v neuron was contrasted with various activation functions.PAL2v-based RPNNs effectively overcome traditional activation function.	Results may not apply to other systems; more testing is required for underactuated mechanical systems.
8	[64]	2018	This research centers on applying the Data-driven Model method in situations when some model parameters are imprecise or unclear. This was created by using a linearized state-space model and performing an algebraic Riccati Equation to determine the optimal state feedback control optimal state feedback control rule. A Radial Basis Function (RBF) neural network was used to model the Inverted Pendulum system's dynamic behavior.	The Radial Basis Function was learned using a modest set of data samples.

9	[65]	2014	This study presents a data-driven modeling technique and control strategy for severely nonlinear and fast-responding Magnetic Levitation (Maglev) systems. Neural networks using Gaussian radial basis function (RBF) were used to approximate the model coefficients. The system was successfully stabilized by the proposed predictive controller, which was based on the recognized model.	The neural network model was trained using a small number of data samples, which is insufficient to fully represent the dynamics.
10	[66]	2018	The use of PID Neural Networks to the control of Inverted Pendulum is the main topic of this paper. Using the Levenberg-Marquardt method, weights were modified.the simulation's findings demonstrated how effective and adaptable the control is in managing the inverted pendulum. The study highlights the neural network's ability to handle the system's nonlinear dynamics, showing superior performance in maintaining stability and achieving desired responses and the potential for further enhancement.	Three neurons and one hidden layer were utilized, which is insufficient to adequately describe the system's dynamics.

11	[67]	2016	The Furuta pendulum which is the two-degree-of-freedom underactuated system, was used in the study to demonstrate a unique adaptive neural network-based control method. Real-time experiments were carried out to validate the performance of the proposed controller.	The non-linear system was modeled using a single hidden Layered Feed-forward neural network with six neurons.
12	[68]	2016	Using an ANN control approach, this study demonstrated the flexibility of an on-line ADALINE controller for the difficult nonlinear and unstable Inverted Pendulum system. By combining LQR with ADALINE, the study showed increased control efficiency.	In the context of one system, the suggested control method that makes use of the Online ADALINE controller and ANN was tried and discussed. It's unclear how far the method can be expanded upon and used with other highly nonlinear systems.

2.6 Research Gap

The existing literature provides valuable insights into the application of neural network-based controllers for nonlinear systems, particularly in the context of inverted pendulum dynamics. Nonetheless, a significant research void appears in the little investigation of control schemes that specifically tackle these systems' sensitivity to varying initial conditions. Many studies primarily focus on training controllers with a small number of samples or a single initial condition, ignoring the wider range of initial conditions. This constraint makes it more difficult to fully comprehend how resilient and adaptive the controller is in a variety of settings. It is also necessary to compare and assess the suggested controllers' adaptability to nonlinear systems and their potential for use in scenarios not covered in detail in previous research. To develop control strategies that are capable of handling varying initial conditions in non-linear systems, it is imperative to bridge this research gap.

2.7 Problem Statement

Designing controllers for nonlinear systems presents challenges due to their sensitivity to initial conditions. Both traditional control methods and neural network-based controls face difficulties when trained on a single initial condition. This struggle arises from the complex relationship between the system's behavior and its starting conditions. Solving this challenge requires innovative approaches to ensure stability and reliability in controlling nonlinear systems.

As a result, a controller that can manage a range of varying initial conditions must be developed. This research aims to use Neural State Space Model-based Predictive Control to create a sophisticated control strategy. To enable the controller to adjust and function at its best across a variety of system states, the goal is to train it on a wide range of initial conditions. The proposed approach seeks to improve control systems' resilience and flexibility by tackling the problems that come with nonlinear dynamics and their sensitivity to initial conditions.

Chapter 3

Mathematical Model of Non-Linear Systems

3.1 Background

This chapter focuses on the evolution of the dynamics of the inverted pendulum system, which is nonlinear and unstable. The mathematical model provides the necessary details to understand the workings of the system.

3.2 Inverted Pendulum System

The inverted pendulum system consists of a cart designed to balance a pendulum. Because the pendulum would begin to fall at the smallest disturbance, this mechanism is inherently unstable. Thus, to keep a pendulum in equilibrium, some kind of control is required. A perfect controller would hardly alter the pendulum angle or cart position to maintain the pendulum's equilibrium. Developing a near-optimal controller is a difficult design task [69]. Figure 3.1 depicts an inverted pendulum system, wherein the pendulum is positioned in an unstable equilibrium state with its mass above its pivot point, challenging control systems to maintain its upright position through continuous adjustments.

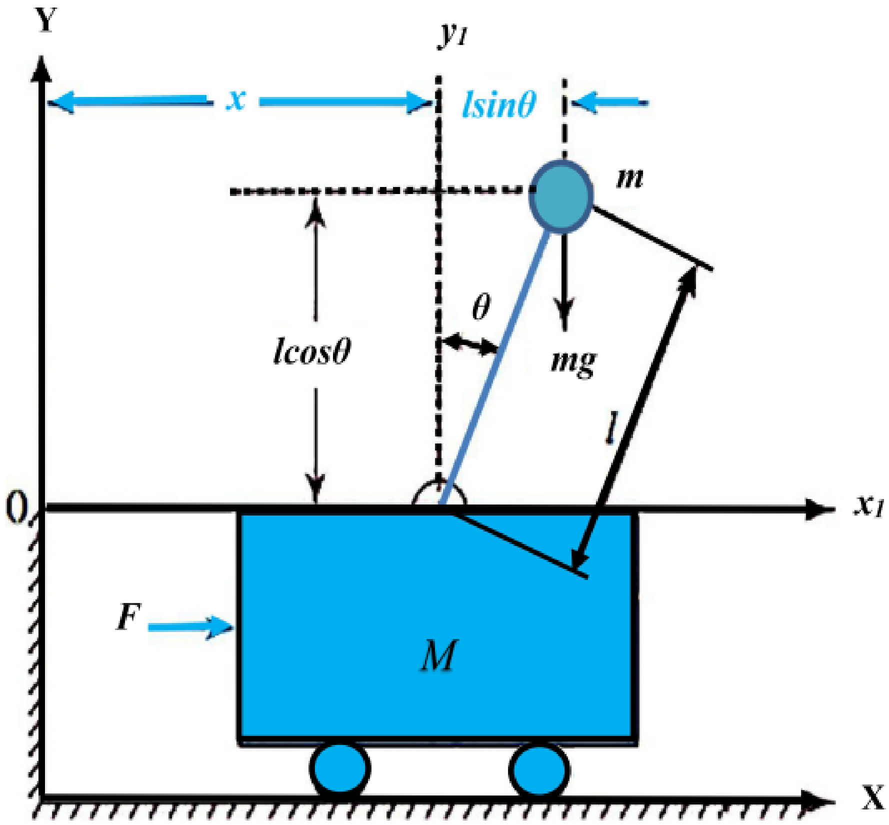

FIGURE 3.1: Inverted Pendulum System [70]

Figure 3.1 depicts the free-body diagram for an inverted pendulum placed on a cart. In the given system, **M** represents the mass of the Cart, **m** represents the mass of the pendulum, and **l** represents the length of the pendulum which is located on the top end of the inverted pendulum. A force from outside the system, represented by **F**, acts on the cart.

3.3 Mathematical Model of Inverted Pendulum

The mathematical model of an inverted pendulum represents a dynamic system that captures the relationships between the cart's position, the pendulum's angle, and external forces, governing the motion of a pendulum inverted above its unstable equilibrium. The equations typically involve terms representing the inertia of the cart and pendulum, gravitational forces acting on the pendulum, frictional forces resisting motion, and control inputs applied to the cart to stabilize the pendulum. These equations may be nonlinear due to the trigonometric terms involved in describing the pendulum's angle and the coupling between the cart's position and the pendulum's motion.

3.3.1 Sum of Forces

The external force **u** acting on the system must be equal to the mass times the acceleration of the cart plus the mass times the acceleration of the point mass in the x-direction, as determined by force balancing in the x-direction. [71]

$$M\frac{d^2}{dt^2}x + m\frac{d^2}{dt^2}x_p = u \tag{3.1}$$

The distance from the reference to the center of the pendulum is denoted as $\mathbf{x_p}$, and the distance from the center of the pendulum to the center of the cart is denoted by $\mathbf{y_p}$. The expressions for these distances are given by:

$$x_p = x + l\sin(\theta) \tag{3.2}$$

$$y_p = l\sin(\theta) \tag{3.3}$$

Putting equation(3.2) into(3.1), we get:

$$(M+m)\ddot{x} - ml\sin(\theta)(\dot{\theta}^2) + ml\cos(\theta)\ddot{\theta} = F \tag{3.4}$$

This is the final equation derived, representing the force balance in the x-direction for the cart-pendulum system. It considers the masses of the cart and pendulum, the pendulum angle, its angular velocity ($\dot{\theta}$), and angular acceleration ($\ddot{\theta}$). The external force u acting on the system is also included.

3.3.2 Sum of Torques

Similarly, the system is subjected to a torque balance, where torque is defined as the product of the distance to the pivot point and the force's perpendicular component. In this case, the torque that the acceleration force applies to the mass is equal to the torque that the gravity force applies to the mass.The acceleration force applies to the mass is equal to the torque .This torque balance equation accounts for the rotational dynamics of the pendulum, balancing the torques acting on the pendulum mass to maintain its equilibrium position or describe its motion about the pivot point. By considering these torque balances, along with other factors such as inertia, friction, and control inputs, the complete dynamics of the inverted pendulum system can be modeled and analyzed to develop effective control strategies.[71] The resulting torque balance is given as: The equation is given by:

$$(F_x \cos(\theta))l - (F_y \sin(\theta))l = (mg \sin(\theta))l \tag{3.5}$$

where the force components F_x and F_y are determined as:

$$F_x = m\ddot{x} - ml \sin(\theta)(\dot{\theta}^2) + ml \cos(\theta)\ddot{\theta} \tag{3.6}$$

$$F_y = -ml \cos(\theta)(\dot{\theta}^2) - ml \sin(\theta)\ddot{\theta} \tag{3.7}$$

Substituting equations (3.6) and (3.7) into equation (3.5), we have:

$$\begin{aligned} m\ddot{x} \cos(\theta) - ml \sin(\theta) \cos(\theta)(\dot{\theta}^2)+ \\ ml \cos^2(\theta)\ddot{\theta}\ ml \sin(\theta) \cos(\theta)(\dot{\theta}^2) + ml \sin^2(\theta)\ddot{\theta} = m \sin(\theta) \end{aligned} \tag{3.8}$$

The sum of torques balancing along the y-axis is given by:

$$m\ddot{x} \cos(\theta) + ml\ddot{\theta} = mgsin(\theta) \tag{3.9}$$

Putting equation (3.9) into equation (3.4) , we get:

$$\ddot{x} = \frac{F + ml \sin(\theta)(\dot{\theta}^2) - mg \cos(\theta) \sin(\theta)}{M + m - m \cos^2(\theta)} \tag{3.10}$$

Put the above equation into equation (3.9), we get:

$$\ddot{\theta} = \frac{F \cos(\theta) - (M + m)g \sin(\theta) + ml \cos(\theta) \sin(\theta)\dot{\theta}^2}{ml \cos^2(\theta) - (M + m)l} \tag{3.11}$$

These two equations clearly indicate the presence of a nonlinear system, posing certain mathematical challenges. These equations may now be stated in state space form, considering the following state variables:

$$x_1 = \theta$$
$$x_2 = \dot{\theta}$$
$$x_3 = x$$
$$x_4 = \dot{x}$$

In the given system, x_1 represents the position of the pendulum, x_2 corresponds to the angular velocity of the pendulum, x_3 represents the position of the cart, and x_4 denotes the velocity of the cart, denoted by $\dot{x}$. The final state space equations for the inverted pendulum system may be stated as follows:

$$\dot{x}_1 = x_2 \tag{3.12}$$

$$\dot{x}_2 = \frac{F\cos(x_1) - (M+m)g\sin(x_1) + ml\cos(x_1)\sin(x_1)x_2^2}{ml\cos^2(x_1) - (M+m)l}$$
$$\dot{x}_3 = x_4$$
$$\dot{x}_4 = \frac{F + ml\sin(x_1)x_2^2 - mg\cos(x_1)\sin(x_1)}{M + m - m\cos^2(x_1)}$$

3.3.3 Linearization of Inverted Pendulum

The goal is to place the pendulum in an upright position with $\theta = 0$. The Jacobian approach is used to linearize the system around this equilibrium position. [72]. The system is linearized with $x_0 = 0$ and $u_0 = 0$. The linearized system is in the form:

$$\dot{x} = Ax + Bu \tag{3.13}$$

The terms of the Jacobian matrix are obtained in a systematic manner by following linearization.By evaluating the partial derivatives of each equation with respect to the state variables x, we obtain the Jacobian matrix A. Linearization provides a linear approximation of the system dynamics near the equilibrium point, enabling the application of linear control techniques such as LQR (Linear Quadratic Regulator) or pole placement to design stabilizing controllers.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{Ml} & 0 & 0 & 0 \\ \frac{-mg}{M} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -\frac{1}{Ml} \\ 0 \\ \frac{1}{M} \end{bmatrix}$$

The output equation is represented by:

$$y = Cx$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

So, the equation y=Cx essentially means that the output vector y, which contains the positions of the pendulum and the cart, is determined by linearly combining the elements of the state vector x using the output matrix C.

3.4 Parameters of System

The simulation of the Inverted Pendulum is done in MATLAB/Simulink 2023b. The parameters of the system are listed in the table below, which includes the mass of the cart, the mass of the pendulum, the length of the pendulum, and the gravitational acceleration. These parameters are crucial for accurately modeling and simulating the dynamics of the inverted pendulum system.

Parameters	**Detail**	**Values**
M	Cart mass	0.6 kg
m	Pendulum mass	0.25 kg
l	Length of Pendulum	0.3 m
g	Gravitational acceleration	$9.8\,\mathrm{m/s^2}$

TABLE 3.1: Parameters of the system

3.5 MATLAB Simulation

s-function (short for system function) is a fundamental component in Simulink. It is used to represent the dynamics of an inverted pendulum. This includes creating the inverted pendulum's equations of motion and adding variables such as cart and pendulum lengths and masses.

In the context of modeling an inverted pendulum, S-functions are used to create the system's equations of motion, incorporating variables such as the cart and pendulum lengths and masses. The dynamics of the inverted pendulum are nonlinear due to the pendulum's angle dependence on trigonometric functions like sine and cosine. The S-function begins by defining the system's parameters, including the cart mass (M), pendulum mass (m), pendulum length (l), and gravitational acceleration (g). These parameters are crucial for accurately capturing the system's behavior.

3.6 Step Response

s-function is used to analyze the response of the System. Simulink provides tools for linear and nonlinear system analysis, allowing us to study the system's behavior under different scenarios. Simulink also offers linearization tools that allow you to linearize your model around an operating point, enabling linear system analysis techniques such as frequency response analysis, pole-zero analysis, and state-space analysis.

This can be particularly useful for studying the small-signal behavior of your system and designing linear control strategies.The step response of the Inverted Pendulum System is given in the Figure 3.2

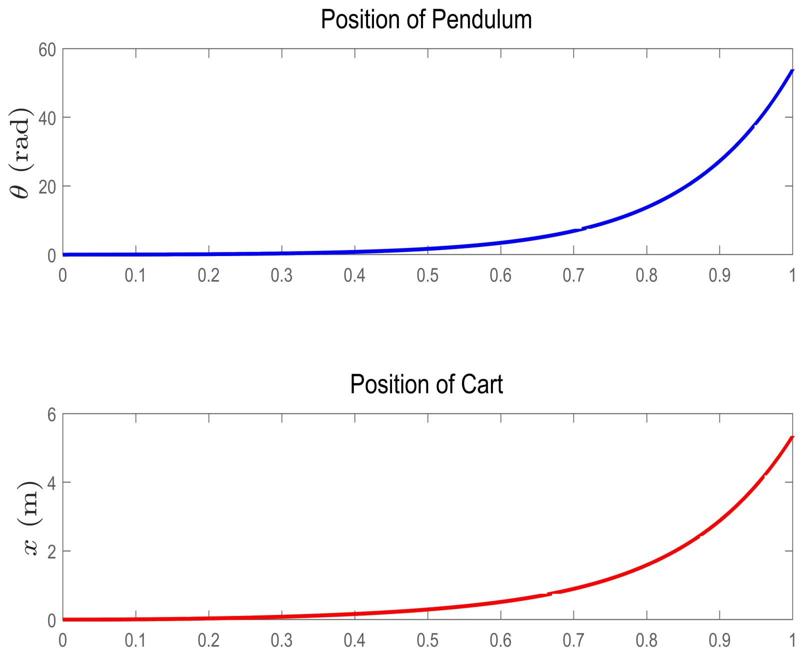

FIGURE 3.2: Step Response for an Inverted Pendulum System

3.7 Chapter Summary

In this chapter, the dynamics of the Inverted Pendulum are discussed in detail. The behavior of the System is shown using step input. The mathematical model of the System is discussed and equations of the System are derived using Newton's Laws. Then all simulations are shown in MATLAB®/Simulink

Chapter 4

Methodology

We introduce the control strategy in this chapter. In section 4.1, we discussed the design of our system using two distinct control laws, and then we talked about how to get data for it. We talked about the process of building a neural state-space-based model in section 4.2. Moreover, we describe the MPC policy used in our system in section 4.3. Then we outline the optimization problem formulation and the steps involved in generating control actions over a finite time horizon. In section 4.4, we conclude all the implementation steps.

4.1 Design of Experiments

The design of Experiments is a critical step in any data-driven modeling application. Before deploying an artificial neural network, it must be trained to solve specific problems. There are three main learning paradigms:

- Supervised Learning: [68, 73–76]
- Unsupervised Learning: [77–81]
- Reinforcement Learning: [54, 82–85]

We emphasized the importance of supervised learning by using it in our strategy to train the Neural Networks.

This technique is based on labeled data, which gives the network explicit guidance and assures that exact outputs match input signals exactly.

4.1.1 Feedback Linearization Technique

To provide the supervised training data for the neural network controller, we use the direct adaptive control technique described in [56], [68], [86]. A control law for the inverted pendulum controller was designed based on the following equations. The force, u, needed to maintain the stability of pendulum stability is determined by the primary equation.

$$h_1 = \frac{3}{4} lg \sin(\theta) \tag{4.1}$$

$$h_2 = \frac{3}{4} l \cos(\theta) \tag{4.2}$$

$$f_1 = m(l \sin(\theta)\dot{\theta}^2 - \frac{3}{8} g \sin(\theta)) - \dot{x} \tag{4.3}$$

$$f_2 = M + m \left(1 - \frac{3}{4}\cos^2(\theta)\right) \tag{4.4}$$

$$u = \frac{f_2}{h_2}\left[h_1 + c_1(\theta - \theta_d) + c_2\dot{\theta} + n_1(x - x_d) + n_2\dot{x}\right] - f_1 \tag{4.5}$$

x_d and θ_d are the desired position of the cart and the angle of the pendulum, respectively. The numeric values for c_1, c_2, n_1, and n_2 used for the simulation and the parameters are given in table 4.1.

Parameters	**Values**
M	0.6 kg
m	0.25 kg
l	0.3 m
g	9.8 kg m/s^2
c_1	50
c_2	20
n_1	60
n_2	38
x_d	0.1
θ_d	0

TABLE 4.1: Parameters for Feedback Linearization

The implementation of the above control law in MATLAB is shown in Figure 4.1.

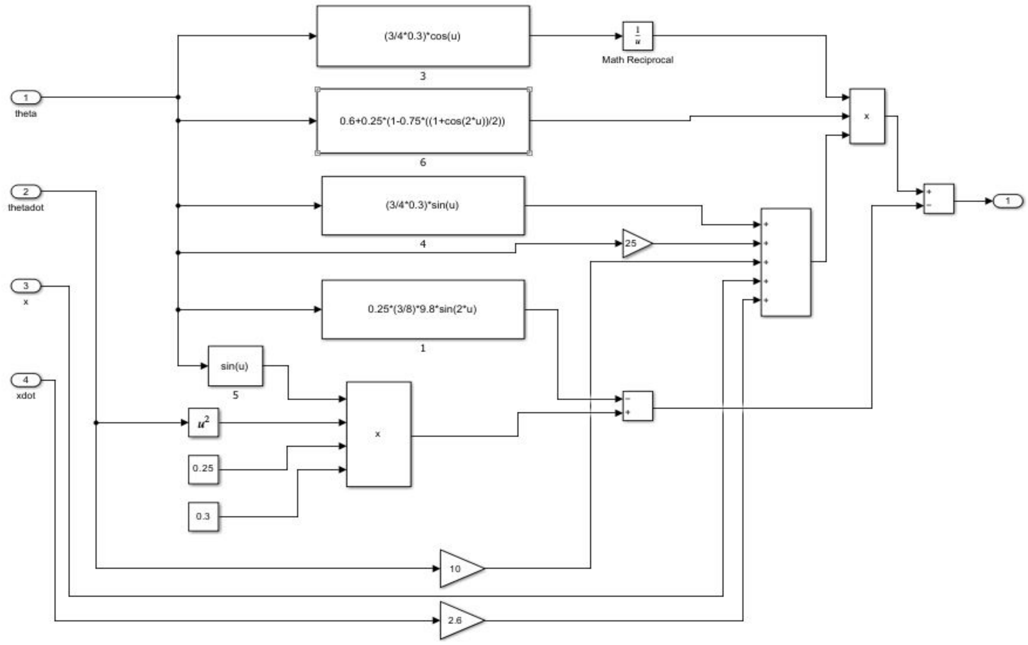

FIGURE 4.1: Feedback Linearization Control Law

The above control law is integrated with our nonlinear inverted pendulum system and the arrangement of both is shown in Figure 4.2

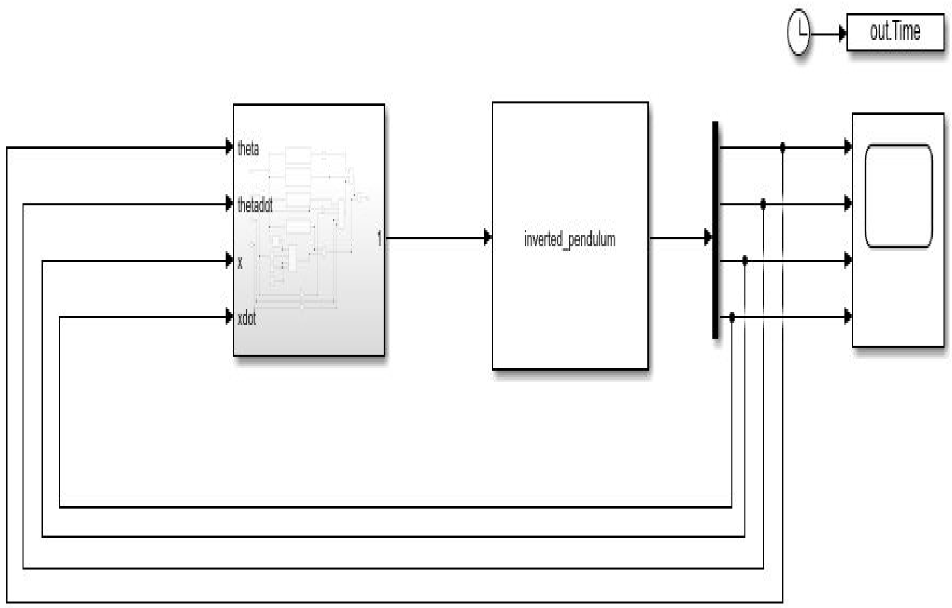

FIGURE 4.2: Feedback control law with inverted pendulum system

After simulating the above feedback linearization model, one hundred and one trials were carried out. When collecting data for neural state-space modeling, it is essential to make sure that the state trajectories explore the entire state space, considering the diverse operating conditions.

4.1.2 Linear Quadratic Regulator

Linear quadratic regulator, [51, 69, 70, 87]is a state-feedback controller that is used in many control systems. LQR controller utilizes a linear model to generate constant gains and it also ensures the global asymptotic stability of the equilibrium point. [14] The nominal model is obtained by linearizing the nonlinear dynamics without considering the disturbance term about the equilibrium point. The linear state-space equation of the inverted pendulum system is obtained as:

The state-space equation is given by:

$$\dot{x} = Ax + Bu \tag{4.6}$$

The state-feedback control signal is selected as:

$$u_{\text{LQR}} = -Kx \tag{4.7}$$

where K is the state feedback gain vector given by:

$$K = R^{-1}B^TP \tag{4.8}$$

Here, P is a positive definite symmetric matrix obtained from the solution of the Algebraic Riccati Equation, which is given below:

$$A^TP + PA - PBR^{-1}BP + Q = 0 \tag{4.9}$$

where R is a positive constant and Q is also a positive semi-definite symmetric constant matrix, minimizing the state cost matrix function. The cost function is given by:

$$J = \int_{-\infty}^{\infty} (x^TQx + Ru^2)\, dx \tag{4.10}$$

Remark: By using MATLAB software, a full-state feedback LQR controller is developed by solving the Algebraic Riccati Equation based on the weighting matrix Q and state penalty matrix R. The non-linear dynamical equations expressed in the linear state-space form serve as a foundation for this process.

4.2 Neural State Space Model

Systems show nonlinear behavior in many real-world situations that linear models are unable to adequately represent. Building state-space models for such systems is a good use for neural networks because they provide a strong foundation for learning and capturing nonlinear interactions. [88]

State-space models are used in classical control theory to express the link between the inputs, outputs, and internal states of dynamic systems, hence describing their behavior.[89]

Consider the mathematical form of the State Space representation of the System: The continuous-time Neural State Space equation is given by:

$$\dot{x}(t) = f(x(t) + h(x(t))u(t)) \tag{4.11}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector. The functions f and h are used for static non-linear mapping [20]

The discrete formulation of the Neural State Space has the following form [57]:

$$x(k+1) = f(x(k) + h(x(k))u(k)) \tag{4.12}$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the input vector, and these are typically represented by linear functions [90].

The state equation may be rewritten for the sampling instant k. The model becomes:

$$x(k) = f(x(k-1) + h(x(k-1))u(k-1)) \tag{4.13}$$

4.2.1 Training of NSSM Model

The use of a neural state-space model as a novel modeling technique is presented. The Neural Network (NN) was trained using data collected from the simulated plant, optimizing its parameters to minimize the difference between model predictions and actual observations. This model incorporates various neural network topologies, including wide, shallow, narrow, and deep architectures.

4.2.2 Performance Parameters

There are some performance matrices like mean squared error, mean absolute error, and stability time which are used to study the performance of the model which are discussed below in detail.

4.2.2.1 Mean Squared Error

MSE is a measure of the average squared difference between the actual and the predicted values. It is calculated by taking the average of the squared differences between the actual value and the predicted value. [91]

MSE is calculated as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.14}$$

where N represents the number of data points, y_i represents the actual values of the data points, and $\hat{y}_i$ represents the predicted values.

4.2.2.2 Mean Absolute Error

MAE is a measure of the average absolute difference between predicted and actual values. It is computed by taking the average of the absolute differences between each predicted and actual value. [91]

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.15}$$

where n represents the number of data points, y_i represents the actual values of the data points, and $\hat{y}_i$ represents the predicted values.

4.2.2.3 Stability Time

Settling time or stability time is a term used to refer to the period that is needed for a system's response to reach and stay within a certain range of its final equilibrium value after a disturbance or change in input. It quantifies how fast a model helps in returning the system back into balance.

4.3 MPC for the Baseline Model

To ensure regulation of the pendulum system, the Model Predictive Control (MPC) policy for the Neural State Space Model (NSSM) [92] optimizes control inputs over a given period. This means that there is a need to minimize the cost function defined as follows:

$$J_N(x^0, u^0) = \sum_{k=0}^{N-1} Q(x(k), u(k)) + T(x(N)) \tag{4.16}$$

The cost of the stage is represented by $Q(x(k), u(k))$, reflecting the current state x and control input u at each step. The aim is to drive all states of the system to zero subject to system constraints. Moreover, there exists a terminal cost term $T(x(N))$ that measures the expense associated with a final state.

This study does not apply a terminal cost. The MPC algorithm optimizes control inputs u over a prediction horizon N, thereby affecting the future behavior of the system.

The optimization problem defining an initial policy for MPC can be formulated as follows:

$$(x^0, u^0) = \min_{\bar{x}, \bar{u}} J_N(\hat{x}, u) \tag{4.17}$$

subject to:

$$\hat{x}(k+1) = \hat{f}(\hat{x}(k)) + \hat{h}(\hat{x}(k))u(k)$$

$$\hat{x}(0) = \hat{x}_0$$

where $\hat{f}$ is the Neural State Space Model dynamics. Additionally, constraints on control inputs and states are considered:

$$u_{\min} \leq u(k) \leq u_{\max} \tag{4.18}$$

$$x_{\min} \leq u(k) \leq x_{\max}$$

The above expressions represent the constraints on the input force and the system, ensuring that the system operates within the safe operating region.

4.4 Methodology

Step 1: Data Collection and Model Training

- For Neural State Space Model (NSSM), experimental data was collected from the simulated plant.
- Train the NSSM using the collected data, providing a data-driven representation of the system dynamics.

Step 2: MPC Design

- Define the MPC cost function based on the states of the Nonlinear State Space Model (NSSM).
- Formulate the optimization problem to minimize the cost function considering constraints and dynamics of the system.
- Specify constraints on states, inputs, and other variables such as cart position, pendulum angle, velocity, and control input limits.

- Choose prediction and control horizons for the MPC.
- Formulate the optimization problem as a finite-horizon constrained optimization problem.
- Use numerical optimization techniques like quadratic programming (QP) or nonlinear programming (NLP) solvers to solve the optimization problem online at each time step.
- Adjust controller parameters and constraints as needed to achieve desired performance and stability.

Step 3: Integration with NSS

- Use the learned NSSM to predict the system behavior inside the MPC framework.
- Establish the control and state constraints in compliance with the system's physical bounds.

Step 4: Simulation and Closed-Loop Control

- Simulate the MPC baseline policy in a closed-loop fashion.
- Implement real-time adjustments based on the established policy.

Step 5: Performance Analysis

- Analyze the performance of the MPC baseline policy in stabilizing the pendulum system.
- Analyze the convergence of the states toward the desired values.

The procedure for stabilizing a pendulum system is depicted in the flowchart, as shown in Figure 4.3. It emphasizes a systematic approach from data collection to real-time adjustment.

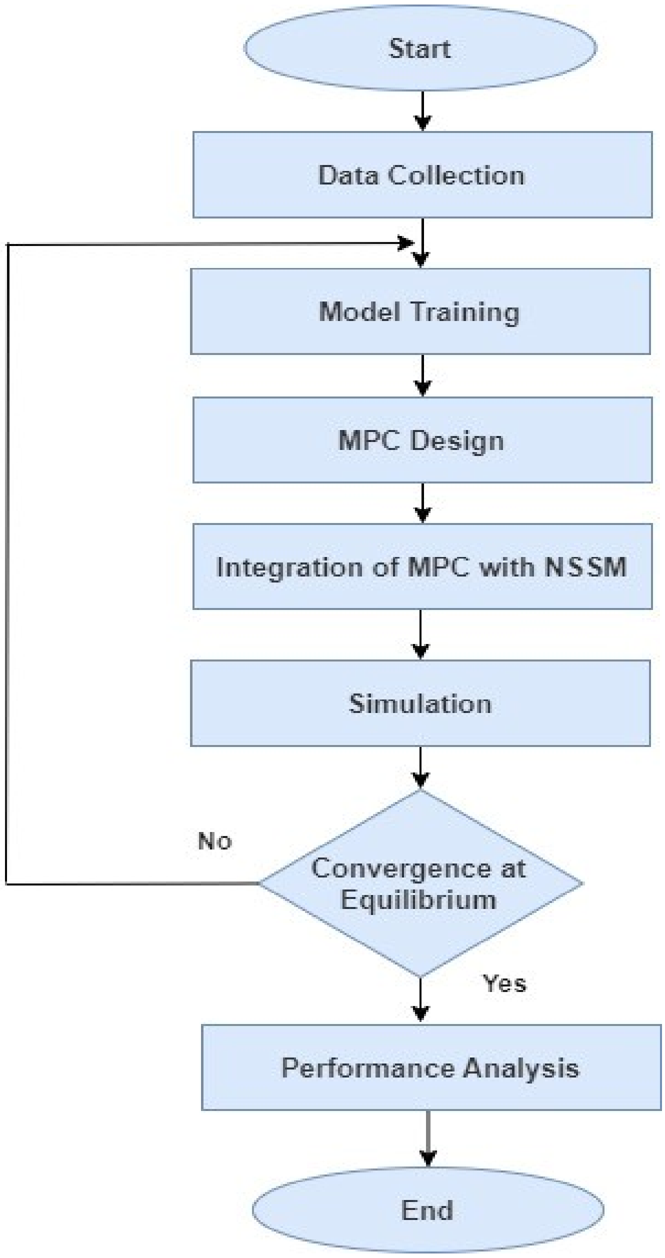

FIGURE 4.3: Flowchart of the Methodology

4.5 Chapter Summary

This chapter covers the design with two distinct control laws, data acquisition, neural state-space model construction, and the implementation of Model Predictive Control (MPC). The optimization problem formulation and steps for generating control actions over a finite time horizon are outlined, providing a holistic overview of the chapter's key contributions.

Chapter 5

Implementation and Results

5.1 Configurational Aspects of Simulations

MATLAB 2023b was used for the simulation. Each experiment lasted for 10 seconds, and state trajectories were recorded after every 0.01 seconds. The Nonlinear Model Predictive Controller (NL-MPC) is initialized using the **nlmpcMultistage** function, indicating a 10-stage MPC with 4 state variables and 1 manipulated variable. State constraints are set for the pendulum angle (θ) with a minimum of $-\pi$ and a maximum of π. The manipulated variable was constrained to the range $[-2, 2]$. The cost function for NL-MPC is defined as the sum of squares of all four state variables of the pendulum. The cost function is expressed in 5.1

$$J = x_1^2 + x_2^2 + x_3^2 + x_4^2 \tag{5.1}$$

After implementing the above model, one hundred and one experiments were conducted against a simulated plant, each starting at different initial conditions $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0]$.

After that, we load the data, and it consists of two cell arrays of timetables – the state trajectories Y and the input trajectories U. Set N, which is the number of training experiments, equal to 100, less one experiment, and reserve the last experiment for validation. The Neural State Space Model (NSSM) was used for control purposes by the NL-MPC model. These simulation settings and parameters give a detailed account of how to set up a nonlinear pendulum control system with constraints using the Neural State Space model with NL-MPC. The total numbers of signals is calculated as:

$$\text{Total experiments} = 101 \text{ experiments}$$
$$\text{Total variables} = 4 \text{ states}$$
$$\text{Samples per experiment} = 1000 \text{ samples}$$
$$\text{Total training signals} = 100 \times 4 = 400 \text{ signals}$$
$$\text{Validation signal} = \text{Last Experiment}$$
$$\text{Total samples for training} = 400 \times 1000 = 4,00,000 \text{ samples}$$

Designing an efficient Neural State Space Model (NSSM) for Model Predictive Control (MPC) requires careful consideration of different architectures and hyperparameters. This involves systematically evaluating various designs to identify which gives accurate and efficient predictions for the system. Detailed NSSM architectures and hyperparameters are given in the next section.

5.2 Neural State Space Model Parameters

Finding the best design for the prediction model is possible by experimenting with different topologies. Model generalization from training data to unknown data is influenced by the topology selected. Several topologies are considered and listed in Table 5.1.

Topology	**Layers**	**Neurons**
Shallow	1	64
Deep	3	64
Narrow	1	10
Wide	1	1000

TABLE 5.1: Topologies of NSSM

The prediction model requires the activation functions, number of hidden layers, and hidden nodes in the NSSM, and other hyperparameters to be configured are

S.N.	Parameters	Values
1	Optimizer	Adam
2	Decay Factor	0.900
3	Loss Function	Mean Absolute Error
4	Learning Ratio	0.001
5	Epochs	500, 1000
6	Activation Function	Sigmoid, tanh
7	Input Interpolation	Spline
8	Batch Size	101

TABLE 5.2: Training Parameters for NSSM

listed in Table 5.2. The 'Glorot' initializer is used to set the weights of the neural network, and zeros are set as the biases. Additionally, it is important throughout the learning phase of training epochs. The network is trained using 500 and 1000 training epochs in our NSSM model designs. The ADAM optimizer is used to obtain the ideal weights.

5.3 FBL-based Neural State Space Model

In this section, we explore the results obtained through the application of the Neural State Space technique, delving into a detailed examination of the simulation setup. The evaluation utilizes crucial metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), Relative Absolute Error, and stability time. These metrics collectively contribute to a thorough assessment of the suggested models, revealing insights into their effectiveness in capturing system dynamics and ensuring stability. The comprehensive analysis presented herein aims to offer nuanced perspectives on the performance of the proposed designs, shedding light on their suitability for addressing the intricacies of the underlying system dynamics.

The simulation setup involves carefully calibrated parameters to reflect real-world scenarios as closely as possible, ensuring that the results are both reliable and applicable. Various test cases, including different initial conditions and external disturbances, were used to evaluate the robustness of the models. MSE and MAE provide a measure of the average deviation between the predicted and actual values, offering a clear indication of the model's accuracy. The Relative Absolute Error offers a normalized view, making it easier to compare performance across different scales and conditions. Stability time, a key performance indicator, measures how quickly the system returns to equilibrium after a disturbance, thus reflecting the model's effectiveness in maintaining control and stability.Through rigorous testing and analysis, we identified strengths and potential areas for improvement in the proposed models. For instance, models that excelled in minimizing MSE and MAE demonstrated high precision in predicting the system's behavior under standard conditions. However, some models showed variability in stability time, indicating a need for further refinement to enhance their responsiveness to sudden changes. The insights gained from these evaluations are instrumental in guiding future iterations of model development, focusing on optimizing the balance between accuracy and stability.

5.3.1 Shallow NSSM

The Shallow Neural State Space Model (NSSM) configuration comprises a single layer with 64 neurons, offering simplicity with moderate capacity. This design choice reflects a deliberate consideration of the trade-off between model complexity and computational efficiency. By opting for a shallow architecture, the model maintains a relatively straightforward structure while still possessing sufficient capacity to capture the underlying dynamics of the system.

Furthermore, the choice of a shallow architecture aligns with the goal of achieving a balance between model complexity and interpretability. By keeping the model relatively simple, it becomes easier to interpret its internal workings and understand how input features are transformed into output predictions. This transparency is particularly valuable in domains where explainability and insight into the model's decision-making process are essential.Overall, the configuration of the Shallow NSSM with a single layer of 64 neurons reflects a thoughtful approach to model design, aiming to strike a pragmatic balance between complexity, capacity, and interpretability. This configuration serves as a practical compromise, offering a model that is both capable of capturing the system's dynamics and tractable for analysis and interpretation. The performance parameters for Shallow NSSM are shown in table 5.3.

	X1	X2	X3	X4	Total
MSE	0.050	0.285	6.186	2.399	2.23
MAE	0.128	0.206	1.781	1.010	0.781
Relative Absolute Error	1.1775	1.0054	1.0264	-0.1024	2.04
Stability time	9	9	9	9	9

TABLE 5.3: FBL-based Shallow NSSM Results

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [0, 1, 1, 0]$ is shown in Figure 5.1. All the states and phase portraits are shown in Figure 5.2

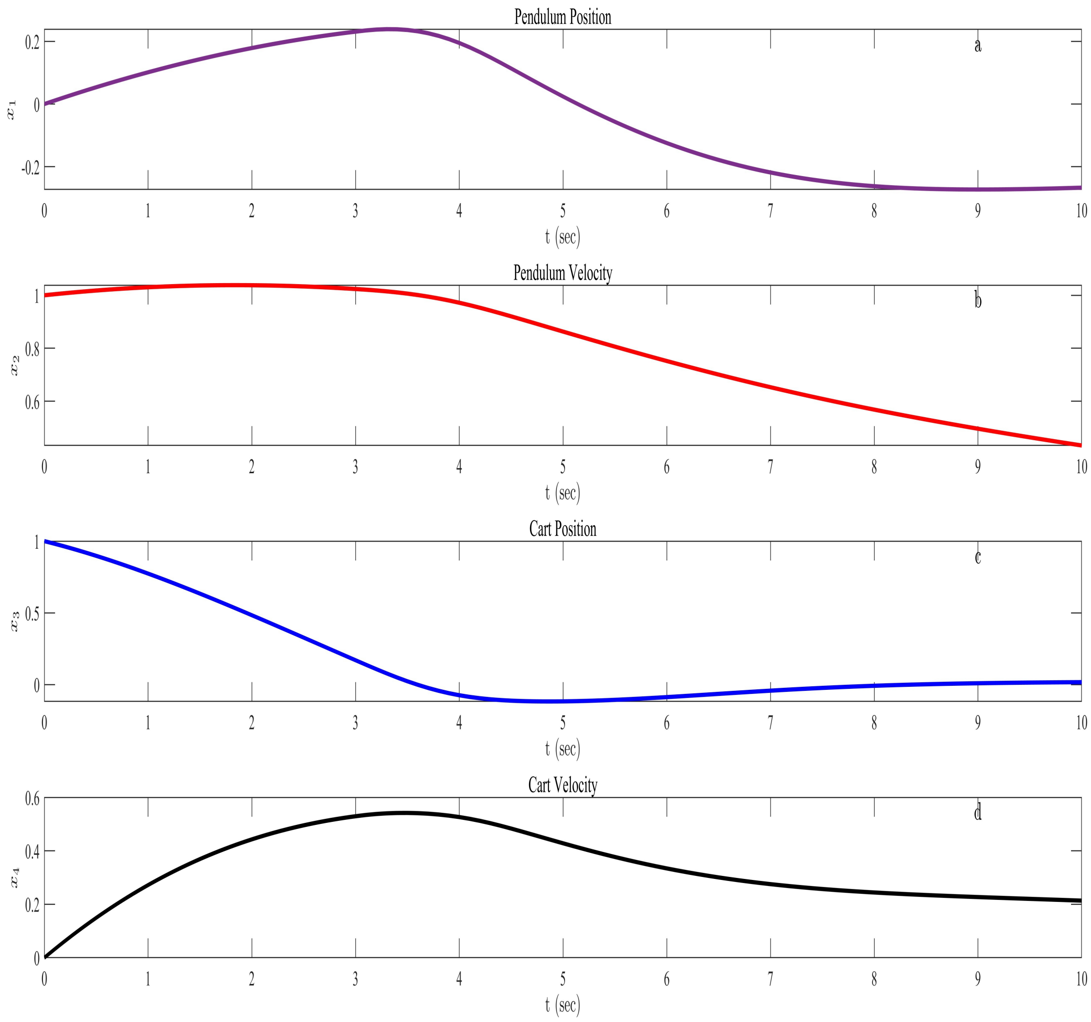

FIGURE 5.1: System states for IC = [0, 1, 1, 0]

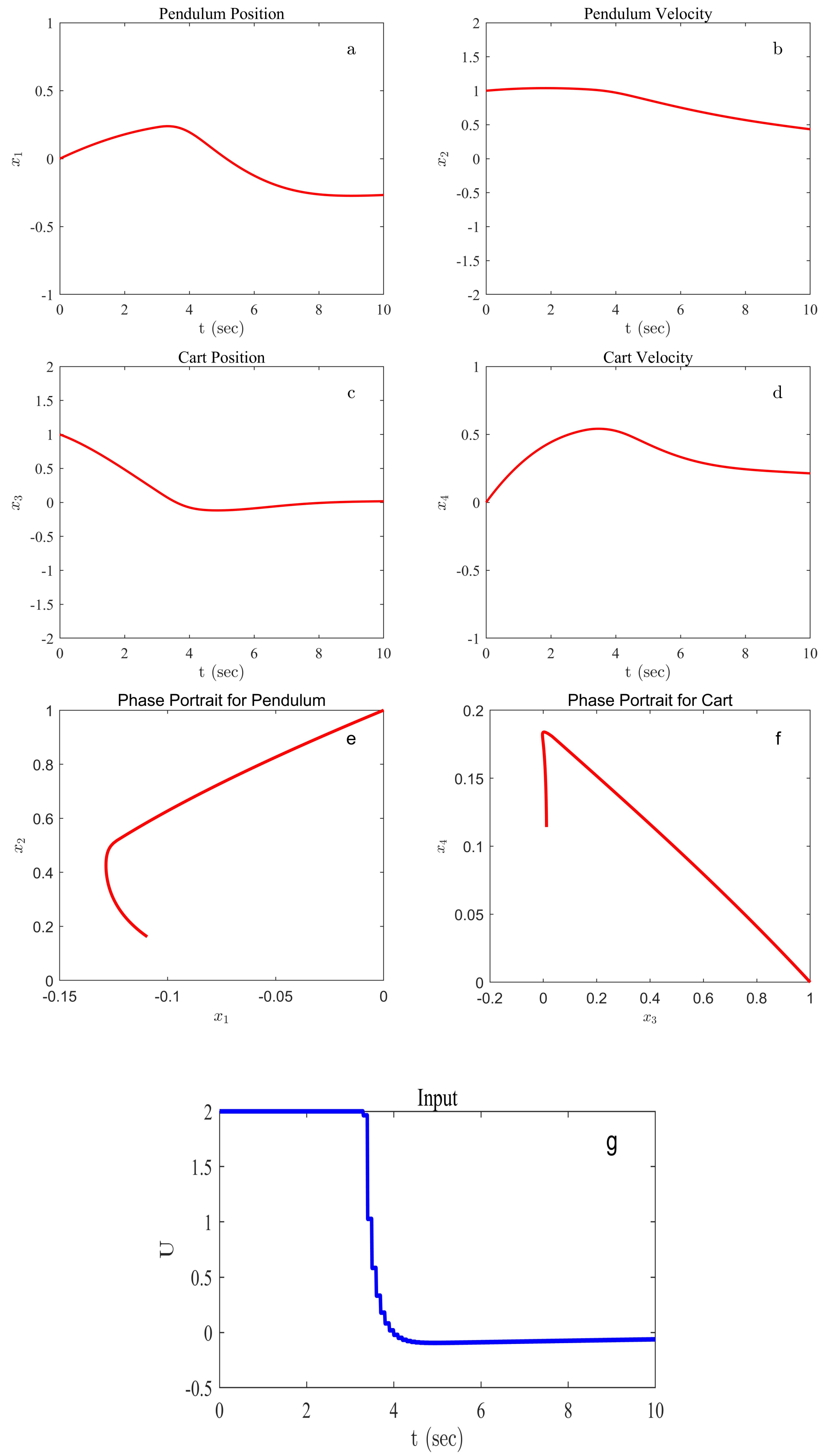

FIGURE 5.2: FBL-based Shallow Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.3.2 Deep NSSM

The Deep Neural State Space Model (NSSM) stands out for its multi-layered architecture comprising three layers, each consisting of 64 neurons are shown in table 5.4. The Narrow Network's characterization as a single-layer architecture with only 10 neurons. This tailored design is engineered to effectively capture complex, hierarchical features inherent in the data.

TABLE 5.4: FBL-based Deep NSSM Results

	X1	X2	X3	X4	Total
MSE	0.032	0.287	6.982	2.464	2.44
MAE	0.130	0.211	1.886, 1.050	0.819	
Relative Absolute Error	1.080	1.00	0.98	-1.03	2.01
Stability time: 9	9	4.5	7	7.375	

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [0, 2, 1, 0]$ is shown in Figure 5.3. All the states and phase portraits are shown in Figure 5.4

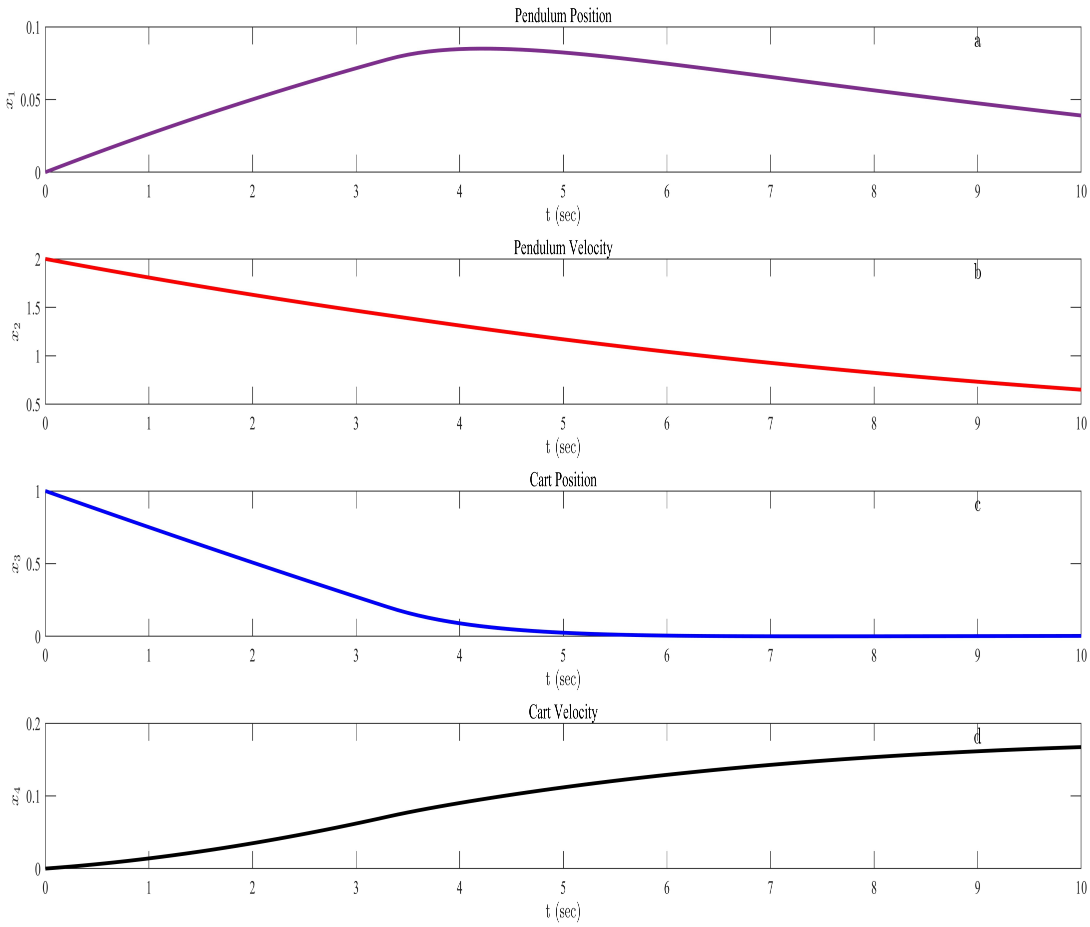

FIGURE 5.3: System states for IC = [0, 2, 1, 0]

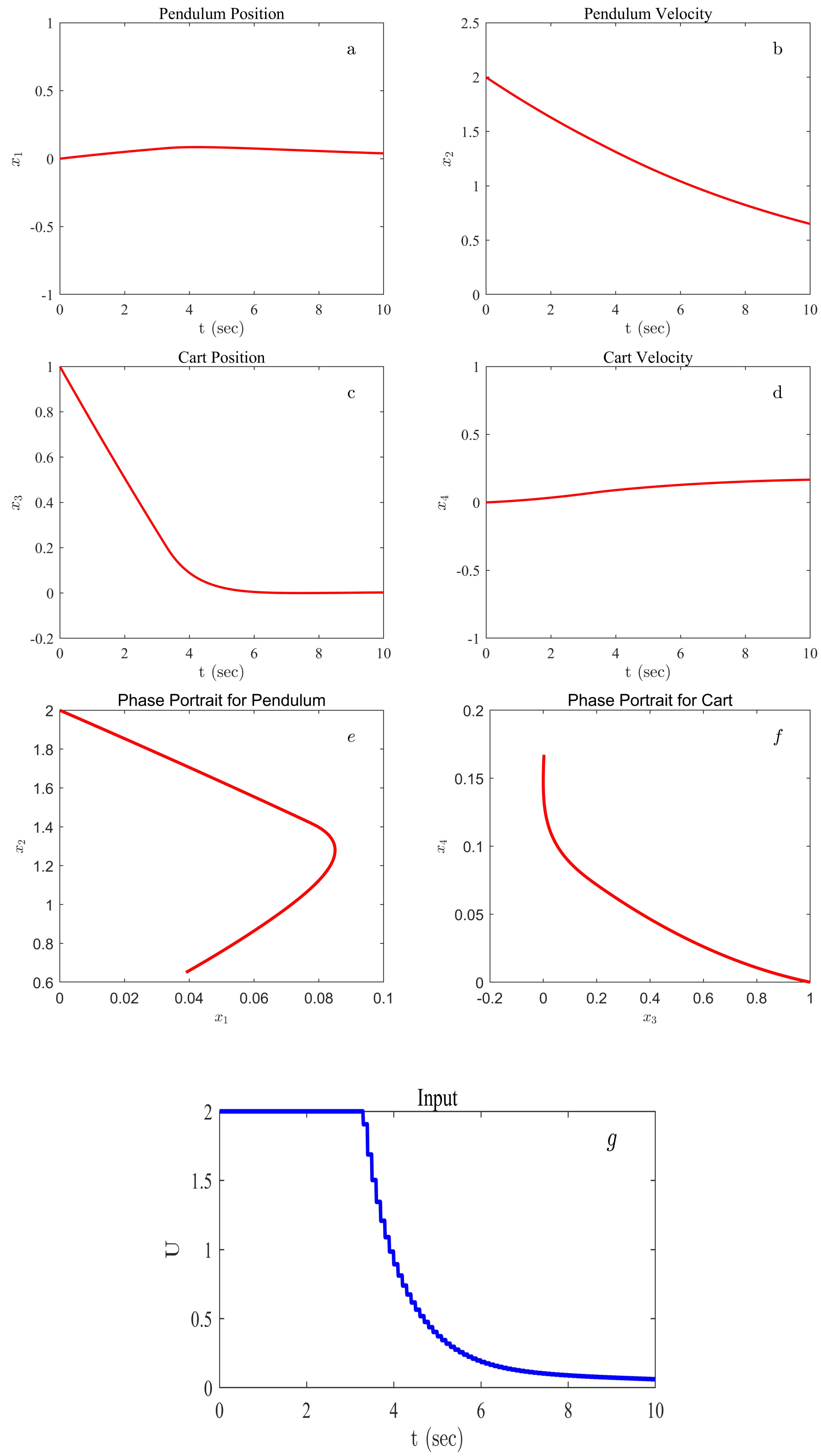

FIGURE 5.4: FBL-based Deep Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.3.3 Narrow NSSM

The Narrow Network is characterized by a single layer comprising only 10 neurons. Table 5.5 shows the performance metrics for the FBL-based Narrow NSSM. In the

TABLE 5.5: FBL based Narrow NSSM Results

	X1	X2	X3	X4	Total
MSE	0.033	0.280	9.278	2.852	3.11
MAE	0.111	0.199	2.192	1.201	0.92
Relative Absolute Error	1.0945	1.0002	1.5164	-0.87	1.6
Stability time	-	-	-	-	-

provided context, the presence of '-' indicates that stability has not been achieved. Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [1, 1, 1, 0]$ is shown in Figure 5.5. All the states and phase portraits are shown in Figure 5.6

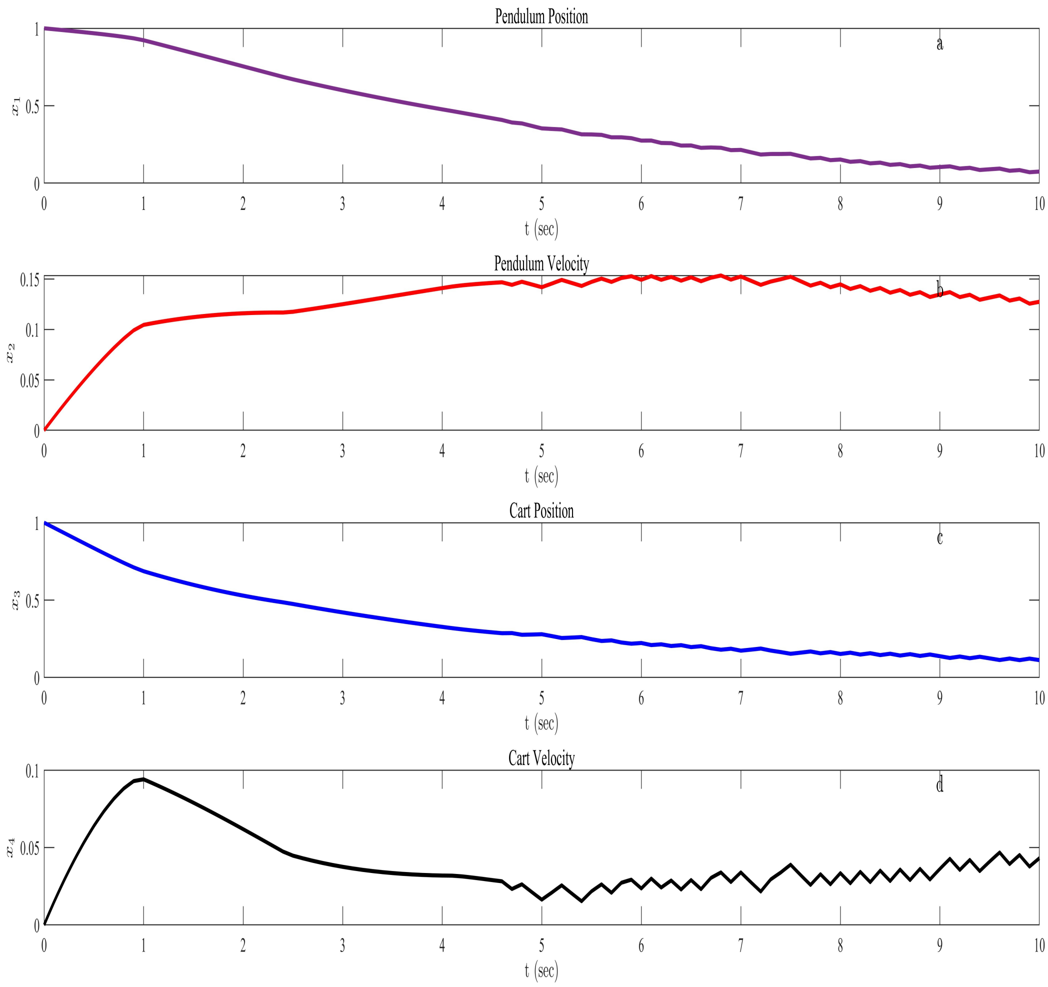

FIGURE 5.5: System states for IC = [1, 1, 1, 0]

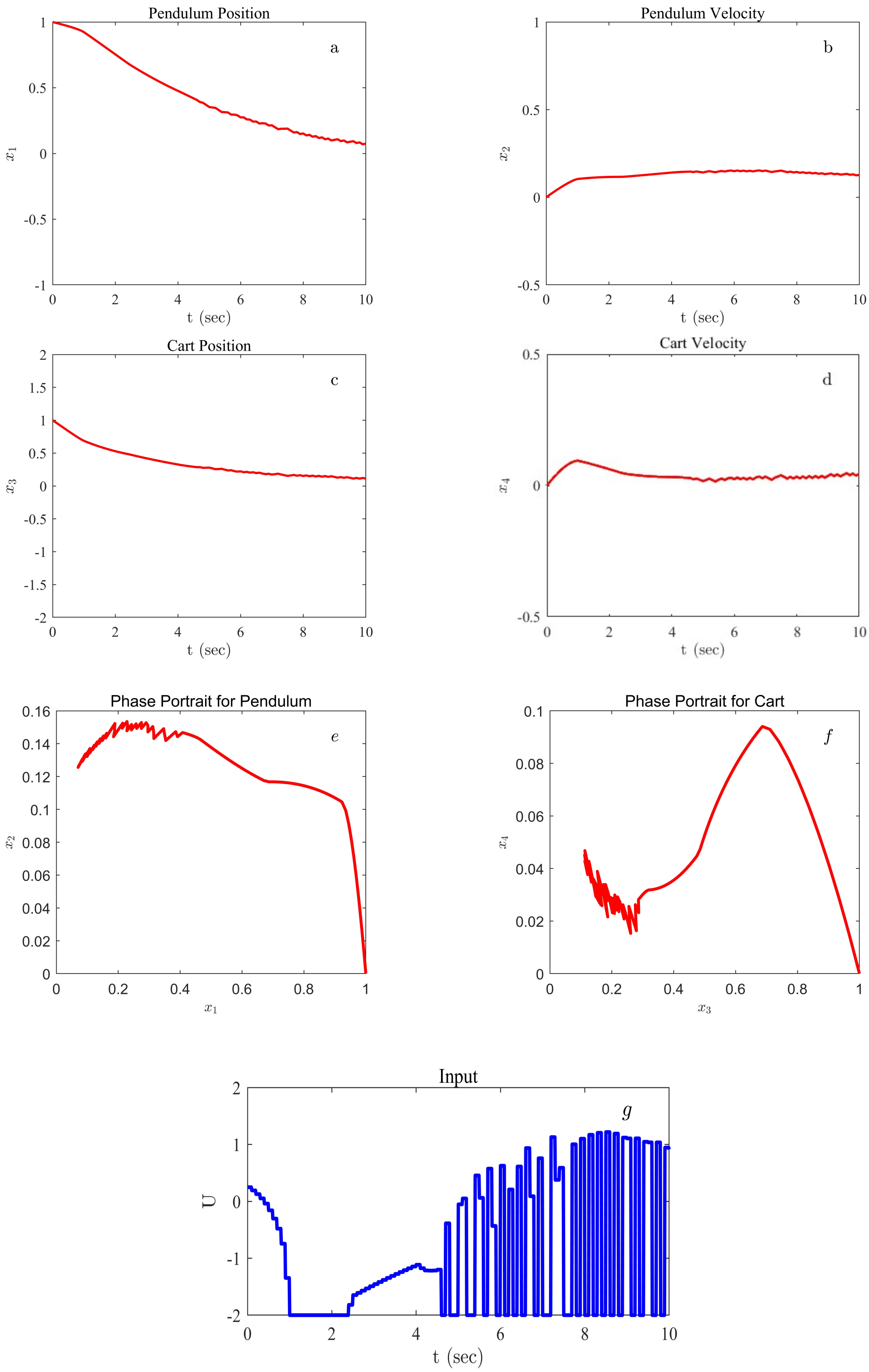

FIGURE 5.6: FBL-based Narrow Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.3.4 Wide NSSM

The Wide Network features a single layer with an extensive 1000 neurons. This architecture aims to capture a broad spectrum of features and relationships within the data.

Table 5.6 shows the performance metrics for the FBL-based Narrow NSSM.

TABLE 5.6: FBL based Wide NSSM Results

	X1	X2	X3	X4	Total
MSE	0.058	0.293	6.200	2.299	2.21
MAE	0.185	0.317	1.782	1.019	0.82
Relative Absolute Error	1.1384	1.0129	1.1489	-0.695	0.88
Stability time	8	4	9	9	7.5

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [3, 0, 1, 0]$ is shown in Figure 5.7. All the states and phase portraits are shown in Figure 5.8

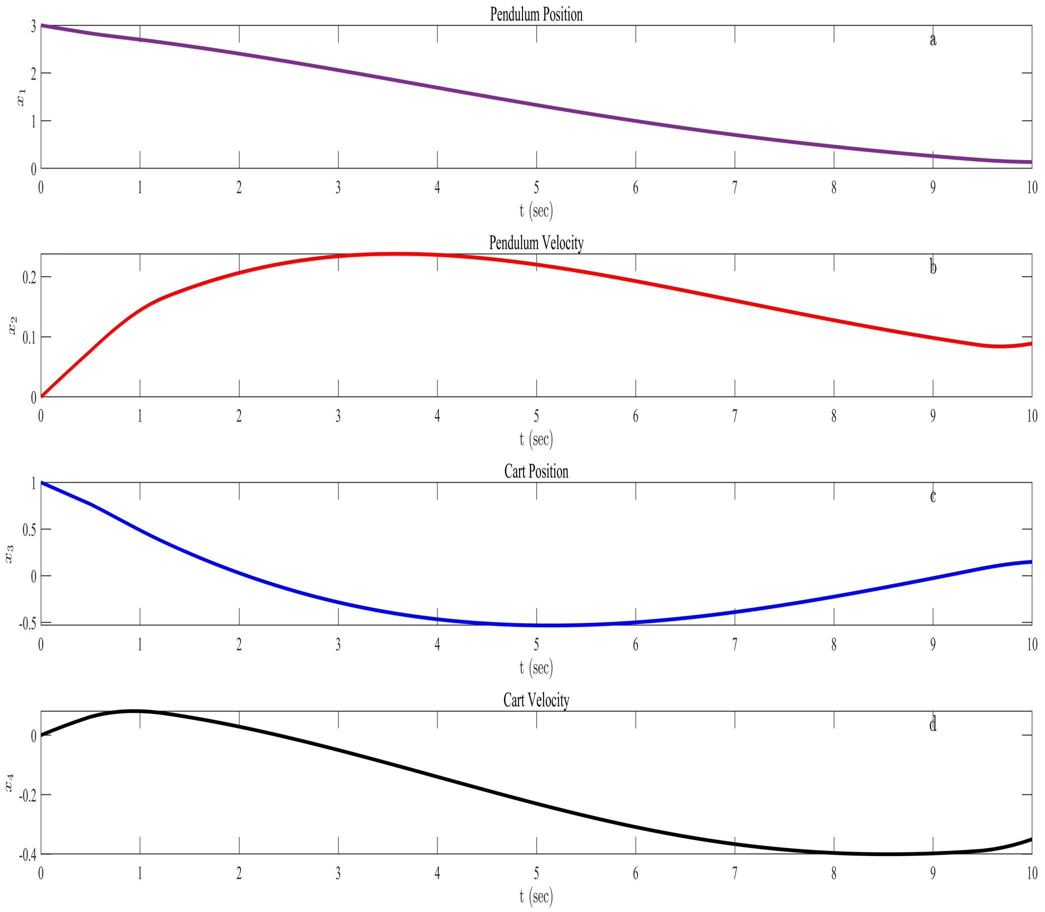

FIGURE 5.7: System states for IC = [3, 0, 1, 0]

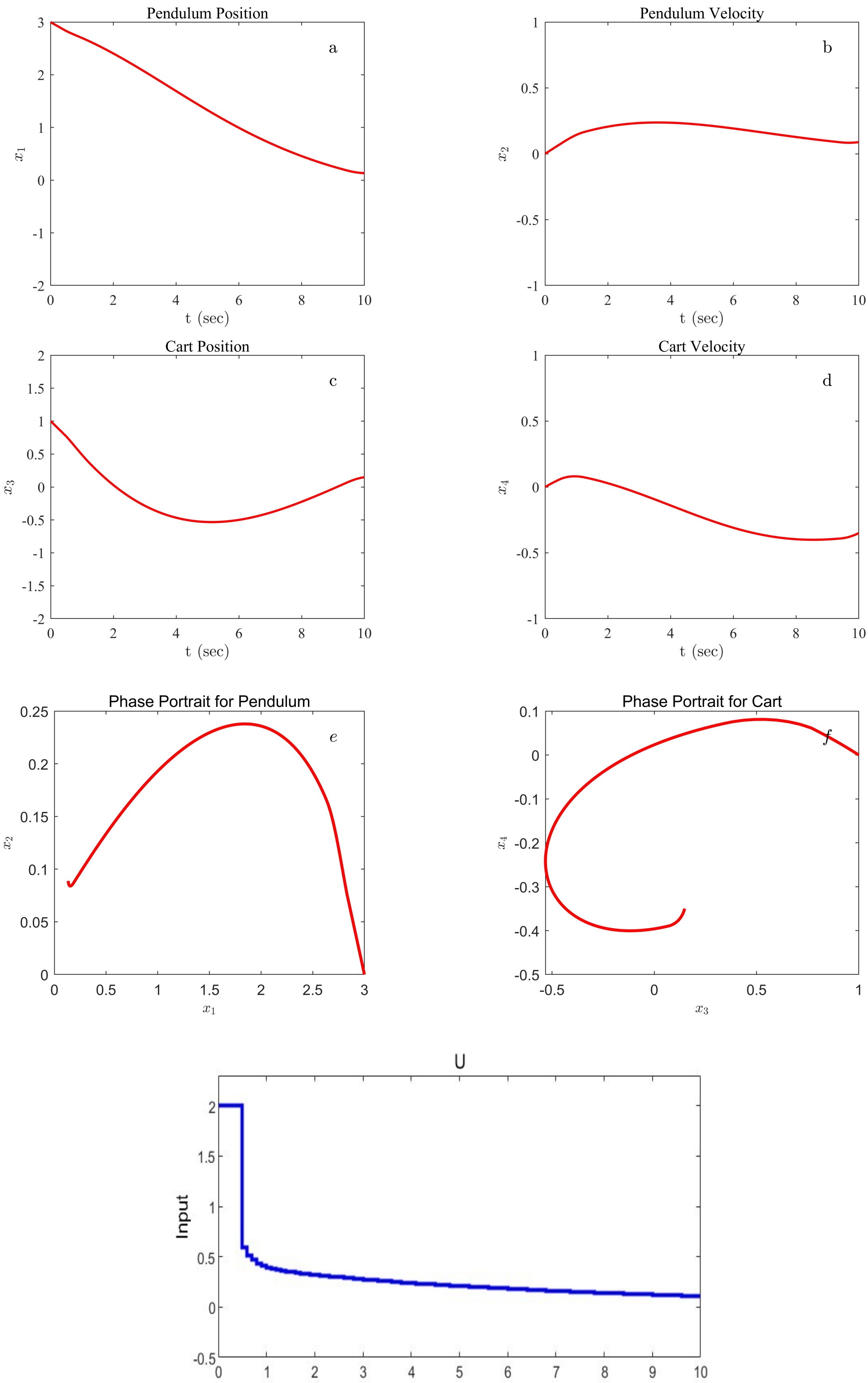

FIGURE 5.8: FBL-based Wide Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.4 LQR-based Neural State Space Model

This section discusses the results of simulations using the LQR-based NSSM.

5.4.1 Shallow NSSM

The Shallow Neural State Space Model (NSSM) is characterized by a simplified architecture, consisting of a single layer with 64 neurons. Table 5.7 shows the performance metrics for the LQR-based Shallow NSSM.

TABLE 5.7: LQR-based Shallow NSSM Results

	X1	X2	X3	X4	Total
MSE	0.033	0.052	0.062	2.848	0.99
MAE	0.138	0.099	0.119	0.480	0.209
Relative Absolute Error	1.4713	1.0106	0.7599	-0.303	1.204
Stability time	3.8	4	8	6	5.45

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [2, 1, 2, 3]$ is shown in Figure 5.9. All the states and phase portraits are shown in Figure 5.10

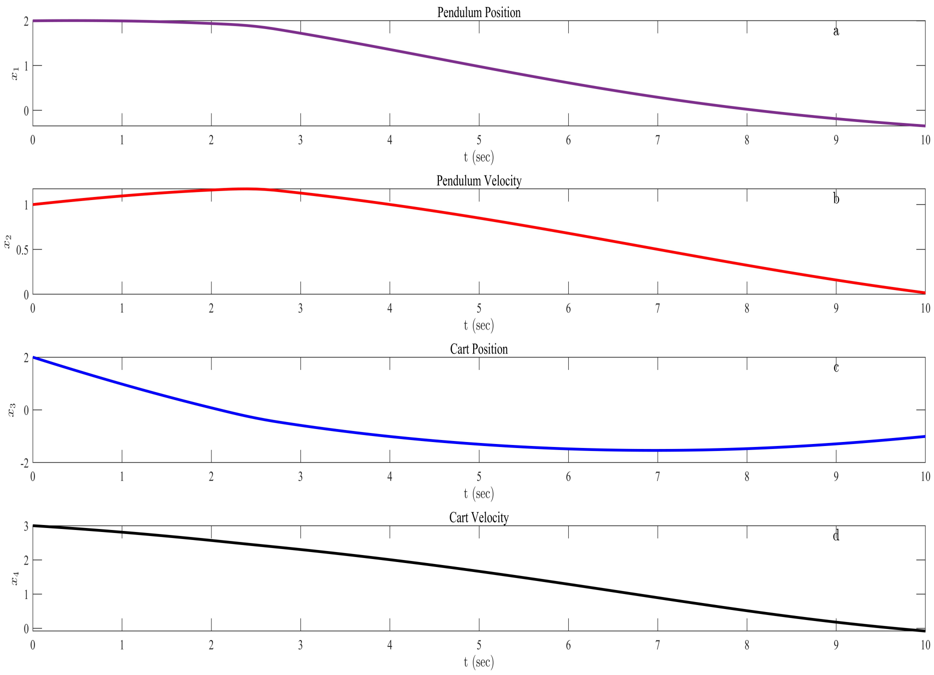

FIGURE 5.9: System states for IC = [2, 1, 2, 3]

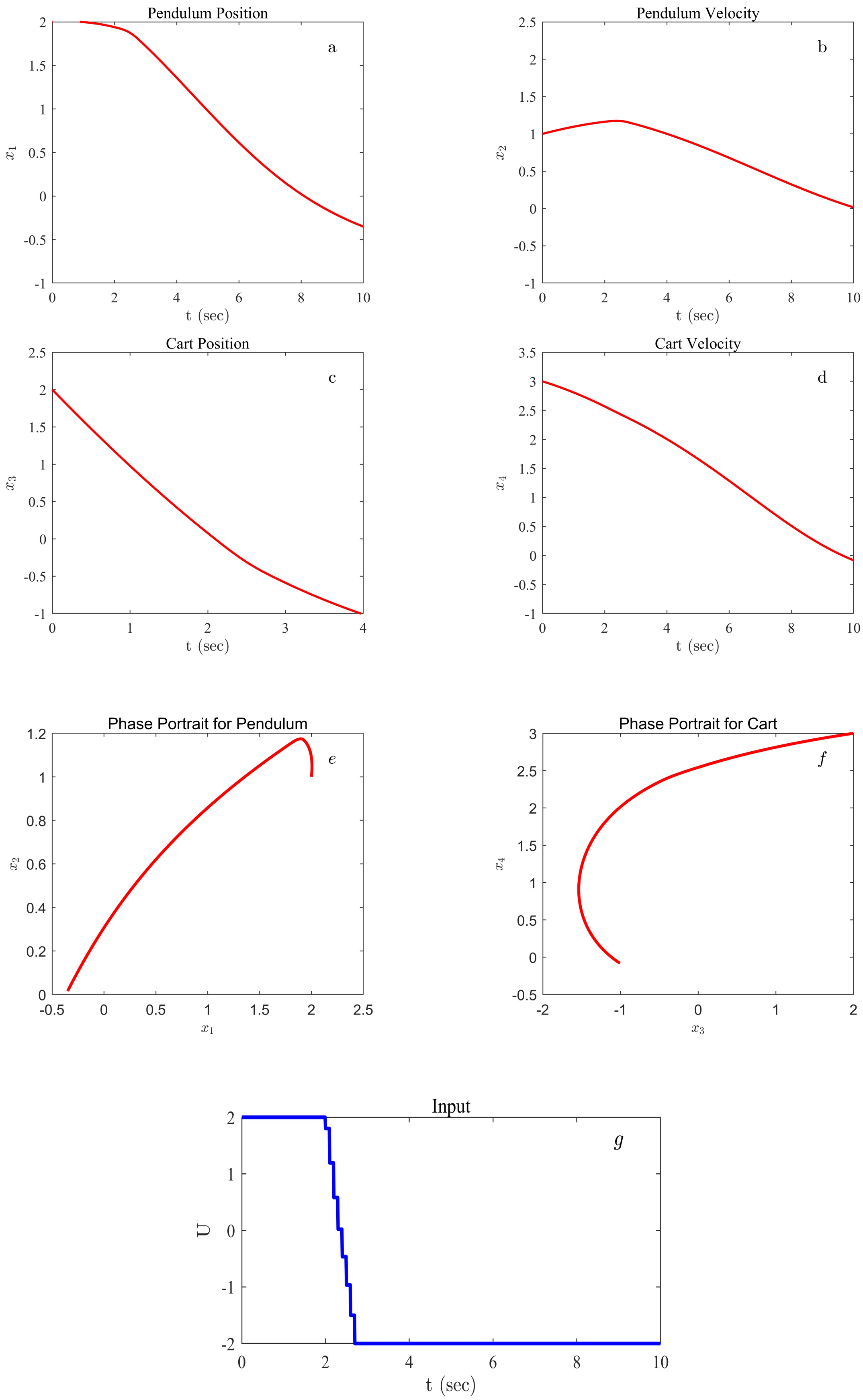

FIGURE 5.10: LQR-based Shallow Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.4.2 Deep NSSM

The Deep Neural State Space Model (NSSM) is distinguished by its multi-layered architecture, featuring three layers with 64 neurons each. This design is specifically crafted to capture intricate and hierarchical features within the data. Table 5.8 shows the performance metrics for the LQR-based Deep NSSM.

TABLE 5.8: LQR-based Deep NSSM Results

	X1	X2	X3	X4	Total
MSE	0.011	0.051	0.218	2.964	0.811
MAE	0.075	0.054	0.202	0.526	0.214
Relative Absolute Error	1.5136	1.0276	0.7387	-0.208	1.14
Stability time	2.5	5	3.5	3.5	3.6

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [2, 1, 2, 3]$ is shown in Figure 5.11. All the states and phase portraits are shown in Figure 5.12

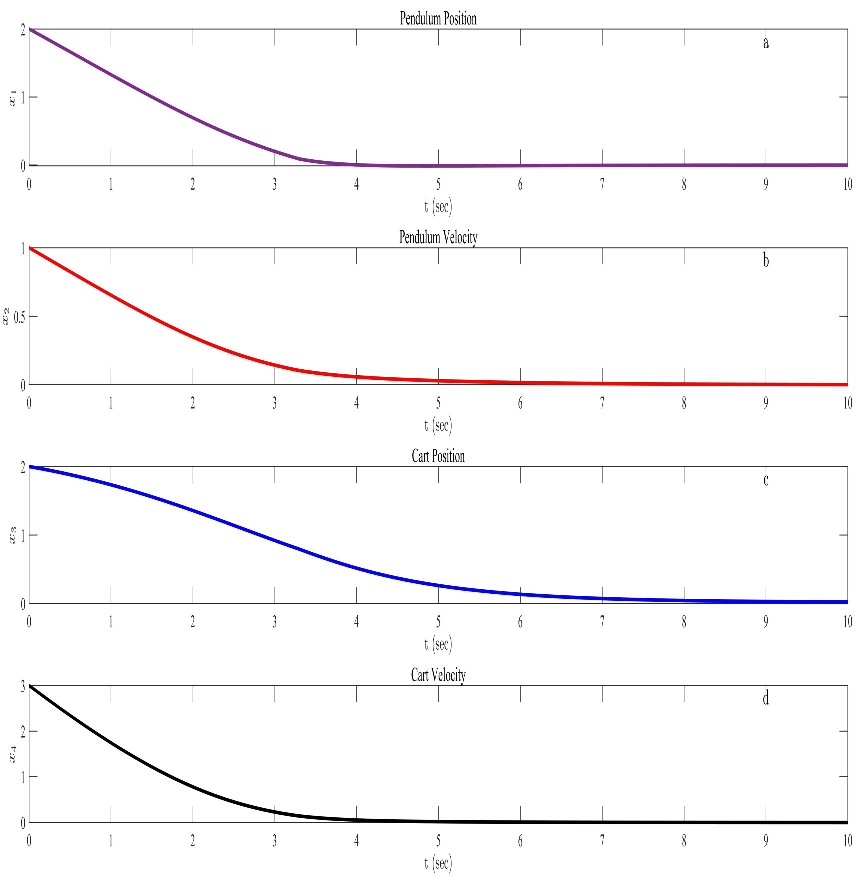

FIGURE 5.11: System states for IC = [2, 1, 2, 3]

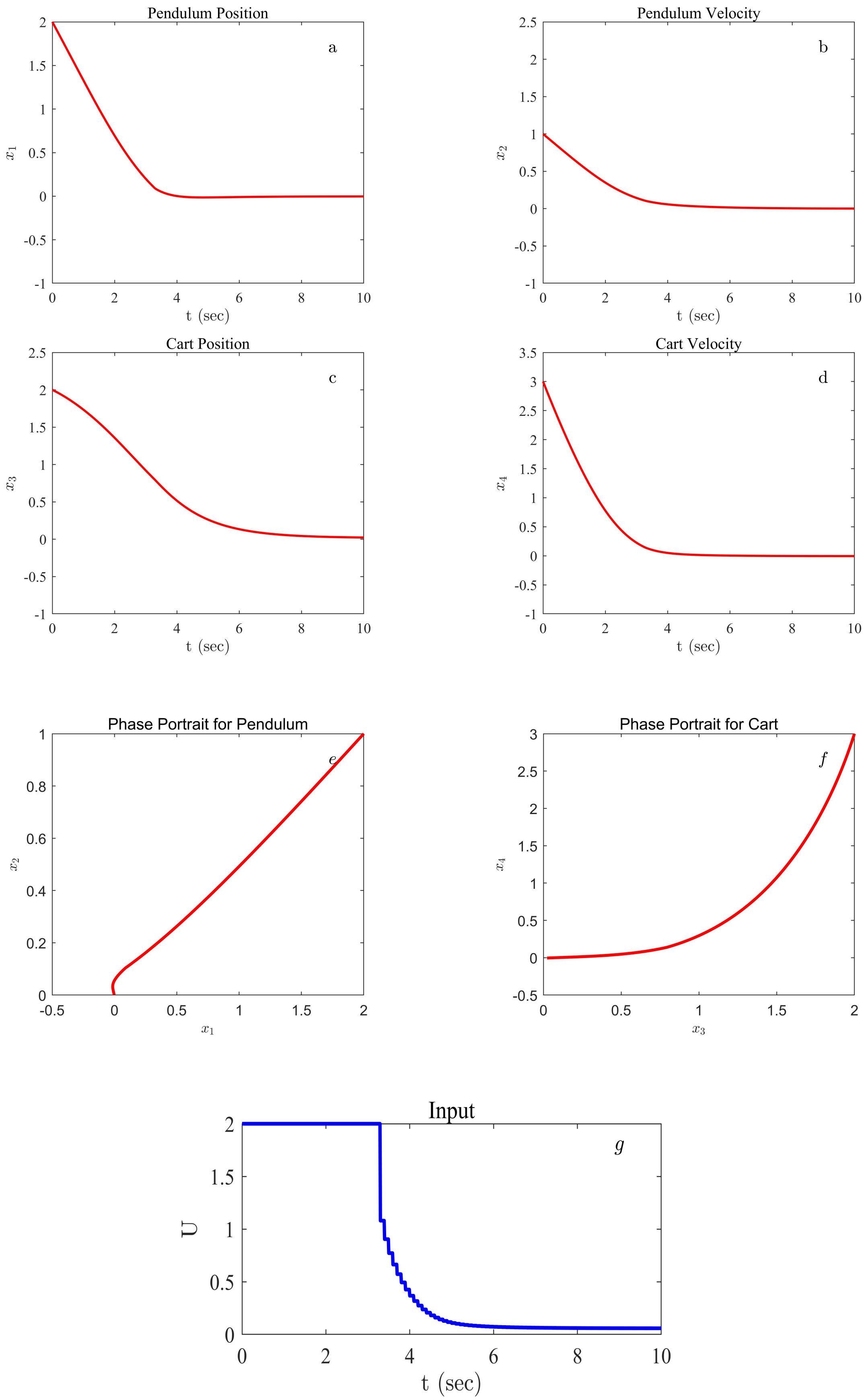

FIGURE 5.12: LQR-based Deep Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.4.3 Narrow NSSM

The Narrow LQR NSSM is characterized by a streamlined configuration, incorporating a single layer with a limited set of neurons. This simplified architecture is tailored for scenarios where efficiency and simplicity are prioritized.

Table 5.9 shows the performance metrics for the LQR-based Deep NSSM.

TABLE 5.9: LQR based Narrow NSSM Results

	X1	X2	X3	X4	Total
MSE	0.01	0.04	0.211	2.98	0.81
MAE	0.08	0.05	0.222	0.54	0.22
Relative Absolute Error	1.54	1.02	0.72	0.04	0.83
Stability time	6	4	7.5	7	6.12

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [2, 1, 2, 3]$ is shown in Figure 5.13. All the states and phase portraits are shown in Figure 5.14

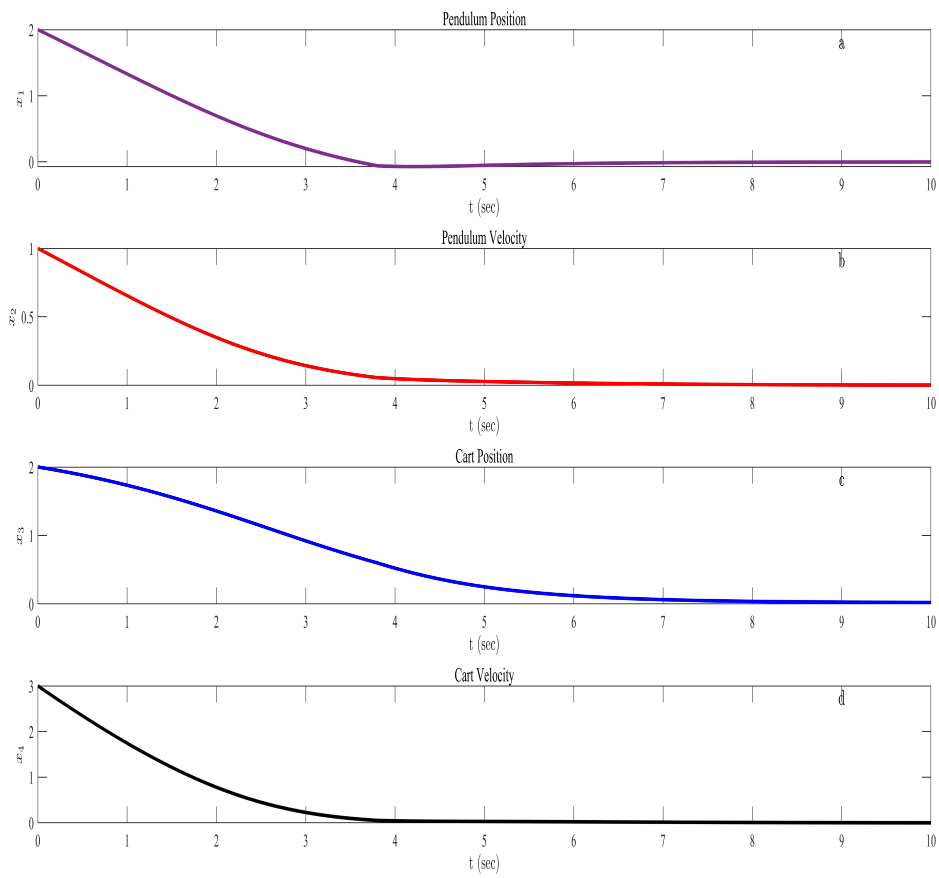

FIGURE 5.13: System states for IC = [2, 1, 2, 3]

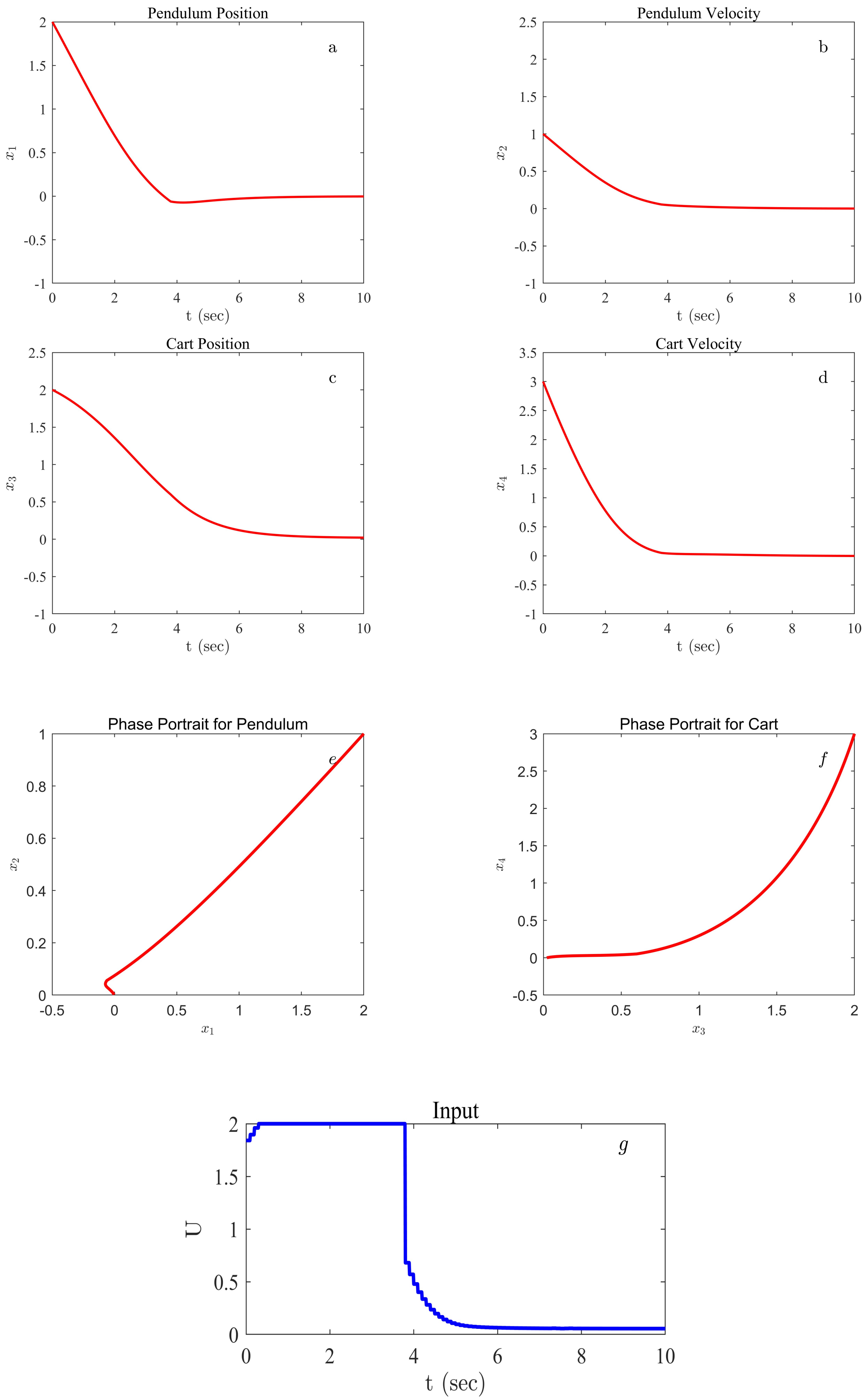

FIGURE 5.14: LQR-based Narrow Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.4.4 Wide NSSM

The Wide Neural State Space Model (NSSM) is characterized by a single layer with an extensive configuration of 1000 neurons. This architecture is designed to capture a diverse and intricate range of features and relationships within the data.

Table 5.10 shows the performance metrics for the LQR-based Wide NSSM.

TABLE 5.10: LQR based Wide NSSM Results

	X1	X2	X3	X4	Total
MSE	0.073	0.038	0.907	3.716	1.18
MAE	0.204	0.066	0.492	0.902	0.411
Relative Absolute Error	1.3377	1.0392	1.0525	-0.073	0.030
Stability time	6.5	3.5	4.5	3	4.37

Output response for the initial condition $[\theta_0, \dot{\theta}_0, x_0, \dot{x}_0] = [1, 3, 0, 2]$ is shown in Figure 5.15. All the states and phase portraits are shown in Figure 5.16

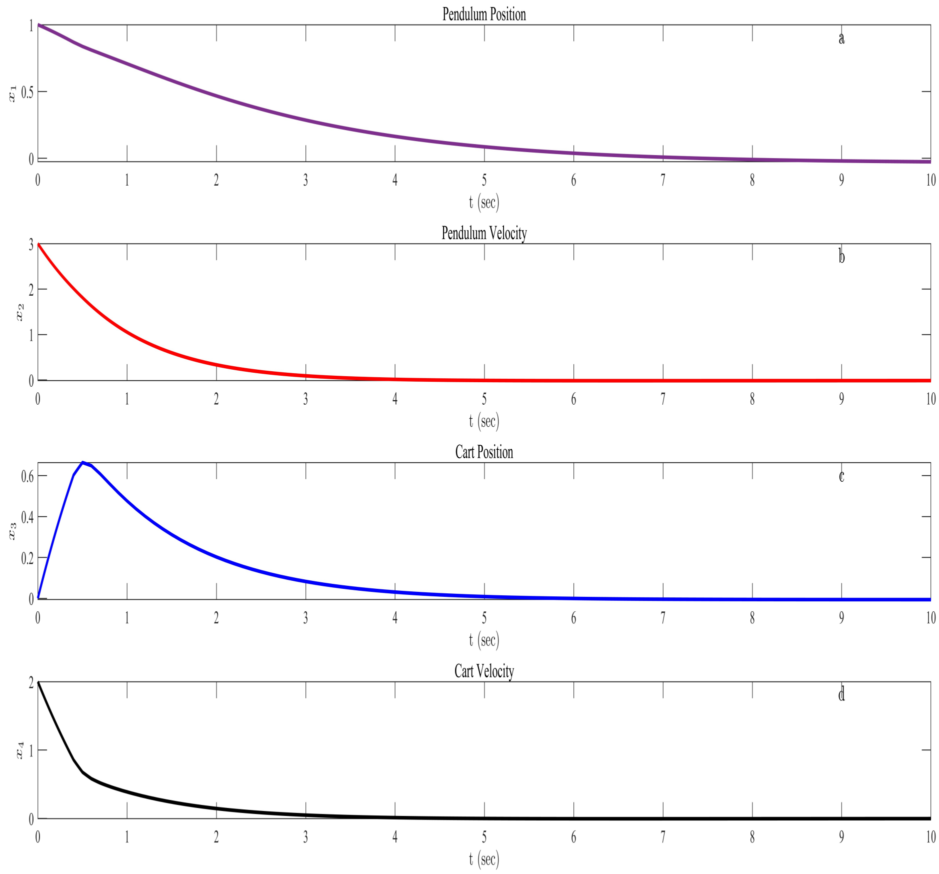

FIGURE 5.15: System states for IC = [1, 3, 0, 2]

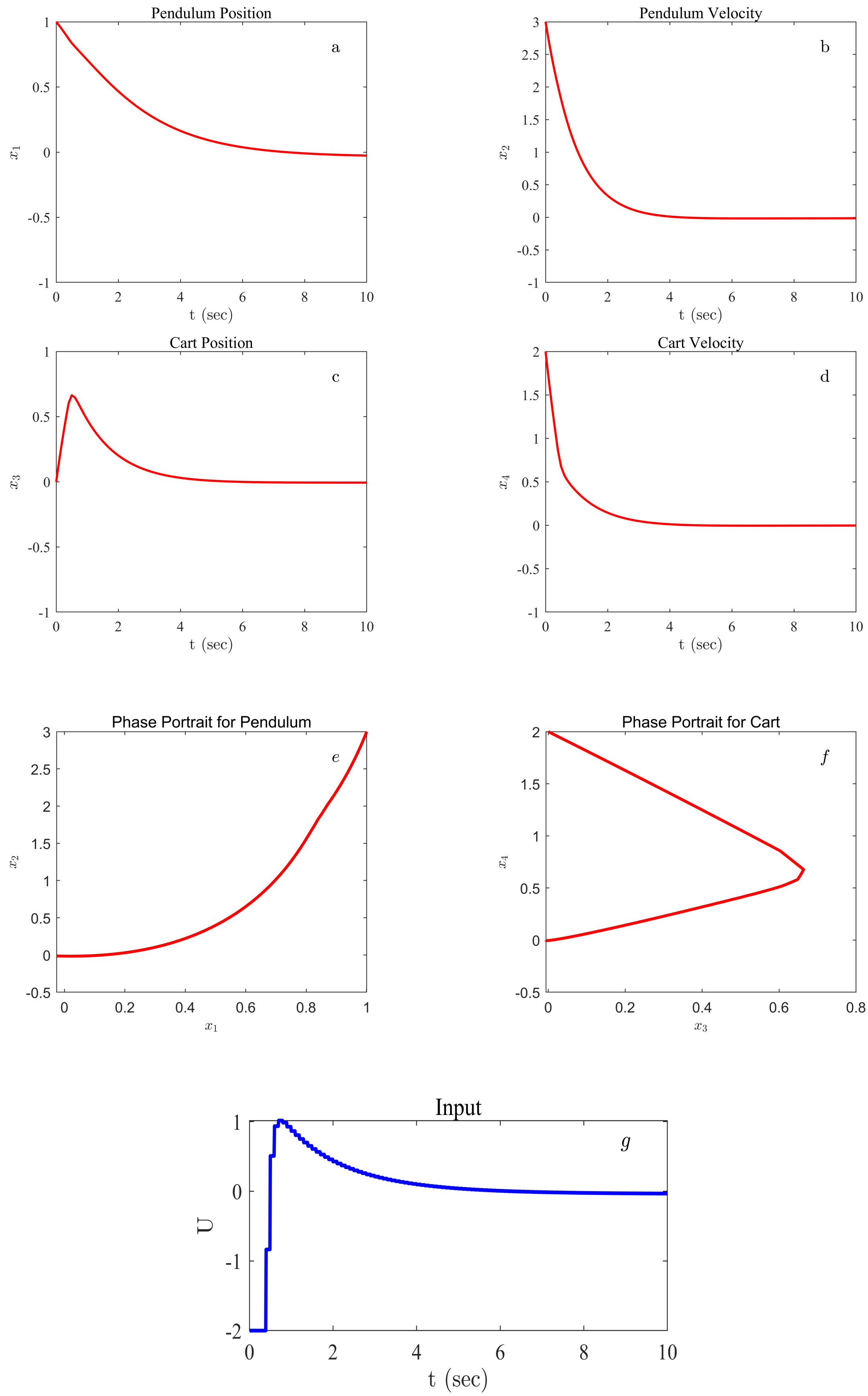

FIGURE 5.16: LQR-based Wide Neural State Space Model Response (a) Pendulum Position (b) Pendulum Velocity (c) Cart Position (d) Cart Velocity (e) Pendulum Phase Portrait (f) Cart Phase Portrait (g) Control Input

5.5 Discussion and Comparison

5.5.1 Comparative Analysis of FBL NSSM

A thorough comparative study of Feedback Linearization Neural State Space Models (FBL NSSM) is included in this section. The relative accuracy, stability, and adaptability of several models including shallow, deep, narrow, and wide are considered.

5.5.1.1 MSE and MAE

Wide and shallow models often have lower MSE and MAE, indicating better performance. Increased MSE and MAE values suggest that the narrow model has trouble capturing system dynamics.

5.5.1.2 Relative Absolute Error

The deep model displays a relatively balanced performance with lower relative absolute error, indicating robustness in capturing system dynamics. In contrast, the shallow, wide, and narrow models exhibit differences in relative absolute error, suggesting varying degrees of adaptability and effectiveness in modeling the underlying dynamics.

5.5.1.3 Stability Time

In the evaluation, both deep and shallow models demonstrate consistent stability times, indicating reliable performance in maintaining system stability. However, the narrow model exhibits oscillations and lacks stability. In contrast, the stability times vary across different states in the wide model, suggesting nuanced dynamics that may require further investigation.

In conclusion, the choice of model depends on the specific requirements of the control system, considering factors such as stability, accuracy, and adaptability to system dynamics. The shallow and wide models generally show better performance across the metrics considered. The narrow model, despite exhibiting higher error metrics, might still have practical utility depending on the specific application and trade-offs between accuracy and stability. The deep model demonstrates competitive accuracy with reduced MAE and MSE but there are stability trade-offs, as seen by the differences in stability times between states.

5.5.2 Comparative Analysis of LQR NSSM

This section conducts a detailed comparative analysis of LQR-based Neural State Space Models (LQR NSSM). Evaluating various model topologies, including shallow, deep, narrow, and wide architectures, provides insights into their relative performance. Metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), relative absolute error, and stability times across different states are examined to discern the strengths and limitations of each LQR NSSM model.

5.5.2.1 MSE and MAE

Both shallow and deep models demonstrate lower Mean Squared Error (MSE) and Mean Absolute Error (MAE), suggesting higher accuracy in predicting system behavior. Surprisingly, the narrow model also exhibits competitive accuracy with lower error metrics despite its minimalist architecture. However, the wide model displays higher error metrics, particularly in MSE, indicating potential challenges in accurately capturing the system's dynamics with its extensive architecture.

5.5.2.2 Relative Absolute Error

The narrow model stands out with the lowest relative absolute error, indicating enhanced adaptability and precision in capturing system dynamics. In contrast, the shallow, deep, and wide models display differing levels of relative absolute error across various states, suggesting variations in their ability to effectively model the complexities of the system. These results underscore the importance of considering the trade-offs between model complexity and adaptability when designing neural network architectures for dynamic system modeling and control.

5.5.2.3 Stability Time

Shallow and deep models show consistent stability times across states. The narrow model exhibits stability with varying times. The wide model shows variations in stability times, with higher values for X1 and X3.

In summary, the choice of LQR NSSM model depends on the specific control system requirements, with considerations for accuracy, stability, and adaptability. The narrow model stands out for its lower relative absolute error, indicating better adaptability, while shallow and deep models showcase competitive accuracy and stability. The wide model, despite higher error metrics, may still find utility depending on the specific application requirements.

5.5.3 Comparison of FBL & LQR NSSMs

5.5.3.1 Shallow Models

The FBL NSSM demonstrates overall stability with a total MSE of 2.23 and maintains stable stability times across different states. In comparison, the LQR NSSM showcases competitive performance with a lower total MSE of 0.99 and consistent stability times. These results suggest that both models are effective in capturing the system dynamics, with the LQR NSSM exhibiting slightly better accuracy and stability.

5.5.3.2 Deep Models

The FBL NSSM demonstrates competitive accuracy with a total MSE of 2.44, showcasing its capability to approximate system dynamics effectively. However, it exhibits variations in stability times across simulations. In contrast, the LQR NSSM displays a lower MSE of 0.811 and maintains consistent stability times, highlighting superior accuracy and stability in capturing the system's behavior. These findings underscore the importance of considering both accuracy and stability when evaluating different modeling approaches for dynamic systems.

5.5.3.3 Narrow Models

FBL NSSM experiences instability, indicated by unstable behavior and oscillations. LQR NSSM performs well with a total MSE of 0.81, demonstrating stability and accuracy.

5.5.3.4 Wide Models

FBL NSSM shows stability with variations in stability times and a total MSE of 2.21. LQR NSSM presents a lower MSE of 1.18 and varying stability times, suggesting a trade-off between accuracy and stability.

FBL NSSM demonstrates flexibility with varying model performances across different topologies. LQR NSSM shows consistent stability and competitive accuracy across various model architectures.
The choice between FBL NSSM and LQR NSSM depends on specific control system requirements. In our case, FBL NSSM offers flexibility with trade-offs between accuracy and stability, while LQR NSSM presents consistent stability and competitive accuracy.

5.5.4 Comparison of LQR NN With Other Techniques

In our study, we employed controllers to compare their response with the Neural Network Controller, aiming to identify linear behavior while enhancing accuracy and response time through MATLAB simulation. Our observations indicate that the proposed design outperforms other controllers, demonstrating improved performance in terms of accuracy and response time. This finding underscores the effectiveness of our approach in achieving superior control outcomes compared to conventional controller designs.

5.5.4.1 Comparison With [56]

The observed performance of the LQR NSSM trained with varying initial conditions, showcasing faster stabilization and return to the equilibrium position (averaging between 2 to 3 seconds), underscores its remarkable control capabilities compared to the alternative technique, Feedforward Neural Network (FFNN), trained with a single initial condition.

This compelling evidence suggests that the LQR NSSM, with its inherent adaptability to diverse initial conditions, outperforms the FFNN in achieving more rapid and stable control responses for the pendulum system. The ability of the LQR NSSM to swiftly adapt to changing conditions not only enhances its performance but also exemplifies its robustness and suitability for real-time control applications.

In Figure 5.17, the phase portraits depict the trajectories of the system dynamics under different control techniques. It is evident from the comparison that the trajectories associated with the LQR NSSM exhibit smoother and more stable behavior compared to those of the alternative technique.

Figure 5.18 further corroborates these findings by showcasing the system response under both control methodologies. The response plots clearly indicate that the LQR NSSM achieves faster stabilization and return to the equilibrium position, demonstrating its superior control capabilities.

These visual representations provide compelling evidence of the effectiveness of the LQR NSSM approach in managing the dynamics of the pendulum system. The smoother trajectories and faster response times exhibited by the LQR NSSM highlight its ability to achieve stable and efficient control, underscoring its superiority over alternative techniques.

Overall, the phase portraits and system response plots presented in Figures 5.17 and 5.18 serve as concrete evidence supporting the conclusion that the LQR NSSM outperforms alternative methods in terms of stability and control effectiveness in the context of the inverted pendulum system.

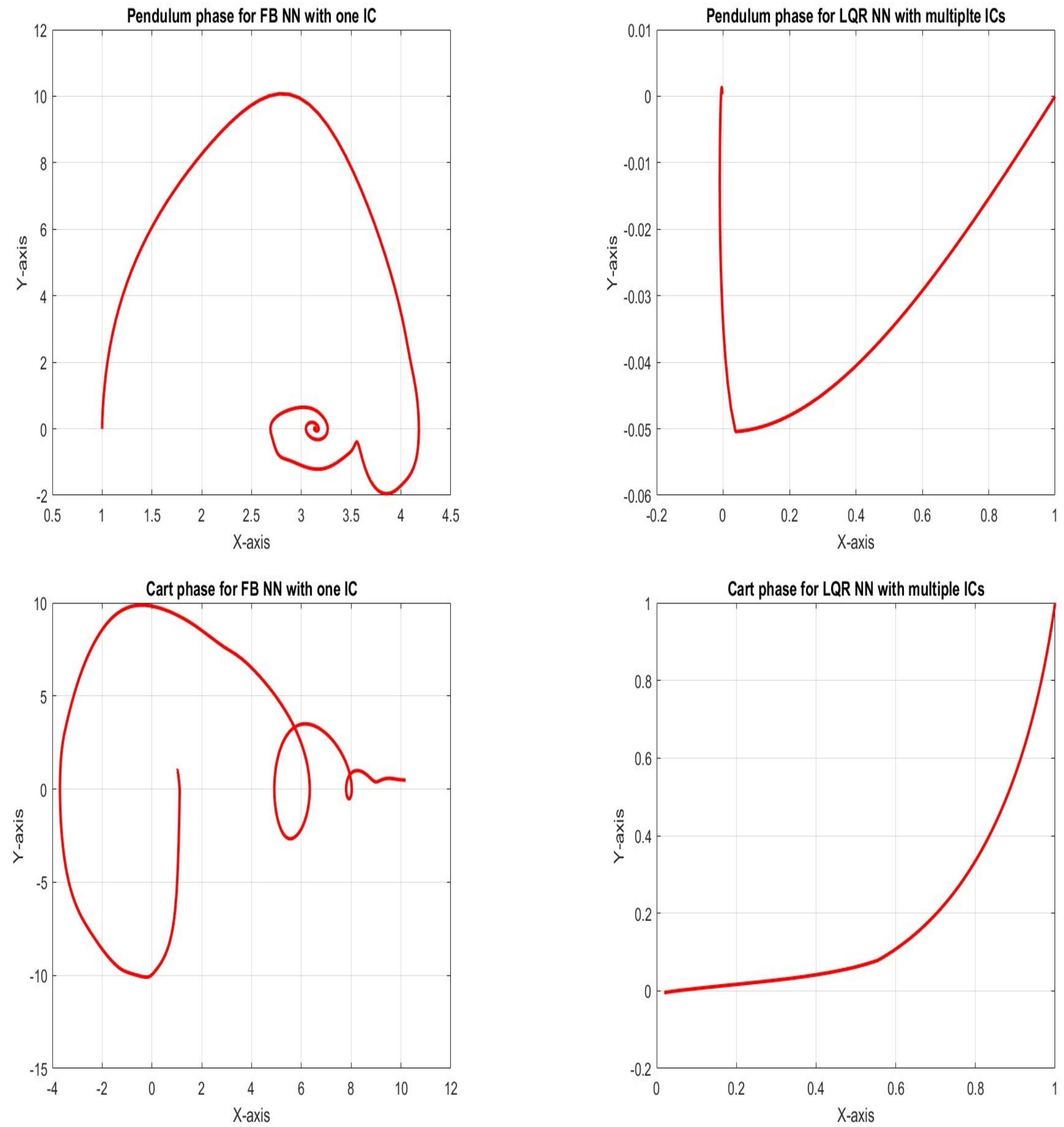

Figure 5.17: Phase portraits for FBNN and LQR

Furthermore, the demonstrated effectiveness of the LQR NSSM in managing diverse initial conditions serves as a testament to its adaptability and versatility. Beyond its application in the pendulum system, this method holds promise for addressing a multitude of control challenges across different domains. By providing stable and efficient control even in the face of uncertainties and variations, the LQR NSSM opens doors to applications in industries ranging from robotics and autonomous vehicles to aerospace and industrial automation. Its robust performance under varying conditions not only enhances system stability but also improves overall operational efficiency, making it a valuable asset in the quest for advanced control solutions in complex real-world scenarios.

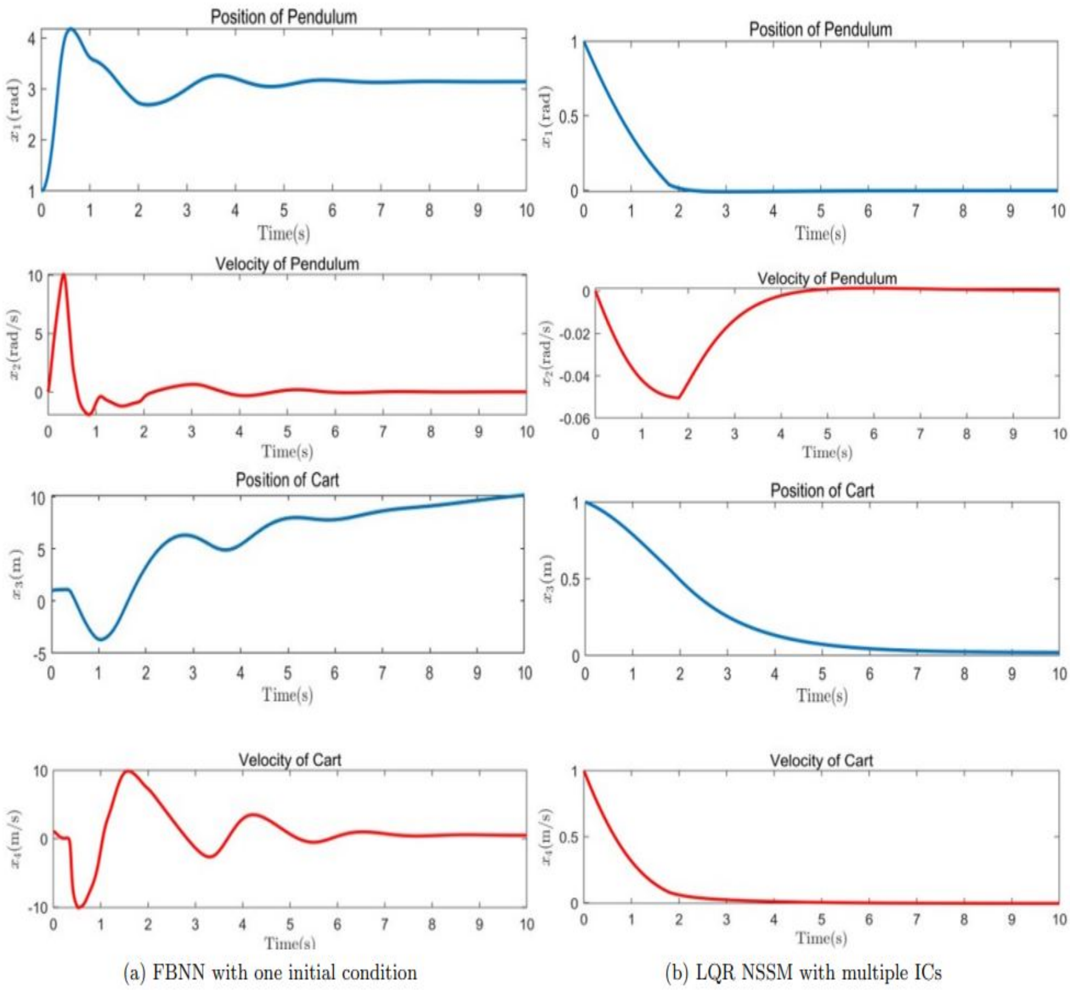

FIGURE 5.18: Pendulum response for FBNN and LQR NSSM

5.5.4.2 Comparison of LQR NN With [93]

The performance evaluation reveals that the LQR NSSM, trained with varying initial conditions, excels in achieving quicker stabilization and returning to the equilibrium position, with an average time of two to three seconds. In contrast, the feedback linearization method demonstrates comparatively slower performance. The performance metrics showcased in Figure 5.19 clearly indicate that the LQR NSSM achieves faster stabilization and smoother response trajectories compared to the FBL technique. This superiority is particularly evident when observing the system's behavior under varying initial conditions. Despite fluctuations and uncertainties, the LQR NSSM consistently demonstrates robust control, ensuring rapid stabilization and return to the desired equilibrium position.

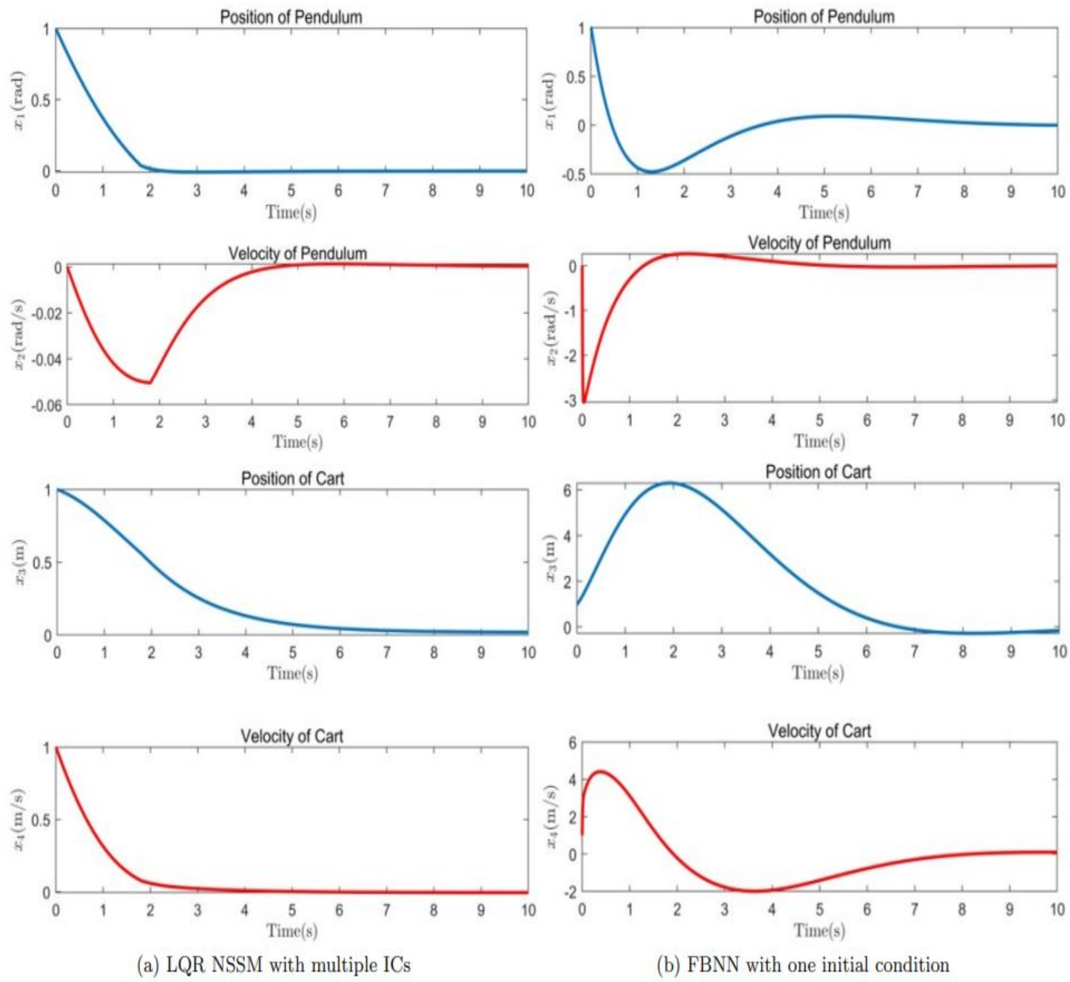

FIGURE 5.19: Pendulum response for LQR NSSM and FBL

The performance evaluation elucidates that the LQR NSSM, trained with diverse initial conditions, exhibits remarkable proficiency in achieving rapid stabilization and swiftly returning to the equilibrium position, typically within an average time frame of two to three seconds. Conversely, the feedback linearization method demonstrates comparatively slower performance across the spectrum of initial conditions. This stark contrast underscores the superior control capabilities of the LQR NSSM, highlighting its effectiveness in adeptly handling a wide range of initial conditions and surpassing the feedback linearization method in terms of both speed and stability. Moreover, these findings highlight the LQR NSSM's robustness across diverse environments, reinforcing its status as a dependable solution for real-world applications. Its adaptability and reliability position it as a frontrunner in the realm of dynamic system control, promising transformative impacts across a spectrum of industries and scenarios.

5.5.5 Comparative Analysis for Varying Initial Conditions

In this section, a comprehensive comparison was conducted by systematically varying initial conditions to assess the performance of each controller. A total of 10 distinct initial conditions were considered, providing a diverse set of scenarios to generalize our evaluation. The specific details of these 10 initial conditions, assessed for each controller, are outlined in the table provided. Each initial condition is represented by a set of values corresponding to the pendulum position (x_1), pendulum velocity (x_2), cart position (x_3), and cart velocity (x_4). These initial conditions cover a wide range of possible starting states for the system, allowing for a comprehensive assessment of each controller's ability to handle various scenarios.

By systematically varying these initial conditions and analyzing the controllers' responses, we gain valuable insights into their performance across different operating conditions and their robustness in real-world applications. The specific details of these 10 initial conditions, assessed for each controller, are outlined in the table below.

TABLE 5.11: Multiple Initial Conditions

S.N.	Initial Conditions	[x_1, x_2, x_3, x_4]
1	IC-1	[2, 3, -2, 2]
2	IC-2	[1, -3, -1, 0]
3	IC-3	[3, 3, -2, 3]
4	IC-4	[3, 0, 2, -2]
5	IC-5	[0, 3, 2, 3]
6	IC-6	[1, -3, 2, 3]
7	IC-7	[1, 2, 1, -1]
8	IC-8	[1, -2, 1, -3]
9	IC-9	[-1, -3, -2, 2]
10	IC-10	[1, -1, 3, -3]

Here x_1 represents the pendulum position, x_2 represents the pendulum velocity, x_3 represents the cart position, and x_4 represents the cart velocity.

Table 5.12: Multiple Initial Condition Response for FBL Technique

ICs	ST(x_1)	ST(x_2)	ST(x_3)	ST(x_4)	SV(x_1)	SV(x_2)	SV(x_3)	SV(x_4)
IC-1	-	-	-	-	-	-	-	-
IC-2	4	4.2	6.9	7.5	0	0	-75	0
IC-3	7	5.5	-	8	0	0	-75	0
IC-4	7.5	6.9	8.9	9	0	0	-76	0
IC-5	8.9	3.6	9.2	7.2	0	0	0	0
IC-6	8	4.4	9.3	7.4	0	0	0	0
IC-7	7	4.2	9	7.4	0	0	0	0
IC-8	4	3.6	7	9	0	0	0	0
IC-9	8	4	9.5	7.7	0	0	0	0
IC-10	6.7	3.8	9.3	7.2	0	0	0	0

Table 5.13: Multiple Initial Condition Response for FFNN control

ICs	ST(x_1)	ST(x_2)	ST(x_3)	ST(x_4)	SV(x_1)	SV(x_2)	SV(x_3)	SV(x_4)
IC-1	8	5.4	9.0	7.2	0	0	10.3	0
IC-2	6.0	4.0	9.4	9.1	0	0	7.6	0
IC-3	6.9	6.0	9.1	7.3	0	0	10	0
IC-4	6.8	6.4	9.1	6.5	0	0	10.2	0
IC-5	6.0	3.2	9.2	8.1	0	0	10.2	0
IC-6	3.2	3.9	-	7.0	-4	0	77	7.7
IC-7	6.5	4.8	9.6	7.5	0	0	8.9	0
IC-8	6	3.3	9.6	8.6	0	0	7.5	1.4
IC-9	3.9	4.8	-	-	-3	0	38.4	4.7
IC-10	6	5.7	-	7.3	0	0	8.7	0

Table 5.12 presents the Settling Time (ST) results for ten different initial conditions using the FBL Technique. In Table 5.13, the Settling Time outcomes for the same initial conditions are provided, this time employing FFNN. Additionally, Table 5.14 displays the Settling Time results for the LQR technique under the same initial conditions. In cases where stability is not attained within 10 seconds, it is denoted by '-'. These tables offer a comparative analysis of the settling behavior

achieved by each technique across various starting conditions, providing valuable insights into their respective effectiveness and efficiency in achieving stable system responses.

TABLE 5.14: Multiple Initial Condition Response for LQR control

ICs	ST(x_1)	ST(x_2)	ST(x_3)	ST(x_4)	SV(x_1)	SV(x_2)	SV(x_3)	SV(x_4)
IC-1	4.2	5.2	3.2	3.9	0.007	0	0	0
IC-2	4.2	5.8	5.0	6.0	0	0	0	0
IC-3	3.6	3.6	4.1	4.1	0	0	0	0
IC-4	3.2	9.2	5.3	5.7	0	0	0	0
IC-5	7.0	5.3	3.1	3.5	0	0	0	0
IC-6	5.4	5.6	4.6	4.0	0	0	0	0
IC-7	4.6	4.6	3.7	3.9	0	0	0	0
IC-8	5.3	6.0	3.1	3.2	0	0	0	0
IC-9	5.6	5.0	4.4	4.3	0	0	0	0
IC-10	5.1	6.8	3.8	4.0	0	0	0	0

5.6 Overall Performance

FFNN achieved faster settling times on average compared to Feedback Linearization and LQR NSSM. However, it showed inconsistent settling values with larger deviations in the equilibrium points in some cases. Feedback Linearization had moderate settling times but remained stable for all initial conditions. Settling values were generally closer to the desired values compared to FFNN. LQR NSSM achieved the fastest settling times and stable behavior for all initial conditions. Settling values were consistently close to the desired values.

5.6.1 Sensitivity to Initial Conditions

FFNN: Was more sensitive to initial conditions. While faster on average, settling times varied significantly and sometimes diverged depending on the initial values.

Feedback Linearization: Showed moderate sensitivity. Settling times varied moderately between different initial conditions. LQR NSSM: Was the least sensitive to initial conditions. Settling times were consistently fast and stable regardless of the starting values.

5.6.2 Trade-offs and Limitations

FFNN: Offers potentially faster settling times but requires careful training and hyperparameter tuning to ensure stability and consistent performance across different initial conditions. Feedback Linearization: Offers a good balance between settling time and stability but might not achieve the fastest performance compared to other techniques. LQR NSSM: Achieves the fastest and most stable performance but requires accurate system modeling and might be more complex to implement compared to other methods.

5.7 Conclusion

In a comparative analysis of the three controllers based on the provided data, the Feedback Linearization Technique and the Linear Quadratic Regulator with Nonlinear State Space Model (LQR NSSM) exhibit overall stability across various initial conditions, with the Feedback Linearization showing effectiveness in returning to the zero-equilibrium point. The FFNN (Feedforward Neural Network) demonstrates competitive stability and efficiency in returning to zero, except for instability observed in IC-6, which affects its performance. Both Feedback Linearization and LQR NSSM consistently return to zero for stable scenarios. However, the choice between the controllers should be tailored to specific application requirements, considering factors such as stability, settling times, and the ability to return to the equilibrium point. Further investigation and refinement may be needed, particularly addressing instability issues, to enhance the overall efficiency of the controllers in real-world applications. The choice of the best technique depends on your specific priorities and system characteristics. If minimizing settling time is your primary concern: FFNN could be a good option, but be aware of its potential instability and sensitivity to initial conditions. If seeking a balance between performance and stability: Feedback Linearization might be a suitable choice. If stability and consistent performance are critical: LQR NSSM is the recommended option, even though it might involve more complex implementation. Ultimately, the best approach is to evaluate each technique on your specific system and initial conditions to determine the one that delivers the most desirable results.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In conclusion, this study presents a novel and pioneering approach to addressing the challenges associated with maintaining stability and achieving desired states in nonlinear systems, with a specific focus on the inverted pendulum system. Through the integration of a data-driven Model Predictive Control (MPC) framework built upon the Neural State Space Model (NSSM).

By leveraging Neural Networks to capture the nonlinear dynamics of the system, our framework demonstrates exceptional capacity to handle varying initial conditions effectively. The fusion of NSSM with MPC enables seamless integration of system dynamics identification and control policy establishment, leading to improved stability across diverse initial conditions.

Extensive simulations provide strong evidence of the efficacy of our proposed approach, showcasing superior performance even amidst varying starting parameters. Moreover, the potential for real-world implementation and practical scenarios is substantial, suggesting promising avenues for future applications beyond simulation environments.

In essence, this innovative approach not only promises enhanced stability in inverted pendulum systems but also sets a precedent for future advancements in nonlinear control systems, thus contributing significantly to the field of control theory and engineering.

6.2 Future Work

About the future prospects, many directions for research become apparent. The extension of the proposed framework to real-world, complex, and high-dimensional systems would give room to investigate its scalability and applicability to various domains. Searching the use of modern machine learning methods or various combined controller solutions could improve the agility and performance of the proposed approach.

Besides, investigating real-world implementations and running tests on physical systems would confirm the practicality of the proposed methodology. The robustness to uncertainties and disturbances, as well as practical implementation, can be further explored to make its more general applicability.

Additionally, the embedding of online learning elements or adaptive strategies in the MPC framework may lead to ongoing improvement and adjustment to changing system dynamics. Collaboration with control theory and machine learning experts can lead to the generation of valuable interdisciplinary insights and the realization of more advanced and robust control systems.

Bibliography

[1] A. W. Leung, *Nonlinear Systems Of Partial Differential Equations: Applications To Life ...* World Scientific Publishing Company, 2015, accessed on Feb. 06, 2024.

[2] A. Haddow, "Testing, nonlinear system," 2001.

[3] P. Waltman, "A brief survey of persistence in dynamical systems," in *Delay Differential Equations and Dynamical Systems: Proceedings of a Conference in honor of Kenneth Cooke held in Claremont, California, Jan. 13–16, 1990.* Springer, 2006, pp. 31–40.

[4] W. R. Perkins *et al.*, "Feedback control systems," in *Reference Data for Engineers.* Elsevier, 2002, pp. 15–1.

[5] L. B. Prasad, B. Tyagi, and H. O. Gupta, "Modelling and simulation for optimal control of nonlinear inverted pendulum dynamical system using pid controller and lqr," in *2012 Sixth Asia Modelling Symposium.* IEEE, 2012, pp. 138–143.

[6] S. M. Pia, S. Audoly, and L. D'Angio, "Parameter identifiability of nonlinear systems: the role of initial conditions," *Automatica*, vol. 39, no. 4, pp. 619–632, 2003.

[7] S. B. Javed, V. I. Utkin, A. A. Uppal, R. Samar, and A. I. Bhatti, "Data-driven modeling and design of multivariable dynamic sliding mode control for the underground coal gasification project thar," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 153–165, 2021.

[8] A. Bhatti, S. Spurgeon, and X. Lu, "A nonlinear sliding mode control design approach based on neural network modelling," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 9, no. 7, pp. 397–423, 1999.

[9] S. Chen, S. A. Billings, and P. Grant, "Non-linear system identification using neural networks," *International journal of control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[10] S. N. Kumpati, P. Kannan *et al.*, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[11] M. T. Hagan, H. B. Demuth, and O. D. Jesús, "An introduction to the use of neural networks in control systems," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 12, no. 11, pp. 959–985, 2002.

[12] F. Bonassi, M. Farina, J. Xie, and R. Scattolini, "On recurrent neural networks for learning-based control: recent results and ideas for future developments," *Journal of Process Control*, vol. 114, pp. 92–104, 2022.

[13] E. Petlenkov, *Neural networks based identification and control of nonlinear systems: ANARX model based approach.* TUT Press, 2007.

[14] H. V. Nghi, D. P. Nhien, and D. X. Ba, "A lqr neural network control approach for fast stabilizing rotary inverted pendulums," *International Journal of Precision Engineering and Manufacturing*, vol. 23, pp. 45–56, 2022.

[15] Z. Pengpeng, Z. Lei, and H. Yanhai, "Bp neural network control of single inverted pendulum," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology.* IEEE, 2013, pp. 1259–1262.

[16] A. A. Ahmed and A. S. Alshandoli, "On replacing a pid controller with neural network controller for segway," in *2020 International Conference on Electrical Engineering (ICEE).* IEEE, 2020, pp. 1–4.

[17] W. Yu and R. Jafari, *Modeling and control of uncertain nonlinear systems with fuzzy equations and Z-number.* John Wiley & Sons, 2019.

[18] A. Iqbal and A. Bhatti, "Developement of framework for structured uncertainty analysis for an aircraft," in *2006 IEEE International Multitopic Conference.* IEEE, 2006, pp. 444–450.

[19] N. Lasemi and H. R. Shaker, "Spacecraft attitude control: Application of fine trajectory linearization control," *Advances in Space Research*, vol. 68, no. 9, pp. 3663–3676, 2021.

[20] M. Spong and M. Vidyasagar, "Robust linear compensator design for nonlinear robotic control," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 345–351, 1987.

[21] S. Bano, M. R. Azam, A. A. Uppal, S. B. Javed, and A. I. Bhatti, "Robust p53 recovery using chattering free sliding mode control and a gain-scheduled modified utkin observer," *Journal of Theoretical Biology*, vol. 532, p. 110914, 2022.

[22] J. Wankar, N. G. Kotla, S. Gera, S. Rasala, A. Pandit, and Y. A. Rochev, "Recent advances in host–guest self-assembled cyclodextrin carriers: Implications for responsive drug delivery and biomedical engineering," *Advanced Functional Materials*, vol. 30, no. 44, p. 1909049, 2020.

[23] J.-J. E. Slotine, W. Li *et al.*, *Applied nonlinear control.* Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[24] ——, *Applied nonlinear control.* Prentice hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[25] P. G. Drazin, *Nonlinear systems.* Cambridge University Press, 1992, no. 10.

[26] R. Bellman, I. Glicksberg, and O. Gross, "On the "bang-bang" control problem," *Quarterly of Applied Mathematics*, vol. 14, no. 1, pp. 11–18, 1956.

[27] D. Recker, P. Kokotovic, D. Rhode, and J. Winkelman, "Adaptive nonlinear control of systems containing a deadzone," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control.* IEEE, 1991, pp. 2111–2115.

[28] D. Liu and A. N. Michel, "Stability analysis of state-space realizations for two-dimensional filters with overflow nonlinearities," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 2, pp. 127–137, 1994.

[29] B. Zhou, W. X. Zheng, and G.-R. Duan, "An improved treatment of saturation nonlinearity with its application to control of systems subject to nested saturation," *Automatica*, vol. 47, no. 2, pp. 306–315, 2011.

[30] ScienceDirect Topics. Backlash - an overview. Accessed on Feb. 06, 2024. [Online]. Available: https://www.sciencedirect.com/topics/engineering/backlash

[31] Electrical4U. Describing function: Analysis of nonlinear systems. Accessed on Feb. 06, 2024. [Online]. Available: https://www.electrical4u.com/different-types-non-linearities-in-control-system/

[32] J. Zhou, C. Wen, and T. Li, "Adaptive output feedback control of uncertain nonlinear systems with hysteresis nonlinearity," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2627–2633, 2012.

[33] E. Wilson, I. Farhoomand, and K. Bathe, "Nonlinear dynamic analysis of complex structures," *Earthquake Engineering & Structural Dynamics*, vol. 1, no. 3, pp. 241–252, 1972.

[34] World Scientific, *Stabilities of Nonlinear Systems (I)*. WORLD SCIENTIFIC, 2023, pp. 37–77.

[35] ScienceDirect Topics. Stability of control system - an overview. Accessed on Feb. 06, 2024. [Online]. Available: https://www.sciencedirect.com/topics/engineering/stability-of-control-system

[36] Cadence Design Systems. (2023) BIBO stability. Accessed on Feb. 06, 2024. [Online]. Available: https://resources.pcb.cadence.com/blog/2023-bibo-stability

[37] J. LaSalle, "The extent of asymptotic stability," *Proceedings of the National Academy of Sciences*, vol. 46, no. 3, pp. 363–365, 1960.

[38] G. Chen, "Nonlinear systems: Stability, dynamics and control," 2023.

[39] R. I. Leine, "The historical development of classical stability concepts: Lagrange, poisson and lyapunov stability," *Nonlinear Dynamics*, vol. 59, pp. 173–182, 2010.

[40] The inverted pendulum benchmark in nonlinear control theory: A survey. Accessed on Feb. 06, 2024. [Online]. Available: https://www.researchgate.net/publication/236012685_The_Inverted_Pendulum_Benchmark_in_Nonlinear_Control_Theory_A_Survey

[41] Velocity control of a wheeled inverted pendulum by partial feedback linearization. Accessed on Jan. 24, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1429366

[42] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 505–513, Jun. 2005.

[43] B.-C. Chang, H. Kwatny, and S.-S. Hu, "An application of robust feedback linearization to a ball and beam control problem," in *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*, vol. 1. IEEE, 1998, pp. 694–698.

[44] W.-D. Chang, R.-C. Hwang, and J.-G. Hsieh, "A self-tuning pid control for a class of nonlinear systems based on the lyapunov approach," *Journal of Process Control*, vol. 12, no. 2, pp. 233–242, 2002.

[45] A. Siuka and M. Schöberl, "Applications of energy based control methods for the inverted pendulum on a cart," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1012–1017, 2009.

[46] W. Zhong and H. Rock, "Energy and passivity based control of the double inverted pendulum on a cart," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01)(Cat. No. 01CH37204)*. IEEE, 2001, pp. 896–901.

[47] M. Kelly and A. Ruina, "Non-linear robust control for inverted-pendulum 2d walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4353–4358.

[48] R. Khan, F. M. Malik, A. Raza, N. Mazhar, H. Ullah, and M. Umair, "Robust nonlinear control design and disturbance estimation for ball and beam system," in *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2020, pp. 1–6.

[49] A. K. Yadav, P. K. Pathak, S. V. Sah, and P. Gaur, "Sliding mode-based fuzzy model reference adaptive control technique for an unstable system," *Journal of The Institution of Engineers (India): Series B*, vol. 100, pp. 169–177, 2019.

[50] S. Riachy, Y. Orlov, T. Floquet, R. Santiesteban, and J.-P. Richard, "Second-order sliding mode control of underactuated mechanical systems i: Local stabilization with application to an inverted pendulum," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 18, no. 4-5, pp. 529–543, 2008.

[51] P. Ponce, A. Molina, and E. Alvarez, "A review of intelligent control systems applied to the inverted-pendulum problem," *American Journal of Engineering and Applied Sciences*, vol. 7, no. 2, pp. 161–207, 2014.

[52] P. N. Dao and Y.-C. Liu, "Adaptive reinforcement learning strategy with sliding mode control for unknown and disturbed wheeled inverted pendulum," *International Journal of Control, Automation and Systems*, vol. 19, no. 2, pp. 1139–1150, 2021.

[53] R. Pramoditha, "The concept of artificial neurons (perceptrons) in neural networks," *Article from Towards Data Science*, 2021.

[54] C. F. Touzet, "Neural reinforcement learning for behaviour synthesis," *Robotics and Autonomous Systems*, vol. 22, no. 3-4, pp. 251–281, 1997.

[55] Saffron Edge. Feedforward vs backpropagation ann. Accessed on Feb. 13, 2024. [Online]. Available: https://www.linkedin.com/pulse/feedforward-vs-backpropagation-ann-saffronedge1

[56] D. Upadhyay, J. Singh, and L. Kumar, "An improved approach of neural network to non-linear inverted pendulum system," *Turkish Online Journal of Qualitative Inquiry (TOJQI)*, vol. 11, no. 4, pp. 1214–1222, October 2020, research Article.

[57] M. Ławryńczuk and M. Ławryńczuk, "Mpc algorithms based on neural state-space models," *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*, pp. 139–166, 2014.

[58] H. A. Afan, A. Ibrahem Ahmed Osman, Y. Essam, A. N. Ahmed, Y. F. Huang, O. Kisi, M. Sherif, A. Sefelnasr, K.-w. Chau, and A. El-Shafie, "Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques," *Engineering Applications of Computational Fluid Mechanics*, vol. 15, no. 1, pp. 1420–1439, 2021.

[59] B. Mehdi, D. Brahmi-Ingrachen, H. Belkacemi, and L. Muhr, "Development of a mathematical model based on an artificial neural network (ann) to predict nickel uptake data by a natural zeolite," in *Physical Sciences Forum*, vol. 6, no. 1. MDPI, 2023, p. 4.

[60] P. Matthews, S. Greenspan, P. Matthews, and S. Greenspan, "Technology definitions: Level setting for the rest of the book," *Automation and Collaborative Robotics: A Guide to the Future of Work*, pp. 45–67, 2020.

[61] S. D. Hanwate, A. Budhraja, and Y. V. Hote, "Improved performance of cart inverted pendulum system using lqr based pid controller and ann," in *2015 IEEE UP Section Conference on Electrical Computer and Electronics (UPCON)*. IEEE, 2015, pp. 1–6.

[62] H. Gao, X. Li, C. Gao, and J. Wu, "Neural network supervision control strategy for inverted pendulum tracking control," *Discrete Dynamics in Nature and Society*, vol. 2021, pp. 1–14, 2021.

[63] A. de Carvalho Junior, B. A. Angelico, J. F. Justo, A. M. de Oliveira, and J. I. da Silva Filho, "Model reference control by recurrent neural network built with paraconsistent neurons for trajectory tracking of a rotary inverted pendulum," *Applied Soft Computing*, vol. 133, p. 109927, 2023.

[64] X. Tian and H. Peng, "A model predictive control approach to inverted pendulum system based on rbf-arx model," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 3559–3564.

[65] Y. Qin, H. Peng, W. Ruan, J. Wu, and J. Gao, "A modeling and control approach to magnetic levitation system based on state-dependent arx model," *Journal of Process Control*, vol. 24, no. 1, pp. 93–112, 2014.

[66] S. Li, C. Huo, and Y. Liu, "Inverted pendulum system control by using modified pid neural network," in *2008 3rd International Conference on Innovative Computing Information and Control*. IEEE, 2008, pp. 426–426.

[67] J. Moreno-Valenzuela, C. Aguilar-Avelar, S. A. Puga-Guzmán, and V. Santibáñez, "Adaptive neural network control for the trajectory tracking of the furuta pendulum," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 3439–3452, 2016.

[68] P. Gautam, "Optimal control of inverted pendulum system using adaline artificial neural network with lqr," in *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. IEEE, 2016, pp. 1–6.

[69] J. Stang, "A design project report presented to the engineering division of the graduate school of cornell university in partial fulfillment of the requirements for the degree of master of engineering (electrical)," unpublished.

[70] E. Y. Bejarbaneh, A. Bagheri, B. Y. Bejarbaneh, S. Buyamin, and S. N. Chegini, "A new adjusting technique for pid type fuzzy logic controller using psoscalf optimization algorithm," *Applied Soft Computing*, vol. 85, p. 105822, 2019.

[71] E. S. Varghese, A. K. Vincent, and V. Bagyaveereswaran, "Optimal control of inverted pendulum system using pid controller, lqr and mpc," in *IOP Conference Series: Materials Science and Engineering*, vol. 263, no. 5. IOP Publishing, 2017, p. 052007.

[72] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE transactions on control systems technology*, vol. 13, no. 4, pp. 559–576, 2005.

[73] E. I. Knudsen, "Supervised learning in the brain," *The Journal of Neuroscience*, vol. 14, no. 7, p. 3985, 1994.

[74] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia: case studies on organization and retrieval.* Springer, 2008, pp. 21–49.

[75] D. Chaturvedi, "Artificial neural network and supervised learning," *Soft Computing: Techniques and its Applications in Electrical Engineering*, pp. 23–50, 2008.

[76] L. Igual, S. Seguí, L. Igual, and S. Seguí, "Supervised learning," *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*, pp. 67–96, 2017.

[77] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, "Unsupervised learning based on artificial neural network: A review," in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS).* IEEE, 2018, pp. 322–327.

[78] Q. Zhang, Y. Yang, Y. Liu, Y. N. Wu, and S.-C. Zhu, "Unsupervised learning of neural networks to explain neural networks," *arXiv preprint arXiv:1805.07468*, 2018.

[79] S. Becker, "Unsupervised learning procedures for neural networks," *International Journal of Neural Systems*, vol. 2, no. 01n02, pp. 17–33, 1991.

[80] V. Golovko, A. Kroshchanka, and D. Treadwell, "The nature of unsupervised learning in deep neural networks: A new understanding and novel approach," *Optical memory and neural networks*, vol. 25, pp. 127–141, 2016.

[81] A. F. Atiya, "An unsupervised learning technique for artificial neural networks," *Neural Networks*, vol. 3, no. 6, pp. 707–711, 1990.

[82] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *arXiv preprint arXiv:1611.02167*, 2016.

[83] R. Coulom, "Reinforcement learning using neural networks, with applications to motor control," Ph.D. dissertation, Institut National Polytechnique de Grenoble-INPG, 2002.

[84] N. T. Siebel and G. Sommer, "Evolutionary reinforcement learning of artificial neural networks," *International Journal of Hybrid Intelligent Systems*, vol. 4, no. 3, pp. 171–183, 2007.

[85] Z. Miljković, M. Mitić, M. Lazarević, and B. Babić, "Neural network reinforcement learning for visual control of robot manipulators," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1721–1736, 2013.

[86] T. Callinan, "Artificial neural network identification and control of the inverted pendulum," *MEng Project Reports, School of Electronic Engineering Dublin City University*, 2003.

[87] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *Ieee Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[88] H. Jahanshahi, M. Shahriari-Kahkeshi, R. Alcaraz, X. Wang, V. P. Singh, and V.-T. Pham, "Entropy analysis and neural network-based adaptive control of a non-equilibrium four-dimensional chaotic system with hidden attractors," *Entropy*, vol. 21, no. 2, p. 156, 2019.

[89] H. Alsubaie, "A neural state-space-based model predictive technique for effective vibration control in nano-beams," *Frontiers in Physics*, vol. 11, p. 1253642, 2023.

[90] MathWorks. Neural state-space models - matlab & simulink. Accessed on Feb. 01, 2024. [Online]. Available: https://www.mathworks.com/help/ident/neural-state-space-models.html

[91] B. Yegnanarayana, *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[92] H. Alsubaie, "A neural state-space-based model predictive technique for effective vibration control in nano-beams," *Frontiers in Physics*, vol. 11, p. 1253642, 2023.

[93] T. Callinan, "Artificial neural network identification and control of the inverted pendulum," Master's Thesis, University Name, August 2003.