Intelligent Control of Directional Drilling Using GRU Neural Networks

Zahed Ebrahimi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

December 2025

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Zahed Ebrahimi

Entitled: Intelligent Control of Directional Drilling Using GRU Neural Networks

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

_____ Examiner

_____ Examiner

_____ Thesis Supervisor(s)

_____ Thesis Supervisor(s)

Approved by _____
Chair of Department or Graduate Program Director

_____
Dean of Faculty

Abstract

Intelligent Control of Directional Drilling Using GRU Neural Networks

Zahed Ebrahimi

Directional drilling requires precise trajectory control under highly nonlinear and time-varying downhole conditions. Conventional proportional–integral–derivative (PID) controllers are simple to implement but struggle to maintain performance when formation properties change. Model predictive control (MPC) can achieve improved accuracy and constraint handling, yet it remains computationally impractical for real-time downhole processors. To address these challenges, this thesis develops two intelligent control strategies based on gated recurrent unit (GRU) neural networks integrated with a finite-element model (FEM) of the drillstring and a nonlinear polycrystalline-diamond-compact (PDC) bit–rock.

The first approach introduces an adaptive GRU–PID controller that updates PID gains online using a trained GRU network to estimate the local input–output Jacobian of the drilling process. This strategy preserves the interpretability and low complexity of PID while significantly improving the transient response, robustness to uncertainties, and steady-state accuracy. The second approach presents a GRU-based MPC surrogate (GMPC) that learns the optimal MPC control policy offline from input–output data and adapts online through a GRU estimator. The resulting controller achieves MPC-level tracking performance with drastically reduced computation, enabling real-time feasibility.

Both controllers were evaluated using identical simulation scenarios derived from the FEM. The results demonstrate that GRU–PID offers low computational cost, while GMPC provides near-optimal accuracy and superior constraint compliance. Together, the proposed methods illustrate a practical path towards adaptive, high-performance, and computationally efficient directional drilling automation.

Acknowledgments

The author would like to express sincere gratitude to Dr. Rastko R. Selmic, whose guidance, encouragement, and deep insight into control theory and intelligent systems made this research possible. Their mentorship throughout the development of this work — from theoretical formulation to practical validation — has been invaluable and instrumental in shaping both the technical depth and professional rigor of this thesis.

The author would also like to acknowledge the members of the Examination Committee for their constructive feedback and helpful discussions that improved the quality of this research.

Special thanks are due to the Department of Electrical and Computer Engineering and Concordia University for providing a strong academic environment and the computational resources necessary for simulation and experimentation.

Finally, heartfelt appreciation goes to the author's family for their unwavering encouragement, patience, and belief throughout the graduate studies journey. Their constant support and motivation have been the cornerstone of this accomplishment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Directional drilling is the key enabling technology that allows oil, gas, and geothermal wells to reach targets that cannot be accessed by vertical drilling alone. It enables operators to drill curved and horizontal boreholes that pass through multiple reservoir zones while minimizing the surface footprint. In modern practice, drilling control involves the continuous coordination of downhole sensors, actuators, and surface controllers to maintain the desired trajectory, optimize the rate of penetration (ROP), and suppress unwanted vibration phenomena such as stick–slip, bit–bounce, and lateral whirl [7, 9].

1.1 Motivation

The directional drilling process exhibits strong nonlinear coupling between torsional, axial, and lateral dynamics of the drill string. Changing formation properties and operational parameters introduce large uncertainties. Conventional control systems, typically based on proportional–integral–derivative (PID) controllers, are simple to implement but often fail to maintain performance when parameters drift away from their nominal values [10]. Model predictive control (MPC) is a theoretically attractive alternative because it explicitly handles multivariable interactions and constraints [9]. However, solving an optimization problem at each control step is computationally expensive, particularly for downhole embedded processors with strict real-time requirements (typical sampling times on the order of 10–20 ms).

With the rapid progress of machine learning, neural networks have emerged as powerful tools for modeling and control of complex nonlinear systems. Among various architectures, the gated recurrent unit (GRU) [11, 12] network has gained attention for its ability to capture temporal dependencies in sequential data with relatively few parameters. GRUs combine the advantages of recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, but are more computationally efficient [13,14]. These features make GRUs well suited for the directional drilling application, where temporal correlation between vibration states, torque-on-bit (TOB), and weight-on-bit (WOB) signals plays a critical role.

1.2 Problem Statement

The drilling process can be represented by a high-order, nonlinear dynamic model derived from FEM [7, 8] discretization of the drill string. Let the generalized coordinates and velocities be collected in a state vector $\mathbf{x}(t)$, and let the surface controllable variables such as top-drive torque and hook load be represented by $\mathbf{u}(t)$. The general dynamic equation of motion is

$$M(\theta)\ddot{\mathbf{x}}(t) + D(\theta)\dot{\mathbf{x}}(t) + K(\theta)\mathbf{x}(t) = \mathbf{f}_{\text{ext}}\big(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), t\big), \tag{1}$$

where M, D, and K are the inertia, damping, and stiffness matrices, respectively, and $\mathbf{f}_{\text{ext}}$ represents the external forces and torques due to the bit–rock interaction, fluid drag, and gravity. The parameters θ vary with formation type, configuration of the drill-string, and the operation conditions. Designing a controller that can track the desired trajectory despite such variations, while being computationally feasible for real-time implementation, is the central challenge addressed in this thesis.

1.3 Research Objectives

This work aims to develop and validate two complementary control strategies that combine model-based understanding with data-driven adaptation:

(1) **Adaptive GRU–PID Control.** An intelligent PID controller whose proportional, integral,

and derivative gains are adjusted online by a GRU network that learns the relationship between control inputs and measured drilling responses. The approach preserves the interpretability and simplicity of PID while introducing adaptive learning capability [11].

(2) **GRU-Based MPC Emulation Control (GMPC).** A controller that learns the input–output behavior of a model predictive controller offline and then adapts online using real-time feedback through a GRU model estimator. The surrogate reproduces MPC-level performance without the heavy optimization burden [12].

Both controllers are designed and evaluated using a detailed FEM-based model of the drill-string and bit–rock interaction. They are compared with classical PID and MPC baselines in terms of tracking accuracy, robustness, and computational efficiency.

1.4 Scope and Limitations

The thesis focuses on the rotary steerable system (RSS) configuration commonly used in push-the-bit and point-the-bit technologies. Only axial and torsional dynamics are explicitly modeled, as lateral bending effects are considered uncertainties. The bit–rock interaction follows the deterministic model of Detournay [7]. Fluid–structure coupling and high-frequency drill-collar vibrations are neglected for computational efficiency.

1.5 Methodology Overview

Figure 1.1 illustrates the workflow adopted in this research. First, the downhole system is modeled using the FEM, yielding a high-order nonlinear model that is discretized to a control-affine form. The adaptive GRU–PID and GMPC controllers are then trained and validated using simulation data generated from this model. The methodology proceeds through the following steps:

(1) Develop and verify an FEM of the drill-string, including PDC bit–rock interaction and Rayleigh damping.

(2) Formulate the model in state-space form suitable for control design.

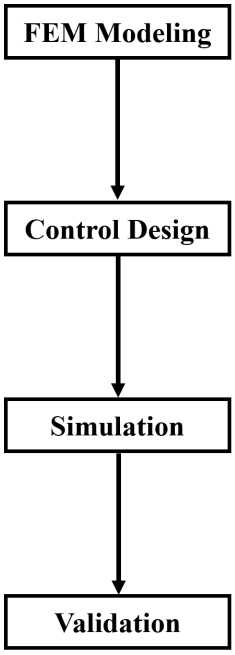

Figure 1.1: Overall workflow of the thesis: modeling, controller design, and validation.

(3) Implement the adaptive GRU–PID controller and tune hyperparameters.

(4) Train the GMPC using data generated from an MPC reference policy.

(5) Compare each controllers in identical simulation scenarios.

1.6 Thesis Organization

The remainder of this thesis is organized as follows.

- **Chapter 2**: This chapter presents an in-depth review of drilling dynamics modeling, vibration mechanisms, bit–rock interaction models, and conventional and control methodologies. Highlight limitations in existing control strategies, motivating the need for adaptive and neural-network-based approaches.

- **Chapter 3**: This chapter develops a high-fidelity modeling framework for the drill string using the FEM. Axial–torsional coupling, Rayleigh damping, drillstring–wellbore contact forces, and detailed PDC bit–rock interaction models are presented. The resulting dynamic

equations are expressed in a control-affine state-space form suitable for future control design. Model discretization and time-delay approximations for depth-of-cut are also introduced.

- **Chapter 4**: This chapter introduces a GRU-based adaptive PID controller that dynamically adjusts PID gains by learning temporal error patterns and system responses. The controller is trained using simulation data and updated online to handle uncertainties. Adaptation mechanisms, and real-time performance aspects are analyzed.

- **Chapter 5**: To address MPC's computational challenges, this chapter presents a GRU-based surrogate model trained to emulate MPC control policies. The GRU is used to replace the MPC optimization step, providing rapid control decisions with significantly reduced computational demand. An adaptive update mechanism improves robustness against time-varying disturbances and modeling errors.

- **Chapter 6**: The final chapter summarizes the findings, highlights the advantages of integrating GRU-based adaptive control with FEM-based drilling models, and identifies future research directions, including real-time hardware implementation, hybrid physics–ML models, and potential extension to full 3D directional drilling.

Chapter 2

Literature Review

2.1 Overview

Directional drilling integrates mechanical, hydraulic, and control subsystems that must operate in coordination under highly uncertain downhole environments. This chapter reviews the major technologies and control methodologies that have evolved to support precise wellbore trajectory tracking and vibration suppression. It begins with a discussion of conventional and rotary–steerable drilling technologies, followed by a review of mathematical modeling approaches and control strategies ranging from classical PID to modern intelligent controllers.

2.2 Advancements in Directional Drilling Technologies

In oil and gas drilling engineering, perfectly vertical boreholes are practically nonexistent. Wells that aim at the subsurface location directly beneath their surface point are defined as vertical wells. As illustrated in Figure 2.1, the deviation angle between the borehole and vertical axis is the critical measure, while azimuth deviation is generally less significant [15].

Directional drilling has dramatically transformed modern petroleum engineering, especially in unconventional resources such as shale gas and coal-bed methane (CBM). It enables precise placement of well trajectories in complex formations, combining multiple advanced technologies [16].

One of the major milestones is the advent of rotary steerable systems (RSS), which are categorized into whipstock, bent-sub, point-the-bit, push-the-bit, lead-the-bit, and jetting/nudging techniques [1, 17].

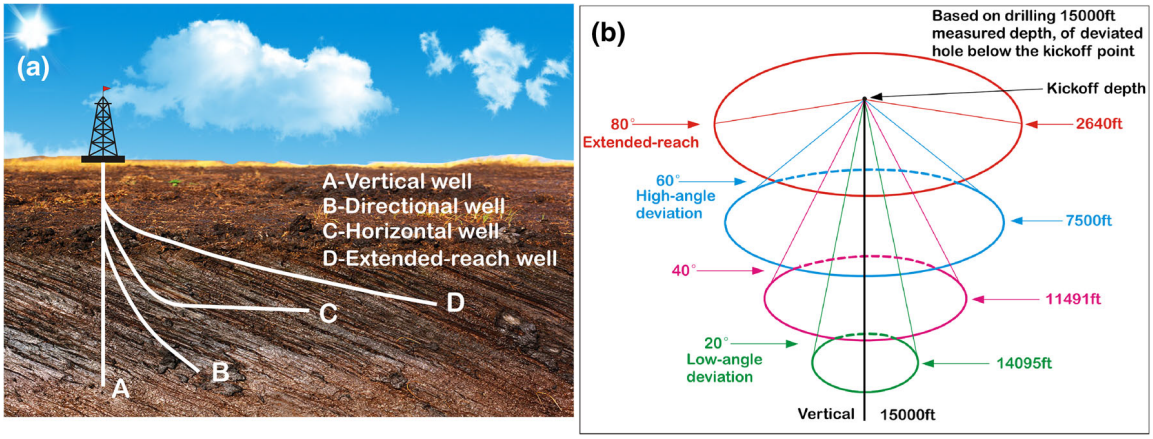

Figure 2.1: The schematic plot of different kinds of wells. (a) Four typical kinds of wells and (b) the classification of well types based on deviated angle [1].

Key foundational directional drilling tools include the whipstock, used in sidetracking, and the bent-sub assembly, which introduces a controlled bend in the drill string [1]. Lead-the-bit technology further enhanced drilling control by integrating steering mechanisms into the bit itself [4]. Jetting techniques add steering capabilities by fluid erosion in soft formations [4]. Together, these technologies significantly improve precision, efficiency, and environmental compatibility.

2.3 Advanced Techniques and Procedures in Whipstock Drilling

The whipstock has long been a foundational tool in directional drilling, primarily utilized in modern times for well sidetracking from the casing. In the era before the widespread adoption of turbine-type and positive-displacement mud motors, whipstocks, in conjunction with jetting methods, served as the primary tools for directional control in drilling. The whipstock, typically made of steel, is wedge-shaped with a tapered concave groove. This design is crucial for guiding the drill bit to initiate a change in the drilling path by directing it towards the wellbore wall. Figure 2.2 is depicted Retrievable (Removable) Whipstock and Permanent Whipstock which are stated below

2.3.1 Whipstock Varieties and Operational Context

Whipstock technology has evolved to include two main types, each suited to different drilling scenarios.

- **Retrievable Whipstock** – Designed for temporary use, the retrievable whipstock is deployed into the well and removed once the desired directional change is achieved. This type is particularly useful in scenarios where the sidetrack is a temporary requirement, or in wells where leaving equipment behind is not feasible.
- **Permanent Whipstock** – As the name suggests, this whipstock type is left in place in the well after it has served its purpose in changing the well's direction. This is often used in permanent wellbore trajectory alterations.

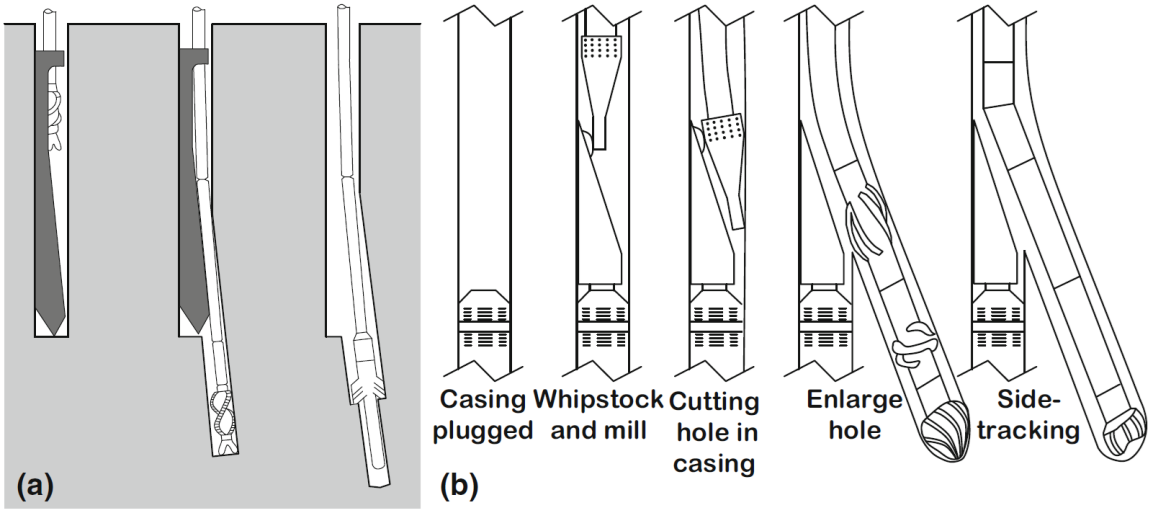

Figure 2.2: The schematic diagram of removable and permanent whipstock. (a) Removable whipstock and (b) Permanent whipstock [1]

2.3.2 Operational Procedure

The process of deploying a whipstock begins with selecting the appropriate edge angle based on the intended deflection trajectory. The diameter of the drill bit is then chosen to complement the whipstock, ensuring a proper fit in the wellbore. The whipstock is mounted on the drill string above

the drill bit and then lowered to the Kick-Off-Point (KOP). At this depth, the toe of the whipstock is precisely aligned in the desired new drilling direction.

2.3.3 Use in Cased Holes

In cased holes, fixed whipstocks are used through three main steps:

(1) **Window Cutting**: The initial phase involves cutting a window in the casing. This is accomplished using a milling tool designed specifically for this task.

(2) **Window Extension**: After the initial window cutting, the mill is replaced with a sidetracking mill. This tool extends the window to a length of approximately 8 to 12 feet, providing enough space for subsequent drilling operations.

(3) **Window Enlargement**: In the final phase, the sidetracking mill is exchanged for a taper mill. Alongside this, a Bottom Hole Assembly (BHA) equipped with watermelon mills is used to further expand the casing window. This expansion is necessary to fit a conventional BHA for the continuation of the drilling operation. Depending on the project requirements, this stage might require multiple trips and extensive rotation hours to achieve the desired sidetrack.

2.4 Downhole Motor and Bent-Sub in Directional Drilling

In the realm of directional drilling, one of the most prevalent techniques for altering wellbore trajectories is the use of a downhole motor in conjunction with a bent sub. This assembly, as illustrated in Figure 2.4, strategically positions the bent sub directly above the motor, creating an effective deflection mechanism. The bent sub, characterized by its inclined lower thread—typically set at an angle ranging from 0.5° to 3° from the axis of the sub body—serves as the pivotal component in this assembly.

2.4.1 Function and Mechanics

The bent sub effectively acts as the fulcrum in a lever system, inducing both lateral and downward movements of the drill bit. When the downhole motor rotates the drill bit without rotating the

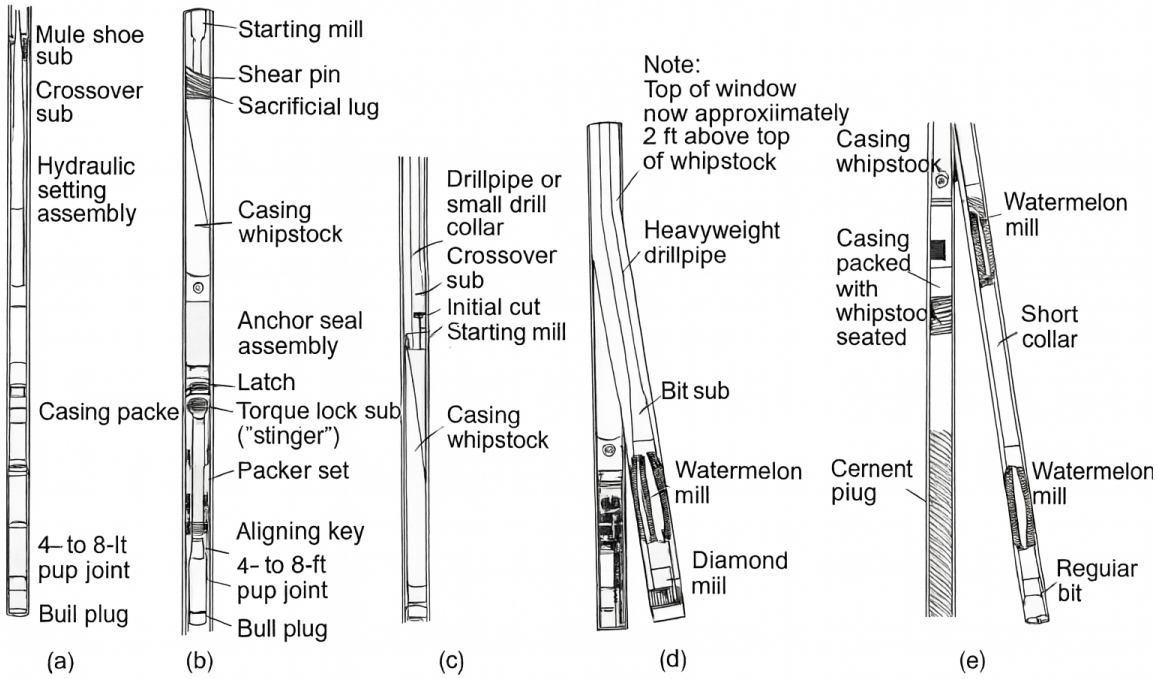

Figure 2.3: (a) Setting the packer and whipstock seat; (b) locking the whipstock into the packer assembly; (c) cutting the casing with the starting mill; (d) cutting a window in the casing with a side-packing mill; (e) drilling ahead with a tricone bit [2].

entire drill string, the lateral force exerted at the bit causes the motor to follow a naturally curved trajectory. This curvature, referred to as the dogleg severity, is a critical aspect of directional drilling. It is influenced by several factors, including the angle of the bent sub, the outer diameters (ODs) of the motor, bent sub, and drill collars in relation to the hole diameter, and the motor's length.

2.4.2 Applications

This combination of a downhole motor and a bent sub is versatile, serving multiple roles in the drilling process. It is instrumental in initiating well kicks, where the direction of the wellbore is altered, in executing correction runs to adjust the trajectory, and in facilitating sidetracking operations when creating new paths from the existing wellbore.

2.4.3 Stabilization Considerations

A notable feature of this assembly is the absence of stabilizers in its lower section, generally for a distance of at least 90 feet above the bent sub. This lack of stabilizers helps in creating a

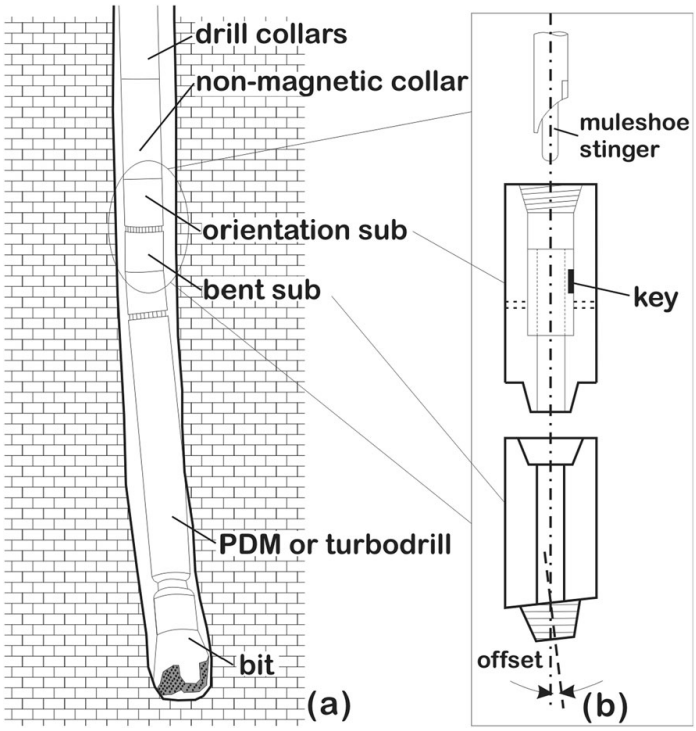

Figure 2.4: Downhole motor and bent sub assembly [1].

more pronounced bend, which is essential for directional control. In scenarios where the motor and bent-sub assembly are employed for initiating directional changes at relatively shallow depths, the entire BHA is often configured to be ”slick.” This term implies the absence of stabilizers throughout the BHA, allowing for greater flexibility and responsiveness in altering the well trajectory.

2.5 Point-The-Bit Technology

The RSS recognized as a groundbreaking innovation in drilling technology, particularly in the exploration of shale gas, has revolutionized the approach to directional drilling. Unlike traditional methods, RSS allows for continuous rotation from the surface, thus facilitating directional, horizontal, and extended-reach drilling operations with greater efficiency [1, 18].

2.5.1 Mechanism

The ”point-the-bit” system operates on a similar principle to that used in bent-housing motor systems. In these ”point-the-bit” setups, the bent housing is integrated within the collar, enabling its

orientation towards the desired drilling direction while the drill-string is rotating. Typical examples of ”point-the-bit” systems include the Halliburton Sperry-Sun Geo-Pilot system and the Gyrodata CDAL system, with the Geo-Pilot system often cited as a representative model for explaining the operational mechanics of these systems. As depicted in Figure 2.5, the Geo-Pilot system is primarily composed of a non-rotating outer housing, an internal rotary shaft, and dual eccentric rings. One of these rings is placed within the other, creating a controllable eccentric unit. The internal ring can cause the internal rotary shaft to deflect, mechanically inducing a dogleg, thereby tilting the bit in relation to the rest of the tool to follow the desired path. In essence, ”point-the-bit” systems alter the well’s trajectory by modifying the tool face angle, with the trajectory shifting in the direction of the bend. This bending orientation is managed by a servo motor that rotates in tandem with the drill-string but in the opposite direction, allowing the tool face to maintain a geostationary, or non-rotating, position while the collar rotates [1].

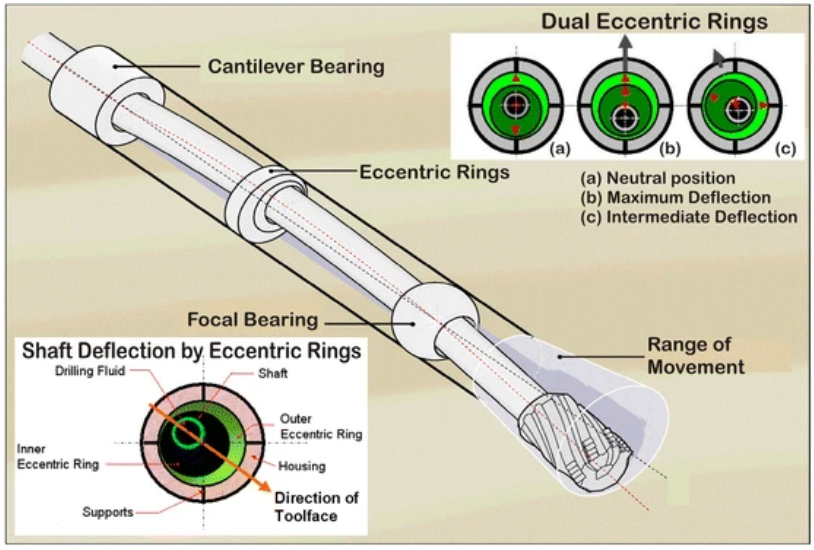

Figure 2.5: The Halliburton Sperry-sun Geo-Pilot system [1].

2.5.2 RSS Integration

In the point-the-bit approach, the RSS steers the drill bit by slightly tilting it in the desired direction while the entire drill string continues to rotate. This tilt is achieved through intricate mechanisms within the RSS tool, which adjust the angle of the bit without hindering the rotation of

the drill string [4, 19] as representing the tin Figure 2.6. The result is a smooth, curved trajectory that can be precisely controlled to navigate complex subsurface formations.

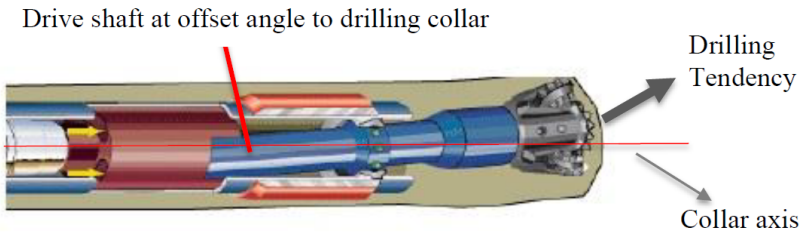

Figure 2.6: Drive shaft set at a collar-offset angle [3].

2.6 Lead-The-Bit Technology

The lead-the-bit methodology within the Rotary Steerable System (RSS) framework represents a significant advancement in directional drilling technology. This approach integrates a specialized steering unit directly within the drill bit.

2.6.1 Integration and Function of the Steering Unit

At the heart of the lead-the-bit system is an innovative steering mechanism, embedded within the drill bit itself. This mechanism typically consists of a shaft that can dynamically tilt, thereby altering the drill bit's heading in response to real-time drilling conditions. By adjusting the angle of this shaft, operators can precisely steer the drill bit, enabling fine control over the wellbore's direction. This ability to make intricate adjustments is crucial for navigating complex geological formations and achieving targeted well placements. Figure 2.7 is depicted this approach [4].

2.7 Push-The-Bit Technology

The push-the-bit methodology is one of remarkable evolution in the technology used for directional drilling. This approach is designed to alter the trajectory of the Bottom Hole Assembly (BHA) in a controlled and precise manner.

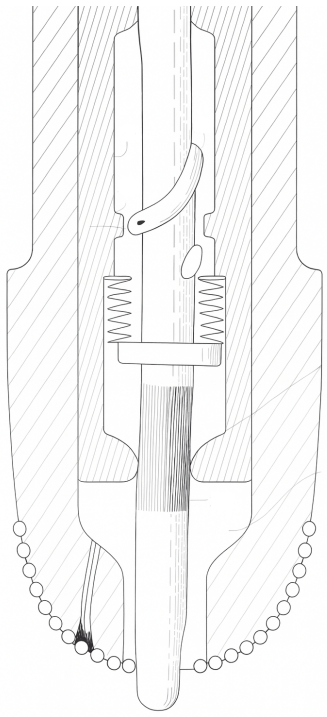

Figure 2.7: RSS Lead- The-Bit [4].

2.7.1 Operational Mechanism

The ”push-the-bit” system operates on the principle of exerting lateral force on the drill bit, pushing it against the wall of the borehole to steer it along the intended path. Prominent examples of ”push-the-bit” systems include Schlumberger’s PowerDrive and Baker Hughes’ AutoTrak. The PowerDrive system is often used to illustrate the mechanics of ”push-the-bit” technology. As depicted in Figure 2.8 and Figure 2.9, the PowerDrive system is both simple in design and compact, consisting of a bias unit and a control unit, adding only about 12.5 feet to the BHA. The bias unit, positioned right behind the drill bit, applies directional force to the bit as the entire drill string rotates. The control unit, located behind the bias unit, is equipped with self-powered electronics, sensors, and a mechanism that controls the average strength and direction of the side loads applied to the bit to follow the planned trajectory. The bias unit features three external, hinged pads, which are operated by controlled mud flow through a valve. This valve utilizes the pressure difference in the drilling mud inside and outside the bias unit. A three-way rotary disk valve in the unit activates the pads by directing mud flow into the piston chamber of each pad as it aligns with the desired push

point — the point opposite the intended trajectory — in the wellbore [1, 19].

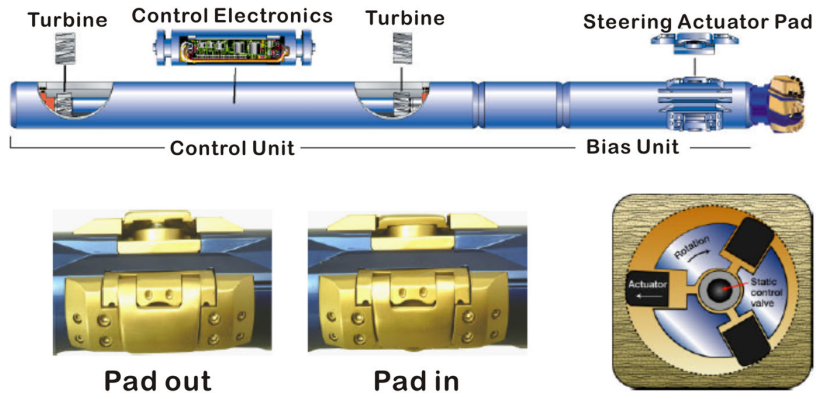

Figure 2.8: The Schlumberger PowerDrive system [1].

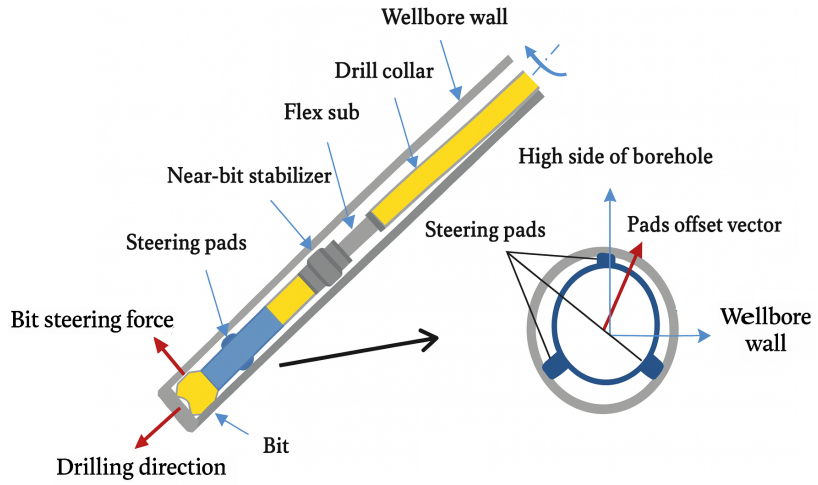

Figure 2.9: Directional drilling of push-the-bit Rotary Steerable Bottom Hole Assembly (RSBHA) [5].

2.7.2 Dynamic Directional Control

The dynamic nature of this system lies in its ability to change the direction of the BHA through the active and deliberate application of force by these pads or blades. Each pad can extend or retract based on real-time downhole measurements and surface commands, allowing for precise adjustments in the drilling trajectory. This active steering mechanism enables the drill bit to be pushed in the desired direction, effectively redirecting the wellbore's path [20].

2.8 Jetting/Nudging Method

Jetting or Nudging is a technique used to alter the direction of a borehole without relying on traditional directional drilling assemblies. This method is particularly effective in softer geological formations and is suitable for creating angles with low build rates. While it can be a moderately effective approach to directional drilling under the right conditions, its use is not very widespread. In holes with low drift angles, the maximum achievable angle increase is typically around 0.5° to 1.5° per 100 feet. This results in a long, gently curving section of the wellbore under semi-normal drilling conditions. The technique is often employed to gradually diverge a cluster of wells from one another. Additionally, it is useful for shifting the kickoff point towards the target, thereby reducing the need for more pronounced angles in subsequent directional drilling efforts [1].

2.8.1 Principle and Mechanism of the Jetting Method

At the core of the jetting method is the strategic use of high-pressure water or fluid jets, which are directed towards the formation at the front of the drill bit as is depicted in Figure 2.10. These jets serve to erode, weaken, or fracture the rock or soil ahead of the bit, making it easier for the drill bit to remove the material. This targeted application of fluid not only softens the ground but also can help steer the drill bit in the desired direction [4].

2.8.2 Combination with Other Techniques

Often, the jetting method is used in combination with other drilling techniques, such as rotary drilling or downhole motors. The addition of jetting can enhance the rate of penetration, reduce wear on the drill bit, and improve the ability to steer the wellbore [1, 21].

2.8.3 Environmental Considerations

While the jetting method offers many advantages, it also requires careful consideration of environmental and operational factors. The management of the high-pressure fluid, potential erosion of the formation, and the handling of the resulting slurry are important aspects to consider in the planning and execution of drilling operations using this method.

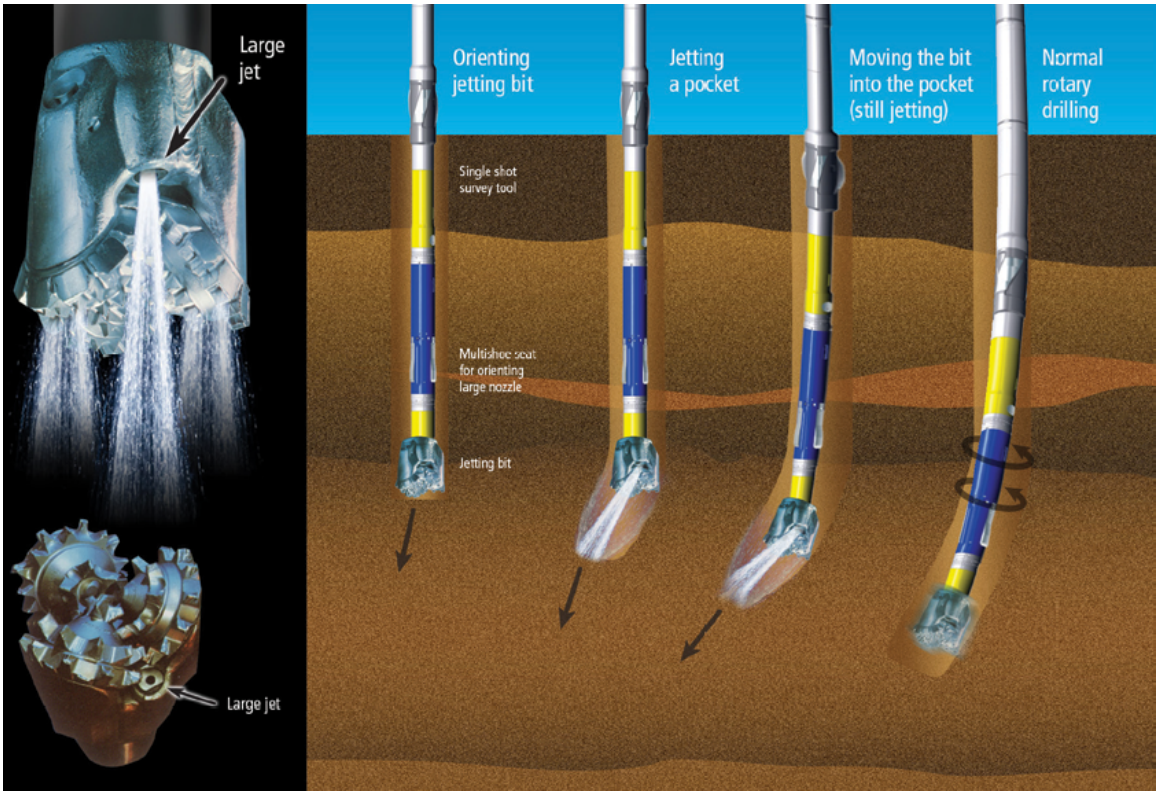

Figure 2.10: The jetting technique [6].

2.9 Neural Network Approach

In recent decades, the field of control systems has witnessed a significant transformation due to the integration of neural network techniques. Traditional control systems, which rely heavily on mathematical models and linear system theories, often struggle to handle the complexities and nonlinearities inherent in many real-world systems. Neural networks, with their capacity for learning and adaptation, offer a powerful alternative. Neural networks are computational models inspired by the human brain, consisting of interconnected nodes (neurons) that work in unison to process information. They are particularly adept at identifying patterns, approximating nonlinear functions, and learning from data, which makes them highly suitable for complex and dynamic control environments. The ability of neural networks to learn from examples allows them to model systems that are difficult to describe mathematically, thereby enhancing the performance and robustness of control strategies [22].

2.9.1 Some Classes of Neural Networks and Their Suitability for Control Systems

We describe here several types of artificial neural networks that are used in modern control systems.

Feedforward Neural Networks (FNN)

FNNs are the simplest type of artificial neural networks Figure 2.11. They consist of an input layer, one or more hidden layers, and an output layer. The connections between the nodes are uni-directional, meaning information moves in one direction: from the input layer, through the hidden layers, and finally to the output layer. There are no cycles or loops in these connections, which is why they are called ”feedforward” networks [23].

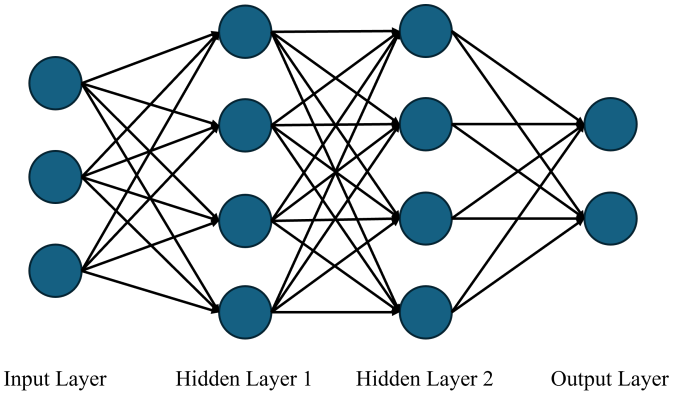

Figure 2.11: FNN Structure

FNNs are primarily used for tasks where the relationship between the input and output is static and does not change over time. Common applications include:

- **Classification**: Assigning input data to one of several predefined categories. For example, classifying images of handwritten digits.
- **Regression**: Predicting a continuous value based on input data. For example, predicting house prices based on various features like size, location, etc.
- **Pattern Recognition**: Identifying patterns in data, such as detecting anomalies in network traffic.

The use of FNNs in control systems is somewhat limited due to their inability to handle temporal dependencies and dynamic changes over time [24]. However, they can still be useful in specific scenarios like

- **System Identification**: FNNs can approximate the behavior of a system based on input-output data. This is useful when the mathematical model of the system is unknown or difficult to derive. The network learns to map the input to the output directly, which can then be used for control purposes.
- **Static Control Law Approximation**: In some control problems, the relationship between the control input and the desired output can be approximated using an FNN. Once trained, the FNN can provide control inputs to achieve the desired system performance.
- **Predictive Control**: Although not as effective as RNNs for dynamic systems, FNNs can still be used in MPC for systems where future outputs depend primarily on current inputs rather than past states.

The limitation for FNNs can be identified as

- No Temporal Dependencies: FNNs cannot capture temporal dependencies since they do not have feedback loops. This makes them unsuitable for tasks that require an understanding of sequences or time-series data.
- Static Behavior Modeling: They are best suited for systems where the relationship between input and output remains constant over time. Dynamic systems with changing behaviors are better handled by recurrent architectures like RNNs, LSTMs, or GRUs.

Recurrent Neural Networks (RNN)

RNNs are a class of neural networks designed to recognize patterns in sequences of data [23,25]. Unlike FNNs, RNNs have connections that form directed cycles, allowing them to maintain a hidden state and handle sequential data Figure 2.12.

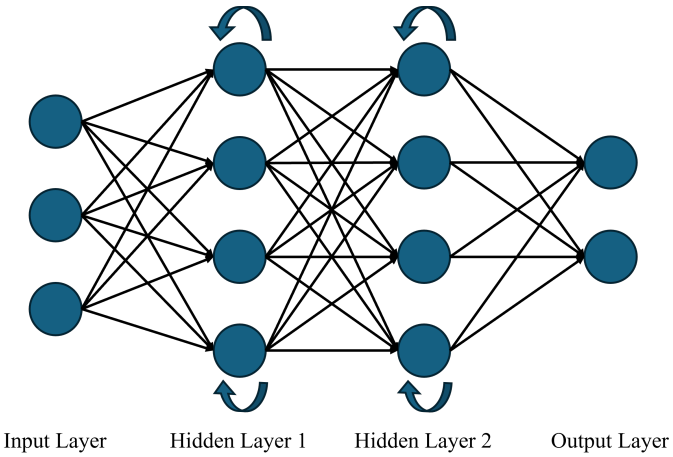

Figure 2.12: RNN Structure

RNNs are particularly effective for tasks where the order of data matters. They can process sequences of arbitrary length by maintaining a hidden state that captures the information from previous time steps.

- Time Series Prediction: Forecasting future values based on historical data, such as stock prices, weather data, and sensor readings.
- Speech Recognition: Converting spoken language into text by understanding the sequential nature of spoken words.
- Sequence Generation: Generating sequences of data, such as text generation, music composition, and handwriting generation.
- Language Modeling: Predicting the next word in a sentence based on the previous words.

RNNs are highly suitable for dynamic systems where control actions depend on the previous states of the system [26]. This ability to retain information over time makes RNNs useful for various control tasks:

- **Dynamic System Modeling**: RNNs can model systems with temporal dependencies, capturing the relationship between current and past states. This is essential for systems where the behavior at any given time depends on its history.

- **Trajectory Prediction**: In robotics and autonomous vehicles, RNNs can predict future trajectories based on past movements, enabling better planning and control.
- **Adaptive Control**: RNNs can adapt to changing system dynamics by continually updating their hidden states based on new input sequences, making them suitable for environments where conditions change over time.

Here are the limitations for RNN,

- **Vanishing and Exploding Gradients**: RNNs can struggle with long-term dependencies due to issues with gradient propagation during training, where gradients can either vanish (become too small) or explode (become too large). This makes it difficult for standard RNNs to learn from long sequences.
- **Computational Complexity**: Training RNNs can be computationally intensive, especially for long sequences, due to the need to backpropagate through time.

Long Short-Term Memory Networks (LSTM)

LSTMs are a specialized type of RNN designed to address the limitations of standard RNNs, particularly the issues of vanishing and exploding gradients. LSTMs introduce a more sophisticated memory mechanism using memory cells and gating units, enabling them to capture long-term dependencies in sequential data [13].

LSTMs excel in tasks that require learning long-term dependencies and handling sequences with complex temporal dynamics. They are widely used in various applications where standard RNNs fall short.

- **Language Modeling and Translation**: Capturing long-range dependencies in text, such as understanding context in sentences for tasks like language translation and sentiment analysis.
- **Speech Recognition**: Converting spoken language into text, where understanding long-term dependencies is crucial for accurate transcription.

- **Time Series Forecasting**: Predicting future values in time series data, such as stock prices, weather patterns, and economic indicators. Sequence Generation: Generating sequences of data, such as composing music, generating text, or creating hand-written digits.

LSTMs are particularly well-suited for control systems that require an understanding of long-term dependencies and delayed effects. Their ability to maintain and update memory over time makes them ideal for dynamic and adaptive control tasks.

- **Predictive Maintenance**: In industrial systems, LSTMs can analyze sequences of sensor data to predict equipment failures before they occur. This helps in planning maintenance activities and avoiding unplanned downtime.
- **Adaptive Control**: LSTMs can adapt to changes in the system dynamics by continuously updating their internal states based on new data. This is useful in environments where the system characteristics change over time, such as in adaptive flight control for drones or adaptive cruise control in vehicles.
- **Trajectory Planning**: In robotics, LSTMs can be used to predict the future trajectory of a robot based on its past movements. This enables more accurate and efficient path planning and obstacle avoidance.
- **Fault Detection and Diagnosis**: LSTMs can monitor system behavior over time and detect anomalies or faults by learning the normal operating patterns and identifying deviations from these patterns.

The LSTM limitation can be described below,

- **Computational Complexity**: LSTMs are more computationally intensive than standard RNNs and other simpler neural network architectures. Training LSTMs on large datasets or long sequences can require significant computational resources.
- **Training Time**: Due to their complexity and the need to learn long-term dependencies, LSTMs typically take longer to train compared to simpler models.

- **Hyperparameter Tuning**: LSTMs have several hyperparameters, such as the number of layers, number of units per layer, and learning rates, which require careful tuning to achieve optimal performance.

GRU Networks

GRU networks are a type of RNN that were introduced to address some of the complexities and inefficiencies associated with LSTM networks. GRUs simplify the LSTM architecture while maintaining, and sometimes improving, performance in certain tasks. They achieve this by combining the functions of the LSTM's multiple gates into fewer gates, reducing computational complexity [13]. A GRU network structure consists of the following:

- **Update Gate**: Combines the input and forget gates of the LSTM into a single gate. This gate decides what portion of the past information (from the previous hidden state) and the current input should be used to update the hidden state.
- **Reset Gate**: Determines how much of the previous hidden state should be forgotten or reset before calculating the new hidden state.
- **Hidden State**: Like LSTMs, GRUs maintain a hidden state that is updated at each time step based on the current input and the previous hidden state, but they do so with fewer gates, making the computation more efficient.

GRUs are particularly effective for tasks that require capturing temporal dependencies in sequential data, like LSTMs, but with potentially lower computational costs. They are used in a variety of applications where sequence data is involved as

- **Time Series Prediction**: Forecasting future values in a sequence of data points, such as financial market trends, weather patterns, and sensor data.
- **Speech Recognition**: Converting spoken language into text by processing the sequential nature of audio signals.

- **Language Modeling and Translation**: Predicting the next word in a sentence and translating text from one language to another by understanding the context within sequences of words.
- **Sequence Generation**: Creating new sequences based on learned patterns, such as generating text, music, or even video frames.

GRUs are well-suited for control systems that require handling temporal dependencies and adapting to changing system dynamics [13]. Their efficiency and simplicity make them an attractive alternative to LSTMs in many control applications.

- **Dynamic System Modeling**: GRUs can model systems where the current output depends on both current and past inputs, which is essential for understanding and controlling dynamic systems.
- **Adaptive Control**: In environments where system parameters change over time, GRUs can adapt by updating their hidden states based on new input sequences. This makes them useful for adaptive control applications, such as real-time process control and adaptive flight control.
- **Predictive Maintenance**: GRUs can analyze time-series sensor data to predict equipment failures and schedule maintenance before failures occur, reducing downtime and maintenance costs.
- **Trajectory Planning**: In robotics and autonomous vehicles, GRUs can be used to predict future movements based on past trajectories, aiding in path planning and obstacle avoidance.

The Advantages over GRUs will be addressed as

- **Computational Efficiency**: GRUs have fewer gates and parameters than LSTMs, which makes them computationally less expensive and faster to train.
- **Simplicity**: The simpler architecture of GRUs often makes them easier to implement and tune compared to LSTMs, without significantly compromising performance.
- **Performance**: In some cases, GRUs perform as well as or better than LSTMs, particularly on datasets where the complexity of LSTMs is not necessary.

Additionally, GRU networks have some limitation as

- **Long-Term Dependencies**: While GRUs are designed to handle long-term dependencies better than standard RNNs, they may not always perform as well as LSTMs in tasks requiring extremely long-term memory.
- **Hyperparameter Sensitivity**: Despite having fewer parameters than LSTMs, GRUs still require careful tuning of hyperparameters, such as learning rates and the number of units in each layer, to achieve optimal performance.

The GRU network (Figure 2.13) used in this system is based on the following governing equations [25]:

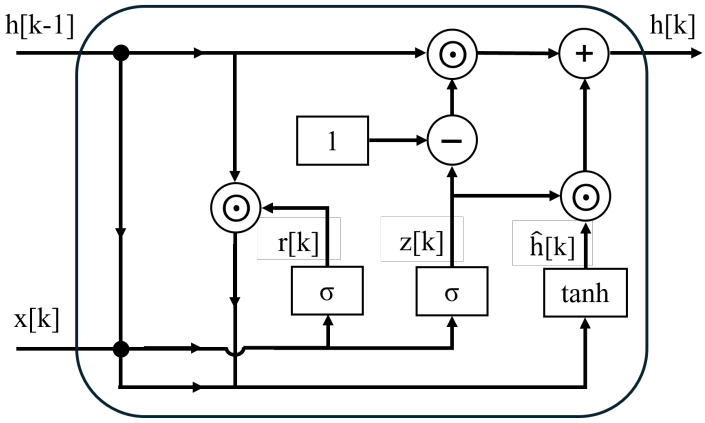

Figure 2.13: GRU variation of LSTM.

$$
\begin{aligned}
r[k] &= \sigma(W_r x[k] + U_r h[k-1] + b_r) \\
z[k] &= \sigma(W_z x[k] + U_z h[k-1] + b_z) \\
\hat{h}[k] &= \tanh(W_h x[k] + U_h(r[k] \odot h[k-1]) + b_h) \\
h[k] &= (1 - z[k]) \odot h[k-1] + z[k] \odot \hat{h}[k],
\end{aligned}
\tag{2}
$$

where $r[k]$, $z[k]$, and $h[k]$ represent the reset gate, the update gate, and the hidden state respectively, $x[k]$ is the input vector at step k. The matrices W_r, W_z, and W_h are weight matrices and U_r, U_z, and U_h are recurrent weight matrices. The neural network bias terms are b_r, b_z, b_h, σ is the sigmoid activation function, and $\odot$ denotes element-wise multiplication.

2.10 Conclusion

This chapter establishes the technical landscape of advances in directional drilling and introduces some classes of neural networks, including an explanation of the GRU equation.

Chapter 3

System Modeling

3.1 Overview

This chapter develops the detailed physical model of the directional drilling system used as the foundation for controller design and simulation. The model combines an FEM representation of the drill string with a nonlinear bit–rock interaction law. The resulting equations of motion are expressed in control–affine state–space form suitable for both model–based and learning–based control strategies.

3.2 Drill–String Dynamics

The drill string is modeled as a slender, elastic shaft that undergoes torsional and axial deformation during drilling with an FEM which is a commonly employed modeling approach for down-hole drilling systems. It offers numerous benefits in representing geometrically complex structures with intricate boundary conditions. Hence, an FEM dynamics model for control design has been developed. Illustrated in Figure 3.1, the complete drill string (Segment AD) can be effectively modeled using the finite element method, employing a finite number of nodes [7, 8].

3.2.1 Finite Element Model

The local motion vector U_i^l for each node i is

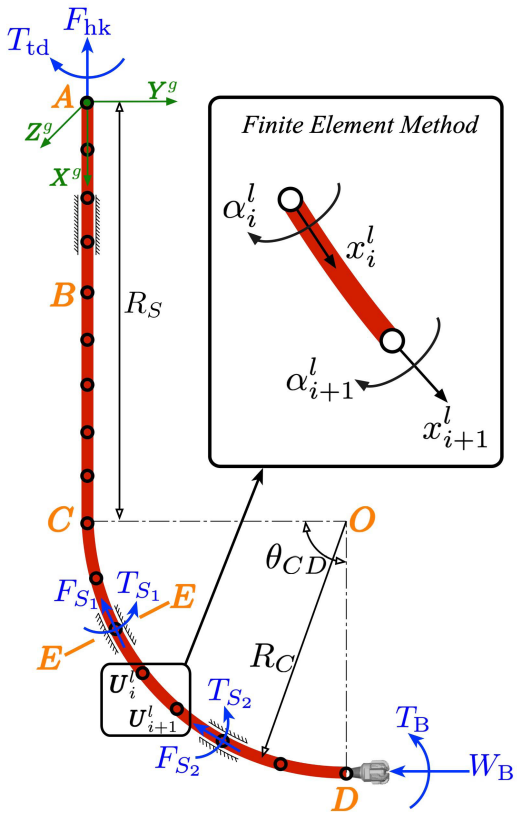

Figure 3.1: Schematics of the computationally efficient directional drilling dynamics model using the finite element method [7].

$$U_i^l = \begin{bmatrix} x_i^l(t) \\ \alpha_i^l(t) \end{bmatrix}. \tag{3}$$

Where superscript l denotes the local coordinate variable; subscript $i \in [1, 2, \ldots, N]$ denotes the ith node of the FEM nodes, x_i^l and α_i^l represent the local axial displacement and the local torsional displacement for node i, and the orientation of variables above can be found in Figure 3.1. Dynamics for an element between nodes i and $i+1$ is given by

$$M_l \ddot{U}_{i,i+1}^l + D_l \dot{U}_{i,i+1}^l + K_l U_{i,i+1}^l = F_{i,i+1}^l, \tag{4}$$

where $U_{i,i+1}^l = [U_i^l, U_{i+1}^l]^T \in R^4$, M^l, D^l, $K^l \in R^{4\times4}$ are the inertia, damping, and stiffness matrices respectively, under the local coordinates for nodes i and $i+1$, and $F_{\{i,i+1\}}^l \in R^{4\times1}$ is the local external force and torque vector exerted on node i and node $i+1$.

Stiffness in drilling is a measure of how rigid the drill string is and how it affects the direction and quality of the hole. Stiffness in drilling depends on several factors, such as the borehole diameter, the bench height, the burden, the bit side force, the bit tilt angle, the formation properties, and the stabilizer types and placement. There is no definitive value range for stiffness in drilling, as different applications may require different levels of stiffness. However, some general guidelines are [27]:

- A higher stiffness ratio (bench height divided by burden) leads to better performance and less oversize, but there is a limit to how high it can be.
- A lower stiffness ratio may cause more deviation and vibration, which can reduce the hole quality and bit life.
- A pre-load pressing force can improve the stiffness of the drilling plane and enhance the stability and accuracy of the hole.
- The optimal value of the pre-load pressing force depends on the robot drilling posture and the thrust force.
- The stiffness of the drill string can be increased by adding more stabilizers, but this can also increase the friction and torque.
- The type of stabilizer should be chosen based on the geology and the wear resistance.

Inertia in drilling is a measure of how resistant the drill string is to changes in its rotational motion. Inertia in drilling depends on several factors, such as the mass, length, and cross-sectional area of the drill string, the rotational speed, the torque, and the friction with the borehole wall. There is no definitive value range for inertia in drilling, as different applications may require different levels of inertia. However, some general guidelines are [28]:

- A higher inertia ratio (mass moment of inertia divided by mass) leads to better stability and less vibration, but also increases the energy consumption and the risk of stick-slip
- A lower inertia ratio may cause more fluctuations and oscillations, which can reduce the drilling efficiency and quality

- The inertia of the drill string can be reduced by decreasing the mass or increasing the cross-sectional area of the drill string
- The inertia of the drill string can also be affected by the type and placement of stabilizers, which can increase or decrease the contact area and friction with the borehole wall

Rayleigh damping is assumed for D^l that is given by

$$D^l = \alpha_0 M^l + \beta_0 K^l, \tag{5}$$

where α_0 and β_0 are constants.

The local coordinate motion vector $U^l_{i,i+1}$ can be related to the global coordinate motion vector $U^g_{i,i+1}$ by

$$U^l = R_i U^g, \tag{6}$$

where $R_i = \text{diag}\,[T_i, T_i, T_i, T_i]$, and $T_i = \cos(x^l_i, X^g)$. Therefore, (4) can be expressed in the augmented global coordinates by

$$M^g \ddot{U}^g_{i-i+1}(t) + D^g \dot{U}^g_{i-i+1}(t) + K^g U^g_{i-i+1}(t) = F^g_{i-i+1}(t), \tag{7}$$

where $M^g = \sum_{i=1}^{N} R_i^T M^l R_i$, $K^g = \sum_{i=1}^{N} R_i^T K^l R_i$, $D^g = \sum_{i=1}^{N} R_i^T D^l R_i$, and M^g, K^g, D^g are the global inertia, damping, and stiffness matrices, respectively. Here $U^g \in R^{N_0 \times 1}$, ($N_0 = 2N$) is the state vector that includes all the nodes from point A to point D, and $F^g_{\text{ext}} \in R^{N_0 \times 1}$ denotes the global external force vector, which includes surface control inputs (hook load and top drive torque), drill string/well-bore contact force and torque, bit/rock interaction force and torque, and the gravitational force.

To model the system dynamics for control purposes, the finite element model is expressed in a state-space control-affine form:

$$\dot{z}(t) = A^g z(t) + B^g F^g_{\text{ext}}(t), \tag{8}$$

where $A^g = \begin{bmatrix} 0 & I \\ -(M^g)^{-1}K^g & -(M^g)^{-1}D^g \end{bmatrix}$, $B^g = \begin{bmatrix} 0, (M^g)^{-1} \end{bmatrix}^T$, and $z(t) = [z_1, z_2, \cdots, z_{2N_0}]^T$ represent the state vector of the system, which includes the positions and torsional of the drill string elements. The external force $F^g_{\text{ext}}(t)$ is given by

$$F^g_{\text{ext}}(t) = F(z(t)) + Hu(t), \tag{9}$$

where $F(z) \in R^{N_0}$ is the external force except for the control input vector, $u \in R^2$ is the control input vector that contains hook load force and top drive torque, and $H \in R^{N_0 \times 2}$ is the matrix corresponding to the input $u(t)$:

$$H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T_{2\times N_0}. \tag{10}$$

3.2.2 Bit-Rock Interaction

A polycrystalline diamond compact (PDC) bit/rock interaction model developed by [29] is given by

$$\begin{cases} W_B(t) = W_{B_f}(t) + W_{B_c}(t) \\ T_B(t) = T_{B_f}(t) + T_{B_c}(t), \end{cases} \tag{11}$$

where W_B and T_B are weight-on-bit and torque-on-bit, which can be defined as their frictional components W_{B_f} and T_{B_f}, and cutting components W_{B_c} and T_{B_c} as

$$\begin{cases} W_{B_f}(t) = \gamma l_w R_b \dfrac{d(t)}{d^*} \\ W_{B_c}(t) = \varsigma \epsilon R_b d(t) H(d(t)) \\ T_{B_f}(t) = 0.5\,\mu\,\delta\,R_b\,W_{B_f}(t) \\ T_{B_c}(t) = 0.5\,\epsilon\,R_b^2\,d(t)\,H(d(t)), \end{cases} \tag{12}$$

where γ stands for the maximum contact pressure at the wear-flat interface, l_w is the equivalent wear-flat length, R_b represents the bit radius, ς is the bit geometry constant, ϵ is the intrinsic specific

energy, μ is the coefficient of friction at the wear-flat rock interface, δ represents the geometric parameter of the bit, d^* is the threshold of the axial depth of cut (DOC), and d is the axial depth of cut. The axial depth is given by

$$\begin{cases} d(t) = n_0 d_n(t) = n_0[x_N^l(t) - x_N^l(t-\tau_0)] \\ \alpha_N^l(t) - \alpha_N^l(t-\tau_0) = \dfrac{2\pi}{n_0}, \end{cases} \tag{13}$$

where n_0 is the number of PDC bit blades and τ_0 is the time delay created by the adjacent cutting blade. In [7, 8], the state-dependent representation of the axial depth of cut (13) is estimated by

$$d(t) = \frac{2\pi\, \dot{x}_N^l(t)}{\dot{\alpha}_N^l(t)}. \tag{14}$$

The bit's torsional velocity, $\dot{\alpha}_N^l(t)$, must remain nonzero to avoid singularities in this approximation. This condition can be ensured by properly controlling the system to prevent operation at that specific velocity [7, 8].

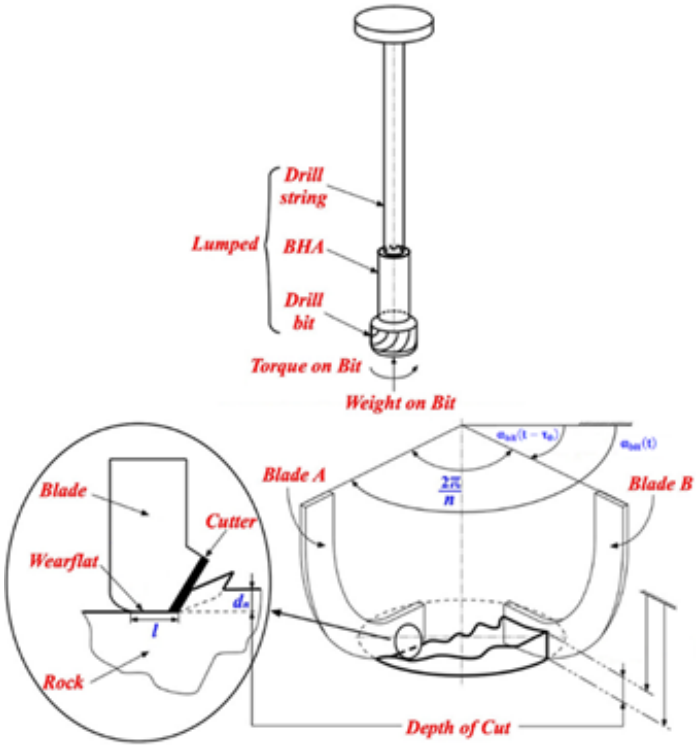

Figure 3.2: Weight-on-Bi $W_B(t)$, Torque-on-Bit $T_B(t)$ and Depth of Cut $d(t)$ [8].

3.3 Control–Oriented Formulation

The measurable outputs include the inclination and azimuth angles of the borehole, which are derived from the bit orientation. The output equation is given by

$$\mathbf{y}(t) = C\,\mathbf{z}(t) + \mathbf{v}(t), \tag{15}$$

where C is the measurement matrix and $\mathbf{v}(t)$ represents sensor noise. The model (8) and (15) forms the basis for subsequent control design.

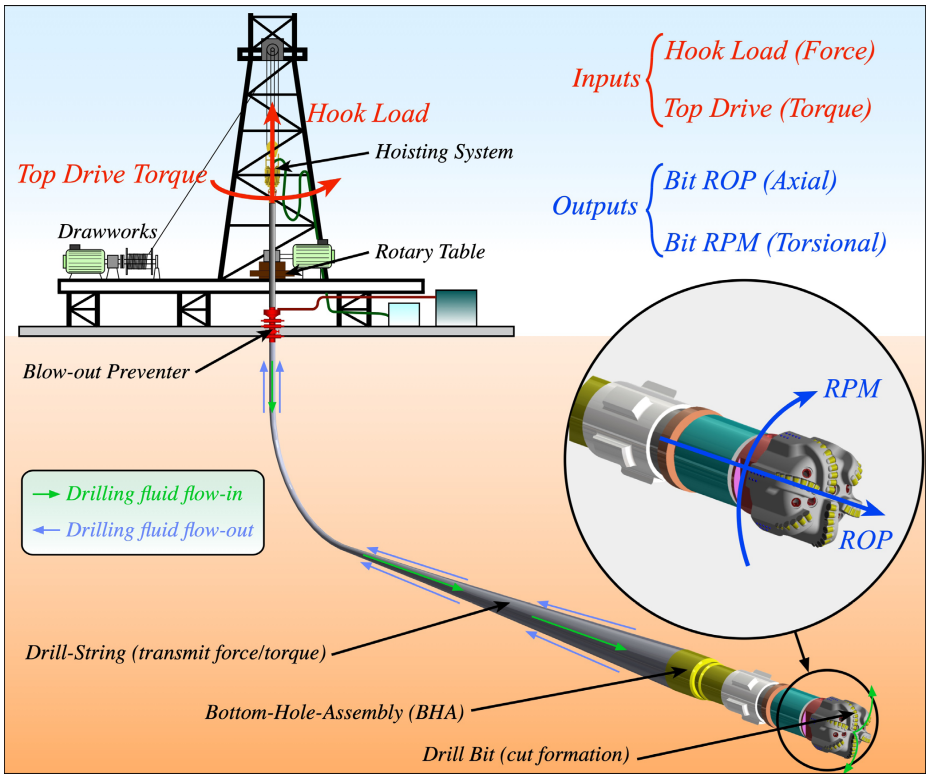

Figure 3.3: Block diagram of the drilling system model, including surface actuation, drill–string dynamics, and bit–rock interaction [7].

To discretize the state space model description it is assumed $F(z(t))$ as a system uncertainty that can be incorporated into the model at a later stage, the resulting continuous-time dynamics are discretized using the Zero-Order Hold (ZOH) method. Under the assumption of ZOH, the input signal $u(t)$ remains constant within each sampling interval of duration T_s, such that $u(t) = u[k]$ for $kT_s \leq t < (k+1)T_s$. The corresponding exact discrete-time state-space representation is given by

$$z[k+1] = A_d^g z[k] + B_d^g u[k], \tag{16}$$

where $A_d^g = e^{A^g T_s}$ and $B_d^g = \int_0^{T_s} e^{A^g \tau} B^g H d\tau$.

3.4 Parameter Values and Simulation Setup

The parameters used for numerical simulation are summarized in Table 3.1. The stiffness and damping coefficients are randomly varied within engineering ranges to emulate formation heterogeneity.

Table 3.1: Nominal model parameters used for simulation.

Symbol	Value	Unit
K^l	$\approx 10^7$	N/m
M^l	$\approx 10^4$	kg/m^2
σ	45	MPa
l_w	3.6×10^{-3}	m
R_b	0.15	m
ς	0.64	–
ϵ	77	MPa
μ	0.7	–
δ	1	–
d^*	1.2×10^{-3}	m
α_0	0.03	–
β_0	0.05	–

3.5 Summary

This chapter developed a comprehensive FEM of the directional drilling system, incorporating Rayleigh damping and nonlinear bit–rock interaction. The model was reformulated into a discrete–time control–affine form suitable for both classical and learning–based control design. It serves as the simulation environment for the adaptive GRU–PID and GRU–based MPC surrogate controllers presented in the following chapters.

Chapter 4

Adaptive PID Control Using GRU Learning

4.1 Overview

This chapter presents the first control strategy developed in this thesis: an adaptive PID controller assisted by a GRU neural network. The method preserves the well–understood structure and interpretability of a PID controller while introducing data–driven adaptation of its gains in real time. The GRU serves as an online estimator of the local input–output Jacobian of the drilling dynamics, providing the sensitivity information required to adjust the gains for changing operating conditions.

4.2 Conventional PID Control

The conventional PID controller computes the control input as

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau + k_d \dot{e}(t), \tag{17}$$

where $e(t) = y_{\mathrm{ref}}(t) - y(t)$ is the tracking error and k_p, k_i, k_d are proportional, integral, and derivative gains, respectively. With digitally realized form at sampling T_s:

$$u[k] = K_P\, e[k] + K_I\, T_s \sum_{i=0}^{k} e[i] + K_D\, \frac{e[k] - e[k-1]}{T_s}. \tag{18}$$

Although simple to implement, the fixed–gain PID struggles to maintain optimal performance when system parameters such as bit stiffness or formation hardness vary during drilling. It's needed to make (18) by considering the discretized PID controller as bellow

$$PID(z) = K_P + K_I\, T_s \frac{1}{z-1} + K_D\, \frac{n}{1 + n\, T_s \frac{1}{z-1}}, \tag{19}$$

where n is the derivative filter bandwidth.

4.3 Motivation for Adaptation

The drilling system modeled in Chapter 3 exhibits time–varying dynamics due to changes in the properties of the rock, friction coefficients, and fluid pressure. An adaptive controller capable of adjusting its gains online can maintain consistent performance under these variations. Gradient–based adaptation is attractive, but requires knowledge of the Jacobian $\partial y(t)/\partial u(t)$, which is generally unavailable. A GRU network is used (Figure 4.1) to approximate this Jacobian from temporal input–output data.

4.4 Adaptive Gain Update Law

Let the instantaneous tracking loss be

$$e[k] = y_{REF}[k] - y[k]. \tag{20}$$

The goal is to minimize $e[k]$ by adjusting the PID gains in the direction of the negative gradient. Using the chain rule and the Jacobian estimate $\hat{J}$, the gradient of tracking error with respect to each gain is approximated as

$$\Delta K_{P,I,D}[k] = \alpha \frac{\partial e[k]}{\partial K_{P,I,D}}, \tag{21}$$

where α is the learning rate. The partial derivative $\frac{\partial e[k]}{\partial K_{P,I,D}}$ is given by

$$\frac{\partial e[k]}{\partial K_{P,I,D}} = \frac{\partial e[k]}{\partial \hat{y}[k]} \times \frac{\partial \hat{y}[k]}{\partial u[k]} \times \frac{\partial u[k]}{\partial K_{P,I,D}}, \tag{22}$$

where $e[k]$ is the error between the reference y_{REF} and the system output y, $\hat{y}[k]$ is GRU prediction at the step k, $u[k]$ is the control signal, and $K_{P,I,D}$ are the proportional, integral, and derivative gains of the PID controller. The term $\frac{\partial \hat{y}[k]}{\partial u[k]}$ is the estimated Jacobian matrix ($\hat{J}$) given by

$$\hat{J} = \frac{\partial \hat{y}[k]}{\partial u[k]} = \begin{bmatrix} \frac{\partial \hat{y}_1[k]}{\partial u_1[k]} & \frac{\partial \hat{y}_1[k]}{\partial u_2[k]} \\ \frac{\partial \hat{y}_2[k]}{\partial u_1[k]} & \frac{\partial \hat{y}_2[k]}{\partial u_2[k]} \end{bmatrix}. \tag{23}$$

Based on equations (20) through (23), the PID gains are updated as follows

$$K_{P,I,D}[k] = K_{P,I,D}[k-1] + \Delta K_{P,I,D}[k] \tag{24}$$

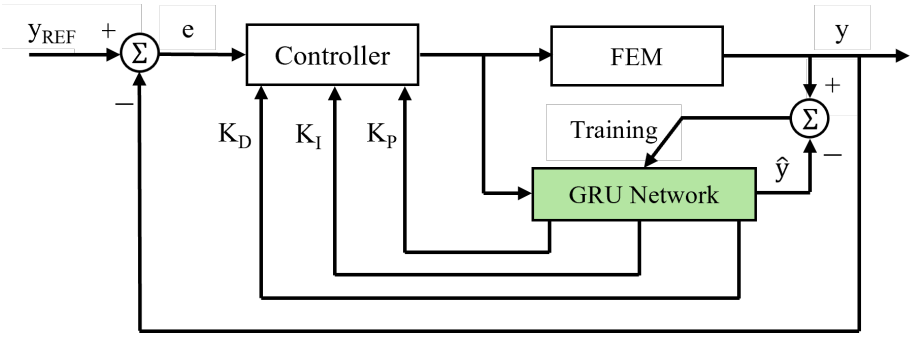

Figure 4.1: Architecture of the GRU–assisted adaptive PID controller.

4.4.1 Hyperparameter Optimization for GRU Model Learning

The selection of appropriate hyperparameters is critical for the adequate performance of a GRU neural network in learning the dynamics of directional drilling [26,30,31]. The key hyperparameters and their impact on the model will be identified below.

Hidden layers and hidden size: The depth of a GRU network is determined by the number of hidden layers it contains. Adding more layers allows the model to capture increasingly complex

hierarchical patterns. However, excessive depth can result in overfitting or higher computational demands, especially in real-time control applications. The hidden size, which represents the number of units in each hidden layer, affects the model's ability to learn and preserve temporal dependencies. A smaller hidden size may cause underfitting, limiting the model's ability to represent system dynamics, while a larger size enhances learning capacity but may lead to overfitting if training data are limited or lack diversity.

Dropout regularization: Dropout helps mitigate overfitting by randomly disabling a portion of neurons during the training process. In GRU models, dropout rates between 0.2 and 0.5 are generally effective, depending on the complexity of the underlying dynamics and the diversity of the dataset. However, excessively high dropout values can reduce model learning capacity and cause underfitting, particularly in smaller network architectures.

Learning rate: The learning rate determines the step size used to update the weights during training. It has a direct impact on how quickly and effectively the model converges. A high learning rate may cause unstable training and prevent convergence, whereas a low learning rate promotes stable convergence, but can significantly slow down the training process.

Hyperparameter tuning: A structured hyperparameter optimization approach, such as grid search or Bayesian optimization, should be employed. This process helps the model achieve an optimal balance between training accuracy and its ability to generalize effectively to unseen data or conditions.

4.4.2 Importance of Rich Data For Training

To effectively train a GRU network for a model, the dataset must meet certain qualities:

Diversity in input conditions: The dataset should include a various range of operational scenarios, including different drilling depths, rock formations, and tool configurations. This diversity enables the model to learn robust patterns that generalize well to real-world conditions.

Temporal continuity: Since GRUs are particularly effective in learning temporal dependencies, training data should include sequential patterns that accurately represent the dynamics of directional drilling. Missing or unevenly sampled time steps can negatively affect the model's performance.

Data quality and preprocessing: Sensor data used for training should be carefully preprocessed to eliminate noise, outliers, and missing values. Applying standardization or normalization to input features ensures consistent scaling across variables, promoting more stable and efficient model learning.

Balanced representation: The dataset should be free from biases, such as overrepresentation of particular geological conditions or operational scenarios, as these could distort the model's predictions and reduce its generalizability.

4.4.3 Integrating Hyperparameter Tuning and Data Insights

By iteratively tuning hyperparameters and assessing performance metrics such as loss, accuracy, and robustness, the GRU model can be effectively optimized to capture system dynamics. Combining cross-validation for hyperparameter assessment, real-time feedback for adaptive learning, and continuous dataset expansion to include a variety of operational scenarios ensures the model remains well-adapted for deployment in directional drilling systems.

4.4.4 Implementation Details

The closed-loop framework comprises the following components [32]:

- A GRU-based neural network model that learns the drilling process dynamics, continuously updates using FEM-generated data
- An FEM system that represents the mechanical and structural characteristics of the drilling setup, providing both the system response and training data for the GRU model
- An adaptive PID controller that refines its control law through error-based training. The GRU network facilitates real-time tuning of the PID controller.

Figure 4.1 shows the interactions within this real-time closed-loop framework, illustrating how the GRU network and FEM model jointly enhance the control strategy through adaptive learning and dynamic compensation for system uncertainties.

Algorithm 1 Adaptive GRU–PID Control Algorithm

Require: initialize GRU weights and biases

1: **for** $step = 1 \rightarrow N$ **do** ▷ N the last node
2: Measure $y[k]$, compute error $e[k]$
3: Compute control input using (18)
4: Update GRU hidden state
5: Estimate Jacobian $\hat{J}$
6: Update gains using (24)
7: Apply $u[k]$ to drilling system
8: **end for**

4.5 Simulation Results

The adaptive GRU–PID controller was evaluated using the FEM described in Chapter 3. The reference trajectory consisted of a sequence of Axial Displacement and torsional factor representative of a horizontal drilling path. For comparison, a fixed–gain PID controller is implemented under identical conditions.

The PID controller for FEM model is given by

$$\begin{bmatrix} PID_1(z) & 0 \\ 0 & PID_2(z) \end{bmatrix} = \begin{bmatrix} K_{P1} + K_{I1}\,T_s \frac{1}{z-1} + K_{D1}\,\frac{n_1}{1+n_1\,T_s\frac{1}{z-1}} & 0 \\ 0 & K_{P2} + K_{I2}\,T_s \frac{1}{z-1} + K_{D2}\,\frac{n_2}{1+n_2\,T_s\frac{1}{z-1}} \end{bmatrix}. \quad (25)$$

A simulation study will be carried out with $n_1 = 100$ and $n_2 = 100$.

4.5.1 Tracking Performance

The system's performance was evaluated by comparing a conventional fixed-gain PID controller with the GRU-based adaptive PID controller. Figure 4.5 and Figure 4.3 demonstrate that the adaptive controller achieved a faster response and minimized steady-state error. The GRU-based system also produced smoother control actions and reduced fluctuations compared to the fixed PID controller.

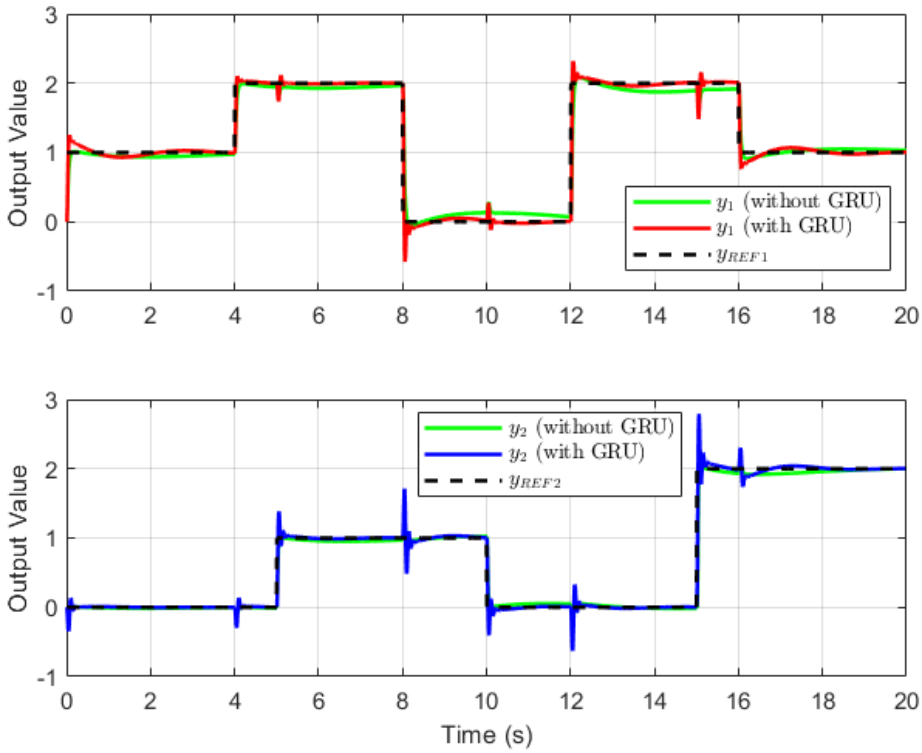

Figure 4.2: Closed-loop system response.

4.5.2 Gain Evolution

As shown in Figure 4.4, the GRU-predicted system output closely followed the FEM response, maintaining low prediction errors even during dynamic transitions. Figure 4.5 illustrates the real-time adjustment of PID gains, emphasizing the controller's adaptability in responding to changing system conditions.

In summary, the GRU-based adaptive PID controller outperformed the conventional approach by improving trajectory tracking, minimizing control signal variations, and enhancing overall system stability.

4.6 Conclusion

This study presented a GRU-based adaptive PID control system integrated with FEM to enable real-time trajectory control in directional drilling operations. The proposed framework continuously refines PID parameters based on GRU predictions and real-time error gradients, resulting in

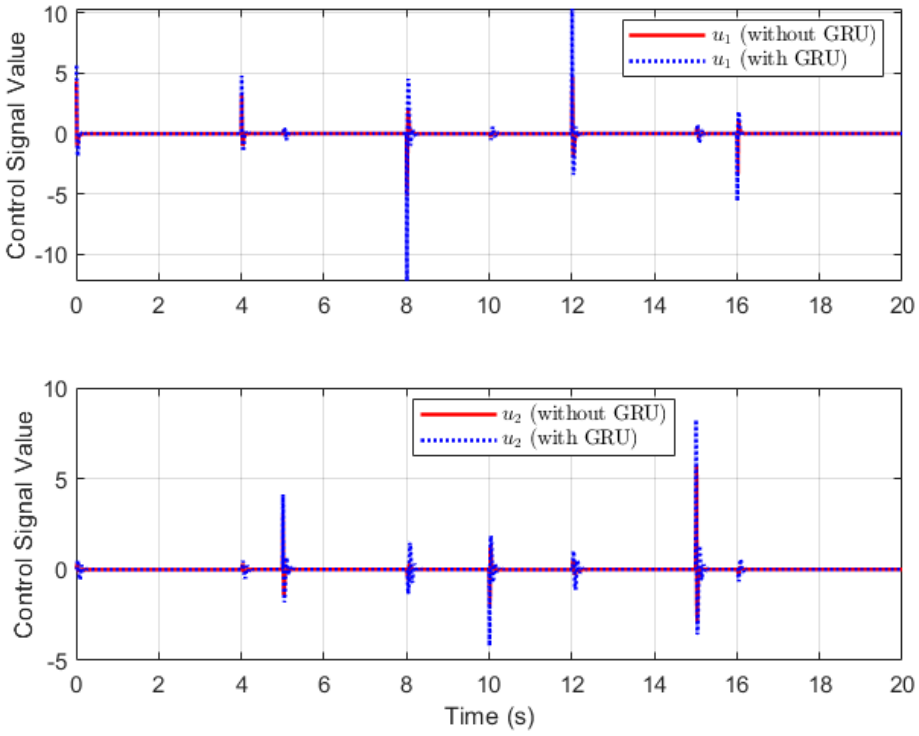

Figure 4.3: Control signal.

improved trajectory tracking and reduced control effort compared to conventional fixed-gain PID controllers.

Simulation results demonstrated that the GRU-based adaptive control approach delivers faster response times, lower steady-state errors, and smoother control actions, even under rapidly changing downhole conditions. The integration of FEM modeling with deep learning provides an innovative mechanism for dynamically capturing drilling system behavior and proactively adjusting control strategies.

However, the method has certain limitations. The accuracy and generalizability of the GRU model are strongly based on the diversity, completeness, and quality of the training dataset. Insufficient representation of specific operating scenarios during training can lead to degraded control performance or delayed adaptation.

Future work will focus on broadening the training dataset to include more extreme and varied drilling conditions, performing experimental validation in real-world environments, and exploring

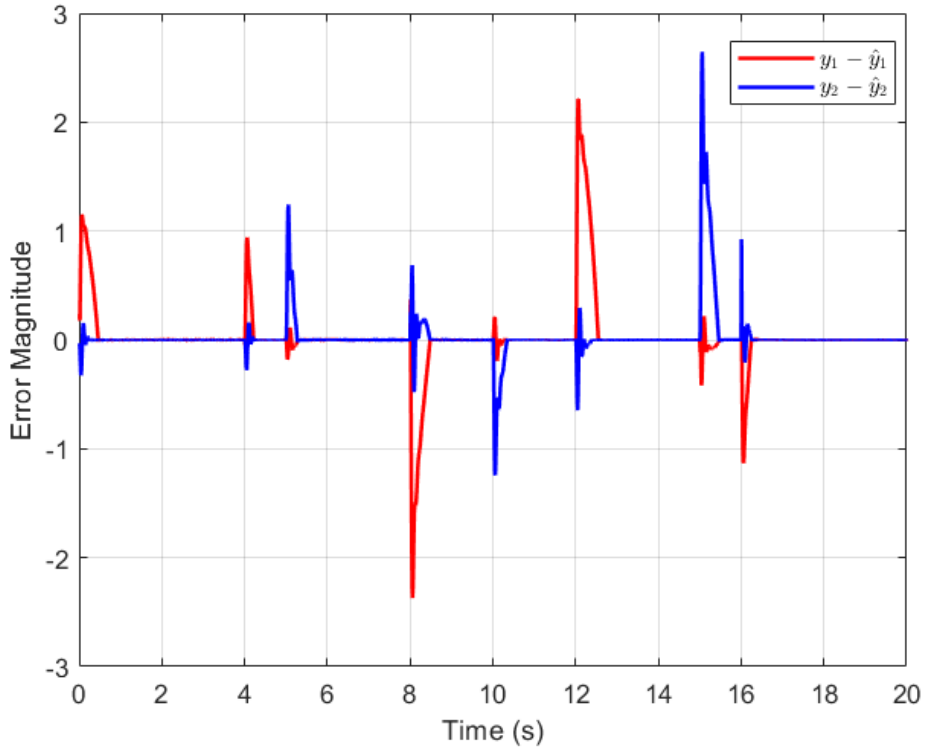

Figure 4.4: Prediction error between system response and GRU-predicted output.

hybrid approaches that combine GRU-based control with deep reinforcement learning or robust H-infinity control techniques. Additionally, efforts will be directed toward real-time computational optimization to reduce onboard processing demands and improve deployment feasibility in embedded downhole systems.

In general, the GRU-based adaptive control framework marks a substantial step toward intelligent, self-adaptive control solutions that improve efficiency, minimize operational risks, and advance autonomy in directional drilling applications.

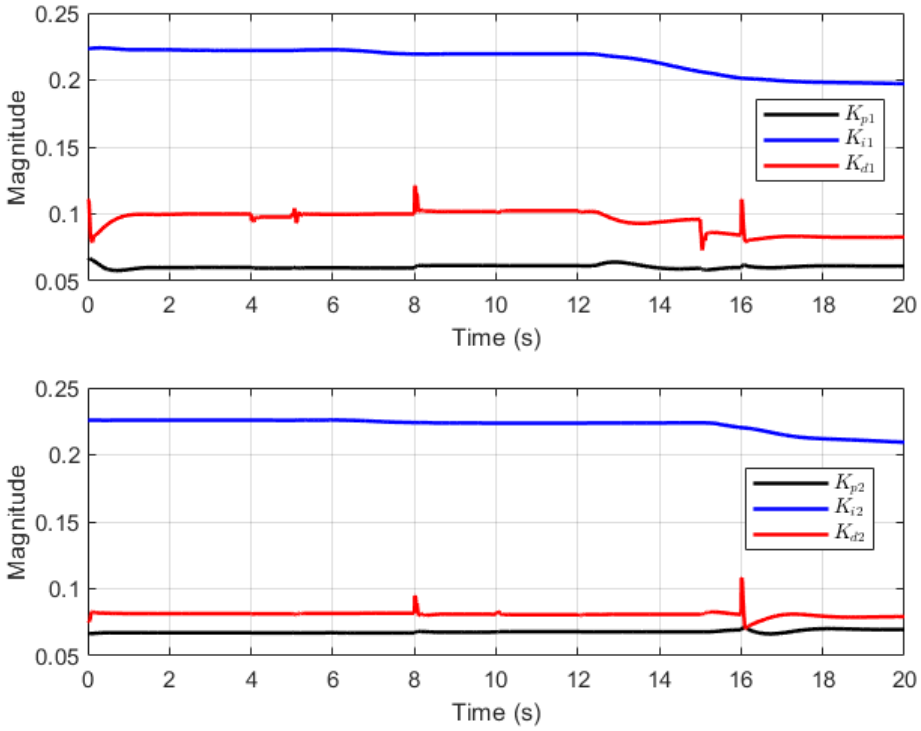

Figure 4.5: Evolution of the adaptive PID gains during simulation.

Chapter 5

GRU-Based MPC Emulation Control

5.1 Overview

MPC provides a principled framework for optimizing control inputs over a finite horizon while respecting system constraints. In drilling applications, however, an online solution of the MPC optimization problem at every sampling instant is computationally infeasible for embedded downhole processors. This chapter presents a GRU-based MPC surrogate (GMPC) that emulates the input–output behavior of a conventional MPC controller using recurrent neural networks. The surrogate is trained offline on MPC data and adapted online through a GRU-based estimator, enabling real-time deployment with minimal computation cost.

5.2 Model Predictive Control Background

MPC has demonstrated strong effectiveness in drilling systems because of its capability to optimize performance while respecting both input and output constraints [23, 33]. The control law is derived by solving

$$\min \sum_{k=0}^{N_P-1} (y_{REF}[k] - y[k])^T Q (y_{REF}[k] - y[k]) + u[k]^T R u[k] + \Delta u[k]^T S \Delta u[k], \tag{26}$$

subject to constraints on discrete control inputs ($u[k]$) and discrete system outputs ($y[k]$), where Q, R and S are weighting matrices, $\Delta u[k]$ is defined as the change in discrete control input between the current step k and the previous time step $k-1$, $y_{REF}[k]$ is the discrete reference trajectory, and N_P represents the prediction horizon.

This optimization process achieves a balance between precise trajectory tracking and smooth actuation; however, it is constrained by the classical challenge of high computational demand. Prior studies in industrial process control [14, 23, 24, 34], electromechanical systems [35, 36], and drilling trajectory control [7, 15, 30] have emphasized this limitation, which confines the application of MPC primarily to slower systems or those equipped with high-performance computational resources.

5.2.1 GRU-Based Predictive Controller Design

To alleviate the computational load, a GRU network is trained offline using input–output data generated from MPC simulations. After training, the GRU serves as an approximation of the control policy, effectively removing the need for online constrained optimization. Compared with static feedforward networks [23] or deep LSTM architectures [13, 14], GRUs offer an optimal balance between computational efficiency and temporal learning capability, making them particularly suitable for real-time drilling control applications [11]. The design framework is guided by the following key considerations:

- Emulate MPC control behavior at a significantly lower computational cost [26].
- Continuously update GRU parameters during drilling operations to sustain performance amid varying downhole dynamics [11].

5.3 Adaptive GRU-Based Controller Methodology

The proposed approach utilizes a GRU neural network trained to emulate the MPC control law, augmented with an adaptive updating mechanism that adjusts in real time to accommodate variations in drilling conditions.

5.3.1 Data Collection and Preprocessing

Training data were generated by simulating the MPC controller in various drilling scenarios, including changes in the reference trajectory, the bit-rock interaction parameters and the dynamics of the drill string. Each data set included system outputs, control inputs, and tracking errors, following the standard approach of using MPC-generated data to train neural surrogates [26, 37], as shown in Figure 5.1.

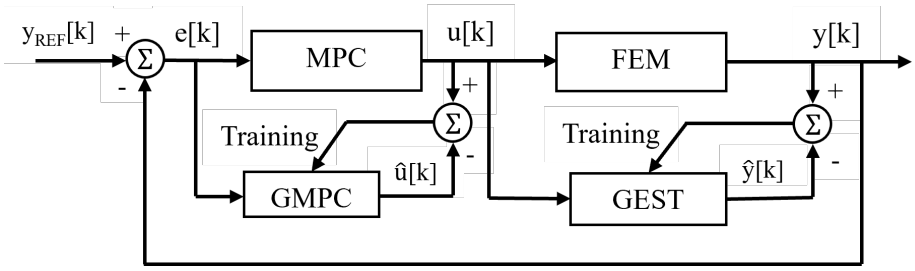

Figure 5.1: Training GMPC and GEST networks using collected data

During the training phase, the collected dataset was divided into 85% for training and 15% for testing, ensuring adequate excitation across various operating conditions while maintaining an independent test set for unbiased performance evaluation. This partitioning approach has proven effective in prior MPC–NN surrogate training studies [26, 37]. The preprocessing and scaling procedures ensured that the outputs of the GRU controller were properly aligned with the physical constraints of the drilling system, while preserving the learning benefits provided by normalized activation functions.

5.3.2 GRU Network Architecture

Two separate GRU-based architectures were developed: one dedicated to system output estimation and the other to approximating the MPC control law. Both networks were trained using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) as loss functions, optimized with the Adam algorithm, which is commonly employed in neural network–based predictive control applications [13, 14, 23, 26].

GRU Estimation (GEST) Network

The GEST network architecture consists of an input layer with n_{gr} neurons that process h_{ge} steps of historical control actions and prediction errors, followed by l_{gh} recurrent hidden layers containing n_{gh} neurons to model temporal dependencies. The output layer comprises two neurons responsible for generating the estimated outputs of the system:

$$\hat{Y}[k] = F_{\text{GEST}}\big(U[k], \ldots, U[k - h_{ge} + 1], \Delta\hat{Y}[k-1], \ldots, \Delta\hat{Y}[k - h_{ge}]\big), \tag{27}$$

where $U[k]$ is the flattening of $u[k] = [u_{11}[k], u_{21}[k]]^T$ and $\hat{Y}[k]$ is the flattening of $\hat{y}[k] = [\hat{y}_{11}[k], \hat{y}_{21}[k]]^T$, $\Delta\hat{Y}[k] = \hat{Y}[k] - \hat{Y}[k-1]$. This structure allows the estimator to incorporate both the past input dynamics and recent changes in output behavior, which is particularly relevant for nonlinear drilling systems modeled by FEM method [7, 15, 30].

The performance of the GEST network was further assessed within a closed-loop configuration. Figure 5.2 presents the estimated outputs $\hat{y}_1$ and $\hat{y}_2$ alongside the actual closed-loop outputs y_1 and y_2 under pulse excitation with a white noise with maximum 8% of amplitude y_1 and y_2. The noisy signals are filtered by a low pass filter by using MATLAb Toolbox then are used for training purpose. The estimated trajectories closely follow the real system responses, exhibiting only slight deviations during switching events. Figure 5.3 quantifies this accuracy by showing the estimation error $y - \hat{y}$ which remains on the order of 10^{-3} across both channels. These findings confirm that the GEST network effectively reproduces plant dynamics with high precision and responds to control inputs consistently with the FEM model, validating its effectiveness in supporting online adaptation of the GMPC.

GMPC Network

The GMPC network was developed to replicate the control law generated by the MPC. Its architecture includes an input layer with n_{mr} neurons that process h_{gm} steps of previous control actions and tracking errors, followed by l_{mh} hidden layers containing n_{mh} recurrent neurons. The

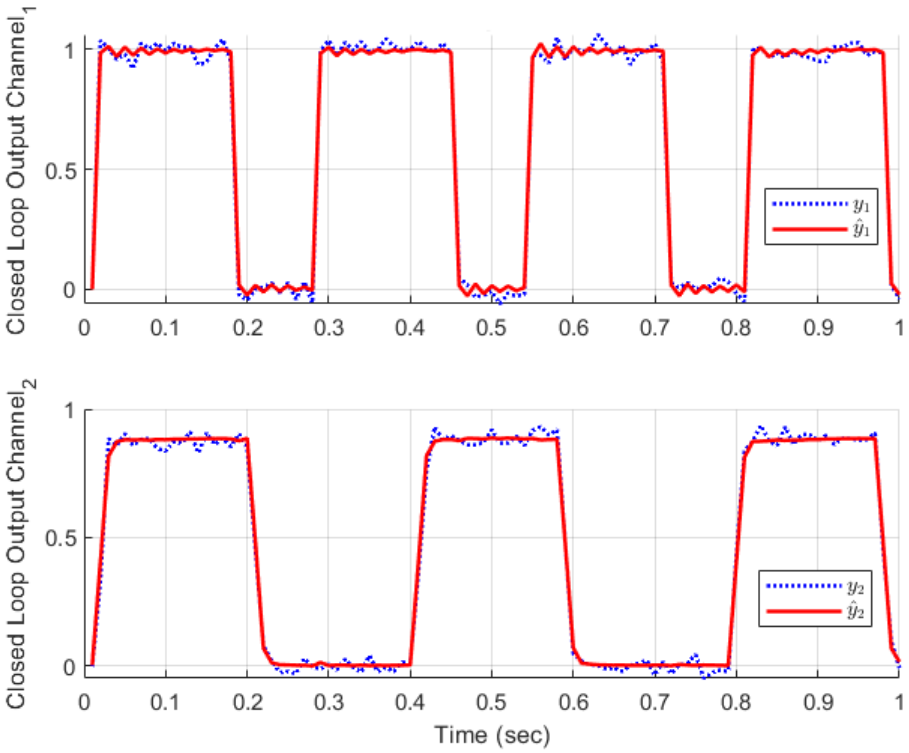

Figure 5.2: GEST ($\hat{y}_1$, $\hat{y}_2$) compared with actual outputs (y_1, y_2)

output layer consists of two neurons responsible for producing the predicted control actions.

$$U[k] = F_{\text{GMPC}}\big(U[k-1], \ldots, U[k-h_{gm}], E[k-1], \ldots, E[k-h_{gm}]\big), \tag{28}$$

where $E[k] = Y_{REF}[k] - Y[k]$ is the tracking error and $Y_{REF}[k]$ is the flattening of $y_{REF}[k] = [y_{11_{REF}}[k], y_{21_{REF}}[k]]^T$. This design captures the temporal dependencies between recent control actions and output errors, allowing the GMPC to replicate the corrective behavior of the original MPC policy [11, 26, 37].

The capability of the GMPC to emulate the MPC controller was further validated in closed-loop simulations. Figure 5.4 compares the closed-loop outputs obtained with MPC and GMPC against the reference trajectory. Both controllers achieve nearly identical tracking performance, with only minor deviations at transients periods. To better visualize these differences, Figure 5.5 shows the error between the two responses, which remains very small (on the order of 10^{-2}), confirming that the GMPC effectively replicates the closed-loop dynamics of the MPC.

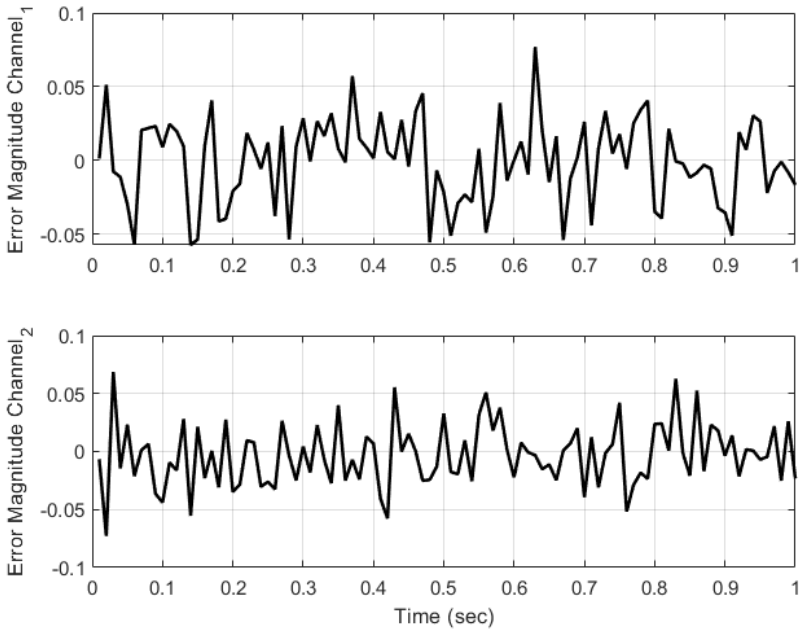

Figure 5.3: Error between actual outputs y and GEST outputs $\hat{y}$ for both channels

The control signal generated by both the MPC and GMPC is shown in Figures 5.6, indicating strong agreement between the two controllers. As illustrated in Figure 5.7, the difference between their outputs remains minimal, indicating that the GMPC accurately reproduces not only the output response of the MPC but also its control actions. These findings confirm that the GMPC has effectively learned the MPC control policy and can function as a dependable model for real-time drilling operations.

5.3.3 Adaptive Update Mechanism

To ensure accurate control inputs during drilling, the GMPC network incorporates an online weight update mechanism that dynamically adapts to variations in drilling conditions. Unlike networks trained solely offline, this adaptive approach allows the controller to sustain high accuracy despite changing formation properties, external disturbances, and unmodeled dynamics. The adaptation strategy is inspired by the framework of adaptive neural network controllers [10, 38], but is implemented here with significantly lower computational complexity compared to more sophisticated hybrid intelligent schemes [39].

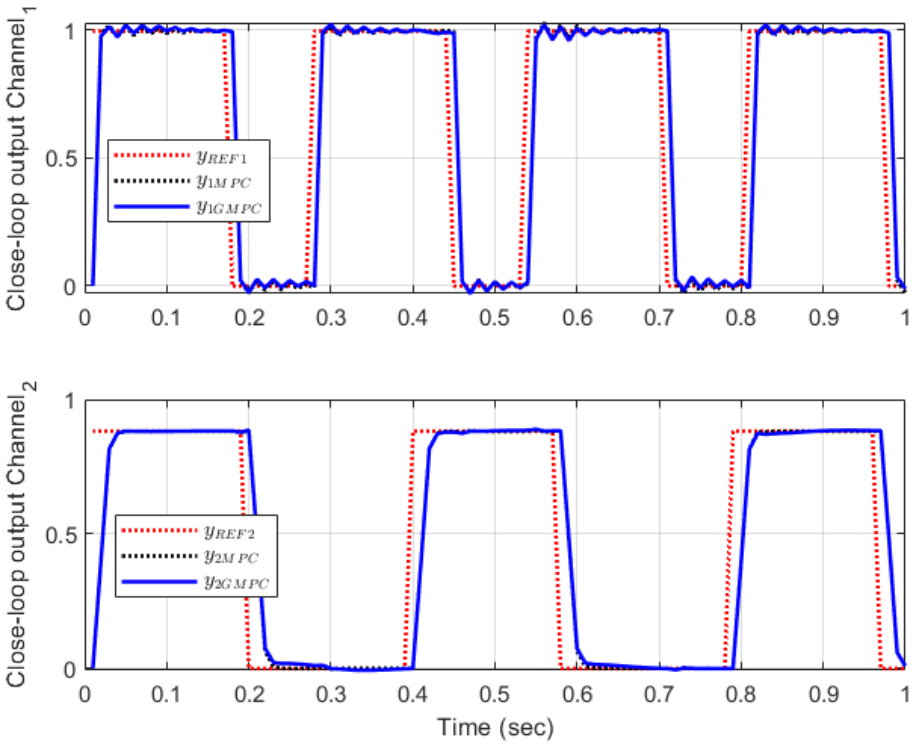

Figure 5.4: Closed loop MPC response compared to GMPC response.

The adaptive update mechanism utilizes a gradient-based approach, where the error is backpropagated through the closed-loop system using a chain derivative structure. Specifically, the parameter updates are given by

$$\frac{\partial E[k]}{\partial \text{GMPC}_P} = \frac{\partial E[k]}{\partial \hat{Y}[k]} \times \frac{\partial \hat{Y}[k]}{\partial U[k]} \times \frac{\partial U[k]}{\partial \text{GMPC}_P}, \tag{29}$$

where $E[k]$ is the instantaneous error between the flattened reference trajectory $Y_{REF}[k]$ and the flattened measured system output $Y[k]$, $\hat{Y}[k]$ is the flattened GEST output at step k, $U[k]$ is the flattened control input and $GMPC_P$ denotes the trainable parameters of the GMPC, including weight matrices, recurrent weight matrices, and biases.

This formulation enables the GMPC to continuously refine its control law by considering how deviations in the predicted output propagate backward through the control input and ultimately influence the GRU parameters. In practical terms, this allows the network to adapt to gradual changes in the drilling environment while mitigating the risk of instability that can arise from abrupt weight adjustments.

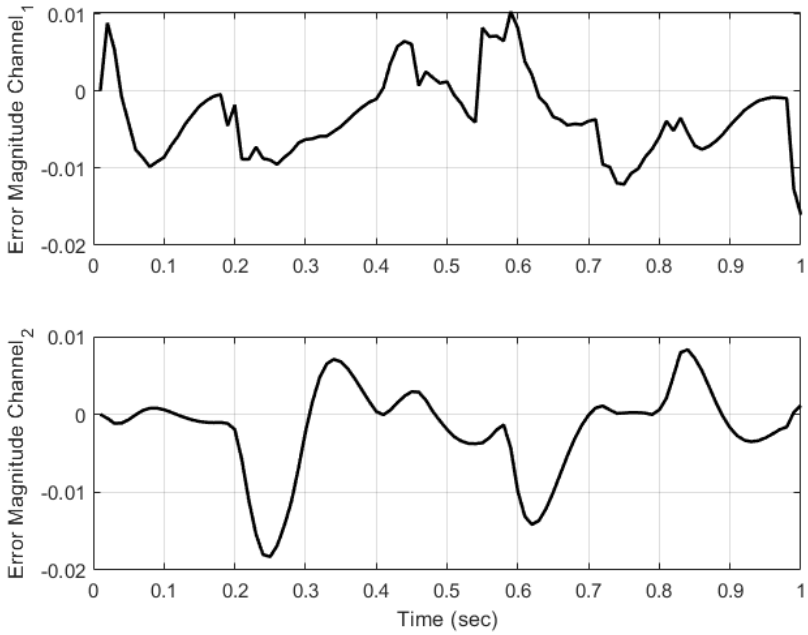

Figure 5.5: Estimation error

By integrating the GEST network with the GMPC within this adaptive framework, the system achieves a closed-loop functionality comparable to dual-learning architectures, where one network models the dynamics of the system and the other adjusts the control law. This dual-network configuration ensures that the controller maintains both computational efficiency and competitive performance under real-time drilling conditions.

Figure 5.8 shows the complete adaptive GMPC strategy. In this framework, the GMPC generates the control signal $u[k]$ based on the tracking error $e[k]$, while the GEST network predicts the plant response $\hat{y}[k]$. The discrepancy between the FEM plant output $y[k]$ and the estimated output $\hat{y}[k]$ is used to update the weights of the GEST network online, ensuring that the estimator remains in synchronization with the actual dynamics of the system . Once the GEST network is corrected, it guides the subsequent update of the GMPC network based on the error between the reference trajectory $y_{REF}[k]$ and the closed-loop output $y[k]$. This sequential adaptation process—first at the estimation level and then at the control level—allows the controller to incrementally converge toward the true drilling dynamics as the operation progresses.

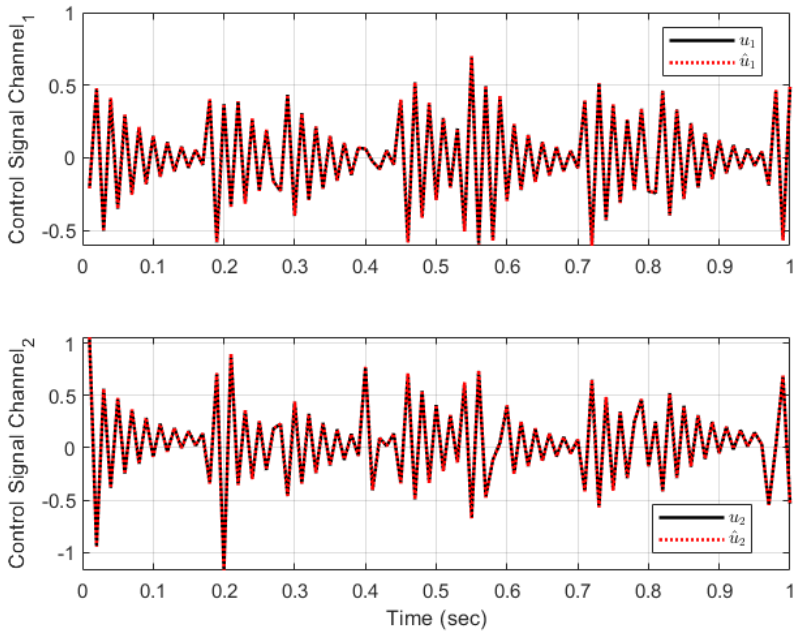

Figure 5.6: MPC and GMPC control signals

5.3.4 Design Considerations and Limitations

The performance of the GRU-based controller is highly dependent on the quality and diversity of the training dataset [13]. The network must be trained on valid and representative data that accurately reflect the dynamics of the drilling system, contain minimal measurement noise, and preserve realistic input–output relationships that the GRU can effectively learn. In contrast, invalid or excessively noisy data can affect generalization and result in poor control performance.

Beyond data quality, network architecture also plays a crucial role. The number and size of the hidden layers directly determine the capacity of the GRU to model system dynamics. An undersized network may underfit to fail to capture the true system behavior, whereas an oversized network may overfit, learning noise, or spurious correlations that compromise closed-loop stability during online adaptation. Consequently, careful architectural design is essential to balance accuracy and generalization—an objective achieved in this work through systematic hyperparameter tuning.

Transient discrepancies between the GRU and MPC may still arise [26], especially under unmodeled or rapidly varying conditions. Additionally, overly aggressive adaptation can destabilize

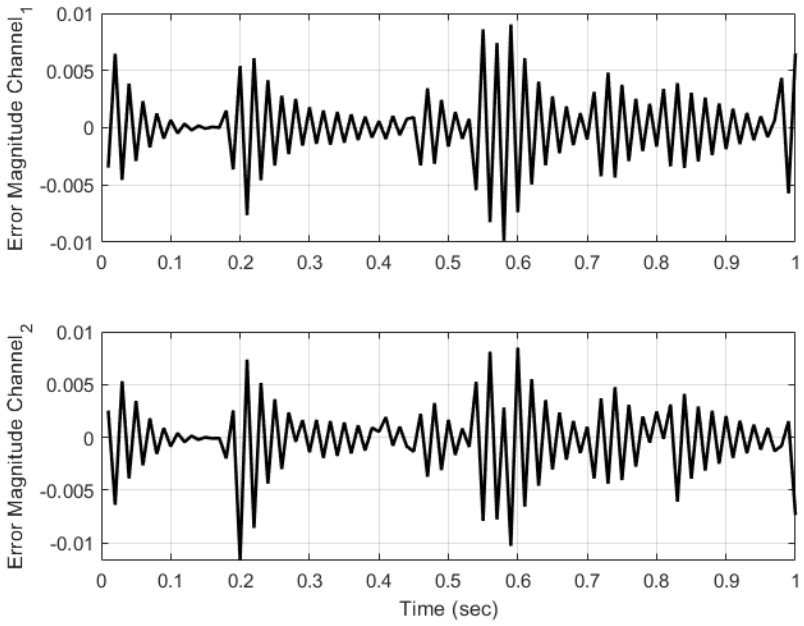

Figure 5.7: Difference between MPC and GMPC control signals

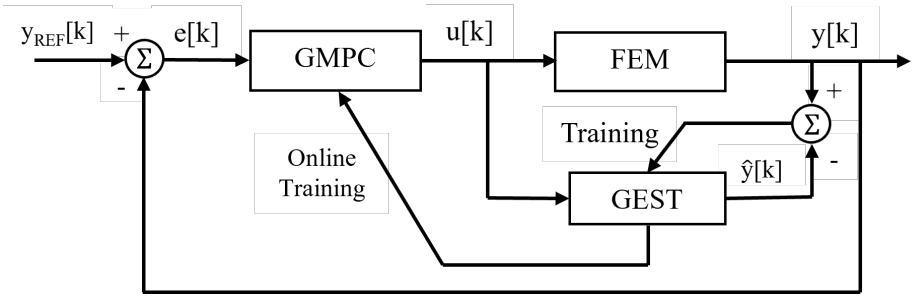

Figure 5.8: Adaptive GMPC framework

the system [38], highlighting the importance of employing conservative learning rates and conducting robust validation across multiple drilling scenarios.

5.4 Simulation Results and Analysis

5.4.1 Simulation Setup

A numerical simulation was conducted using the finite element directional drilling model to assess the closed-loop performance of the proposed adaptive GRU-based predictive controller. The total simulation duration was set to 1 second.

To replicate realistic downhole drilling conditions, three penetration stages were defined within this 1-second window. At each stage, the FEM parameters were updated to capture variations in boundary conditions and bit–rock interactions associated with progressive drilling advancement. The system matrices (M^l, D^l, K^l) were updated every 26 ms, resulting in three distinct operating regimes throughout the simulation. This dynamic variation posed a significant challenge to the controller, requiring continuous adaptation to evolving drilling dynamics. The key simulation parameters are summarized in Table 5.1.

Table 5.1: GRU network parameters.

Symbol	Value	Unit
g_{ii}	10^8	–
n_{gr}	20	–
l_{gh}	1	–
n_{gh}	5	–
h_{ge}	5	–
n_{mr}	12	–
l_{mh}	1	–
n_{mh}	5	–
h_{gm}	3	–

To evaluate performance, the pulse response of the closed-loop system was analyzed. This pulse excitation enabled a direct comparison between the transient behavior of the proposed GMPC and that of a conventional MPC baseline, emphasizing differences in tracking accuracy, response speed, and adaptability across the three penetration stages.

5.4.2 Closed-Loop Performance

Figure 5.9 presents the closed-loop output $y[k]$, the GEST-estimated output $\hat{y}[k]$ and the reference signal $y_{REF}[k]$, which is a pulse applied to both output channels. Throughout the one-second simulation, as the plant parameters vary across the three penetration stages, the system outputs accurately track the reference signals with minimal transient deviations and negligible steady-state error following each transition. The estimator output $\hat{y}[k]$ remains closely aligned with the measured output $y[k]$, exhibiting only brief discrepancies during transition moments corresponding to model or command changes. This indicates that:

- the GMPC produces the appropriate corrective action fast enough to keep the system on target;
- the GEST remains well aligned with the FEM dynamics during operation.

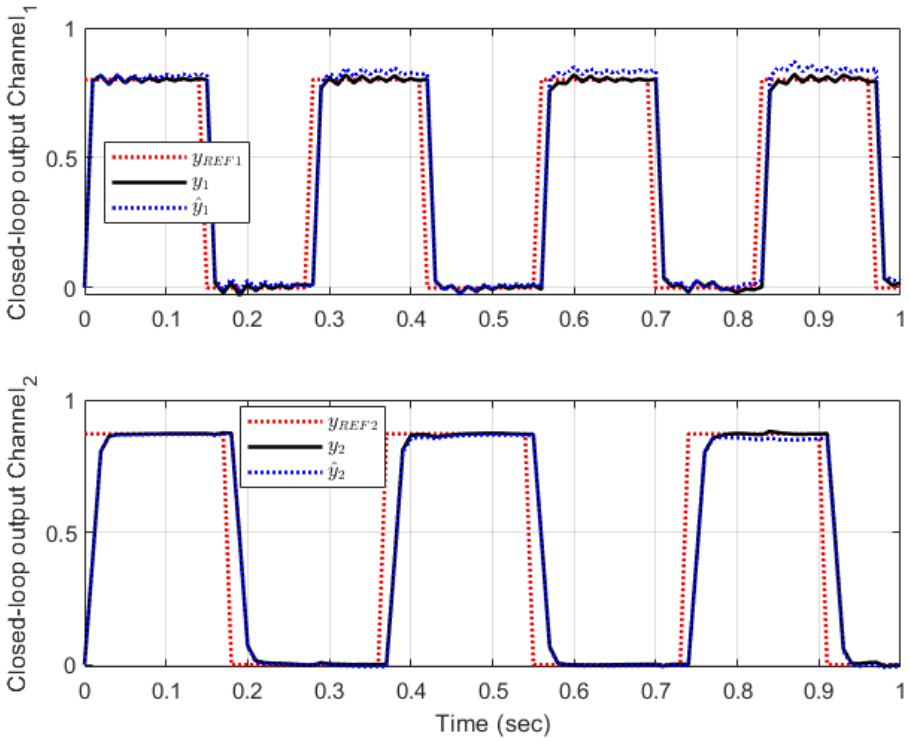

Figure 5.9: Adaptive GMPC pulse response with model changes

Figure 5.10 illustrates the control efforts for both input channels. As expected under pulse excitation and varying plant conditions, the inputs exhibit brief corrective spikes followed by rapid stabilization. No sustained oscillations are observed, and the control signals remain bounded throughout the simulation—indicating effective closed-loop damping and demonstrating the controller's real-time feasibility.

The accuracy of the estimation is shown in Figure 5.11, which shows the absolute difference $|y[k] - \hat{y}[k]|$. For $channel_1$ the error envelope remains below ~ 0.05, and for $channel_2$ below ~ 0.04 throughout the duration of the test. The error peaks occur immediately after reference or plant transitions and decay rapidly, consistent with the intended two-stage adaptation mechanism.

(1) the GEST is updated first with $y - \hat{y}$;

(2) then the GMPC is refined using $y_{REF} - y$.

The gradual decrease in error following each transition demonstrates the effectiveness of the online estimator updates and highlights the controller's ability to stay aligned with the evolving FEM dynamics throughout the simulation.

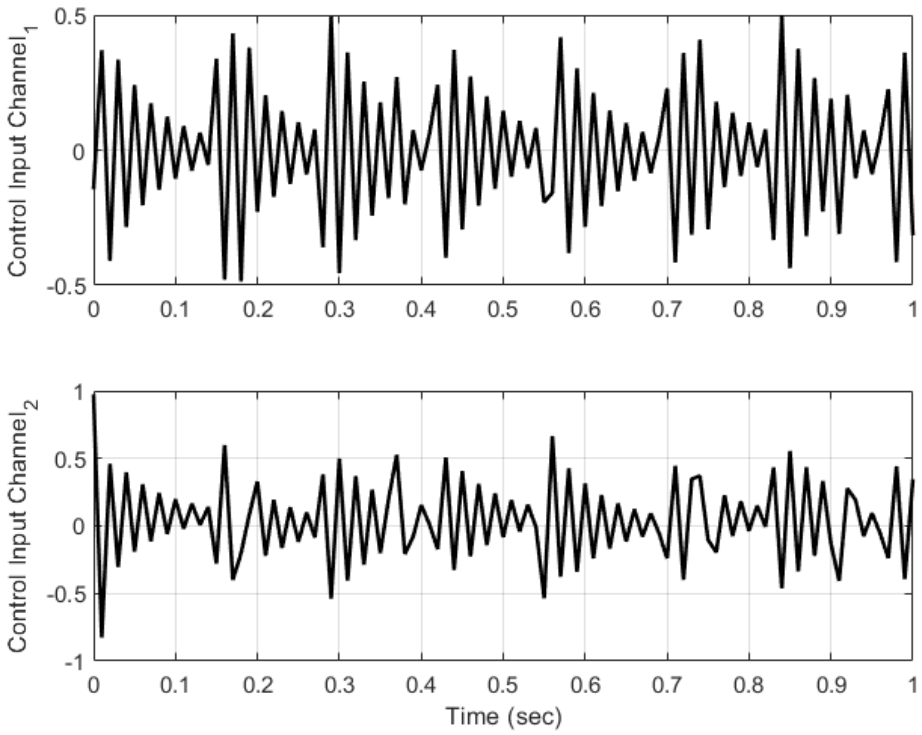

Figure 5.10: Control input signals generated by GMPC

5.4.3 Comparison with a PID Controller

To further validate the effectiveness of the proposed adaptive GRU-based predictive controller, its performance was compared to that of a standard PID controller. The two controllers were subjected to identical simulation settings, with the drilling model updated at 1.25-second intervals to incorporate step changes in penetration. The PID controller is given in (18). Table 5.2 provides the PID coefficient values.

Figure 5.12 illustrates the closed-loop responses of both control strategies relative to the reference trajectories. The PID controller exhibits pronounced overshoot and reduced convergence speed at each model transition, particularly in the presence of abrupt parameter variations. In contrast, the adaptive GRU-based controller achieves more accurate reference tracking, characterized

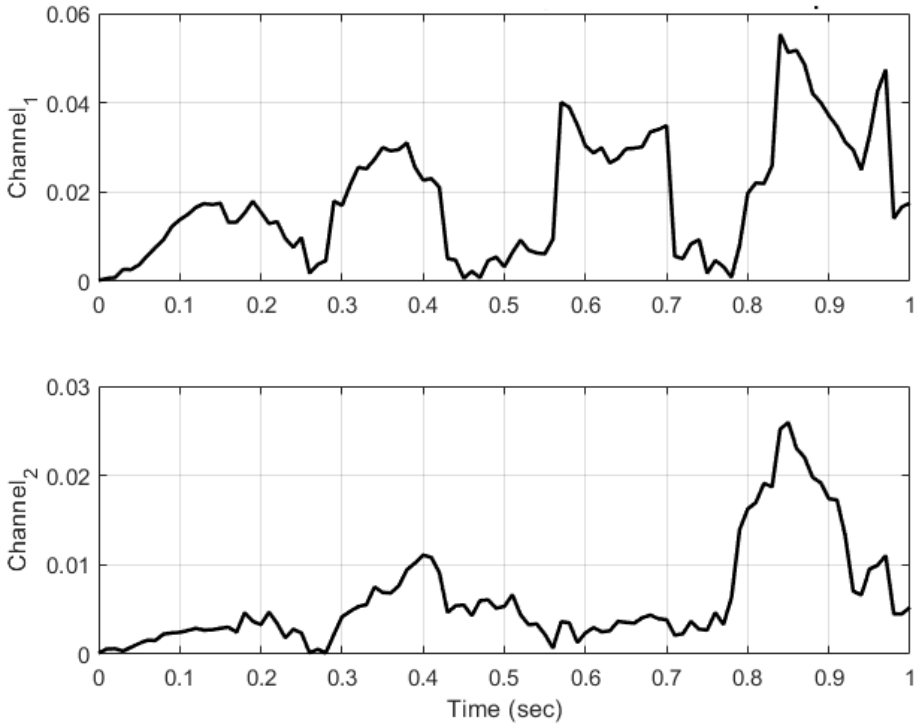

Figure 5.11: Tracking error with GMPC in closed loop

by faster transient recovery and minimal overshoot in both channels. This performance underscores the ability of the GRU controller to dynamically adjust to changing drilling conditions.

The error plots in Figure 5.13 corroborate these findings. In both channels, the absolute tracking error $|y_{REF} - y_{GRU}|$ remains consistently lower than $|y_{REF} - y_{PID}|$. The PID controller displays repeated error surges after each model transition, while the GRU-based controller promptly mitigates deviations. These results highlight the enhanced adaptability of the GRU framework when confronted with regime changes and nonlinear drilling behavior.

Figures 5.14 and 5.15 depict the applied control inputs for both methods. Although the PID controller operates with smaller input magnitudes relative to the GMPC, this reduced actuation is accompanied by a slower dynamic response and increased steady-state error. In contrast, the GRU-based controller delivers more assertive corrective inputs immediately after each penetration change, enabling more effective realignment with the reference trajectory. Importantly, these stronger inputs remain bounded and free of persistent oscillatory behavior, thereby maintaining closed-loop stability.

Table 5.2: PID Parameters.

Coefficients	Value
K_{P1}	-9.61×10^{-3}
K_{P2}	43.37×10^{-3}
K_{I1}	-0.139
n_1	100
K_{I2}	0.581
K_{D1}	3.69×10^{-4}
K_{D2}	-2.64×10^{-3}
n_2	100

5.5 Summary

This chapter presented an adaptive GRU-based predictive controller tailored for directional drilling operations, aimed at mitigating the computational demands of traditional MPC while retaining its predictive capabilities and constraint-handling features. A high-fidelity finite element model of the drill string was developed that incorporates penetration-dependent dynamics to support comprehensive simulation studies. The proposed methodology combined an MPC-trained GRU architecture with an online adaptation strategy, in which a GEST network was updated using real-time feedback, and the GMPC controller was subsequently refined based on closed-loop tracking errors.

Simulation studies demonstrated that the GRU-based controller achieved tracking performance comparable to that of MPC, but with dramatically reduced computational requirements, thereby enabling real-time deployment. The adaptive mechanism was validated through one-second pulse experiments in multiple penetration stages. Compared to a classical PID controller, the GMPC offered a superior transient response, reduced tracking error, and improved adaptability, although with slightly increased control effort.

Future research will investigate robustness under measurement noise and downhole uncertainties, and examine hybrid GMPC paradigms that integrate data-driven learning with physics-based models to improve generalization across operating conditions.

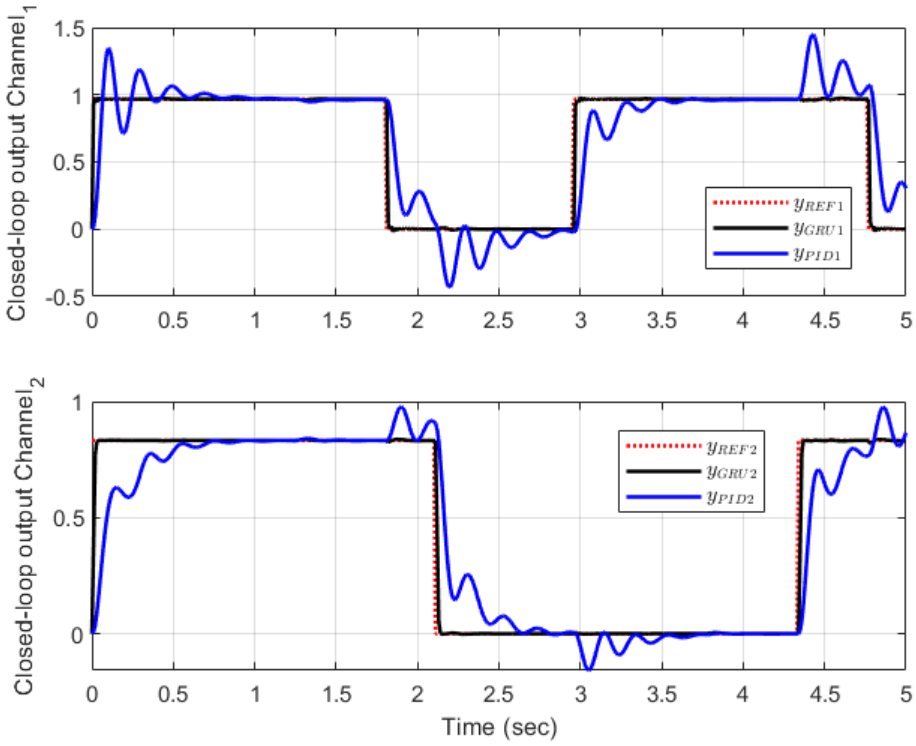

Figure 5.12: Closed loop response generated by PID and GMPC

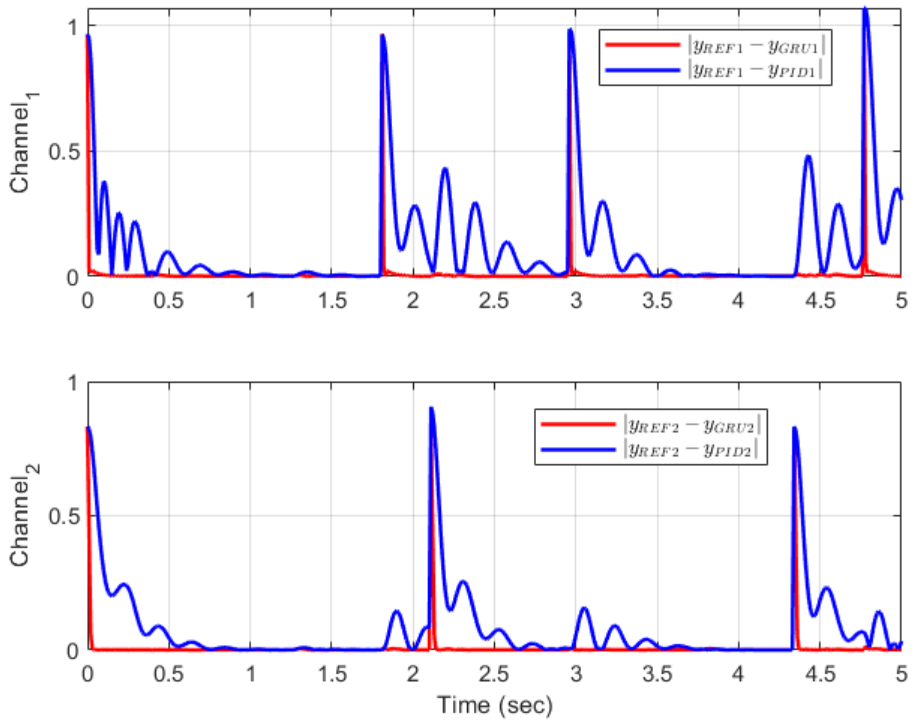

Figure 5.13: Difference in closed-loop response for PID and GMPC responses.

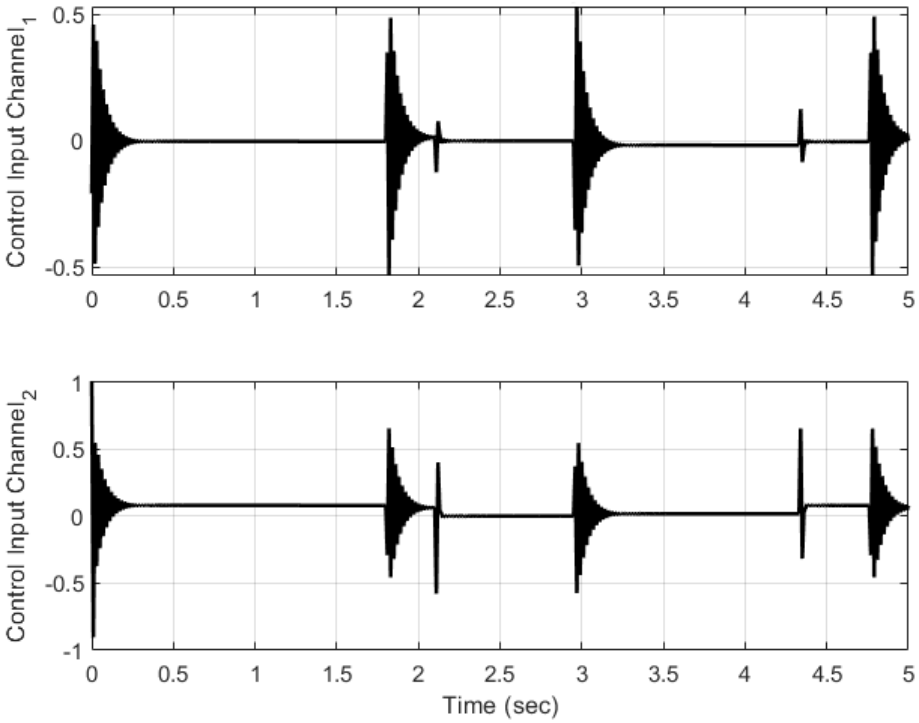

Figure 5.14: Control input signals generated by GMPC

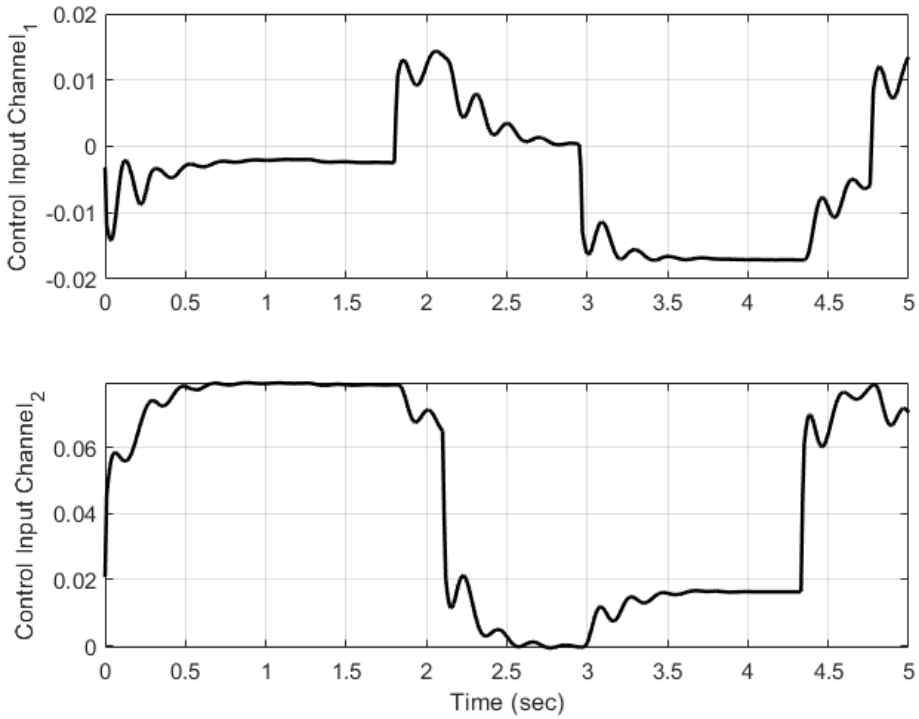

Figure 5.15: Control input signals generated by PID

Chapter 6

Conclusions and Future Work

6.1 Summary of Contributions

This thesis investigated intelligent control of directional drilling using GRU neural networks, built on a high-fidelity FEM with nonlinear bit–rock interaction. Two complementary strategies were developed and validated:

- An adaptive GRU–PID controller that updates PID gains online using a GRU-estimated sensitivity
- a GMPC that emulates the optimal behavior of a reference MPC while remaining computationally lightweight.

Both controllers were evaluated in a consistent simulation framework derived from the FEM and compared against classical baselines.

6.2 Key Findings

6.2.1 Modeling Basis and Control Affine Form

Chapter 3 developed a control-oriented FEM incorporating Rayleigh damping and a PDC bit–rock interaction law, then reformulated it into a discrete-time control-affine form for controller design.

This model served as the common testbed for all comparative studies, ensuring a fair assessment of the proposed controllers.

6.2.2 Adaptive GRU–PID Controller

The adaptive GRU–PID controller retained the interpretability and simplicity of PID while introducing data-driven adaptation via a GRU that approximates the local input–output Jacobian. Simulations showed a faster response, smaller steady-state error, smoother actuation, and overall improved stability relative to fixed-gain PID. These benefits were obtained under time-varying conditions representative of heterogeneous formations.

6.2.3 GMPC: GRU-Based MPC Surrogate

To address the computational burden of online MPC optimization, Chapter 5 introduced the GMPC surrogate: a GRU learned from offline input–output pairs MPC and adapted online through a GEST. The approach preserved MPC-level tracking while removing the per-step optimization bottleneck, enabling real-time feasibility for embedded targets.

Closed-loop experiments with parameter variations demonstrated that:

- GMPC tracked pulse references with negligible steady-state error after transitions
- GEST stayed aligned with the plant except for short-lived discrepancies at switching
- Control actions remained bounded with no sustained oscillations—consistent with good damping and real-time feasibility.

6.2.4 Comparative Insights

Across identical scenarios, GRU–PID offered low computational cost; GMPC achieved near-MPC accuracy and superior constraint handling by imitating the MPC policy.

6.3 Practical Implications

The results indicate that intelligent GRU controllers can be embedded in downhole or surface automation where sampling periods on the order of 10–20 ms are typical. GMPC is particularly suitable where constraint handling and optimality are paramount, while GRU–PID provides a low-complexity alternative when computational headroom is minimal. The shared FEM foundation and control-affine formulation simplify migration between strategies within the same automation stack.

6.4 Limitations

Despite their strong performance, the controllers rely on representative training data and appropriate hyperparameter choices. Insufficient diversity or quality in training sequences can degrade generalization; overly aggressive adaptation can risk loss of transient performance. Furthermore, the scope of the modeling emphasized axial–torsional dynamics with simplified treatment of lateral effects and fluid–structure coupling to maintain computational tractability.

6.5 Future Research Directions

6.5.1 Experimental and Hardware-in-the-Loop (HIL) Validation

A priority is closed-loop validation on HIL platforms or laboratory drilling testbeds that incorporate realistic communication delays, sensor noise, and actuation limits. This would stress-test adaptation under practical constraints and guide embedded implementations.

6.5.2 Robustness and Stability

Formal robustness analysis for the online adaptation laws is necessary, including small-gain or ISS-style arguments for GRU–PID weight updates and GMPC/GEST coupled adaptation. Such results would complement the empirical stability observed in the simulations.

6.5.3 Richer Physics and Multi-Physics Coupling

Extending the plant model to include lateral bending, drillstring–borehole contact, and fluid–structure coupling will enable evaluation in more aggressive regimes. These effects can be incrementally introduced to assess tradeoffs between fidelity and real-time feasibility.

6.5.4 Hybrid Physics–ML Control

Promising directions include hybrid observers that fuse FEM-based predictions with GRU estimators, and constraint-aware surrogates that embed safety margins learned from MPC into the network architecture (e.g., smooth saturation layers or barrier-loss regularization).

6.5.5 Dataset Design and Continual Learning

Systematic dataset curation (coverage of formation variability, operational modes, and transients) and continual learning protocols can reduce distribution shift during long drilling campaigns. Careful controls on learning rates and confidence-triggered updates should mitigate instability during online adaptation.

6.5.6 Field Deployment and Benchmarking

Finally, side-by-side field trials comparing GRU–PID and GMPC against tuned industrial baselines would quantify benefits in ROP, NPT, and tool wear. Benchmark datasets and open-source reference implementations could accelerate reproducibility and adoption.

6.6 Closing Remarks

The thesis demonstrates that synergizing physics-based modeling with GRU neural networks yields practical, high-performance controllers for directional drilling. The adaptive GRU–PID and GMPC offer complementary tradeoffs between optimality, and computational cost, and together provide a scalable path toward autonomous drilling systems capable of learning and operating reliably under uncertainty.

Appendix A

GEST and GMPC Weight and Bias Matrices

The weights and biases of the GEST model are represented by

$$EW_z = \begin{bmatrix} -0.5614 & 0.2966 & -0.3050 & 0.5417 & 0.1850 \\ -0.9552 & 0.0069 & 0.1450 & 0.9349 & 0.4547 \\ -0.8369 & 0.4173 & -0.4689 & 0.7075 & 0.6907 \\ -1.1439 & 0.7837 & -1.0275 & -0.0433 & 0.4524 \\ -0.0570 & -0.0169 & 0.0319 & -0.1091 & 0.0628 \\ 0.1685 & -0.0070 & 0.0354 & 0.1987 & 0.1863 \\ 0.4070 & -0.1528 & 0.1780 & 0.1679 & 0.5227 \\ 0.6022 & -0.4937 & 0.4732 & 0.3683 & 0.6064 \\ 0.4083 & -0.3915 & 0.3724 & 0.1686 & 0.2568 \\ 0.0223 & 0.0044 & -0.0832 & 0.0526 & -0.0010 \\ 0.1099 & -0.2007 & 0.0127 & -0.0960 & 0.0911 \\ 0.1868 & 0.2722 & -0.0650 & 0.2622 & -0.1663 \\ 0.4369 & -0.3824 & 0.3592 & -0.1867 & 0.1042 \\ 0.0141 & 0.1869 & 0.1057 & 0.2281 & -0.0372 \\ 0.9220 & -0.8455 & 0.2023 & 0.0563 & 0.1430 \\ -0.1416 & -0.0146 & 0.0223 & -0.0377 & 0.0282 \\ 0.1351 & 0.0110 & -0.3814 & 0.1533 & -0.0537 \\ -0.2953 & -0.0781 & 0.2459 & 0.2091 & 0.4476 \\ 0.2062 & -0.1938 & 0.1834 & -0.0854 & -0.0287 \\ -0.4407 & -0.1982 & -0.0764 & 0.1136 & 0.0386 \end{bmatrix}^T_{20\times 5}, \tag{30}$$

$$EW_r = \begin{bmatrix} -0.1994 & 0.0974 & 0.4458 & 0.0088 & -0.2360 \\ -0.4462 & -0.1758 & 0.0510 & 0.2904 & 0.0958 \\ 0.0028 & -0.1251 & 0.2357 & 0.0580 & 0.5733 \\ -0.3062 & 0.2139 & 0.2976 & -0.1185 & 0.0234 \\ -0.7732 & -0.1331 & -0.7075 & 0.2108 & -0.1167 \\ -0.0896 & 0.1260 & 0.1355 & -0.0719 & -0.0046 \\ -0.1868 & 0.1608 & 0.1407 & -0.0705 & -0.2926 \\ -0.3305 & -0.0441 & -0.2041 & -0.1200 & -0.3398 \\ -0.1129 & -0.0205 & -0.0173 & -0.1300 & -0.0847 \\ 0.1687 & 0.0244 & 0.1643 & 0.0330 & 0.0181 \\ 0.2201 & 0.5752 & -0.5080 & -0.0146 & 0.0043 \\ 0.0501 & 0.0636 & 0.0845 & -0.0413 & 0.0376 \\ -0.0902 & -0.0761 & -0.3555 & -0.0353 & 0.0790 \\ 0.0749 & -0.0635 & 0.2337 & -0.1030 & -0.2503 \\ -0.6869 & -0.4809 & -0.2195 & 0.1168 & 0.1869 \\ 0.0581 & 0.0239 & 0.0141 & 0.0877 & 0.0249 \\ -0.1088 & -0.0335 & -0.0587 & -0.0447 & 0.0278 \\ -0.0958 & 0.2911 & 0.3222 & -0.0726 & -0.2158 \\ 0.0465 & -0.4209 & -0.6941 & 0.0267 & -0.0565 \\ -0.0030 & 0.2024 & 0.2328 & -0.2761 & 0.1131 \end{bmatrix}^T_{20\times 5}, \tag{31}$$

$$EW_h = \begin{bmatrix} 0.3269 & 0.7193 & -0.3208 & -0.3481 & 0.5804 \\ 0.8517 & 1.1478 & -0.1519 & -0.7973 & 0.9437 \\ 1.1310 & 1.5855 & -0.5085 & -0.6116 & 0.9611 \\ 0.5824 & 1.3116 & -0.7646 & -0.1040 & 0.7473 \\ 0.0102 & -0.0196 & 0.0372 & 0.0471 & 0.0435 \\ -0.2446 & -0.0872 & 0.0512 & -0.1998 & 0.2116 \\ -0.3876 & -0.4785 & 0.1898 & -0.5999 & 0.3525 \\ -0.2730 & -0.6952 & 0.3002 & -0.8129 & 0.3076 \\ -0.1904 & -0.3039 & 0.1958 & -0.4043 & 0.2179 \\ -0.0036 & -0.0023 & -0.0013 & 0.0109 & 0.0074 \\ -0.0892 & 0.0316 & 0.1778 & -0.0042 & -0.0208 \\ 0.0058 & -0.0149 & 0.3319 & 0.0098 & -0.0179 \\ -0.2487 & -0.2736 & 0.1977 & 0.0169 & -0.1514 \\ -0.0648 & 0.0301 & -0.1340 & -0.0701 & 0.0441 \\ -0.0094 & -0.3121 & 0.2990 & -0.0789 & -0.0832 \\ -0.0201 & -0.0029 & 0.0081 & 0.0119 & 0.0060 \\ 0.0469 & -0.0007 & -0.0399 & -0.0327 & -0.0229 \\ -0.3068 & 0.1015 & 0.0086 & -0.3325 & 0.5045 \\ 0.3534 & -0.5328 & 0.2473 & -0.2646 & -0.3984 \\ -0.1150 & 0.5004 & -0.2137 & 0.2591 & 0.2372 \end{bmatrix}^T_{20\times5}, \tag{32}$$

$$EU_z = \begin{bmatrix} -0.2343 & -0.6566 & 0.3125 & -0.1205 & -0.4953 \\ 0.2757 & 0.2172 & -0.5478 & 0.2150 & -0.0314 \\ -0.2747 & -0.0396 & 0.0864 & 0.2291 & 0.0600 \\ -0.5169 & 0.4516 & -0.0836 & -0.3296 & 0.2743 \\ -0.7238 & -0.0281 & -0.2000 & -0.0338 & 0.1577 \end{bmatrix}_{5\times5}, \tag{33}$$

$$EU_r = \begin{bmatrix} 0.8416 & 0.0502 & -0.3299 & -0.4392 & -0.1156 \\ 0.8584 & -0.0405 & -0.8420 & 0.1688 & 0.3988 \\ 0.0531 & -0.8032 & -0.0486 & -0.6548 & -0.7570 \\ 0.4600 & -0.1193 & 0.1737 & 0.0812 & -0.6525 \\ 0.2247 & 0.0357 & 0.0599 & -0.0079 & 0.1374 \end{bmatrix}_{5\times5}, \tag{34}$$

$$EU_h = \begin{bmatrix} 0.2842 & 0.2808 & -0.4147 & 0.6176 & 0.6248 \\ -0.0653 & 0.6199 & -0.1941 & -0.4233 & -0.2084 \\ -0.4729 & -0.7034 & 0.6650 & 0.1149 & -0.0816 \\ 0.2988 & 0.1293 & -0.0504 & 0.4132 & -0.3598 \\ 0.2892 & 0.1740 & -0.2427 & -0.6725 & 0.2514 \end{bmatrix}_{5\times5}, \tag{35}$$

$$Eb_z = \begin{bmatrix} -0.0303 & -0.1626 & 0.5175 & 0.1911 & 0.0948 \end{bmatrix}^T_{1\times5}, \tag{36}$$

$$Eb_r = \begin{bmatrix} 0.9501 & -0.3357 & 1.2014 & 0.0508 & 0.3299 \end{bmatrix}^T_{1\times5}, \tag{37}$$

$$Eb_h = \begin{bmatrix} 0.0370 & -0.0080 & 0.0133 & -0.0325 & -0.0361 \end{bmatrix}^T_{1\times5}, \tag{38}$$

where EW_r, EW_z and EW_h are the reset gate, the update gate and the hidden state weight matrices respectively, EU_r, EU_z and EU_h are the reset gate, the update gate and the hidden state recurrent matrices respectively, Eb_r, Eb_z and Eb_h are the reset gate, the update gate and the hidden state bias matrices respectively.

The weights and biases of the GMPC model are represented by

$$CW_z = \begin{bmatrix} 0.3801 & 0.3446 & 0.2317 & -0.0871 & 0.9405 \\ -0.0813 & 0.9514 & 0.5966 & 0.8107 & 0.1312 \\ 0.5044 & -0.3209 & -1.5080 & -1.2955 & -0.1007 \\ 0.1418 & -0.0419 & -0.3085 & -0.1111 & -0.0974 \\ -0.0790 & -0.2129 & -0.1369 & 0.2576 & 0.2913 \\ 0.3938 & 0.4877 & 0.5285 & 0.7118 & -0.1141 \\ -0.0433 & 0.2618 & -0.1828 & 0.2590 & -0.3276 \\ 0.1298 & 1.3023 & 1.0306 & 1.1963 & -1.0498 \\ -0.0666 & 0.5196 & -0.4592 & 0.1388 & -0.5264 \\ 0.2351 & 0.2331 & -0.1030 & -0.3738 & 0.2391 \\ -0.6943 & -0.5010 & 0.7893 & 0.5556 & -0.5189 \\ -0.1982 & 1.0602 & -1.0970 & -0.4018 & 0.1147 \end{bmatrix}^T_{12\times 5}, \tag{39}$$

$$CW_r = \begin{bmatrix} -0.7314 & -0.3372 & -0.0257 & 0.3196 & 0.0952 \\ -1.2558 & 0.0945 & 0.1314 & -0.0251 & -0.1871 \\ -0.8665 & 0.1210 & -0.1656 & -0.5810 & -0.5022 \\ -0.7833 & -0.0722 & 0.0676 & 0.0020 & -0.2515 \\ -1.2311 & -0.0505 & 0.0816 & 0.0003 & -0.4275 \\ -0.7863 & -0.5524 & -0.0471 & 0.0728 & -0.2913 \\ -0.0310 & 0.0809 & -0.0111 & -0.1087 & -0.0648 \\ 0.0663 & -0.2276 & -0.0786 & 0.3128 & 0.1294 \\ -0.0425 & 0.3010 & 0.0344 & -0.1165 & -0.1707 \\ -0.1583 & -0.0622 & -0.0173 & 0.0484 & -0.1000 \\ 0.3128 & 0.0758 & 0.0422 & -0.0835 & 0.1641 \\ 0.0187 & -0.1344 & -0.0614 & -0.0248 & 0.1884 \end{bmatrix}^T_{12\times 5}, \tag{40}$$

$$CW_h = \begin{bmatrix} -0.2319 & 0.1311 & -0.2961 & -0.2594 & 0.1273 \\ -0.2844 & 0.2224 & -0.2956 & -0.3525 & 0.1642 \\ -0.0713 & 0.0376 & 0.4581 & 0.3343 & -0.2267 \\ -0.1957 & -0.0829 & 0.0915 & 0.0943 & -0.1085 \\ -0.2364 & -0.0474 & 0.1232 & 0.0357 & -0.1527 \\ -0.1208 & 0.2039 & 0.0243 & -0.1815 & 0.1387 \\ 0.0106 & -0.0699 & 0.1191 & 0.1029 & -0.0738 \\ -0.0269 & 0.1151 & -0.2748 & -0.2651 & 0.2493 \\ 0.0130 & -0.1445 & 0.1774 & 0.0987 & -0.0996 \\ -0.3465 & -0.0687 & 0.0226 & 0.0424 & -0.1181 \\ 0.7237 & 0.2584 & 0.0047 & -0.1328 & 0.2990 \\ -0.4220 & -0.2414 & -0.0511 & 0.1274 & -0.2034 \end{bmatrix}^T_{12\times5}, \tag{41}$$

$$CU_z = \begin{bmatrix} -0.5899 & 0.3586 & -0.4120 & -0.0030 & -0.0276 \\ -0.7784 & -0.4434 & 0.0812 & -0.2206 & 0.3147 \\ -0.0845 & -0.5154 & 0.2820 & -0.7153 & 0.7584 \\ -0.6846 & -0.0940 & 0.9767 & -1.0618 & 0.3273 \\ -0.0529 & 0.3322 & -0.8542 & -0.7739 & 0.5599 \end{bmatrix}_{5\times5}, \tag{42}$$

$$CU_r = \begin{bmatrix} 0.0209 & 0.0124 & -0.4658 & 0.0291 & -0.4488 \\ -0.2925 & 0.7351 & -0.6567 & 0.8360 & -0.9610 \\ -0.7677 & -0.5408 & 0.4512 & -0.8422 & 0.0577 \\ -0.3800 & -0.7090 & 0.9132 & -0.7072 & 0.6247 \\ -0.1604 & -0.8160 & 0.4167 & -0.8535 & 0.2230 \end{bmatrix}_{5\times5}, \tag{43}$$

$$CU_h = \begin{bmatrix} -0.3354 & -0.2972 & -0.1295 & 0.1233 & 0.0251 \\ 0.1835 & 0.0911 & 0.6170 & -0.2332 & -0.5886 \\ 0.0350 & 0.2158 & -0.3492 & 0.1841 & 0.1132 \\ 0.0412 & 0.1623 & 0.1164 & -0.0993 & -0.2175 \\ 0.0400 & 0.2954 & -0.0344 & 0.0550 & -0.2405 \end{bmatrix}_{5\times5}, \tag{44}$$

$$Cb_z = \begin{bmatrix} 0.1114 & -0.1330 & -0.2701 & -0.4367 & 0.2592 \end{bmatrix}^T_{1\times5}, \tag{45}$$

$$Cb_r = \begin{bmatrix} -1.4831 & -0.9476 & -0.9942 & -0.6937 & -0.8175 \end{bmatrix}^T_{1\times5}, \tag{46}$$

$$Cb_h = \begin{bmatrix} 0.1825 & 0.4010 & -0.0459 & 0.1491 & -0.1939 \end{bmatrix}^T_{1\times5}, \tag{47}$$

where CW_r, CW_z and CW_h are the reset gate, the update gate and the hidden state weight matrices respectively, CU_r, CU_z and CU_h are the reset gate, the update gate and the hidden state recurrent matrices respectively, Cb_r, Cb_z and Cb_h are the reset gate, the update gate and the hidden state bias matrices respectively.

Bibliography

[1] T. Ma, P. Chen, and J. Zhao, "Overview on vertical and directional drilling technologies for the exploration and exploitation of deep petroleum resources," *Geomechanics and Geophysics for Geo-Energy and Geo-Resources*, vol. 2, no. 4, pp. 365–395, 2016.

[2] E. J. Bouki-Matondo, *Miglioramento delle operazioni di perforazione con l'uso di un rivestimento di custodia da 7 pollici, economico e sicuro. Enhancement of drilling operations with the use of 7"casing liner, cost effective and safe.* PhD thesis, Politecnico di Torino, 2023.

[3] S. Schaaf, D. Pafitis, and E. Guichemerre, "Application of a point the bit rotary steerable system in directional drilling prototype well-bore profiles," *SPE Western Regional Meeting*, pp. SPE–62519, 2000.

[4] J. Kim, B. Park, J.-U. Shin, S. Jung, and H. Myung, "A novel steering sections of hybrid rotary steerable system for directional drilling," *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, pp. 1617–1619, 2014.

[5] H. Wang, Z. C. Guan, and G. S. Zhao, "Dynamic characteristic analysis on the bit lateral steering force of push-the-bit rotary steerable bottom hole assembly," *Mathematical Models in Engineering*, vol. 4, no. 4, pp. 201–209, 2018.

[6] S. Bridges and L. Robinson, *A practical handbook for drilling fluids processing*. Gulf Professional Publishing, 2020.

[7] C. Ke and X. Song, "Control of down-hole drilling process using a computationally efficient dynamic programming method," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 10, p. 101010, 2018.

[8] C. Ke and X. Song, "Control design for directional down-hole drilling using dual-heuristic programming with a high-order dynamic model," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1009–1020, 2021.

[9] J. Wu, C. Lu, H. Li, H. Wei, and M. Wu, "Trajectory tracking for directional drilling in underground coal mine using robust model predictive control," *2025 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1–6, 2025.

[10] J. Ding, M. Wu, and M. Xiao, "Nonlinear decoupling control with pi λ d μ neural network for mimo systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 6, pp. 8715–8722, 2022.

[11] Z. Ebrahimi and R. R. Selmic, "Intelligent control system for directional drilling: A gru neural network approach," *2025 33rd Mediterranean Conference on Control and Automation (MED)*, pp. 257–262, 2025.

[12] Z. Ebrahimi and R. R. Selmic, "Adaptive gated recurrent unit-based predictive control for directional drilling operations," *Journal of Dynamic Systems, Measurement, and Control*, 2025.

[13] Y. Gao, N. Wang, and Y. Ma, "L2-ssa-lstm prediction model of steering drilling wellbore trajectory," *IEEE Access*, vol. 12, pp. 450–461, 2023.

[14] H. Tian, J. Tang, and W. Yu, "Lstm neural network-based on model predictive control for furnace temperature in mswi process," *2025 37th Chinese Control and Decision Conference (CCDC)*, pp. 2151–2154, 2025.

[15] W. Li, M. Wu, K. YaO, Q. Li, C. Lu, and S. Du, "Design of an intelligent control system for underground directional drilling trajectory in coal mines," *2022 41st Chinese Control Conference (CCC)*, pp. 2586–2590, 2022.

[16] E. I. Epelle and D. I. Gerogiorgis, "A review of technological advances and open challenges for oil and gas drilling systems engineering," *AIChE Journal*, vol. 66, no. 4, p. e16842, 2020.

[17] B. Park, J. Kim, J. Park, J.-U. Shin, and H. Myung, "Hybrid 4-pad rotary steerable system for directional drilling of unconventional resources," *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 659–660, 2013.

[18] J. Tian and L. Sheng, "Cubature kalman filtering for dynamic pointing rotary steerable system based on dynamic event-triggering mechanism," *2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS)*, pp. 1–6, 2021.

[19] J. Park, J. Kim, B. Park, and H. Myung, "Design and analysis of a new hybrid rotary steerable system for directional drilling," *IEEE ISR 2013*, pp. 1–3, 2013.

[20] J. Kim and H. Myung, "Development of a novel hybrid-type rotary steerable system for directional drilling," *IEEE access*, vol. 5, pp. 24678–24687, 2017.

[21] K. Isaka, K. Tsumura, T. Watanabe, W. Toyama, M. Okui, H. Yoshida, and T. Nakamura, "Soil discharging mechanism utilizing water jetting to improve excavation depth for seabed drilling explorer," *IEEE Access*, vol. 8, pp. 28560–28570, 2020.

[22] K. Gurney, *An introduction to neural networks*. CRC press, 2018.

[23] P. Kittisupakorn, P. Thitiyasook, M. A. Hussain, and W. Daosud, "Neural network based model predictive control for a steel pickling process," *Journal of process control*, vol. 19, no. 4, pp. 579–590, 2009.

[24] J.-Y. Dieulot, "Model predictive control of fuel cells using a neural network based inverted model," *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 864–869, 2023.

[25] S. V. Feofilov and D. Khapkin, "Application of recurrent neural networks in closed loop tracking systems for controlling essentially nonlinear objects," *2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, pp. 467–472, 2021.

[26] E. Calle and J. Oliden, "Recurrent neural network based predictive control applied to 4 coupled-tank system," *2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pp. 1–6, 2021.

[27] Y. Bu, W. Liao, W. Tian, J. Zhang, and L. Zhang, "Stiffness analysis and optimization in robotic drilling application," *Precision Engineering*, vol. 49, pp. 388–400, 2017.

[28] G. Downton, "Challenges of modeling drilling systems for the purposes of automation and control," *IFAC Proceedings Volumes*, vol. 45, no. 8, pp. 201–210, 2012.

[29] E. Detournay and P. Defourny, "A phenomenological model for the drilling action of drag bits," *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, vol. 29, no. 1, pp. 13–23, 1992.

[30] R. Ma, B. Zhang, Y. Zhou, Z. Li, and F. Lei, "Pid controller-guided attention neural network learning for fast and effective real photographs denoising," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 3010–3023, 2021.

[31] R. Ma, B. Zhang, Y. Zhou, Z. Li, and F. Lei, "Pid controller-guided attention neural network learning for fast and effective real photographs denoising," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 3010–3023, 2021.

[32] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*. CRC press, 2020.

[33] Y. M. Ren, M. S. Alhajeri, J. Luo, S. Chen, F. Abdullah, Z. Wu, and P. D. Christofides, "A tutorial review of neural network modeling approaches for model predictive control," *Computers & Chemical Engineering*, vol. 165, p. 107956, 2022.

[34] D. Wang, Z. J. Shen, X. Yin, S. Tang, X. Liu, C. Zhang, J. Wang, J. Rodriguez, and M. Norambuena, "Model predictive control using artificial neural network for power converters," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 3689–3699, 2021.

[35] Y. Liu, Y. Wang, Y. Feng, and Y. Wu, "Neural network-based adaptive boundary control of a flexible riser with input deadzone and output constraint," *IEEE transactions on cybernetics*, vol. 52, no. 12, pp. 13120–13128, 2021.

[36] J. Yu, P. Shi, W. Dong, B. Chen, and C. Lin, "Neural network-based adaptive dynamic surface control for permanent magnet synchronous motors," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 640–645, 2014.

[37] S. Piche, B. Sayyar-Rodsari, D. Johnson, and M. Gerules, "Nonlinear model predictive control using neural networks," *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 53–62, 2000.

[38] A. Theodorakopoulos and G. A. Rovithakis, "A simplified adaptive neural network prescribed performance controller for uncertain mimo feedback linearizable systems," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 589–600, 2014.

[39] M. B. Milovanović, D. S. Antić, M. T. Milojković, and M. D. Spasić, "Adaptive control of nonlinear mimo system with orthogonal endocrine intelligent controller," *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 1221–1232, 2020.