

PHYSICS-INFORMED MACHINE LEARNING
FOR MODELING AND ANALYSIS OF
MECHANICAL SYSTEMS

PHYSICS-INFORMED NEURAL NETWORKS FOR MODELING
AND ANALYSIS OF A MAGNETOREHOLOGICAL DAMPER
SETUP

By YUANDI WU, B.A.Sc

A Thesis Submitted to the School of Graduate Studies in Partial
Fulfillment of the Requirements for
the Degree Master of Applied Science

McMaster University © Copyright by Yuandi Wu, August 2024

McMaster University

MASTER OF APPLIED SCIENCE (2024)

Hamilton, Ontario, Canada (Mechanical Engineering)

TITLE: Physics-Informed Neural Networks for Modeling and
Analysis of a Magnetorehological Damper Setup

AUTHOR: Yuandi Wu
B.A.Sc (Mechanical Engineering),
McMaster University, Hamilton, Canada

SUPERVISOR: Dr. Stephen Andrew Gadsden

NUMBER OF PAGES: xviii, 175

Lay Abstract

This study presents an implementation of Physics-Informed Neural Networks to model complex mechanical systems with hidden variables, exemplified by Magnetorheological (MR) dampers. This research focuses on the development of an experimental setup to analyze real-world data, emphasizing data collection for validation of developed methods. Through Physics-Informed Neural Networks, known dynamics of the system may be incorporated into the training process of machine learning algorithms, allowing for predictions that adhere to physical principles. This is employed to resolve many of the difficulties associated with accurately modelling an MR damper. Overall, this research demonstrates proof of concept for the application and use case of PINNs in mechanical systems.

Abstract

Physics-informed neural networks (PINNs) provide an alternative to traditional solvers for differential equations, specifically in tasks such as system identification and inverse problems. This study applies the recently popularized paradigm of PINNs for system identification and surrogate modelling in Magnetorheological (MR) dampers. A task hindered by the nonlinear behaviour, hysteresis effects and the presence of latent variables in certain interpretations of the MR damper dynamic model. An experimental setup was developed to analyze empirical data collected from MR dampers, incorporating a voltage-controlled MR damper, with motions actuated through a linear actuator, and various sensors for capturing damping forces and motion profiles. A data collection pipeline was developed and allows for synchronous data collection from a multitude of devices, and database storage. Collected data is useful for validating developed models, as well as setting up a foundation for experimental validation of novel methods in future work. A literature review was performed, highlighting the limitations of existing models and the potential of PINNs, cases of deployments, and innovations by authors within the literature. The research additionally constructs and validates a discretized state space model from estimated parameters. Overall, this research demonstrates proof of concept for the application of PINNs in mechanical

systems modelled by differential equations. Results demonstrated satisfactory accuracy in parameter identification, with implications for system behaviour prediction, demonstrating the potential and limitations of PINNs in this context.

*Dedicated to my family and mentors, whose unwavering support and guidance have
been the cornerstone of this journey.*

Acknowledgements

I would like to, first and foremost, thank Brian for his foundational contributions and support throughout this project. His guidance has been instrumental in shaping research endeavours. Gratitude is also extended to Abolfazl for his invaluable assistance and to Waleed and Alessandro for their availability and willingness to provide their insights. I am deeply appreciative of everyone in the ICE lab: Alex, Alessandro, Brett, Patrick, Raveen, and others not specifically named here, thank you for your friendship and for making the lab space a welcoming environment, for the fun times and laughs we had along the way. Heartfelt thanks go to my friends for their encouragement throughout this journey. I am profoundly grateful to my parents, my mother, Haili, and my father, Yifeng, for their unwavering support and sacrifices, which have allowed me to pursue my academic career thus far. Lastly, I extend my sincere thanks to my advisor, Dr. Gadsden for their guidance and encouragement throughout this project. Thank you for all the opportunities that you have opened up for me, and for pushing me to be the best I can be. As always, thanks for keeping it cool & ICE-y, for inspiring this work and me by sliding the filters of innovation over my vision. Your expertise and mentorship have been pivotal in shaping the course of this research.

Table of Contents

Lay Abstract	iii
Abstract	iv
Acknowledgements	vii
Notation, Definitions, and Abbreviations	xvii
1 Introduction	1
1.1 Research Objectives and Motivation	4
1.2 Organization of Thesis	5
2 Background and Literature Review	7
2.1 Physics-Informed Neural Networks	8
2.2 Magnetorheological Dampers and Modelling	36
3 Experimental Setup	55
3.1 Physical Setup	55
3.2 Data Collection	64

4	Surrogate Modelling and System Identification	76
4.1	Physics-Informed Neural Networks Applied to Non-linear Dynamic Systems	78
4.2	Results and Discussion	99
5	Data Analysis and Modelling	121
5.1	Mathematical Model	121
5.2	Simulation	126
6	Conclusion	130
6.1	Future Work	131
6.2	Summary of Contributions	132
A	Compilation of Literature Review Articles	134
B	Components and Specifications	147

List of Figures

2.1	General form of the Physics-informed Neural Network structure, consisting of a NN for the prediction of latent solutions, followed by an automatic differentiation process to obtain derivatives of latent solution. The network is trained based on a reconstruction of differential equations and initial / boundary conditions.	22
2.2	Architectures for the solutions of unknown variables (A) for a unified NN, (B) for independent networks.	24
2.3	Integration of physics-based and data-driven domains through feature fusion: The CNN architecture is employed as a feature extractor. Adapted from [46] and [165]	32
2.4	Cross Data-Physics Fusion, as presented by [150] predictions based on information from both the data domain (comprised of features derived from labelled monitoring data), and physics domain (comprised of features derived from unlabelled data) are simultaneously mapped to a shared space and concatenated. Both are processed through a regression layer for the final prediction.	34
2.5	A labelled depiction of an MR damper.	38
2.6	Depiction of the Bingham model of MR dampers, adapted from [139]	41

2.7	Depiction of the extended Bingham model of MR dampers, adapted from [36]	42
2.8	Depiction of the: (A) Bouc-Wen model of MR dampers, adapted from and (B) The Modified Bouc-Wen model, adapted from the work of [136]	45
3.1	The experimental setup constructed, consisting of an MR damper, linear actuator, force sensor, and programmable power supply. Setup is controlled via a unified program run on the main computer.	56
3.2	Solidworks model of the entirety of the experimental setup, adapted from [95].	57
3.3	Solidworks model of the experimental setup, depicting the Linear actuator, force sensor, brackets and mounting, and the MR damper. . .	60
3.4	Sample data collected for measured force over time for incrementally increasing applied voltages.	62
3.5	Sample data collected for measured force over measured damper velocity for incrementally increasing applied voltages.	63
3.6	Sample data collected for measured force over measured external damper velocity for incrementally increasing applied voltages.	64
3.7	Block Diagram illustrating communication between components in the experimental setup.	65
3.8	GUI display for controlling and readings from the Linear actuator, featuring elements for connection, data-streaming, motion control and recording data from all synced/connected devices.	69
3.9	GUI display for readings from the Force Sensor, featuring elements for connection, data-streaming.	70

3.10	Flowchart illustrating the process of initialization and operation of the controller and created worker threads.	72
3.11	Reading process. Items read from the serial ports are filtered through regular expressions for key information, where it is then appended to a buffer. As the serial port reading process is FIFO, in the scenario where the read information is deemed incomplete, the partial variables are saved and appended to the start of the next iteration.	74
4.1	An illustration of (A) the general Recurrent Neural Network architecture, and (B) the inner computational processes within each RNN cell.	80
4.2	An illustration of (A) the general Recurrent Neural Network architecture, and (B) the inner computational processes within each RNN cell.	87
4.3	Depiction of the normalization layer employed to normalize inputs for processing by the NN, while maintaining the physical meaning of input data, as adapted from [111]	90
4.4	Hyperparameter optimization history utilizing Bayesian optimization.	101
4.5	Hysteresis loop of measured behaviour. Plot generated via varying actuation speed from 0.5291 cm/s to 2.672 cm/s.	103
4.6	Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 0 V.	104

4.7	Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 0.6678 V.	104
4.8	Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 1.6342 V.	105
4.9	Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 2.2703 V.	105
4.10	Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 3.2046 V.	106
4.11	Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 4.003 V.	106
4.12	RMSE and MAPE with increasing applied voltage.	108

5.1 Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at varying applied voltages, representing the nominal state, undervoltage fault state, and overvoltage fault state. 128

List of Tables

2.1	Summary of popular methods within the literature for parametrically modelling MR dampers.	53
3.1	Values measured from experimental setup.	61
3.2	Values measured from the experimental setup.	67
4.1	List of parameters and their corresponding values, as determined by the PINN parameter estimation algorithm outlined in Section 4.1. . .	100
4.2	Learning rates, as determined from hyperparameter optimization, tabulated for physical parameters.	102
4.3	Measured MSE and MAPE values across varying applied voltages. . .	107
5.1	List of parameters and their corresponding values, as determined by the PINN parameter estimation algorithm outlined in Section 4.1. . .	127
A.1	Literature Compiled for physics-guided or physics-informed regularisation technique employed	146
B.1	Typical properties of the LORD RD-8041-1 MR damper, as specified by [20].	148
B.2	Electrical properties of the LORD RD-8041-1 MR damper, as specified by [20].	148
B.3	Properties of the RAS1-500S-S Force Sensor, as specified by [124]. . .	149

B.4 properties of the Ultramotion A1 linear actuator, as specified by [92]. 150

Notation, Definitions, and Abbreviations

Abbreviations

AI	Artificial intelligence
AD	Automatic Differentiation
ADAM	Adaptive Moment Estimation
FCNN(s)	Fully Connected Neural Network(s)
FIFO	First In First Out
GP	Gaussian Process
GUI	Graphical User Interface
LIFO	Last In First Out
MAPE	Mean Absolute Percentage Error
ML	Machine Learning

MR	Magnetorheological
MSE	Mean Squared Error
NN(s)	Neural Network(s)
ODE(s)	Ordinary Differential Equation(s)
PDE(s)	Partial Differential Equation(s)
PIML	Physics-Informed Machine Learning
PINN(s)	Physics-Informed Neural Network(s)
RNN(s)	Recurrent Neural Networks(s)
SGD	Stochastic Gradient Descent
SQL	Structured Query Language

Chapter 1

Introduction

Physics-informed neural networks (PINNs) have emerged as an alternative to classical solvers for differential equations and have seen applications in various engineering systems described by differential equations [27, 149, 5, 168]. Though introduced in recent years, they have rapidly gained traction as a viable alternative for the solutions of inverse problems and system identification tasks [110, 158]. As such, their capabilities may be naturally extended for characterizing the behaviour of systems with latent variables and nonlinear dynamics, such as the parametric models developed for Magnetorheological (MR) dampers utilised in this study.

MR dampers, upon application of an external magnetic field, exhibit variable stiffness and damping characteristics in response, and present inherent challenges in modelling accuracy due to their nonlinear behaviour and the presence of hysteresis effects [148, 1, 108]. Hysteresis is a nonlinear phenomenon defined as the lag between magnetic flux and changes in magnetic field strength, present in ferromagnetic materials. This nonlinearity is extended to the response of the MR damper to excitations

and presents a challenge for the development of accurate mathematical models. Numerous models, parametric and non-parametric, have been developed over the years to capture this behaviour to varying degrees of accuracy. A prominent model within literature is the Bouc-Wen model, characterized by Bouc [11] initially and later extended by Wen [153]. Spencer and colleagues later proposed a further extension that is capable of better capturing the roll-off region behaviour [136]. The increased accuracy of the model, however, comes at the cost of increased complexity with numerous parameters introduced. Addressing this issue, the use of PINNs within the identification and modelling process allows for the incorporation of known physical principles of prior works governing MR damper dynamics as learning biases, ensuring adherence to said principles while maintaining a degree of adaptability due to the ML implementation, presenting a valid alternative for system identification [72, 33, 16].

PINNs are utilized in this study to address the challenge of latent variables in MR damper modelling, which traditional identification may struggle with. The core operations of PINNs are based on deep learning architectures, such as Neural Networks (NNs), renowned for capturing complex, nonlinear mapping through purely sampled data of the system due to capabilities for generalization and universal approximation [42, 123]. However, conventional NNs often require an extensive quantity of representative data for proper generalization and cannot generalize beyond the scope provided. They are referred to as *black box* models due to the non-interpretative nature of their inner operations, which poses questions of reliability. These concerns may be partially alleviated in the PINN model through the incorporation of prior knowledge within the optimization process [110, 21]. The developed model based on

the PINN paradigm in this study involves a hybrid approach utilizing both the measured experimental data and the known physics of the MR damper system. Based on results from the identification algorithm a discretized state space model was constructed and validated on experimental data, as a foundation for future work in the area involving novel machine learning and estimation techniques.

To demonstrate the feasibility and effectiveness of the proposed methodology, a dedicated experimental setup for MR damper analysis was constructed for the collection of real-world empirical data. This setup includes an MR damper powered by a controlled programmable power supply and actuated through a linear actuator. The setup is equipped with various sensors, on-board and external, for the observations of the effective damping force with respect to motion and velocity profiles. A program was developed for the synchronous collection of sampled readings from the plethora of devices involved within the experimental setup, with options for storage in an established database for use in future experiments. Data collected from this experimental setup serve as the foundation for developing and validating techniques employed. Effectively, the constructed setup provides a tangible platform for demonstrating the applicability of PINNs for modelling non-linear systems with latent variables.

In all, the study presented herein has the objective of developing methodologies for the identification of physical parameters in systems characterized by parametric equations under a variety of operating conditions. From this, the developed model is expected to aid in further developments in setting a foundation for validating modern control, estimation and machine-learning strategies.

1.1 Research Objectives and Motivation

In recent years, PINNs have gained popularity as a framework for solving the various forms of differential solutions pertaining to physical systems, as well as system identification through the solution of inverse problems. One such system that this framework may be extended to is the MR dampers, which present challenges in obtaining said solution due to the inherent nonlinear behaviour, as well as the existence of latent variables inherent to certain established physics-based models. As such, the primary motivations for this research may be summarized as:

1. **Development of an experimental setup data collection system:** Developing a data collection system for an MR damper experimental setup, which involves multiple devices, stemming from the requirement to gather empirical data for validating theoretical machine learning and estimation strategies.
2. **Capturing nonlinear dynamics in systems with latent variables:** MR dampers exhibit highly nonlinear behaviour and are characterized in certain models by unobservable variables. Through the expressiveness of conventional neural networks, it is the objective to capture the behaviour of latent variables, as well as nonlinear mappings directly, enabling a more interpretable representation of the system dynamics.
3. **System identification:** Estimating the parameters of MR damper models from experimental data is necessitated to ensure model accuracy and robustness to a variety of operating conditions. PINNs provide a suitable framework for incorporating observational data and prior knowledge into the estimation process for a principled inference of physical parameters.

4. **Developing models for efficient simulation and optimization:** Circumventing certain limitations associated with conventional neural networks and deep learning in general in dealing with scenarios with limited data available. This is performed through the incorporation of physical knowledge of the general form of the governing equations, enforced within the training process of the networks. Through this, the behaviour of MR dampers may be modelled with minimal data requirements.

5. **Empirical validation for developed models:** With the data from the experimental setup developed as detailed in this study, the validation of the accuracy and robustness of the PINN strategy implemented herein may be performed, demonstrating proof of concept in applications to real-world engineering systems. In addition, methodologies developed for future work within this domain may utilize this implementation of the experimental setup for development and validation as well.

Overall, this research endeavours to demonstrate proof of concept for the implementation of PINNs in a broader variety of mechanical systems, providing a foundation for future studies in domains such as condition monitoring and anomaly detection.

1.2 Organization of Thesis

The subsequent sections of this research are structured to offer an examination of the application of physics-informed machine learning in the context of mechanical systems

and are arranged as follows: Section 2 provides background information and a literature review, covering the fundamental concepts of physics-informed machine learning alongside established models of MR dampers. Following this, Section 3 details the experimental setup of the MR damper and describes the data collection pipeline. Moving forward, Section 4 explores system identification and surrogate modelling using physics-informed neural networks, presenting insights into their effectiveness in this domain. Section 5 discusses the state space modelling of the system. Finally, Section 6 concludes the paper, summarizing contributions and outlining avenues for future research.

Chapter 2

Background and Literature Review

This section explores recent developments in PINN techniques employed in real-world engineering scenarios, as well as popular methodologies in the literature for representing the behaviour MR dampers mathematically.

The content of this chapter is structured into two independent sections; The first section examines the prominent literature available for PINN, with the primary focus of the review being the application to physical systems and their implementation for addressing challenges in the physical representation of the system, system identification, and condition monitoring. Within this section, the theoretical foundations of PINNs, variations and algorithmic developments by various authors, and practical applications of the technique are examined in detail. The focus of this review section is primarily on their relevance to mechanical system modelling.

The second section of this chapter explores the existing literature on existing techniques for characterizing the behaviour of MR dampers mathematically. In this section, the review will primarily encompass modelling approaches and their effectiveness in representing the non-linear hysteresis phenomena inherent to MR dampers.

This section will also provide brief comments on the advantages and limitations of each model introduced, framing it within the context of their applicability to accurately represent different aspects of MR damper.

2.1 Physics-Informed Neural Networks

This section detailing current applications of PINNS is an excerpt from a review article by the author of this thesis, which has also been published on arXiv pre-prints, available as reference: [158]. As of writing, the review article has completed the review process with the Expert Systems with Applications (ESWA) Journal by Elsevier, available as reference: [157]

Throughout the last decade, ML algorithms have witnessed rapid development in a variety of industries for their efficacy and ability to extrapolate patterns from data. Through available data, ML models are capable of accurately representing the relation between a given set of inputs and outputs with minimal human interference. This property rendered ML models suitable for the representation of complex systems in which the relation and parameters governing behaviour are not easily obtained, as evident by the influx of interest in recent years [142].

In general, ML algorithms fundamentally operate as data-driven processes designed to map the relationship between specific inputs and their corresponding outputs. These algorithms typically follow a structured optimization procedure, wherein the model's predictions are assessed and iteratively refined to more accurately reflect the provided data. As can be expected, the performance of ML models is heavily reliant on the data upon which they are optimized.

Indeed, restrictions to data quality and availability are amongst the main concerns when choosing to work with ML [85]. For many engineering applications, the collection of sufficient quantities of data to build a reliable model may be challenging, costly, and/or not feasible due to time and resource constraints [85, 112, 125]. A considerable amount of clean, representative, and non-sparse data is required to properly formulate the model [85]. Conventional implementations often leverage synthetic data and data augmentation techniques [131]. Insufficient quantities and/or non-representative data often lead to a skewed representation of system behaviour that is inconsistent with the true underlying physical relationship, ultimately resulting in misleading conclusions.

Furthermore, ML models are considered to be "*black box*" models, in which intermediary information between input and output is not relevant nor required in producing a correlation between some input and output. That is to say, the underlying mechanism of a system is often not considered in the development of these models, and while effective in representing a system, may not further contribute to the understanding of said system [117].

Concerning the representation of systems based on prior knowledge, physics-based modelling has also been traditionally employed. However, models developed purely on the understanding of the system see limited use in modelling real-world systems, due to the many challenges to its applicability. First and foremost, physical models are often computationally expensive to model. Due to the computational complexity of most real-world physical systems, and the variety of governing equations involved for each specific physical agent or phenomenon, the cost required to fully model said systems is considerable [51]. Furthermore, physical models often represent an

imperfect interpretation of the system, due to a missing or incomplete understanding [166].

Researchers have come to the realization that the combination of physical and data-driven models was the next step in the prediction and modelling of system behaviour. Indeed, varying methodologies of varying degrees of interpretability, seeking to combine insights prior knowledge with conventional black-box models have been proposed, ranging from the the combination of ML theories with state space models [2, 167], or alternatively digital twins [129, 132, 130, 133]. The paradigm of PIML was initially conceptualized by Lagaris and colleagues[69], who first demonstrated the use of artificial neural networks for solving ordinary and partial differential equations. Karpatne et al. formally introduced this paradigm in their study of theory-guided data science, outlining various avenues of integration between domain knowledge and data-driven solutions, through which, new physics-informed models are capable of benefiting from both physics-based and data-driven methods concurrently [57]. Since their publication, a plethora of studies regarding the PIML paradigm have been conducted. Various authors, most notably Raissi and colleagues, further advanced the integration between theory and data science with the introduction of PINNs, whereby physical laws in the form of governing equations are encoded within the NNs [110]. The properties of deep learning architectures made it suitable in their use case, for approximating the solutions of PDEs modelling physical systems [110]. Raissi et al. made use of the NN architecture in their demonstration of a systematic methodology for solving non-linear partial differential equations [110]. Karniadakis and colleagues reviewed popular methodologies by which the integration of physics and data-driven

techniques takes place, as well as presented their insights on limitations and potential applications of the technique [56]. Meng and colleagues also surveyed a variety of work in the area of PIML and presented a summary of core motivations behind their development, popular physical governing equations employed in various applications, as well as methods of integration [91]. From the literature, it is evident that applications of PIML have been prominent in a variety of fields.

The paradigm of PIML combines the interpretive capabilities of machine learning algorithms with the foundational understanding of physics, employing prior knowledge to guide the learning process. PIML methods benefit from reduced reliance on labelled training data, as physics-based guidelines for optimization can constrain the solution space and provide further insights with regard to intermediary computations, even in data-scarce scenarios. These methods offer better explainability to the end user in the context of explainable artificial intelligence (XAI), which is a growing consideration for the wide adoption of AI techniques.

Regularization techniques have been fundamental in training ML models since their inception. Conventional regularization techniques, such as the Lasso (L1) or Ridge (L2) regressions, incorporate additional penalty terms within the loss function established, with the objective of reducing the model's capacity to overfit data; Certain interpretations of data by the model that may not be reflective of the general behaviour of the system. Ultimately, this is performed to obtain simpler and more robust solutions.

While this has been used extensively, authors have extended its usage, whereby the usage of physics-based regularizers is employed within conventional deep learning

algorithms, an approach seeking to leverage the explainability and fidelity of physics-based models to enhance the accuracy, interpretability, and robustness of conventional data-driven solutions. Prior knowledge regarding the physical system is integrated as a part of the learning process, either as constraints or regularizers, effectively encoding the physical constraints to aid in guiding the optimization process in producing physically meaningful solutions.

Past implementations of physics-based regularization involved solving the physical equations and incorporating them as constraints in the optimization problem [118, 98]. This approach was computationally expensive and is limited to well-understood physical systems. Techniques have been developed that more effectively utilize physics-based modelling and ML in tandem. In prominent studies, such as the work of Raissi and colleagues [110], regularization based on the residuals of the governing differential equations was proposed, utilizing the approximative capabilities of conventional NNs in tandem. This foundational work, termed PINNs, has seen numerous citations since its inception. A summary of compiled literature employing variations of this technique, with varying engineering applications is provided in Table A in Appendix A.

Physics-guided regularizations consist primarily of tailoring constraints that directly alter the data-driven model in the training phase to favour predictions that are consistent with underlying physics. Constraints of this type are also known as learning biases, as characterized in the work of Karniadakis and colleagues [56], and implemented through physics-informed loss functions. These loss functions penalize deviations from physics, making the model more likely to produce physically plausible solutions.

Conventionally, the loss function employed in ML algorithms is a measure of the empirical difference between the model prediction and ground truth, with the objective of minimizing the loss function through an iterative process. Model loss is optimized by adjusting the parameters of the model to reduce the aforementioned difference in model predictive capabilities versus ground truth. In contrast, a physics-informed loss function incorporates additional information about the system being modelled, such as physical constraints, conservation laws, and other known properties of the system in tandem with the penalization of deviations from ground-truth observations. Through this framework, the ML algorithm may more effectively constrain the prediction space to avoid violations of physical principles.

Algorithms introduced in this format simultaneously minimize errors in both the labelled data and physical constraints through penalization of deviations, reflected in the structure of the loss functions implemented, whereby the physics-informed loss function is comprised of a data-driven loss term and a physics-based loss term. The data-driven loss term measures the error between the predicted output of the model and the observed response. The physics-based loss term enforces the solution that satisfies the prior known physics of the problem through adherence to governing equations specific to the problem.

Conventionally, compliance with observed data (data-driven loss) is achieved by minimizing the residual between predictions of the network and true state and is performed with a variety of distance evaluators such as the mean squared error (MSE) or cross-entropy error (CSE). Compliance with known physical laws is case-specific and varies in implementation, however, the aforementioned methods for evaluation have seen many implementations in the literature. The general form of the loss

function then, may be represented as:

$$Loss_{total} = \lambda_{data}\mathcal{L}_{data}(Y_{prediction}, Y_{target}) + \lambda_{phys}\mathcal{L}_{phys}(Y_{prediction}) \quad (2.1.1)$$

Where the parameters λ_{data} and λ_{phys} are the regularization factors to adjust loss terms to best-fit system characteristics. Thus in this format, authors have introduced a methodology for the incorporation of governing equations to influence the direction of loss minimization in networks. In literature, physics-informed regularization has been employed to incorporate knowledge of the expected fault signatures of the system under different failure modes, to ensure that the model is able to accurately detect and classify faults, even in the presence of noise or other confounding factors.

For instance, Sun and colleagues proposed a methodology for the non-destructive detection and quantification of micro-crack defects, a framework based on the electromagnetic acoustic transducer, which functions by exciting guided waves for crack detection [141]. Sun developed a novel physics-informed architecture that they have termed *GuwNet* [141]. The proposed network seeks to employ various deep learning modules such as convolutional layers, dense layers, and GRU layers in conjunction with the introduction of physical parameters for the approximation of variables of crack propagation. The physical process is represented through various connections within the data-driven and physics-based layers and parameters within the network[141]. The network is optimized by hybrid feed-forward and feedback loss functions, comprised of empirical and physics-informed error terms to integrate the physics of ultrasonic wave testing into the training process of the network. Physics-informed terms are derived from the relationship of defect depth, and quantified by

transmitted wave intensity and reflected wave intensity of the ultrasonic guided wave nondestructive testing method. The method demonstrated great promise in the detection of length, depth, and direction of crack propagation, and was shown to have significant improvements in accuracy in comparison to conventional deep learning approaches[141].

Freeman and colleagues proposed a hybrid approach for anomaly and fault detection in turbine rotor blades, whereby fault features acquired from turbine power signals are combined with environmental data to ensure conformity to the dynamics of the hydro-kinematic rotor[31]. The framework extracts statistical features by means of continuous wavelet transforms, and categorized via multi-nomial regression. The time domain features selected were proven by the authors to be physically significant, accurately reflecting the high-frequency fluctuation behaviour in signals with respect to turbulence intensity[31]. Turbulence intensity is classified with a NN, based on time-domain features extracted from the reduced feature space and physically constrained through a hybrid loss function, whereby deviation from the dynamics of turbulence intensity is penalized [31].

Regularization has also been applied with respect to applications in fatigue stress and life monitoring. Zhang and colleagues constrained the process of creep-fatigue life estimation in a stainless steel specimen with physics-augmented feature engineering and physics-informed regularization [173]. The developed feed-forward model introduces two physics-informed loss terms that take into account and penalize physical violations with regard to fatigue life [173]. From the expected behaviour of creep-fatigue in the specimen, the authors added physical constraints in the form of penalization for negative values, as well as extreme values of creep-fatigue life within the

loss function. The model constructed boasted superior performance when compared with benchmark empirical and purely data-driven methods [173].

In a similar vein, Kim and colleagues adopted a data-driven prognostics model that incorporates low-fidelity physical features in the optimization process [61]. The authors presented an innovative methodology for obtaining training parameters for unlabelled extrapolation data[61]. In general, the process for obtaining the extrapolated region, that is, the target of the prognostics framework, involves the physics-based regularization term that penalizes deviation from the low-fidelity physical model [61]. To this effect, the model is optimized to minimize interpolation error with available data, as well as extrapolation error, from the embedded physical model. The authors validated their approach with their verification of fatigue crack growth with respect to Paris’s law.

De Santos and colleagues built upon conventional frameworks for monitoring the progression of fatigue on off-shore wind turbines by extending the monitoring time period [23]. Conventional evaluation of damage monitoring models is based upon the model’s ability in ten-minute fatigue damage estimations, whereas De Santos et al. have extended this methodology for monitoring long-term fatigue accumulation. The PINN model proposed focuses on minimizing the Minkowski logarithmic error, providing a more conservative estimation of fatigue damage in the form of damage estimation moments [23]. The loss function was derived such that accuracy between the model’s ability to predict short-term and long-term damage is not compromised.

Li and colleagues further extended the physics-informed loss function to meta-learning, in their proposed strategy for estimating tool wear [79]. The method integrates both physically derived model inputs, as well as physics-informed loss terms

with data-driven models over a series of ML models for the purposes of meta-learning. Meta-learning is defined as the systematic observation and learning of learning from meta-data or the observed experience accrued by ML models and their performance on various tasks. Meta-learning may be classified as a sub-field of ML, whereby artificial intelligence models are trained to solve tasks or problems more efficiently and effectively automatically [79]. In their work, the inherent principles of tool wear are learned for applications in tool wear predictions under varying tool wear rates [79]. Through the various parameters derived from the dynamic relationships governing tool wear, the authors derived the input feature space of the various deep learning and ML algorithms tested, for enhanced interpretability and robustness. Individual ML models are constructed on the basis of physics-informed data-driven modelling with *cross physics-data fusion*. Initially conceptualized by Wang et al. [150], the model represents a methodology to fuse data from both the physics and data-driven features. The meta-learning model is employed to learn the experiences of three ML models and their predictions of the degradation state of the asset at different stages of wear [79]. The algorithms tested were optimized via the physics-informed loss function, whereby constraints to the tool wear rate are imposed based on inherent attributes of tool wear and relations governing tool wear and cutting force.

Physics-Informed Neural Networks

PINNs are a rapidly growing field that utilizes NNs to learn patterns and relationships while also incorporating the underlying physics as learning biases, referred to as "physics-informed" in that they incorporate physics-based knowledge or constraints into the model training process. In literature, PINNs have been shown to be capable

of generating accurate predictions with small quantities of labelled data as collocation points, rendering them suitable for applications where data acquisition is expensive or challenging. Furthermore, the implicit constraints to the system and produce physically meaningful predictions, providing a pragmatic alternative for the solution of differential equations, as well as inverse and surrogate modelling.

The concept of employing NNs, with their computational capabilities, for solutions to differential equations was initially presented in the work of Lagaris [69]. Raissi and colleagues [110] popularized the concept through their more recent study, demonstrating the effectiveness of their methodology in solving forward and inverse problems pertaining to the governing partial differential equations of various well-established physical systems. The effectiveness of PINNs, as defined in Raissi’s work [110], is derived in part, from their usage of the universal approximation capability of NNs [43], which states that a NN with a single-layered feed-forward network with an activation function may approximate any function, provided that it is comprised of a sufficient number of neurons. Researchers have extended this property to the solution of a variety of complex, non-linear differential equations, in which numerical or empirical solutions are difficult or impossible; the PINN learns the mapping between the input data and the output variables while enforcing the physical constraints of the system [110]. Through this framework, researchers can build accurate models that provide insights into the underlying physical processes, making them a viable alternative in many scientific and engineering applications [114].

The original PINN architecture by Raissi is based on the feed-forward NN structure, and employed to solve the first-order non-linear PDE [110]. Various names

exist for this structure in literature such as Feed-Forward Neural Networks, Artificial Neural Networks (ANNs), Multi-layer Perceptron Neural (MLP) Networks, and Deep Neural Networks (DNNs). The feed-forward NN consists of multiple layers of interconnected nodes, or neurons, that transmit information through weighted connections. In the context of PINNs, the input layer of the network corresponds to the physical domain, while the output layer represents the solution to the problem of interest. The intermediate layers, also known as hidden layers, provide the necessary computational power to map the input to the output.

An ANN may be described as a series of non-linear transformations. In terms of a mathematical definition of the network: for a given input layer of N neurons, and may be denoted as $X = \{x_1, \dots, x_n\}$, whereby x_i represents a feature within the input space X . The network may be defined to host H hidden layers, with each layer containing M neurons. From this, the output of the I -th hidden layer may be represented as $A^I = \{a_1^I, \dots, a_m^I\}$, where a_j^I represents the j -th neuron in the I -th hidden layer. For each hidden layer, the output A^I is computed through an element-wise application of non-linear activation function σ_i to the weighted sum of inputs from the prior layer $I - 1$, which may be written as:

$$z_j^I = \sum (w_{ji}^I * a_i^{I-1}) + b_j^I \quad (2.1.2)$$

Where w_{ji} represents the weight connecting the i -th neuron in the prior layer $I - 1$ to the j -th neuron in the current layer I , a_i^{I-1} represents the output of the i -th neuron in the prior layer, and B_j represents the bias term associated with the j -th neuron in

the I -th hidden layer. The output of the i -th hidden layer is computed as:

$$a_j^I = \sigma^I(z_j^I). \quad (2.1.3)$$

The output layer is comprised of K neurons, with predicted output denoted as $Y = \{y_1, \dots, y_k\}$. Thus, the output of the NN may be computed as:

$$z_j^{H+1} = \sum (w_{ji}^{H+1} * a_i^H) + b_j^{H+1}, \quad (2.1.4)$$

where w_{ji}^{H+1} represents the weight connecting the i -th neuron in the H -th hidden layer to the j -th neuron in the output layer, a_i^H is the output of the i -th neuron in the H -th hidden layer, and b_j^{H+1} is the bias term associated with the j -th neuron in the output layer. Collectively, this may be referred to as:

$$\mathbf{z}^{H+1} = \mathbf{w}^{H+1} * \mathbf{a}^H + \mathbf{b}^{H+1}. \quad (2.1.5)$$

The PINN employs this existing framework to be an approximator of the solution to the differential equation. In the general case, the non-linear PDE parameterized by γ , as well as its initial and boundary conditions may be represented by the form:

$$\mathcal{F}\left(x, t, u, \nabla u, \dots; \frac{\delta u}{\delta t} \dots; \gamma\right) = 0, x \in \Omega, t \in [0, t] \quad (2.1.6)$$

$$u(x, t = t_0) = g(x), x \in \Omega \quad (2.1.7)$$

$$u(x, t) = h(x, t), x \in \delta\Omega, t \in [0, t] \quad (2.1.8)$$

Defined in the domain Ω , where $\Omega \in R^d$ with boundaries $\delta\Omega$. \mathcal{F} represents the

non-linear function that defines the relationship between unknown function u , its derivatives, and its parameters. The PDE defined has hidden solution $u(x_1 \dots x_n, t)$, with input space that may be composed of spatial variables x and temporal variables t . The PDE has initial conditions g and boundary conditions h . The NN seeks to make a computational approximation of the solution u_{NN} from input space [110, 55]. The approximation of solution space by the NN is denoted as:

$$u_{NN}(x_1 \dots x_n, t) \approx \mathbf{z}^{H+1} \quad (2.1.9)$$

The derivatives of this approximation are calculated through automatic differentiation, functions through the application of the chain rule of calculus to compute the exact derivatives of a function with respect to the spatial and or temporal input variables [7]. Utilizing the predicted solution u_{NN} and its derivatives, the PDEs and their initial and boundary conditions are reconstructed. This deviation of predictions is evaluated with respect to the sampled observations, and with respect to the reconstructed differential equation itself, represented as:

$$Loss_{total} = \lambda_1 Loss_{Data} + \lambda_2 Loss_{PDE} + \lambda_3 Loss_{BC} + \lambda_4 Loss_{IC} \quad (2.1.10)$$

With parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ representing weights for the adjustment of each loss term. Deviations are typically evaluated as an L^2 norm and are minimized during the back-propagation process, whereby NN parameters (weight and biases) are adjusted iteratively following the governing equations, as represented in 2.1. Minimization of the total deviation through the optimization algorithms such as gradient descent

allows the network to learn the mapping between the input and output space, while simultaneously complying with known physical laws and constraints. Through the introduction of learning biases, PINN significantly relaxes restrictions in terms of the quantity of data required to properly train deep learning algorithms [159].

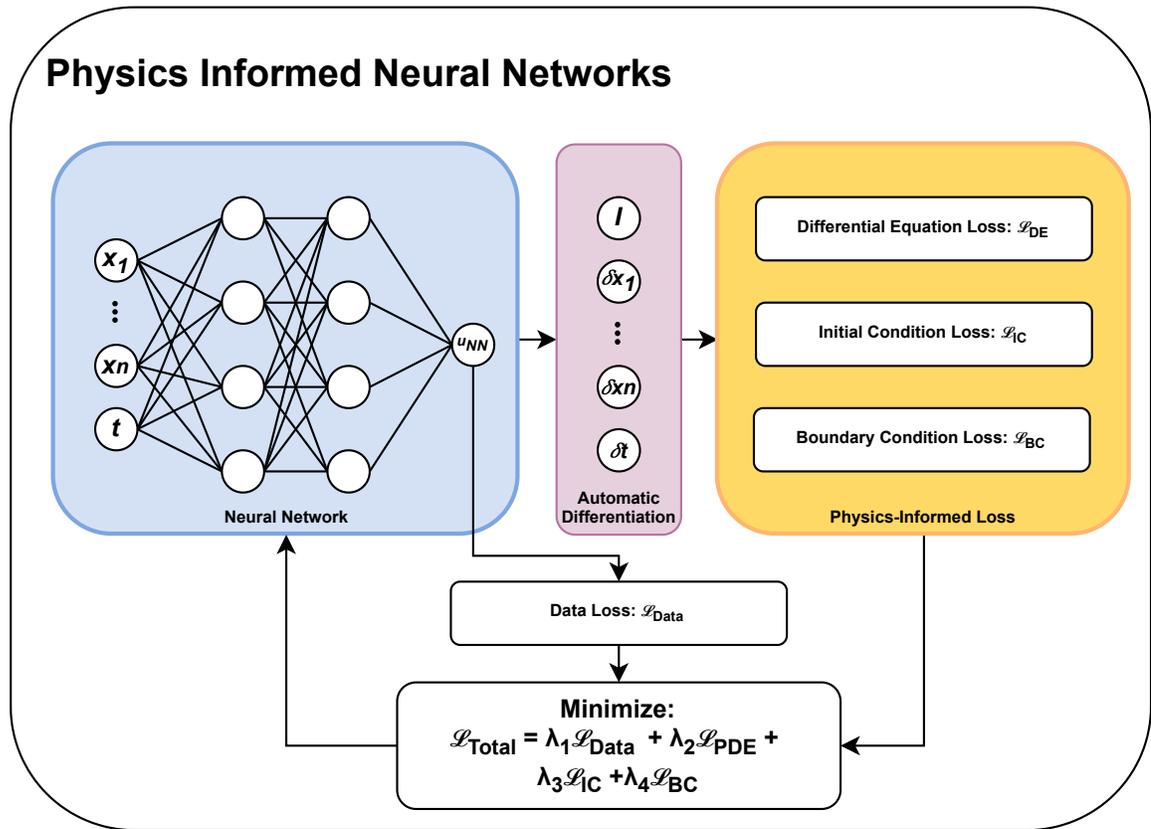


Figure 2.1: General form of the Physics-informed Neural Network structure, consisting of a NN for the prediction of latent solutions, followed by an automatic differentiation process to obtain derivatives of latent solution. The network is trained based on a reconstruction of differential equations and initial / boundary conditions.

In the context of condition monitoring applications, PINNs allow for accurate predictions by incorporating both data-driven and physics-based approaches. PINNs

can handle sparse and noisy data, extrapolate beyond training data [61], and provide interpretable results. They also enable early fault detection, reduce false alarms, and can be used for online monitoring. Since their initial popularization, a plethora of subsequent implementations that followed their publication have employed the same feed-forward architecture. However, experimentation with other popular deep learning architectures, such as the CNN, RNN and its variants, encoder and decoder networks, as well as graph NNs have seen relatively smaller quantities of deployment in literature. The following sections will detail the integration of physics-based regularization with a variety of NN architectures.

Data-Driven Solutions to Differential Equations

Applications of PINNs to the solution to governing differential equations of physical systems methods vary greatly across industries and have been applied to numerous areas in which the system dynamics are known beforehand. For instance, within the domain of solid mechanics, PDEs of physical parameters such as elasticity, deformation, and structural response are determined with the purpose of continued structural health monitoring. One such example is evident in the work of Haghghat and colleagues [41], who developed a method for surrogate modelling and model inversion with respect to behaviour in structures defined by the principles of linear elasticity. This is performed through the incorporation of governing PDEs and various constitutive equations into a PINN for parameter estimations. Through their experimentation, the authors demonstrated the proof of concept through a model of the displacement field under elastic plane-strain conditions. For their use case, the authors compared the effects of a collective network with shared hidden layers 2.2

(A), as opposed to utilizing the PINN framework to solve for individual outputs irrespective of the others 2.2 (B), with each output being solved by a PINN drawing data from a collective input space. The authors have concluded that, while in principle, a wider network will allow individual associations to be made between sections of the network and output, it was more effective for each variable of the solution to be calculated separately. The authors attributed this to the hyperbolic tangent activation function used, being incapable of accurately representing the cross-dependencies of the network outputs in a manner faithful to kinematic relations.

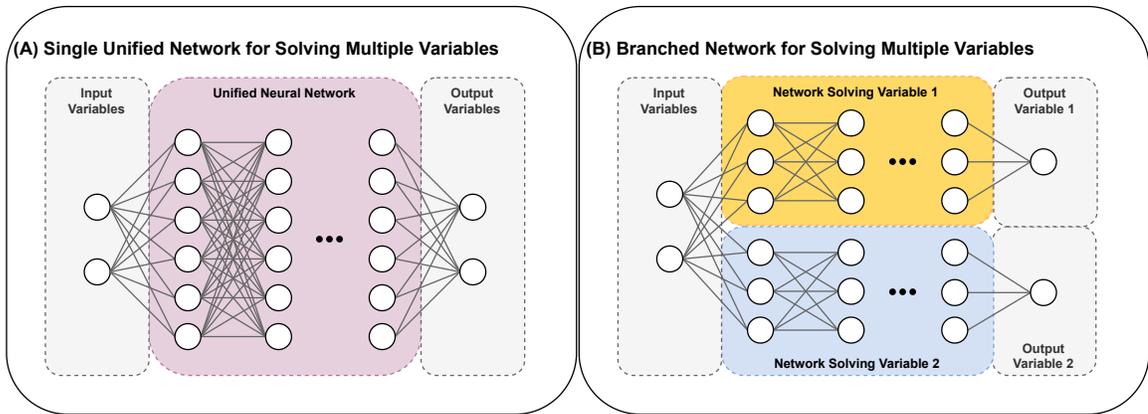


Figure 2.2: Architectures for the solutions of unknown variables (A) for a unified NN, (B) for independent networks.

Anton and colleagues applied the PINN framework for material parameter estimation with inputs in the form of full-field displacement data [4]. With respect to structural health monitoring on existing infrastructure, the estimation of material parameters of structural components may be a method of evaluation of degradation. To that effect, the authors derived the solutions to the momentum balance equation, as well as the constitutive equations for linear-elastic materials with the classic PINN

architecture [4]. Physical regularization was implemented with respect to the PDE established, as well as labelled data available for boundary conditions and observed deformation.

Similarly, Kharazmi and colleagues estimated the structural parameters of a flexible cylinder structure subjected to vortex-induced vibrations from the hydrodynamic force, with the objective of evaluating structural damage due to fatigue [60]. Utilizing the PINN framework, the authors solved the linear beam-string equation, which governs the motion of the cylindrical structure in question [60].

Bharadwaja and colleagues utilize a PINN to model and quantify uncertainty in the elastic deformation of heterogeneous solids [10]. More specifically, isotropic linear elastic behaviour is assumed to solve the governing differential equation for the approximation of momentum balance and constitutive equations governing elasticity. The proposed PINN is optimized via the physics-based loss function, representing model error to governing differential equation, as well as the Dirichlet, Neumann boundary conditions, the boundary conditions associated with fibers and voids, and initial conditions. From their analysis, the proposed physics-informed methodology returned results that are similar to that of the Monte Carlo finite element simulation model, designated as the benchmark model in this scenario [10].

As another example, Rautela simulated guided waves for monitoring structural health with applications in aerospace applications [113]. The framework revolves around using a PINN to solve governing PDEs associated with wave propagation. In their study, the one-dimensional wave equation with Dirichlet boundary conditions is formulated as the target of the loss function, and predictions by the PINN are continuously optimized by the loss function to more accurately reflect the physical

governing PDE [113].

Zhou and colleagues proposed a methodology for fatigue life estimation, physically constrained by a hybrid loss function within a probabilistic PINN framework [176]. Through the feed-forward model, the stress-life relationship is approximated. Physical violations are determined through the evaluation of select collocation points, whereby the ground truths are approximated by the probability distribution out-putted by the feed-forward model [176].

Finally, Mai and colleagues employed the PINN architecture in predicting structural instability in truss structures [86]. The network outlined is a representation of the displacement field of the structure, and analysis of parameters allows for the location of critical points susceptible, given the input load factors [86]. Optimization is performed via the minimization of the physics-informed loss function, which represents, physically, the residual load and stiffness characteristics of the structure. In all, the method yields superior accuracy through the various example validations on several truss structures [86].

With applications to machinery fault detection and classification, Shen and colleagues proposed a novel machine fault classification framework employing a unique PINN framework based on Hamiltonian mechanics, whereby the model is trained to represent the energy conservation of the system in healthy and abnormal states [127]. Hamiltonian systems are those that obey Hamilton’s equations of motion, which describe the time evolution of a system’s state variables in terms of its energy. Based on the principle of Hamiltonian mechanics, the evolution of a physical system is described via the energy of the system as a function of its position and momentum. This network is termed *Hamiltonian Neural Network* (HNN) and may be considered

a class PINNs specifically tailored towards the modelling dynamical systems governed by Hamiltonian equations. This incorporation allows networks to predict the evolution of a system over time [40]. In the work by Shen and colleagues, the authors applied this concept for the classification of faults in rotating machinery. Estimations of system energy signatures are derived from observed sensor measurements through the HNN. Subsequently, parameters of the HNN are extracted to form the total energy function, which is used as the input features for the classification algorithm based on the conventional RF algorithm [127].

An abundance of studies has also been performed in optimizing or complementing the available data from sensors for monitoring applications. Through optimization, the objective of designed systems is to maximize the relevant and informative data for monitoring the system. An example of this optimization process with PINNs may be seen in the work of Zhu and colleagues, who optimized sensor placement locations for the monitoring of low-rise buildings in response to wind pressure [178]. The ML model is trained on data generated from a physical simulation by means of a high-fidelity finite element computational fluid dynamics model [178]. From the data provided, the ML model seeks to construct a surrogate model of pressure-field in real time. This surrogate model is further embedded within a NN for the optimization of sensor placement locations.

For inference of non-observable sensor data, jadhav and colleagues performed condition monitoring for fouling conditions on system health with respect to an air preheating system in thermal power plants [48]. Issues arising from the lack of available sensors on the interior of the system are resolved with the proposed PINN architecture based on the non-dimensionalized governing equations for heat transfer for fluid and

metal interfaces [48]. The authors employed a series of multiple PINNs in parallel, operating from the same set of input features to resolve a plethora of equations governing heat transfer. PINNs are regularized via the physics-informed loss function, composed of the loss components of the governing equations, boundary conditions, and interface conditions.

From the various applications listed, the accuracy of sensor data is critical for the collection of data faithful to the system. Decisions based on inaccurate or incomplete information may lead to sub-optimal outcomes or catastrophic consequences, and as such, one direction of this architecture has been the reconstruction of corrupt sensory data to allow users a holistic view of system operations. In particular, in the work by Peng and colleagues, the authors proposed a PINN structure to reconstruct data with significant corruption from sensor errors [103]. The networks proposed are based upon the Least Absolute Deviation and median absolute deviation, whereby the PINN architecture is continuously optimized by minimizing the residual between data-driven and physical models. The design of the architecture was validated on several classical problems involving PDEs, such as the Navier-Stokes equation, Poisson’s equation, and wave equations, whereby the algorithm was capable of accurately recovering governing equations from corrupted observation data [103].

In other avenues of research, PINNs have been applied for the modelling of dynamic systems, as demonstrated in the work of Zhou and colleagues [177]. The authors applied the PINN framework for the evaluation of reliability in multi-state systems. The governing equations for Markov processes take the form of differential equations, and as such, aligns with the PINN framework. The authors utilized the gradient surgery method for multi-task learning as outlined in the work by Yu et al. [169] to

improve the PINN’s precision in approximating solutions to differential equations by alleviating issues with imbalanced gradients during training phases. For multi-state system reliability evaluation, the PINN solves for the state estimates of systems with the input of time instant. As with the traditional PINN, the network is penalized based on loss with respect to boundary conditions, and with respect to approximation of governing equations.

Physics-Informed Regularization in Tandem with Other Deep Learning Architectures

A plethora of literary works employ the inherent symmetries and invariances encoded by various conventional deep-learning architectures in compliance with the philosophy of physics-guided regularizations. Literary works presented in this section mainly utilize physics-informed regularizations as the primary methodology to encode physical knowledge into the system, taking advantage of the efficacy of certain architectures for specific data types, researchers have drastically innovated upon the structure of the original PINN and employed the framework in their own fields of specialization.

For instance, with respect to the CNN architecture, their unique convolutional layers are valued for their capabilities in automatically extracting features without the need for manual feature engineering, making them invaluable in complex applications whereby the relevant features are difficult to understand or quantify. Studies employing the CNN architecture can be seen in the works of McGowan and colleagues, who monitored the porosity during the additive manufacturing process with their introduction of a set of loss functions [90]. The regularization of the network comprises standard cross-entropy data loss, as well as losses informed by physical

parameters that penalize deviations from ideal simulated melt pool temperature and length-to-width ratio and relative error prior to normalization.

As another example, Zhang and colleagues established a surrogate model for the estimation of structural seismic response, informed via equations of motion representing a dynamic system subjected to ground excitation [172].

Several instances of literature attempt to employ the physics-informed loss function as a methodology to minimize deviations between established physical and data-driven domains. For example, Shen and colleagues adopted a hybrid approach in their development of a physics-informed CNN model for fault detection in bearings under varying rotational speeds [128]. The proposed CNN model and the physics-based threshold model operated co-currently to evaluate the health class of bearings. The threshold model is established based on known limits with regard to the amplitude of envelope spectra of healthy and damaged bearings [128]. Subsequently, a customized physics-informed loss function is implemented, which serves to penalize the model for predictions that deviate from known physics, as represented by the threshold model. Through this format, however, the authors have made the simplifying assumption that predictions of physics-based models are correct, or rather the probability of predictions being correct is very high, due to the extreme thresholds set [128].

Huang and colleagues explored a similar approach for the combination of the physical and data domains [46]. The authors trained a CNN employing a finite element model for applications in structural health monitoring. Through their designed framework, the authors sought to incorporate predictions from both the physics-based finite element model and data-driven method [46]s. The CNN proposed functions as a

set of feature extractors that operates simultaneously based on inputs from the finite element model-driven physics domain, and the data domain [46]. Physical constraints are encoded in a classifier through a novel cross-physics-data domain loss function, whereby predictions of the classifier are evaluated with respect to the labelled data, as well as the discrepancy of features between the physical domain and the data domain.

Of a similar nature, Yin and colleagues monitored structural damage localization in bridge structures due to loads applied by vehicles [165]. The authors developed a numerical simulation of the structure and, using the physics-informed loss function sought to fuse features from the physics and data domains, the workflow of which is visualized in Figure 2.3. Processed data from both domains are fed through the Visual Geometry Group 16 architecture [134], whereby damage features are extracted from the time-frequency map of acceleration signals. The optimization was carried out with a hybrid loss function comprised of data-driven cross-entropy loss and physics-informed loss penalizing deviations from the physical domain established via numerical simulations. Effectively, the network seeks to minimize discrepancies between the physical and numerical models [165].

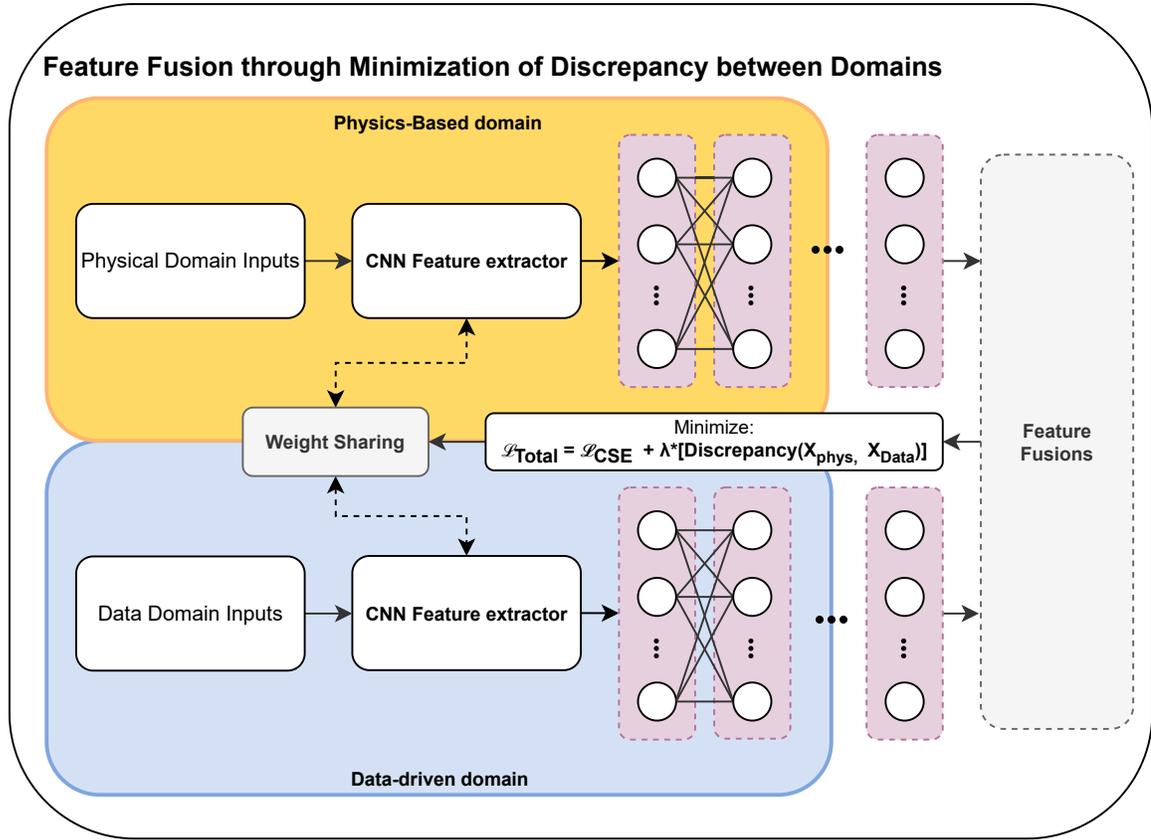


Figure 2.3: Integration of physics-based and data-driven domains through feature fusion: The CNN architecture is employed as a feature extractor. Adapted from [46] and [165]

Another implementation of physics-informed regularization is with structures involving the encoder-decoder style networks, or autoencoders. The structure of networks of this style may be described as two components working in tandem: an encoder and a decoder network. Through the encoder network, input data is compressed through multiple transformations to a low-dimensional representation. This representation is subsequently decompressed and transformed back into the original

representation through various transforms in the decoder, with the objective of accurate reconstruction of input data [37]. Intermediate layers typically consist of lower quantities of neurons, which in effect force the network to learn a compressed representation. In general, autoencoders are well-suited for condition monitoring tasks as they are able to learn the representations of the normal operating state of a system and detect anomalies or deviations from that state [174]. Implementations of the autoencoder learn to identify these changes by encoding the normal behaviour of the system into a lower-dimensional representation, and then detecting anomalies in the reconstruction error when the system deviates from this normal behaviour.

This strategy has been employed in subsequent literary works for the effective detection of deviant behaviour without the need for additional labelled data. For example: Li and colleagues designed a physics-informed convolutional autoencoder for the detection of high impedance faults in power distribution grids to overcome the issue of the lack of labelled data from conventional approaches [78]. The physics hybrid physics-informed loss term featured in the network serves to regularize the prediction of voltage, taking advantage of the physical relationship, the elliptical trajectory between measured voltage and current.

As another example, Russell and colleagues proposed a framework for signal compression and reconstruction of large quantities of data in the setting of industrial CM through a physics-informed deep convolutional autoencoder [119]. A hybrid loss function was developed comprised of the traditional MSE, Pearson’s correlation coefficient loss, and a physics-informed loss term. As the primary objective of an autoencoder is to reconstruct a given signal, dominant frequencies in the signals must be preserved

post-reconstruction [119]. This fact is leveraged by the authors to impose a physical constraint on the data-driven solution through a loss term sensitive to frequency [119]. The authors also elected to learn latent representations of operating conditions individually, effectively isolating the compressed representations, with the objective of optimal representation for individual faults.

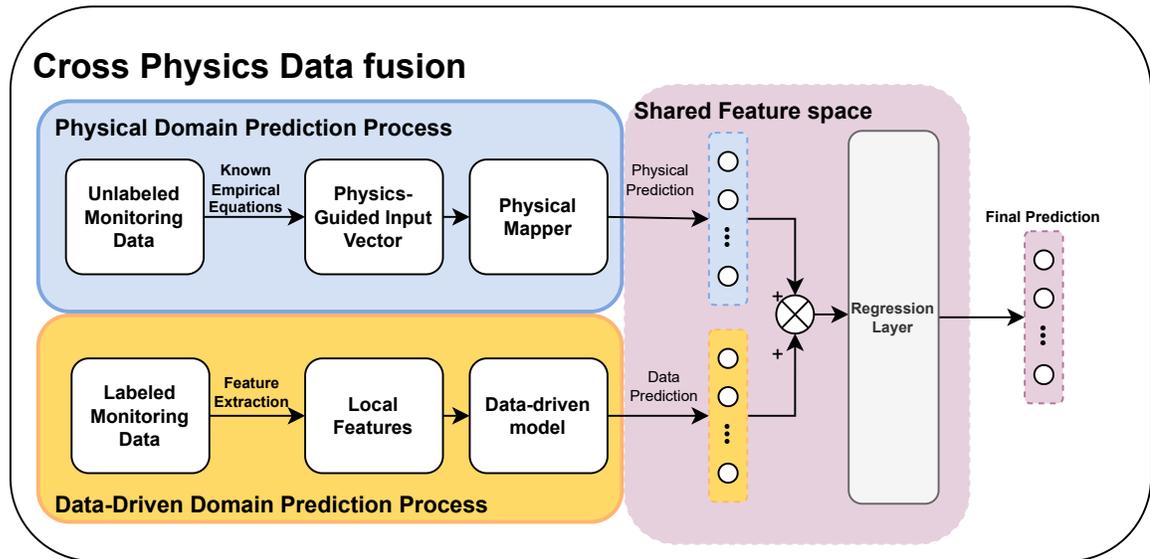


Figure 2.4: Cross Data-Physics Fusion, as presented by [150] predictions based on information from both the data domain (comprised of features derived from labelled monitoring data), and physics domain (comprised of features derived from unlabelled data) are simultaneously mapped to a shared space and concatenated. Both are processed through a regression layer for the final prediction.

Several examples in literature also take advantage of the RNNs’ ability to extract temporally invariant data, for use in applications involving time-domain monitoring. For example, Wang and colleagues fused features from the data-driven and physics domain through their applications of the *cross physics-data fusion*, with application in modelling damage accumulation in tools [150]. Features from the data domain and physics domain are extracted separately, and subsequently mapped to

a shared feature space, representing tool wear [150]. Predictions from both domains are concatenated, and evaluated in the final regression layer of the network whereby a physics-informed loss function is employed to minimize discrepancies between the data-driven Bi-directional Gated Recurrent Unit model and empirical equations [150].

Liu and colleagues proposed a physics-informed RNN for offshore structural monitoring. The methodology proposed employs an optimal singular value decomposition procedure for modal identification of the structure [81]. Through their study, the authors formulated the physics-informed modal identification process into an eigensystem and employed an RNN for the solution of the governing differential equations of the eigensystem. Through their proposed framework, the authors improved upon conventional monitoring methods to devise an efficient strategy for modal identification and monitoring in real-time, and under dynamic environmental conditions [81].

Researchers have also innovated upon the methodology by which the loss is evaluated. Traditionally, the vast majority of literature explores the minimization of deviations from a target value. Chen and colleagues instead proposed an LSTM differentiation strategy for the state of health focusing on maximizing deviations between known states [19]. In their developed strategy for the selection of LSTM hyperparameters in the detection of gearbox faults, rather than the conventional minimization of mean squared error of the labelled data, the selection strategy proposed maximizes the discrepancy, in this case, evaluated by the Mahalanobis distance, between healthy and physics-informed faulty states. Data of vibration signatures correlating to the fault state are generated based on prior knowledge of the system and used to establish the target of evaluation [19].

In all, physics-informed regularization presents a guided process by which the

algorithm can acclimate to the domain of physical feasibility, as illustrated in the numerous works discussed in this particular section. Though effective, the main limitations of this approach are primarily regarding the increased complexity of the loss landscape, and difficulties in achieving generalization, to be detailed in section 4.2.1.

2.2 Magnetorheological Dampers and Modelling

Magnetorheological (MR) fluids are a class of smart materials exhibiting changes in rheological properties when subjected to an external magnetic field and consist of micron-sized ferrous particles suspended in a liquid carrier. The configuration of which enables the rapid manipulation of viscosity and flow characteristics through controlled alignment of the said ferrous particles. The ferrous particles undergo a magnetic flux-induced alignment along field lines with the application of an induced external magnetic field, effectively transitioning from a free-flowing liquid to a semi-solid state [179, 95, 75]. The degree of viscosity alteration is proportional to the strength of the applied magnetic field, allowing for minute adjustments in the fluid's rheological behaviour.

Due to the MR fluid's properties, the MR Dampers allow for the design of engineering systems with precise damping properties through the use of appropriate control systems. The increasing popularity of MR dampers in automotive, industrial machinery, and large-scale civil engineering applications is often attributed to the MR damper's relatively lower power consumption and extensive range of damping control [116]. With the vast majority of practical applications, the design of implemented MR

Dampers are of a semi-active design [95, 75, 50]. MR dampers are capable of functioning in passive, active, or semi-active modes. However, fully active MR dampers are not commonly employed due to their dependency on a constant power supply. With a similar design to conventional dampers, semi-active MR dampers are capable of producing significantly more damping force when power is applied [50]. Semi-active MR dampers have the capability of achieving significant damping force and allow for large changes in damping force with a relatively fast response time [95, 75].

For the operation of the MR damper: the displacement of the piston as it moves through the cylinder causes the MR fluid to flow through narrow channels or orifices in the piston. Electromagnetic coils in the damper may exert an induced magnetic field depending on applied excitation, influencing fluid viscosity. A labelled diagram of the MR damper, consisting of the components of a typical single-ended linear MR damper, is provided in Figure 2.5.

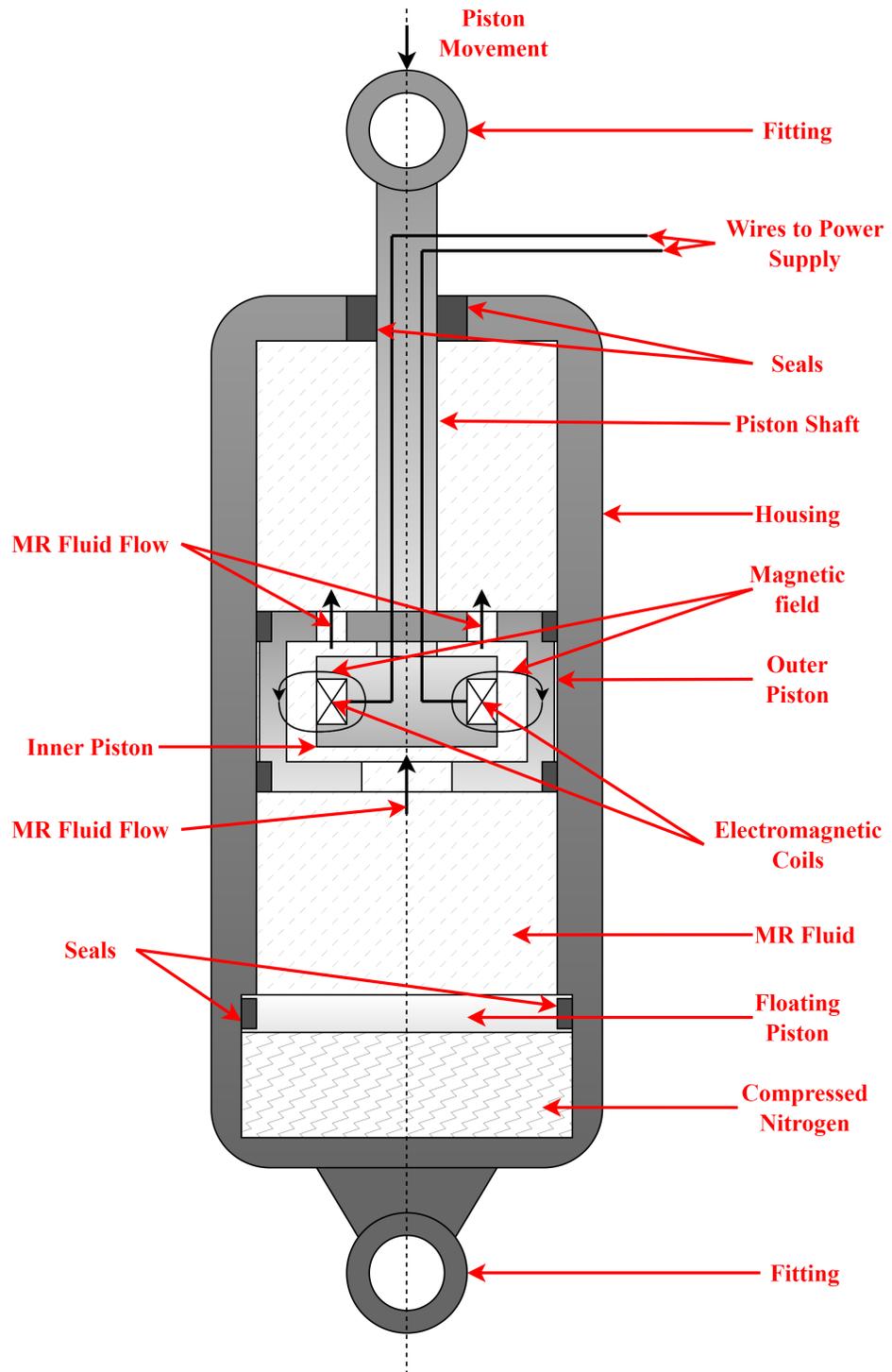


Figure 2.5: A labelled depiction of an MR damper.

The damping capability of MR dampers may be altered via the application of a magnetic field, which is generated by an electromagnetic coil within the damper and controlled through an external power source. The MR damper behaves like a conventional fluid damper with the piston moving freely in the absence of a magnetic field, with minimal damping force in its passive state [50]. With increasing magnetic field strength, ferrous particles within the MR fluid are magnetized and aligned along the magnetic field lines, with the increased viscosity from this alignment creating restrictions to the flow of the fluid through the channels, thereby increasing the resistance [50].

The process of developing mathematical models of MR dampers poses a challenge due effects of hysteresis and the complexity associated with representing said effect mathematically, with the dynamic response of MR fluids exhibiting a high degree of non-linearity [179, 116, 120]. Rather extensive research has been performed with the objective of accurately characterizing the non-linear response observed in MR dampers, aiming to understand and predict MR damper response under varying operating conditions. These include, for example, curve fitting methodologies adapting the response for use in estimation, as seen in the work of Lee [72, 75]. In the literature, MR damper modelling has been explored thoroughly, with methods varying in terms of accuracy and complexity [179, 116, 120, 148]. These dynamic models may be categorized into parametric and non-parametric models.

Parametric models require predefined parameters and initialization with an initial estimation of value for parameters [1, 148], whose values are then adjusted based on the error relative to experimental data. Prominent examples of parametric models developed for the MR damper in literature are the Bingham models, the Biviscous

model, and the Bouc-Wen models, each with their respective extensions to better account for certain phenomena or to characterize certain intricacies in the response. Non-parametric models do not have a set form and are derived from experimental data [1]. The non-parametric model structure is not predetermined but inferred from the data, with the most common forms of non-parametric models being black box models and NN models [1].

2.2.1 Bingham Model and Derivatives

Among the earliest models introduced in this domain is the one proposed by [139]. Derivation of this model, titled the Bingham model is based on the stress-strain relationship characteristic of the Bingham visco-plastic model, which was initially outlined by Shames and colleagues [126].

With respect to the Bingham visco-plastic model, the total stress experienced by the fluid for positive values of the shear rate is represented as a combination of the shear stress induced by the magnetic field and the shear stress arising from fluid movement [126, 138, 139]. The induced shear stress by the applied magnetic field is a function of the applied magnetic field strength, as well as the shear stress resulting from movement through the fluid. Wherein the latter component is dependent upon the material properties of the fluid, namely the viscosity and the shear rate. The relationship outlined above is represented as:

$$\tau = \tau_{field} \cdot \text{sgn}(\dot{\gamma}) + \eta\dot{\gamma}, \quad (2.2.1)$$

where τ denotes the total shear stress within the MR fluid. The term τ_{field} represents the yield stress resulting from the influence of the applied magnetic field. $\dot{\gamma}$ denotes

the shear rate, while parameter η denotes the viscosity of the fluid.

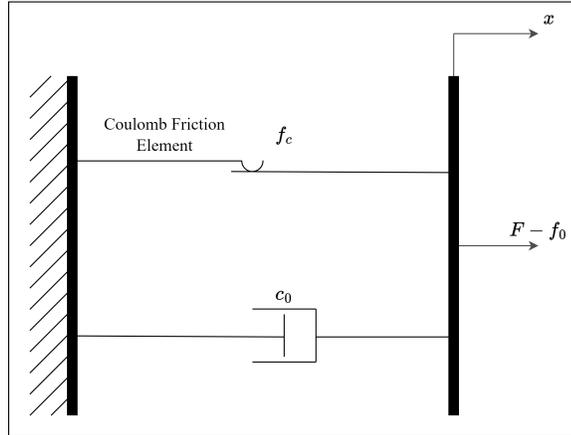


Figure 2.6: Depiction of the Bingham model of MR dampers, adapted from [139]

The Bingham model may be visualized as comprising a Coulomb friction element arranged in parallel with an ideal viscous damper, as depicted in Figure 2.6. Within this model, the Coulomb friction element accounts for the yield stress, in effect, the stress required to initiate flow in the presence of a magnetic field. While the dashpot, characterized by damping coefficient c_0 , captures the fluid’s response to ongoing shear once the yield stress has been surpassed. The effective damping force may be mathematically modelled as:

$$F = c_0 \dot{x} + f_c \operatorname{sgn}(\dot{x}) + f_0, \quad (2.2.2)$$

for damper coefficient c_0 , yield force f_c , and initial offset force from the accumulator f_0 .

From prior works in the domain, the Bingham model has shown adequate efficacy in modelling the forced response of MR dampers [148, 1]. However, the model has notable limitations. It has been demonstrated that the model does not replicate the

nonlinear force-velocity response observed in the experimental data under conditions where the acceleration and velocity have opposite signs and the velocity magnitude is small [136]. Considering that the hysteretic loop developed in the Bingham model is singular, the model encounters difficulties in accounting for the response to the magnetization and or demagnetization of ferrous particles in MR dampers [136].

To alleviate the above limitations, variants of the Bingham model have been proposed. One such extension involves combining the original Bingham model (i.e., a frictional element in parallel with a dashpot) in series with a series of ideal springs and dashpot, as initially described in the work of Gamota and Filisko [36], illustrated in figure 2.7.

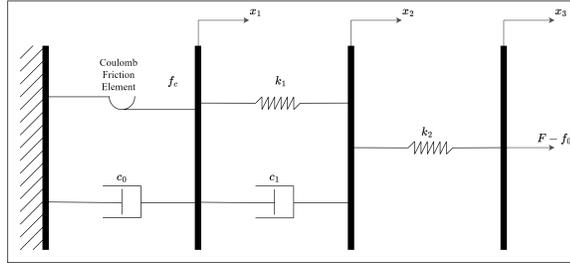


Figure 2.7: Depiction of the extended Bingham model of MR dampers, adapted from [36]

The approach aims to improve the accuracy of the force-velocity relationship by integrating additional mechanical elements that can better simulate the observed behaviours of MR dampers [36]. The resultant damping force is modelled as:

$$F = \begin{cases} k_1(x_2 - x_1) + c_1(\dot{x}_2 - \dot{x}_1) + f_0 = c_0\dot{x}_1 + f_c \operatorname{sgn}(x_1) + f_0 = k_2(x_3 - x_2) + f_0, & \text{if } |F| > f_c, \\ k_1(x_2 - x_1) + c_1(\dot{x}_2) + f_0 = k_2(x_3 - x_2) + f_0, & \text{if } |F| \leq f_c, \end{cases} \quad (2.2.3)$$

for spring constants k_1 , k_2 , as well as damping factors c_0 , c_1 , and c_2 [36].

The force-velocity relationship is more accurately represented and closer to measured values in comparison to the default Bingham model [36, 148]. However, this improved accuracy comes at the cost of increased complexity in calculations, due to the additional elements, and intermediary displacement items introduced.

2.2.2 Biviscous models

Biviscous models conceptualize the MR fluid dynamics by assuming that the fluid exhibits plastic behaviour in both the pre-yield and post-yield phases [154, 101, 54]. These models capture the hysteresis loop by delineating its shape into three discrete linear sections. The inherent structure of biviscous models is thus characterized by piecewise linear segments as:

$$F = \begin{cases} c_{po}\dot{x} + f_y, & \text{if } \dot{x} \geq \frac{f_y}{c_{pr} - c_{pr}} \\ c_{pr}\dot{x}, & \text{if } -\frac{f_y}{c_{pr} - c_{pr}} \leq \dot{x} \leq \frac{f_y}{c_{pr} - c_{pr}}, \\ c_{po}\dot{x} - f_y, & \text{if } \dot{x} \leq -\frac{f_y}{c_{pr} - c_{pr}}, \end{cases} \quad (2.2.4)$$

whereby the model is characterized by parameters c_{pr} and c_{po} , representing pre-yield and post-yield damping coefficients respectively, as well as yield force f_y .

The same authors also devised a hysteresis biviscous model, which introduces an additional parameter compared to the earlier nonlinear bi-viscous model [154]. This alteration, as presented in equation 2.2.5, allows for a more accurate representation of the velocity at which the damping force is zero, taking into account the direction of acceleration [154]. This alteration introduces the parameter of the zero force velocity

intercept, v_0 , and the resultant damping force is modelled as:

$$F = \begin{cases} c_{po}\dot{x} + f_y & \text{if } \dot{x} \leq -\frac{f_y - c_{pr}v_0}{c_{pr} - c_{po}} \text{ and } \ddot{x} > 0, \\ c_{pr}(\dot{x} - v_0) & \text{if } -\frac{f_y - c_{pr}v_0}{c_{pr} - c_{po}} \leq \dot{x} \leq \frac{f_y + c_{pr}v_0}{c_{pr} - c_{po}} \text{ and } \ddot{x} > 0, \\ c_{po}\dot{x} + f_y & \text{if } \frac{f_y + c_{pr}v_0}{c_{pr} - c_{po}} \leq \dot{x} \text{ and } \ddot{x} > 0, \\ c_{po}\dot{x} + f_y & \text{if } \frac{f_y - c_{pr}v_0}{c_{pr} - c_{po}} \leq \dot{x} \text{ and } \ddot{x} < 0, \\ c_{pr}(\dot{x} + v_0) & \text{if } -\frac{f_y + c_{pr}v_0}{c_{pr} - c_{po}} \leq \dot{x} \leq \frac{f_y - c_{pr}v_0}{c_{pr} - c_{po}} \text{ and } \ddot{x} < 0, \\ c_{po}\dot{x} + f_y & \text{if } \dot{x} \leq -\frac{f_y + c_{pr}v_0}{c_{pr} - c_{po}} \text{ and } \ddot{x} < 0. \end{cases} \quad (2.2.5)$$

However, in both cases of the model, their piecewise linear approach does not accurately depict the smooth transitions typically observed in force-velocity curves of real fluids. The simplification inherent in the biviscous model means that accuracy and fidelity to observed response will fall short in regions representing the transition between each piecewise equation; In addition, with scenarios where a detailed, continuous representation is required analysis or simulation.

2.2.3 Bouc-Wen Models and Derivatives

The Bouc-Wen model has emerged as a prominent model in literature with modelling nonlinear hysteretic systems, originally introduced by Bouc [11], and extended by [153]. Since its introduction, the Bouc-Wen model has been recognized for its capability to model the effect of hysteresis in a mathematically tractable form. The capability to represent the shape of hysteretic cycles analytically is achieved through a system of differential equations. Equations incorporate parameters for tuning the model to match the observed behaviour.

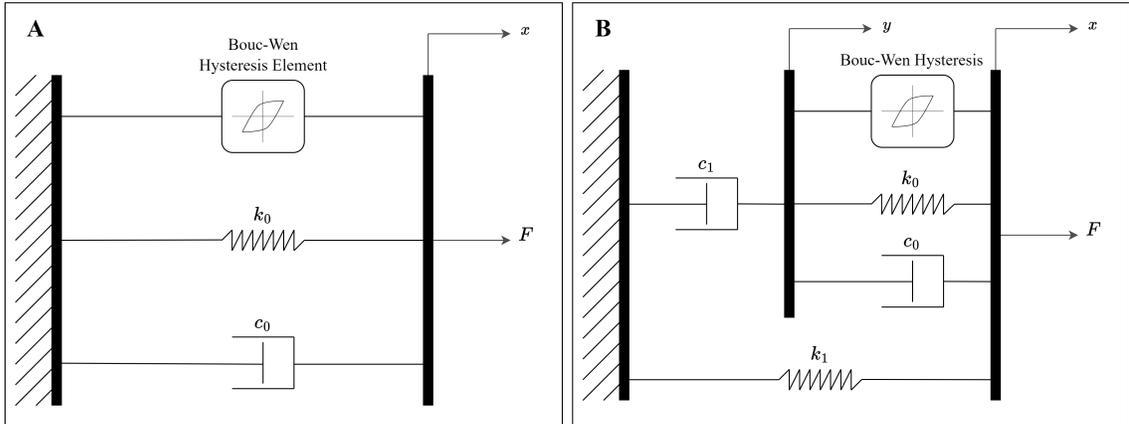


Figure 2.8: Depiction of the: (A) Bouc-Wen model of MR dampers, adapted from [136] and (B) The Modified Bouc-Wen model, adapted from the work of [136]

The simple Bouc-Wen model, depicted in Figure 2.8(A), is commonly employed for its capacity to capture hysteretic behaviours through parameter adjustments. The model expression solving for predicted damping force F for a displacement x is defined by the equations:

$$F = \alpha z + c_0 (\dot{x}) + k_0 (x - x_0), \quad (2.2.6)$$

$$\dot{z} = -\gamma |\dot{x}| z |z|^{n-1} - \beta \dot{x} |z|^n + A \dot{x}. \quad (2.2.7)$$

where $\alpha, \beta, \gamma, A, n, c_0, k_0$ are the model parameters and z is the hysteretic displacement, a variable with the units of length, characterizing the effect of hysteresis. The bias force generated by the accumulator is to be represented via the initial offset x_0 of the ideal spring with spring constant k_0 . The Bouc-Wen model has been demonstrated in various works to be capable of fitting characteristics of MR dampers [136]. However, the presence of a differential term renders parameter identification challenging in this context.

The simple Bouc-Wen model has been shown to have certain limitations in accurately capturing hysteresis at certain points within a cyclical motion of the damper, more specifically, in the specific case in which the damper is operating at low velocity and experiencing acceleration in the opposite direction to its velocity. In the work of Spencer and colleagues, the authors introduced a modified version of the Bouc-Wen model to enhance accuracy, albeit with a higher level of complexity than the original model [136]. This increased complexity in computations is primarily due to the incorporation of an intermediary differential displacement variable, denoted as y , in addition to the various new parameters to be identified before use. The model utilizes the dashpot with damping coefficient c_1 to characterize viscous damping at low velocities, and the dashpot with damping coefficient c_0 to characterize viscous damping at high velocities. Similar to the simple Bouc-Wen model, the model also incorporates the accumulator stiffness through representation by ideal spring with stiffness k_1 with initial equivalent displacement x_0 . An illustration of the aforementioned modifications may be seen in Figure 2.8(B). The modified Bouc-Wen model is defined as follows:

$$F = \alpha z + c_0 (\dot{x} - \dot{y}) + k_0 (x - y) + k_1 (x - x_0), \quad (2.2.8)$$

which may be simplified to:

$$F = c_1 \dot{y} + k_1 (x - x_0). \quad (2.2.9)$$

The evolutionary variable z , and the intermediary displacement y are represented

in the following differential equations:

$$\dot{z} = -\gamma |\dot{x} - \dot{y}| z |z|^{n-1} - \beta (\dot{x} - \dot{y}) |z|^n + A (\dot{x} - \dot{y}), \quad (2.2.10)$$

$$\dot{y} = \frac{1}{c_0 + c_1} (\alpha z + k_0 (x - y) + c_0 \dot{x}). \quad (2.2.11)$$

Alterations to the applied voltage induce changes within the magnetic field strength, resulting in changes to damping characteristics. In their research, Spencer and colleagues have extended the modified Bouc-Wen model to accommodate the impact of voltage application on MR dampers [136]. Through experiments, they observed that parameters c_0 , c_1 , and α exhibit variations with the efficient voltage term u , and is given as:

$$\alpha = \alpha_a + \alpha_b u \quad (2.2.12)$$

$$c_0 = c_{0,a} + c_{0,b} u, \quad (2.2.13)$$

$$c_1 = c_{1,a} + c_{1,b} u. \quad (2.2.14)$$

Commonly referred to in the literature as the efficient voltage, the term u denotes a differential voltage term that signifies a low-pass filtered response to the applied voltage v .

$$\dot{u} = -\eta (u - v). \quad (2.2.15)$$

Spencer and colleagues demonstrated in their work that the dynamics associated with the attainment of rheological equilibrium may be achieved using a first-order filter, wherein the filter system tracks variations in applied voltage, with rapid changes

being attenuated [136].

2.2.4 Dahl models

Dahl developed the Dahl friction model[22], which characterizes friction through a differential equation between stress and strain. A significant advantage of the modified Dahl model is its simplicity in parameterization.

The Dahl model for frictional force is given as:

$$\frac{dF}{dx} = \sigma \left(1 - \frac{F}{f_c} \text{sgn}(\dot{x}) \right)^\zeta, \quad (2.2.16)$$

where f_c represents the Coulomb friction force and σ represents the stiffness coefficient. Parameter ζ controls the shape of the resultant stress-strain curve [22]. For the special case in which $\zeta = 1$, introducing the hysteretic variable z , the Dahl model can be expressed as:

$$F = \sigma z, \quad (2.2.17)$$

$$\dot{z} = \dot{x} - \frac{\sigma}{f_c} |\dot{x}| z, \quad (2.2.18)$$

which bears great resemblance to the simple Bouc Wen model introduced in the prior section. In fact, it is to be acknowledged that the Dahl model can be viewed as a particular case of the more generalized model by Bouc [11] and Wen [153].

Building upon Dahl's foundational work, Zhou and colleagues implemented a modified Dahl model tailored for MR dampers [175]. Similar to the Bouc Wen model, the Dahl model implemented features a hysteresis element in parallel with ideal springs

and dashpots. The damping force of their implementation may be represented as:

$$F = k_0x + c_0\dot{x} + F_cz - f_0, \quad (2.2.19)$$

whereby the hysteretic variable z , dimensionless in this case, has the relation:

$$\dot{z} = \sigma\dot{x}(1 - z \operatorname{sgn}(\dot{x})), \quad (2.2.20)$$

for parameters k_0 , c_0 , representing the spring constant and damping factor respectively. F_c represents the coulomb force, which is subject to change based on the applied magnetic field [175]. f_0 represents the initial offset force from the accumulator. The parameter σ controls the shape of the hysteresis loop formed. The authors have also extended this to account for varying applied voltage, with parameters c_0 and F_c varying linearly with efficient voltage u in equation 2.2.15.

$$c_0 = c_{0,a} + c_{0,b}u, \quad (2.2.21)$$

$$F_c = F_{c,a} + F_{c,b}u. \quad (2.2.22)$$

Ikhouane and colleagues further expanded the application of the Dahl friction element by incorporating it into a viscous Dahl model tailored for shear mode MR dampers [47]. The viscous model is represented by a single dashpot in parallel with the hysteretic element, and the resultant damping force is given as:

$$F = c_0\dot{x} + \alpha z, \quad (2.2.23)$$

with constants α and c_0 being voltage-dependant parameters, governed by the efficient voltage defined in equation 2.2.15. The relation is identical to the one outlined in equations 2.2.13 and 2.2.12.

2.2.5 LuGre Friction Model

The LuGre friction model is a widely used mathematical framework for capturing the behaviour of friction in mechanical systems, and an extension of the Dahl model. Central to the LuGre model is the bristle deformation concept, which represents the microscopic interactions between contact surfaces [24, 80]. This concept visualizes the surfaces as covered in microscopic bristles that deform as the surfaces move relative to each other, whereby the deformation of these bristles accounts for both the static and dynamic friction forces observed in real-world systems. The average deflection of aforementioned bristles, referred to by the authors as internal variable z , is modelled by:

$$\dot{z} = \dot{x} - \frac{|\dot{x}|}{g(\dot{x})}z, \quad (2.2.24)$$

whereby the function $g(\dot{x})$ is positive and dependent on various material properties and state factors [24]. These factors include but are not limited to, lubrication and temperature [24]. The resultant resistant force may be modelled as a function of the resistance from the aforementioned internal variable, and the relative velocity between surfaces, \dot{x} :

$$F = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{x}, \quad (2.2.25)$$

for the model parameters σ_0 , σ_1 and σ_2 .

Jimenez and colleagues presented a modification and extended the application of

the existing LuGre friction model for MR damper modelling specifically [53, 52]. This modification is aimed at accounting for the response of MR dampers under varying operational conditions, also taking into account the applied voltage, v . The resultant damping force of this model by Jimenez is modelled as:

$$F = \sigma_0 z v + \sigma_1 \dot{z} + \sigma_2 \dot{x}. \quad (2.2.26)$$

While in the original LuGre friction model, the internal state z represents the average bristle deflection, for the MR damper model developed by Jimenez, this model has been adapted to account for the specific characteristics of MR fluids. More specifically in this context, the variable z is redefined to represent the transient deformation within the MR fluid caused by the movement of the damper, as modelled by:

$$\dot{z} = \dot{x} - \sigma_0 \eta_0(v) |\dot{x}| z, \quad (2.2.27)$$

where parameters function $\eta_0(v)$ represents a function capturing effects of the applied voltage, defined as:

$$\eta_0(v) = \alpha_0(1 + \alpha_1 v), \quad (2.2.28)$$

for constant parameters α_0 , and α_1 . A significant limitation of the modified LuGre model is the absence of a term that accounts for the Stribeck effect. The LuGre model adapted for MR dampers demonstrates satisfactory accuracy in literature, albeit with notable discrepancies observed primarily at higher velocities [148, 53, 52].

2.2.6 Other Models

Aside from the models covered earlier in this section, several other models have been developed since. In consideration of the study's specific focus, this research has selectively explored the most prominent parametric models pertaining to MR dampers. Table 2.1 summarizes some of the range of models available in the literature, in addition to previously detailed methods.

Model Name	Citation
Bingham Models:	
Simple Bingham model	[138, 139]
Gamota and Filisko: Extended Bingham model	[36]
Hysteretic Bingham model	[135]
Occhiuzzi: Extended Bingham model	[96]
Biviscous models:	
Non-linear Biviscous model	[154, 137]
Hysteresis Biviscous model	[154]
Nonlinear hysteretic arctangent model	[3]
Bouc-Wen models:	
Simple Bouc-Wen Model	[11]
Modified Bouc-Wen	[136, 161]
Bouc-Wen model for shear mode MR dampers	[50]
Current dependent Bouc-Wen Model	[44]
Current-frequency-amplitude dependent Bouc Wen-model	[25]
Non-symmetrical Bouc-Wen model	[68]
Dahl Models:	
Modified Dahl model	[175]
Viscous Dahl model	[47]
LuGre Friction Models:	
LuGre Friction model	[53, 52]
Modified LuGre model	[121, 146]
Viscoelstic plastic:	
Viscoelastic-plastic model	[101]
Stiffness–viscosity-elasto-slide	[155, 44]
Sigmoid model:	
Sigmoid model	[149]
Hyperboic Tangent model:	
Hyperbolic Tangent model	[67]
Equivalent model:	
Equivalent model	[97]

Table 2.1: Summary of popular methods within the literature for parametrically modelling MR dampers.

For the purposes of this study, subsequent sections will focus on the modified Bouc-Wen model. This choice is motivated by the model's enhanced capability to describe the pre-yield and post-yield stiffness, as well as its ability to provide a smooth transition between these states.

Chapter 3

Experimental Setup

3.1 Physical Setup

The experimental setup was constructed to fulfil the academic objective of validating the performance of various estimation and machine learning algorithms through empirical data. The objective is to gather data on the dynamic response of the Lord RD 8041-1 damper, a widely employed Magnetorheological (MR) damper with applications across both academic research and industrial domains [95]. MR dampers, such as the Lord RD 8041-1, are utilized in various engineering applications due to their ability to rapidly adjust damping characteristics in response to dynamic loading conditions, in accordance with an applied magnetic field. A visualization of the physical setup may be seen in Figure 3.1, and its corresponding view in the CAD software Solidworks in 3.2. The physical setup outlined in this section was constructed at McMaster University with reference [95].

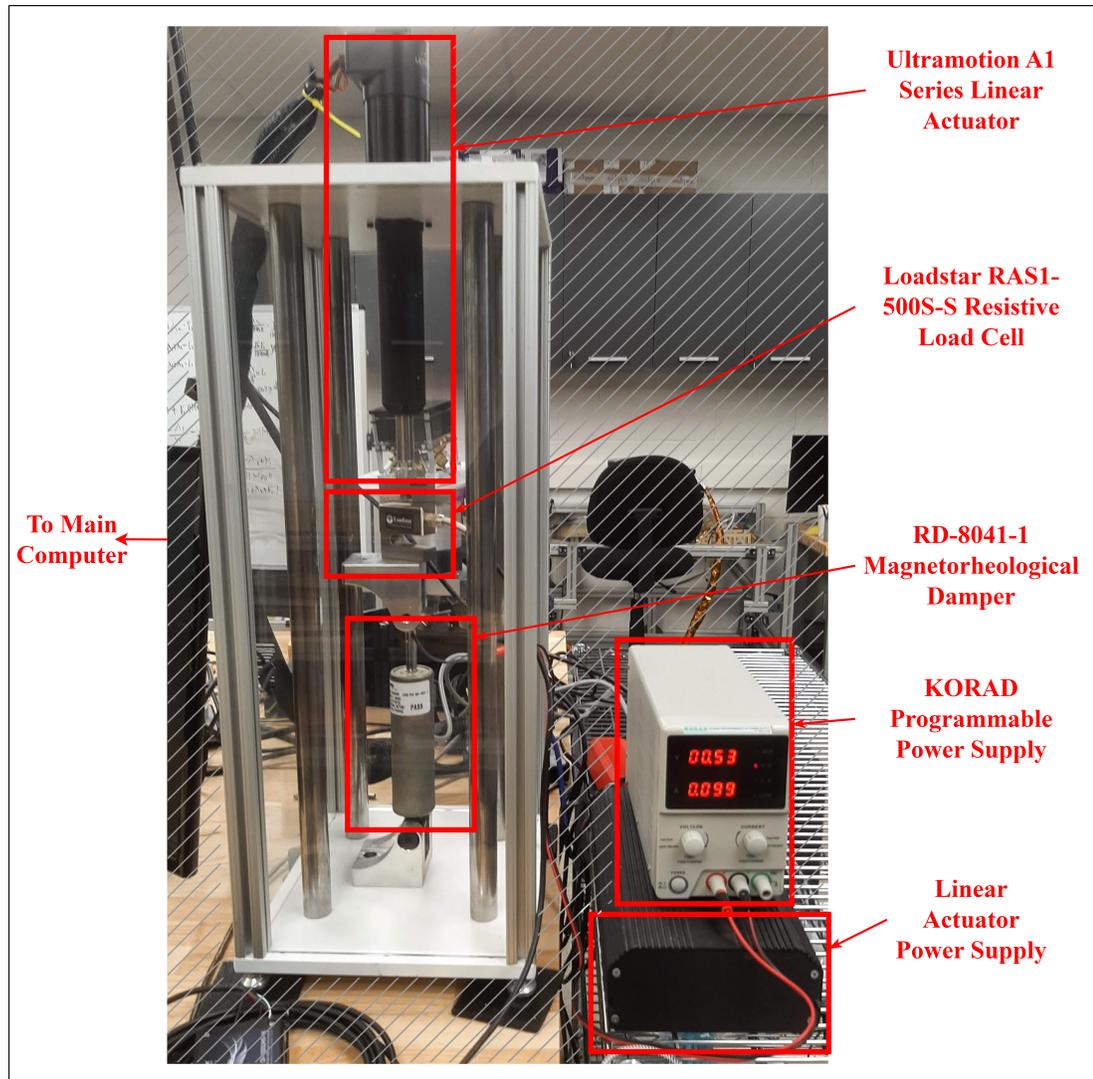


Figure 3.1: The experimental setup constructed, consisting of an MR damper, linear actuator, force sensor, and programmable power supply. Setup is controlled via a unified program run on the main computer.

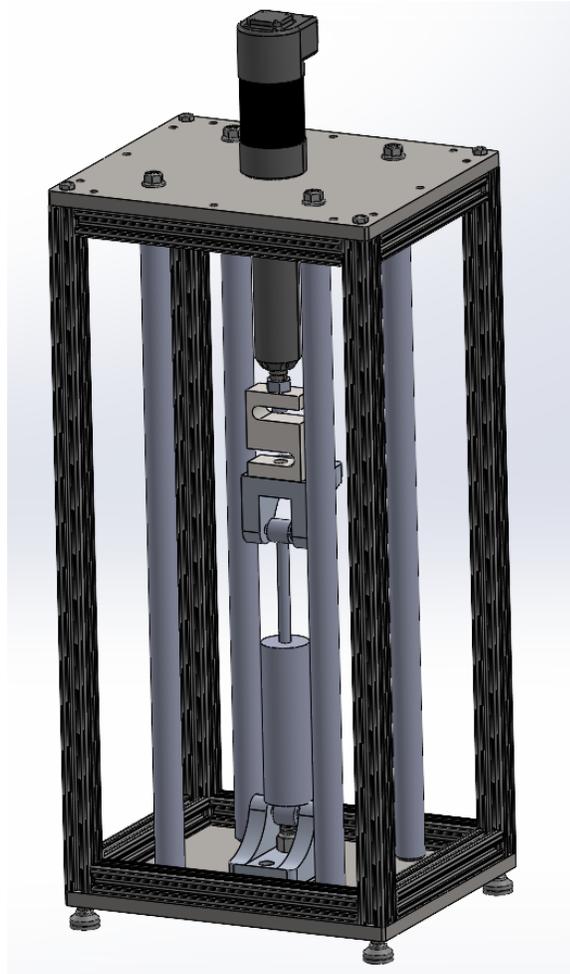


Figure 3.2: Solidworks model of the entirety of the experimental setup, adapted from [95].

MR dampers consist of several sub-components and a detailed visualization may be seen in Figure 2.5. The operation of an MR damper involves the conversion of mechanical energy into friction loss by harnessing the rheological properties of the MR fluid present within the damper [164, 6, 73, 70].

The Lord RD 8041-1 MR damper is characterized as a monotube shock, and is pressurized with high-pressure nitrogen gas to 300 psi [20]. The presence of nitrogen

gas within the accumulator ensures that the piston is fully extended under no-load conditions. An accumulator is required to account for the change in volume due to the displacement of the piston [75, 73].

In operation, the MR fluid flows between chambers through orifices in the piston as an external force is applied to displace the piston. An electrical current may be supplied to the damper, which induces a magnetic field that, in turn, influences the alignment of the ferrous particles suspended within the MR fluid. The presence of the induced field has the effect of altering the viscosity of the fluid and by extension, modulating the damping characteristics [164]. The response time of the Lord RD 8041-1 to changes in the magnetic field is estimated by the supplier to be approximately 15 milliseconds [20].

It is noted that the performance of the MR damper is influenced by ambient temperatures. Operating temperature primarily affects the coil resistance, and by extension, the strength of the magnetic field imposed. For the MR damper employed in this study, the coil resistance is typically measured at 5Ω at room temperature (22.0 degrees Celsius), while at elevated temperatures (71 degrees Celsius), the resistance increases to approximately 7Ω [20]. Variation in resistance directly impacts the efficacy by which the electromagnetic coil generates the magnetic field for particle alignment. The system is operated within standard room temperature conditions, with an operational temperature of 22.0 degrees Celsius [92].

In terms of operational electrical properties: the Lord RD 8041-1 operates at a continuous applied current of 1 A or intermittently at 2 A, at a maximum voltage of 12 VDC[20]. The operating parameters, as well as technical data of the damper utilized, are summarized in Table B.1. Additional information regarding the electrical

properties of the MR damper may also be found in Appendix B in Table B.2.

The actuation system of the experimental setup utilizes a linear actuator linear servo by Ultramotion and incorporates a rod-style actuator paired with a configurable brushless DC motor controller [92]. The actuator employs multi-turn absolute position feedback, with position feedback resolution of 3.1 micrometres. In addition, the linear actuator features a self-locking acme screw mechanism, which is implemented to prevent back drive. [92].

Regarding loading capacity, the actuator demonstrates a dynamic continuous loading capability of 756 Newtons and a dynamic peak loading capacity of 1512 Newtons. Operating with a power rating of 180 watts, it achieves a maximum speed of 356 millimetres per second and offers a total stroke length of 76.2 millimetres, [92].

Control of the linear actuator is performed via the RS-422 serial communication protocol for command transmission. The actuator itself integrates several onboard sensors for monitoring various operational variables, including but not limited to: position, torque, temperature, and humidity. Positional tracking is achieved through a phase index absolute position sensor employing a multi-turn magnetic encoder, delivering an output resolution of 1024 counts per revolution for measurement of the actuator's absolute position [92].

The force sensor utilized in this research is the RAS1-500S-S resistive S-Beam load cell, which has an operating force capacity of 226.80 kgf or 2224.11 N, and is constructed from tool steel [124]. It maintains an accuracy of $\pm 0.02\%$. Calibration measurement equipment is traceable to the National Institute of Standards and Technology (NIST) through Pacific Calibration Services.

Data collection for overall force within the system from the load cell sensor is

achieved via serial communication using the DI-10000UHS-1K USB interface, enabling data streaming at a rate of 1000 Hz [124]. The assembly of the linear actuator, force sensor and MR damper is illustrated in 3.3.

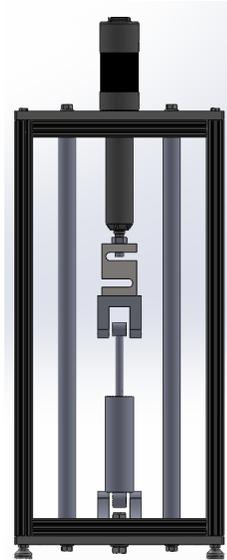


Figure 3.3: Solidworks model of the experimental setup, depicting the Linear actuator, force sensor, brackets and mounting, and the MR damper.

Within the experimental setup, the KORAD programmable power supply serves as the means of delivering a regulated current to the MR damper [63]. Adjustments to the voltage supplied to the damper itself open up exploration for a range of operating conditions and the investigation of varied damping effects. This particular power supply was chosen for its digital control features and programmable capabilities, including its encoder-controlled interfaces and USB control functionality.

A summary of parameters and properties that may be measured from the experimental setup is provided in Table 3.1.

The provided sample data collected for constant oscillation along the entire stroke

Properties	Device
Stroke length	Linear actuator
Velocity (Encoder Counts per 10 ms)	Linear actuator
Acceleration	Linear actuator
Motor torque feedback	Linear actuator
Time Stamp	Linear actuator
Force	Force sensor
Time Stamp	Force sensor
Applied Voltage	Programmable power supply
Applied Current	Programmable power supply
Time Stamp	Programmable power supply

Table 3.1: Values measured from experimental setup.

length of the damper, illustrated in Figures 3.4, 3.5, and 3.6 shows the measured force-time, force-velocity, and force-displacement curves respectively. Of Note, the hysteretic loop of the MR damper response may be seen.

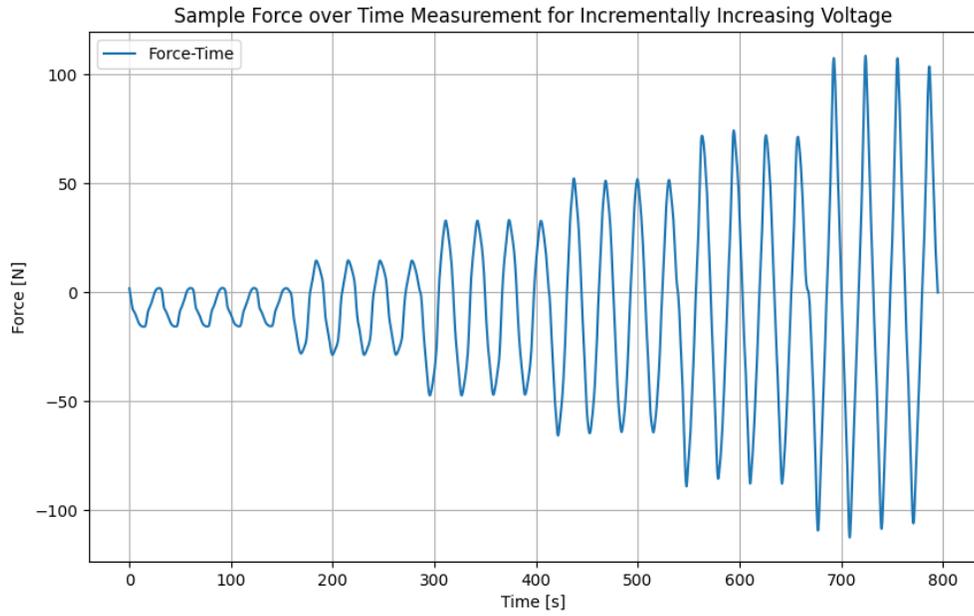


Figure 3.4: Sample data collected for measured force over time for incrementally increasing applied voltages.

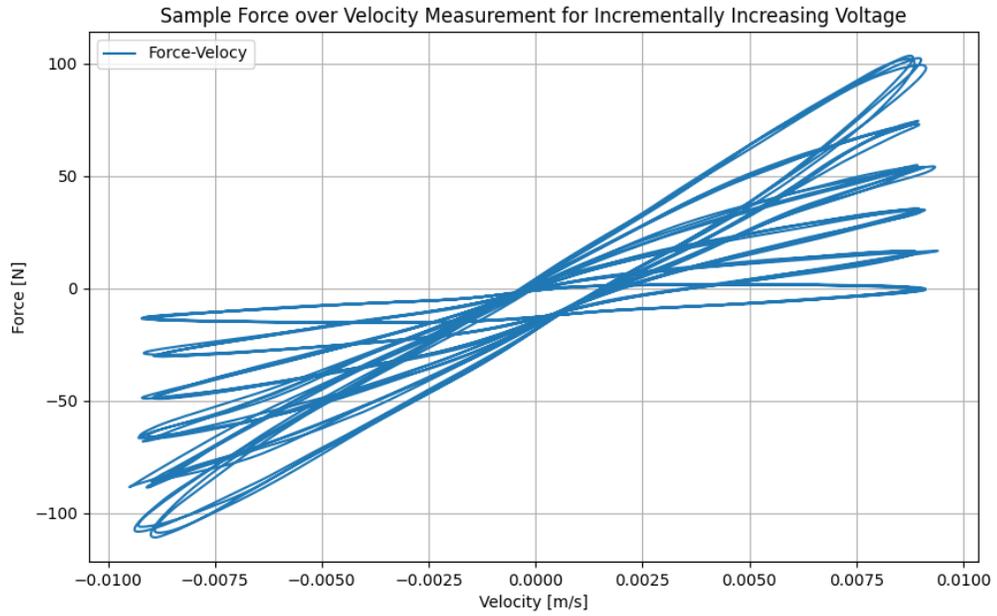


Figure 3.5: Sample data collected for measured force over measured damper velocity for incrementally increasing applied voltages.

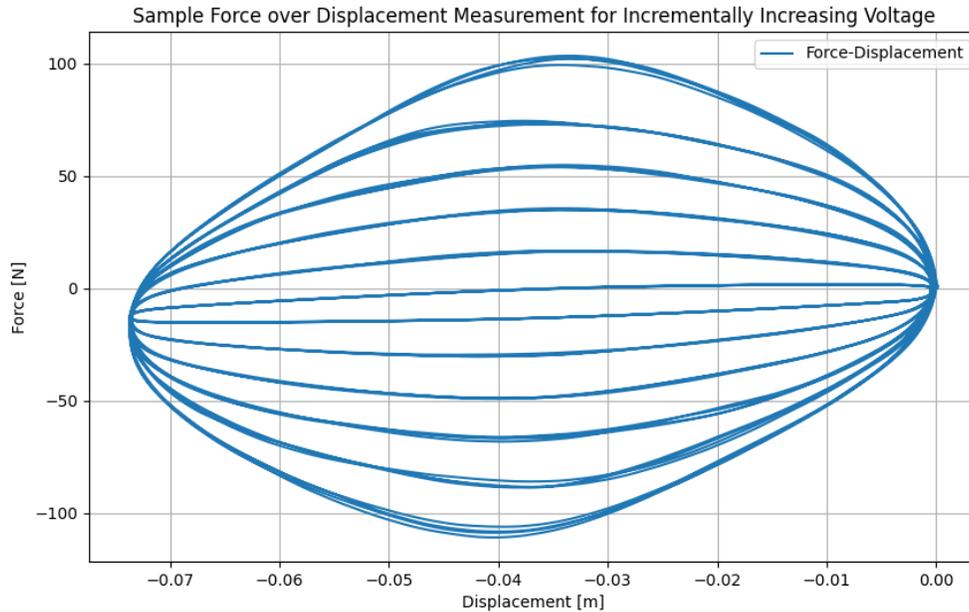


Figure 3.6: Sample data collected for measured force over measured external damper velocity for incrementally increasing applied voltages.

3.2 Data Collection

Within the framework of the experimental setup, an important aspect involves devising a method for efficiently collecting data from diverse devices in a concurrent manner. A block diagram of the experimental setup illustrating directional communication between components may be seen in Figure 3.7.

This subsection details the rationale, as well as the details in employing multi-threading and its integration into the data acquisition process, with the objective of mitigating bottlenecks associated with parallel data collection from the multitude of

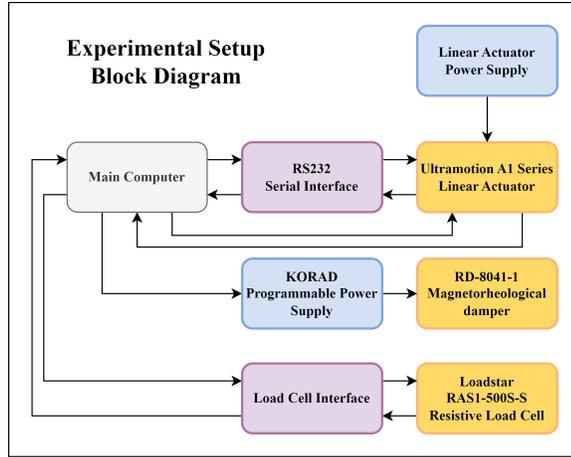


Figure 3.7: Block Diagram illustrating communication between components in the experimental setup.

devices within the experimental setup and managing heterogeneous device environments.

Multithreading represents a technique that allows for multiple sub-processes, also called threads, to exist within a single overall process, which enables the parallel execution of tasks [94]. Each thread operates independently, although threads share the overall process’s resources. For example, resources such as memory and file descriptors [94]. Multithreading is commonly employed in applications requiring simultaneous operations performed in parallel. Due to the existence of multiple processes accessing the same resources at the same time, proper management is a requirement to ensure that shared resources are being accessed safely, avoiding common issues present in fast parallel processes such as race conditions and deadlocks [94, 17]. For this implementation, in the graphical user interface (GUI) and data collection program established, multiple threads are instantiated to handle user input, and background processing, keeping the interface responsive, necessitating the usage of multi-thread operations.

The data collection program code serves the following functions: A data collection system is established, aimed at communication between distinct devices within the experimental setup and the main computer independently. More specifically, devices that the program communicates with are the linear actuator and power supply at a baud rate of 115200 baud and the force sensor at a baud rate of 230400 baud, representing the default rate at which aforementioned devices communicate over serial. Data from devices at a lower baud is interpolated to match that of devices at higher baud rates at each sampling time.

The initialization of the program involves instantiating the configuration of the data collected within a database to organize collected data efficiently. In the presented data collection system, structured query language (SQL) is utilized for the storage and organization of collected data within the database, established using the relational database management system, *MySQL*. Upon initialization, the program establishes connections to the locally hosted *MySQL* server using *Python* connectors, providing a repository for storing collected data. The option for users to create a new database or append to existing databases is provided. Within each created database, data is organized into tables, initialized upon selection of said database, for the organization of data from each device.

Each table within the database is structured to accommodate specific data attributes and types. The table pertaining to information gathered from the force sensor contains the measured force values and the respective timestamps of data collection from the device. In a similar fashion, relevant parameters to the linear actuator's operation such as absolute position, position command setpoint, velocity and acceleration are stored in the table associated with the linear actuator device.

Applied voltage to the MR damper may be obtained from the KORAD programmable power supply. The relevant measurements that may be recorded from this setup are listed in table 3.2.

Measurement	Device	Units
Position	Linear actuator	m
Velocity	Linear actuator	$\frac{m}{s}$
Acceleration	Linear actuator	$\frac{m}{s^2}$
Time Stamp	Linear actuator	s
Force	Force sensor	N
Time Stamp	Force sensor	s
Applied Voltage	Programmable power supply	V
Applied Current	Programmable power supply	A
Time Stamp	Programmable power supply	s

Table 3.2: Values measured from the experimental setup.

To interact with the database and perform data storage operations, the program utilizes SQL queries executed through *MySQL* connectors. These queries create the organizational structure within each database, insert and or delete data as required, and manage database transactions based on the ongoing data collection activities. Data collected from each device is inserted and modified within their respective tables using SQL queries. By default, the process is designed to incorporate mechanisms for batch data insertion to optimize efficiency, more specifically, the throughput and scalability in processing. Batch queries reduce the overhead associated with individual data insertions.

The program utilized a predefined class, or a user-defined constructor for objects in object-oriented programming languages such as *Python*, for each device, encapsulating functionalities specific to their operation. The object instantiated by the *linear_actuator* class orchestrates interactions with the linear actuator device. Within its initialization method, it configures the GUI elements necessary for user interaction. The object instantiated manages serial communication with the actuator device, utilizing threading to facilitate concurrent data retrieval and user interface responsiveness. In terms of functionality, it implements the program capability for reading actuator operating variables and parameters in real-time, toggling and configuring actuation motion profiles, and recording data to the database. A visual of elements from the GUI is depicted in Figure 3.8.

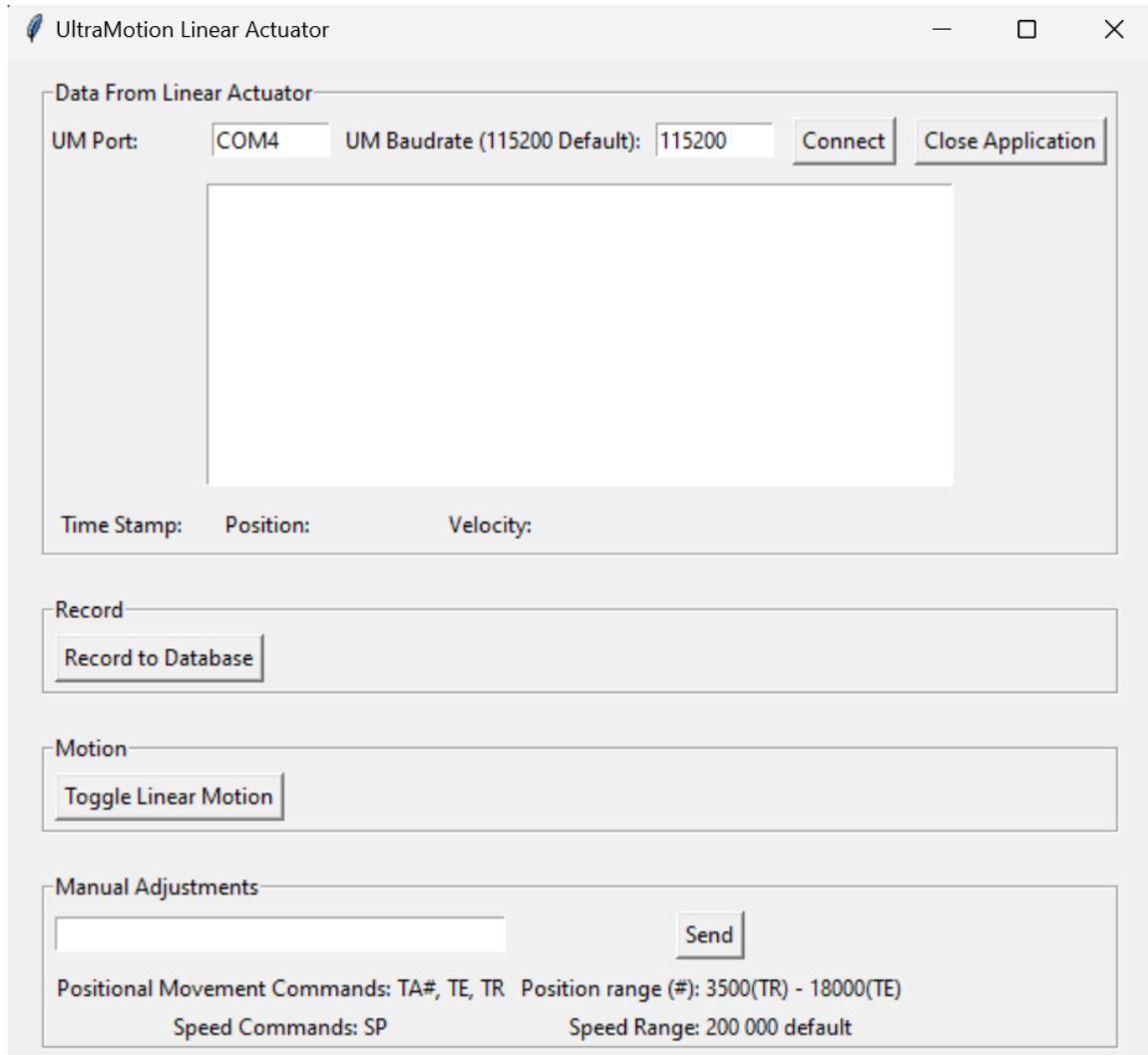


Figure 3.8: GUI display for controlling and readings from the Linear actuator, featuring elements for connection, data-streaming, motion control and recording data from all synced/connected devices.

Similarly, the *force_sensor* and *power_supply* classes encapsulate functionalities relevant to the force sensor device readings and applied voltage readings. As detailed for the linear actuator, the objects instantiated from these constructors establish serial communication, configure GUI elements, and manage the continuous retrieval

and storage of sampled data. Again, threading is required in this implementation as well, to ensure non-blocking execution and real-time data acquisition while simultaneously maintaining GUI responsiveness. The constructors also include institutions for updating the user interface with real-time data for observations. The GUI element of the force sensor is depicted in Figure 3.9.

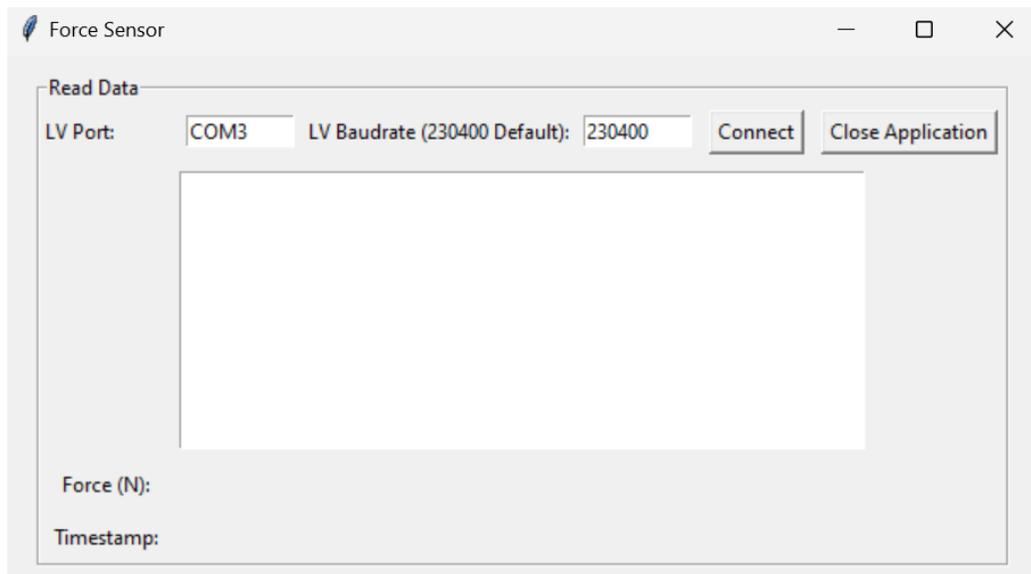


Figure 3.9: GUI display for readings from the Force Sensor, featuring elements for connection, data-streaming.

As alluded to above, within the implemented data collection system, threading orchestrates concurrent tasks and ensures the smooth operation of the data collection process without blocking the user interface interactivenss. Operations such as communication with the connected devices, enabling parallel execution of data retrieval processes, and user interface displayed all rely upon proper management of shared program resources through multithreading. In terms of the organizational structure

of these sub-processes, in general, separate threads are instantiated for individual processes with each device, which enables independent and asynchronous interactions. A visualization of the initialization process of the controller and worker threads may be seen in Figure 3.10.

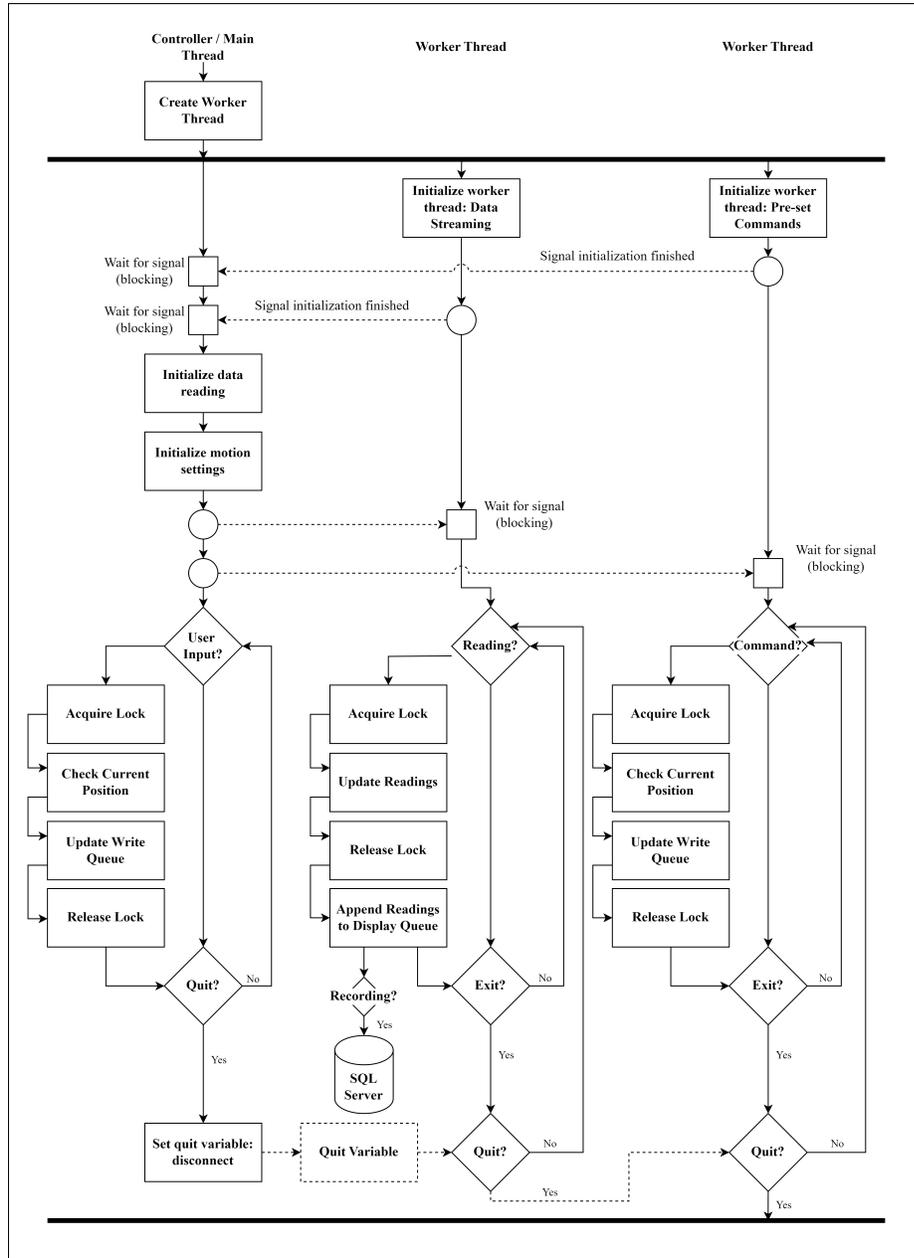


Figure 3.10: Flowchart illustrating the process of initialization and operation of the controller and created worker threads.

For synchronization within the program, locks are utilized to ensure thread safety

and minimize undesirable conditions such as race conditions and deadlocks during concurrent access to shared resources [170, 29, 87]. Due to the multi-device nature of the setup, it necessitates coordination to avoid data corruption or inconsistencies due to concurrent access. Instantiated lock objects serve as synchronization primitives, which allow established threads to coordinate access to resources such as internal data buffers for batch reads and storage, as well as scheduling sent commands. Locks are utilized within the program to manage access to shared resources accessed concurrently by multiple threads.

The primary purpose of this implementation is to maintain parallel access to serial port while maintaining responsiveness to new inputs. Lock objects are further employed to manage access to internal data structures: for example, internal buffers for storing incoming data from the devices. Internal buffers, ie. the data reading method detailed within the *linear_actuator* constructor are preserved by the serial lock object, acquired prior to accessing and modifying the buffer to prevent data corruption from concurrent accesses. This process may be visualized in Figure 3.11.

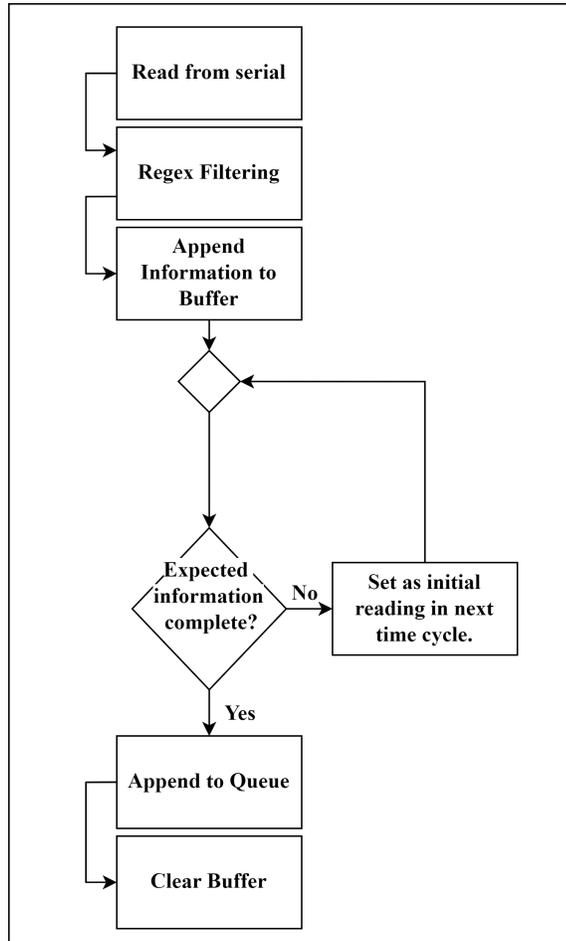


Figure 3.11: Reading process. Items read from the serial ports are filtered through regular expressions for key information, where it is then appended to a buffer. As the serial port reading process is FIFO, in the scenario where the read information is deemed incomplete, the partial variables are saved and appended to the start of the next iteration.

More specifically on the operation of sub-processes within the program, for the object instantiated by the *linear_actuator* class, upon establishing the serial connection with the actuator device, a dedicated thread is initiated to continuously retrieve data from the serial port, in which the linear actuator device continuously streams its readings. This sub-process operates independently in the background and periodically

checks for and stores incoming data in batches, updating internal buffers accordingly. Stored data is discarded on a FIFO basis, under the assumption that the option for data storage is not toggled within the main thread. The main thread remains responsive during this process which primarily serves to maintain the responsiveness of GUI elements. Sub-processes are also created to facilitate pre-set excitation commands for the actuator.

In a similar process with objects instantiated by the *force_sensor* and *power_supply* constructor, continuous data reading is facilitated while maintaining the responsiveness of interface elements. Upon connecting to the device, a separate thread is instantiated to handle retrieval operations, updating the internal buffer from sensor reading and queuing data for processing.

Overall, threading ensures efficient utilization of system resources through parallel execution. Decoupling data retrieval operations from the main thread, the system can allocate processing resources more effectively, and retain the capability to handle user inputs separately. Overall, the program instantiates instances of instantiated objects from defined constructors within the *Tkinter* application, which governs the interaction with its respective devices, as well as facilitates data collection and database recording.

Moving forward, the system data collection pipeline lays a foundation for subsequent experimentation and analysis in the following chapters, where the acquired data will be subjected to be employed to train and validate a model developed based on the PINN methodology.

Chapter 4

Surrogate Modelling and System Identification

This section contains excerpts from a review article by the primary author, which has also been published on arXiv pre-prints, available as reference: [158]. As of writing, the review article has completed the review process with the Expert Systems with Applications (ESWA) Journal by Elsevier, available as reference: [157]

In recent years, PINNs have emerged as a paradigm for the solution of differential equations. This is, in part, due to the universal approximation capabilities of NNs, which have been shown in the literature to be effective for approximating complex functions. The universal approximation theorem, as proven in the work of Hornik and colleagues [43], states that an NN is capable of approximating any continuous function to an arbitrary precision, for a sufficiently large quantity of neurons in its hidden layers [83].

Their approximative capabilities have naturally been extended for identifying the solutions to differential equations, as illustrated in the work of Lagrais and colleagues,

who initially conceptualized the idea [69]. As in many real-world scenarios, the physical parameters of a system are often challenging to directly measure or quantify [171]. Therefore, the specific implementation considered in this study employs the use of PINNs as a solution to the parameter estimation task via the integration of governing ODEs delineated by the modified Bouc-Wen model for the MR damper.

This section, building upon the initial work in the area conducted by Raissi [109], as well as many other authors in the domain, explores the application of PINNs within the context of identifying solutions to differential equations [18, 49, 110, 109]. More specifically, the methodology involves utilizing PINNs to discern optimal physical parameters that effectively describe and characterize the system. The utilization of NNs is advantageous due to their inherent ability to be configured as inverse models [76], which may efficiently learn and approximate the values of parameters of physical importance through optimization. This approach aligns with the inherent capability of NNs to adapt and generalize patterns from data, which is conventionally performed through the minimization of discrepancies between their predictions and the training dataset, and extends this capability to handle additional parameters with inherent physical meaning. Overall, it is the objective of this research to contribute to the growing domain of applied machine learning in physical systems by employing the developed network architecture to estimate and fine-tune the various parameters of parametric models, demonstrating proof of concept, as well as the validity of this approach for system identification.

4.1 Physics-Informed Neural Networks Applied to Non-linear Dynamic Systems

It should be noted that various authors have already recognized the potential of PINNs for parameter estimation and surrogate modelling and have applied it to diverse engineering applications [100, 143, 145, 163, 171, 89]. This application of PINNs for identifying the solutions to inverse problems has gained traction due to the inherent challenges associated with the complexity of models, for example, such as the modified Bouc-Wen model for MR dampers.

More generally, the challenge of parameter estimation is commonly formulated as an inverse problem, wherein the objective is to infer the parameters of a given model based on observed data [144, 84]. For a system defined in domain Ω , with unknown physical parameters θ_p the PINN establishes a surrogate solution to the ODE solution, $u(x)$, as $\tilde{u}(\mathbf{x}; \theta_n) \approx u(\mathbf{x})$, for an input $\mathbf{x} \in T$. The NN parameters θ_n represent the set of weights and biases employed to calculate \tilde{u} .

This surrogate solution is optimized through a loss function that restricts the NN solutions to satisfy the physics imposed by the ODE and is applied by enforcing the solution u on points sampled from the domain. The training data set $T \subset \Omega$ is typically comprised of separate sets, which are fed into the NN independently [110, 84]. Sets typically include the points within the domain $T_f \subset \Omega$, and additionally sampled observed points $T_i \subset \Omega$. The intuition is that the optimized solution should conform to both the ODE and the data observed. Restrictions to the network are typically

imposed, as mentioned in Section 2 as the weighted sum of the MSE loss or L^2 norm:

$$\mathcal{L}(T; \theta_p, \theta_n) = \lambda_{phys} \mathcal{L}_{phys}(T_f, \theta_p, \theta_n) + \lambda_{observed} \mathcal{L}_{observed}(T_i, \theta_p, \theta_n), \quad (4.1.1)$$

for weights λ_{phys} and $\lambda_{observed}$, whereby the optimal parameters, θ_p^*, θ_n^* desired may be extracted by minimizing the loss function.

$$\theta_p^*, \theta_n^* = \underset{\theta_p, \theta_n}{\operatorname{argmin}} (\mathcal{L}(T; \theta_p, \theta_n)). \quad (4.1.2)$$

Recurrent Neural Networks

A popular deep-learning architecture popular within the community is the RNN. RNNs have been prevalent since their inception due to their capabilities in processing sequential data: taking into account the context of the previous inputs in a sequence [122, 93]. For time t , information from the previous time state, $h^{(t-1)}$, is passed along with the conventional input data $x^{(t)}$, as the inputs to a new time state, thereby enabling the network to have access to, and incorporate information from previous inputs into its current processing at time t . RNNs are inherently designed to encode temporal invariance and have been well-established in literature for tasks involving temporal dynamics and relationships [93].

Conventional RNNs map some input $x^{(t)}$ at time t to an output $y^{(t)}$ through possessing information from both the input space $x^{(t)}$ and prior time state $h^{(t-1)}$, also known as the hidden state. An illustration of the RNN architecture may be seen in figure 4.1 (A).

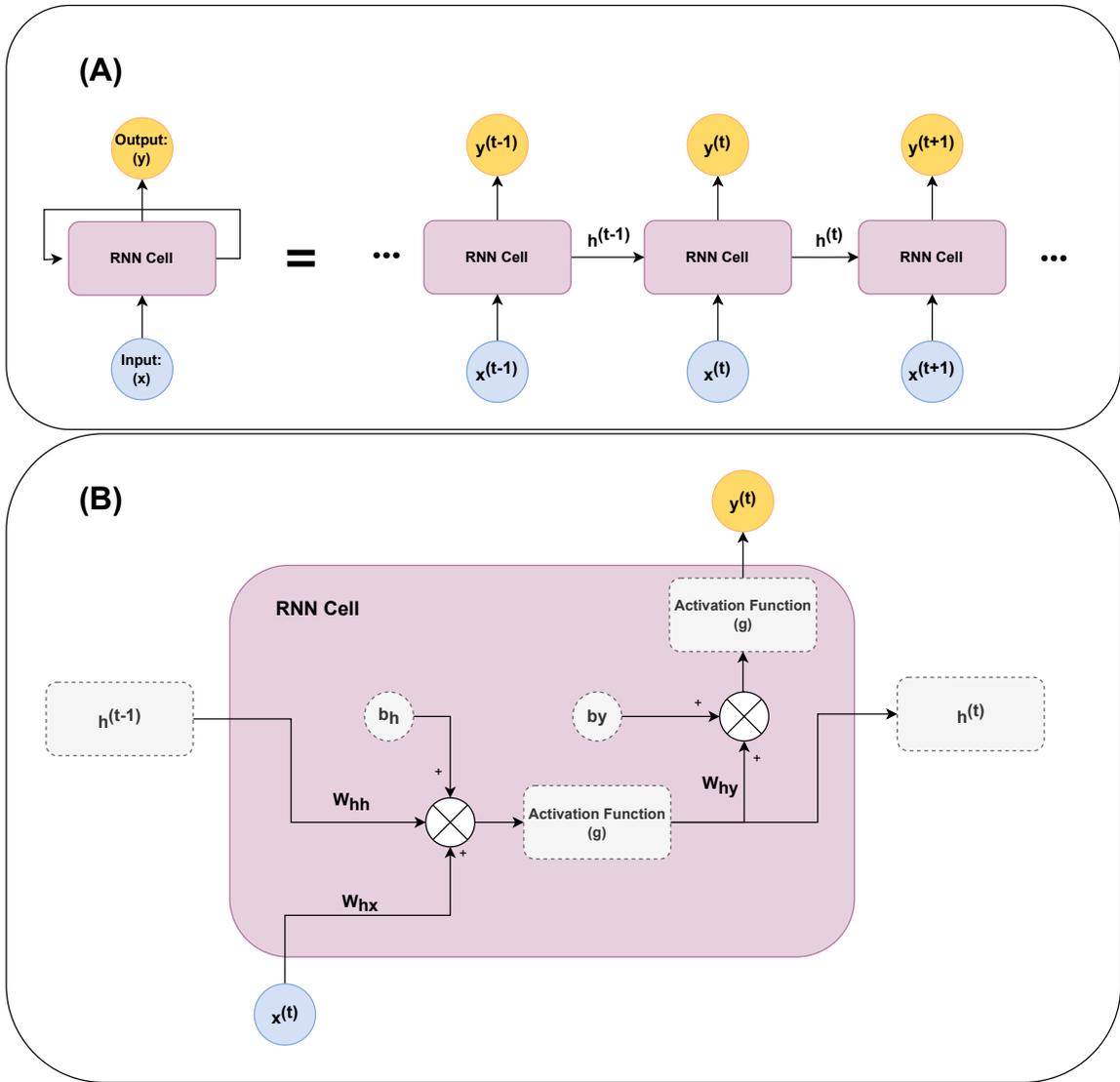


Figure 4.1: An illustration of (A) the general Recurrent Neural Network architecture, and (B) the inner computational processes within each RNN cell.

The mathematical representation of an RNN and its respective computational procedures is presented as follows. At time t , for a given input $x^{(t)}$ and prior hidden

state $h^{(t)}$, the hidden state of a cell is presented as:

$$z^{(t)} = W_{hh}h^{(t-1)} + W_{hx}x^{(t)} + b_h, \quad (4.1.3)$$

where W_{hh} and W_{hx} represent the weight matrix associated with the prior temporal state, and current input state respectively, and b_h represents the associated bias for the current hidden state. A non-linear activation function $g(\cdot)$ is applied element-wise to produce the hidden state of the cell:

$$h^{(t)} = g(z^{(t)}). \quad (4.1.4)$$

Following this, the output at time t , $y^{(t)}$, may be represented as:

$$y^{(t)} = g(W_{hy}h^{(t)} + b_y), \quad (4.1.5)$$

whereby W_{hy} and b_y represent the associated weights and biases respectively. In essence, the activation function $g(\cdot)$ is applied to a linear transformation of the input and prior cell state in order to produce the cell state output [93]. The computational process outlined in equations 4.1.3, 4.1.4, and 4.1.5 is visually represented in Figure 4.1 (B).

As a result of connections from prior time states, the NNs have the capability to maintain hidden cell states with information from the previous time, thereby granting the ability to capture temporal dependencies. In addition, RNNs were selected, as the network and their variants, such as the LSTMs and GRUs, have the flexibility in processing and outputting sequences of varying lengths, which extends their usefulness

in processes involving data with dynamic lengths, a common property in real-world monitoring applications.

4.1.1 System Identification and Surrogate Modelling

The identification of variables and parameters in nonlinear structural systems is facilitated by the incorporation of physics-based terms as regularizers, which may be specifically designed for different aspects of the system. Regularizers provide a favourable condition for facilitating generalization, as well as a bias for adherence to known physics in nonlinear systems with external excitations. An issue that conventional modelling techniques often encounter, in certain nonlinear structural systems is the presence of latent variables, that is, variables that are not directly observable but influence the system’s dynamic response.

Conventional identification methods often encounter issues in the presence of latent variables, resulting in limitations in their predictive capability. For example, a rather prominent method in literature, the Sparse Identification of Nonlinear Dynamical systems (SINDy) has been demonstrated to have great success in system identification for nonlinear systems [12]. However, SINDy requires prior knowledge of the latent variables to act as the ground truth on which to base the identified model, and without this knowledge, the method’s effectiveness is significantly compromised. Thus, this issue emphasizes the requirement for approaches that can infer and incorporate latent variables without requiring extensive prior information.

Utilizing PINNs, the approach outlined in this section addresses this issue with latent variables, enabling the identification and simulation of said variables, as well as the use of said latent variables to construct a training curriculum. This curriculum

guides the training process of the NN, allowing the network to learn and refine its predictions progressively.

System Model, Setup and Data The physics-based knowledge to be incorporated within the PINN developed is the non-linear, time-invariant dynamic model presented as follows. The Bouc-Wen model of hysteresis utilized herein is a prominent example of a model that captures dynamic hysteretic behaviour; where the output does not follow the input path directly but depends on the history of past inputs, leading to a looped response curve seen in Figures 3.5. For a mass m and input excitation $\mu(t)$, the damped system of ODEs from an MR damper modelled by a modified Bouc-Wen model may be expressed by the following equations adapted from Section 2.

$$m\ddot{x} + c_1\dot{y} + k_1x = \mu \quad (4.1.6)$$

$$\dot{z} = -\gamma |\dot{x} - \dot{y}| z |z|^{n-1} - \beta (\dot{x} - \dot{y}) |z|^n + A (\dot{x} - \dot{y}), \quad (4.1.7)$$

$$\dot{y} = \frac{1}{c_0 + c_1} (\alpha z + k_0 (x - y) + c_0 \dot{x}), \quad (4.1.8)$$

For the dynamics governed above, and in a discrete sequence with N samples, parameters $c_0 \in \mathbb{R}^N$, $c_1 \in \mathbb{R}^N$, and $\alpha \in \mathbb{R}^N$ represent the voltage-dependant parameters of the MR damper, and are characterized based on the efficient voltage $u \in \mathbb{R}^N$, characterized by the following equation:

$$\dot{u} = -\eta(u - v). \quad (4.1.9)$$

Equations 4.1.9 effectively represent a low-pass filter. For a response time of 15 ms, as specified in the technical specifications of the MR damper, an η value of 66.6 s^{-1} representing a time constant of $1 * \tau$ is employed [20]. The PINN established serves to approximate the solution of the ODE with given initial conditions:

$$x(t_0) = x_0, \quad y(t_0) = y_0, \quad z(t_0) = 0, \quad v(t_0) = v_0. \quad (4.1.10)$$

where initial conditions vary by test case. For parameter estimation, the above system of ODEs may be expressed in a condensed form as:

$$m\ddot{x} + f(x, y; \theta_x) = \mu(t), \quad (4.1.11)$$

$$\dot{y} = g(x, \dot{x}, z; \theta_y), \quad (4.1.12)$$

$$\dot{z} = h(x, z; \theta_z), \quad (4.1.13)$$

where items $\theta_x, \theta_y, \theta_z$ represent the physical parameters to be identified within the Bouc wen model of hysteresis. Where predictions regarding the system behaviour may be made based on observations of external displacement, velocity, and acceleration x, \dot{x}, \ddot{x} , as well as an externally applied voltage v and excitation μ .

The input data $I = [t, F_{in}, u_{in}] \in \mathbb{R}^{N \times 3}$ for the network consists of sampled collocation points. Collocation points, T , also referred to as residual points in the literature [84], represent the measured data points made available to the PINN to serve as points to identify the particular solution, whose purpose is to assist with identifying

the exact solution that also fits with the ODE simultaneously.

Sampling these residual points is performed in accordance with varying methodologies, of which, the most commonly used techniques include methods such as equispaced uniform sampling and uniformly random sampling [156, 84]. In literature, these aforementioned methods have been used extensively and generally perform adequately in many scenarios [156].

Selecting collocation points for PINNs presents several difficulties, particularly when solutions exhibit steep gradients [84]. The primary challenge lies in ensuring that these points are adequately distributed across the domain to capture the details of more complicated behaviours in the solution. As a direct result of numerical instability introduced by steep gradients and phenomena such as aliasing, a denser concentration of collocation points is often necessary for regions with steep gradients to accurately characterize the exact solution [84, 104, 156]. Of note, Lu and colleagues proposed a residual-based adaptive refinement method for selecting collocation points, which aims to address these challenges by adaptively refining the points based on the solution’s residuals [84]. However, for the purposes of this study, a more straightforward uniformly equispaced sampling method was employed.

Model Architecture Inspired by the work of [82], an extension for the MR damper identification process is presented herein, designed for the extraction of physical parameters from a set of input data via a split identification process. More specifically, the prediction of forced response and latent response.

This approach employs the use of fragmented networks, with each individual network utilized for the identification of differing physically relevant variables, which allows for modularity. This specialization allows for a more flexible identification of

variables as individual networks may be optimized and fine-tuned for the specific characteristics and dynamics of their assigned domains. This is advantageous as the scale of the problem is uncertain, and the added flexibility may be ideal for facilitating convergence [82]. As will be discussed further in the limitation section, PINNs have been shown in the literature to have struggled with multi-scale problems, chiefly due to challenges in resolving fine-scale features, balancing contributions from varying scales in the loss function, handling stiff differential equations, and the high computational resources required for complex network architectures.

The structure of the implemented workflow utilizes several RNNs operating in parallel. For this implementation, RNNs were chosen over the more prominent LSTM networks chiefly due to their relatively more straightforward architecture, and ease of implementation. This property is advantageous when rapid prototyping and iteration, such as the case with hyperparameter tuning strategies, are necessary.

Within this framework, the physical model parameters that require identification are embedded as additional parameters. These parameters are optimized through a training process that leverages a physics-informed loss curriculum, to be detailed later in this section.

A visualization of the workflow of this program is shown in Figure 4.2. The identification process is divided into multiple fragmented networks. Each network is designed to handle a subset of the variables within the MR damper model.

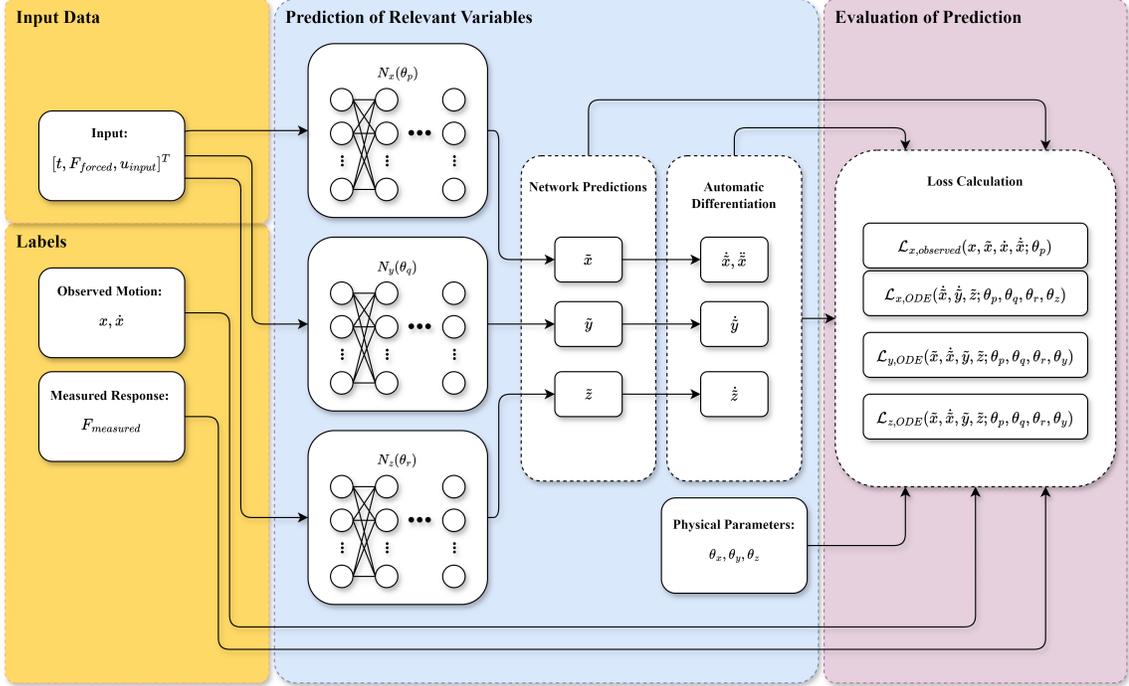


Figure 4.2: An illustration of (A) the general Recurrent Neural Network architecture, and (B) the inner computational processes within each RNN cell.

The NN aims to approximate the individual equations of the system of ODEs modelling the MR damper, defined by NN parameters: $\{\theta_p, \theta_q, \theta_r\}$. For a given input, the network approximates the following equations: $\tilde{x} = N_x(\mathbf{I}, \theta_p) \in \mathbb{R}^N$, $\tilde{y} = N_y(\mathbf{I}, \theta_q) \in \mathbb{R}^N$, and $\tilde{z} = N_z(\mathbf{I}, \theta_r) \in \mathbb{R}^N$ representing the dependent variables of the system. Leveraging the universal approximation property, the NN function is expressed as an approximation for the solution to a system of ODEs. The universal approximation theorem originally established for feedforward NNs states that an NN with a single hidden layer can approximate any continuous function to any desired accuracy, given sufficient neurons in the hidden layer [42]. As an extension of this initial work, the proof for universal approximation has been extended to RNNs in

the work presented by Schafer and colleagues [123]. This is significant as the proof of universal approximation provides a theoretical guarantee that RNNs employed have the capability to approximate any continuous function to any desired degree of accuracy.

Normalization : For data preprocessing in ML algorithms, data normalization is widely employed to scale input and target data features to a comparable level, the primary purpose of which is to enhance the convergence of the learning process and balance the effect of multi-scaled features. Data normalization has been extensively studied and has seen numerous implementations in conventional, data-driven NNs. Both the input and target features are normalized to facilitate convergence during training, then subsequently de-normalized to obtain the original scale for the desired results [45]. Z-score normalization is amongst the most commonly utilized methods for normalization, being a statistical technique that transforms the data to have a mean of zero and a standard deviation of one [102].

This preprocessing step is necessary such that relevant individual features contribute to the learning process, as opposed to the learning process being dominated by a single feature as a result of differences in their original scale. Mathematically, Z-score normalization is defined as follows:

$$X'_i = \frac{X_i - \mu}{\sigma}, \quad (4.1.14)$$

where X_i is the i -th element of the original feature, X'_i is the normalized value, μ is the mean of the feature, and σ is the standard deviation of the feature.

With the case of conventional NNs and deep learning, normalization of both inputs

and target data serves to stabilize the training process when employed with many gradient descent type optimizers, improving convergence rates [46].

However, implementation in PINNs remains scarce, due to the inherent physical meaning attached to data, whereby alterations to the data based on statistical distribution would not be appropriate, as it would result in the loss of said physical meaning inherent. As is the case in PINNs, the physical significance of the inputs is required in the application of governing equations-based learning biases fundamental to the PINN framework. As such, normalizing inputs in PINNs results in severe inaccuracies due to the nature of the framework itself.

While early implementations of PINNs did not rely on the normalization of data due to the nature of their specific implementations, a more pragmatic approach adopted by certain researchers, and in this research, entails normalizing features subsequent to the input layer, which effectively treats normalization as a hidden layer operation [111]. Thus, the network simultaneously preserves the physical meaning of original input values, while still providing some of the benefits of normalization in providing convergence and consistency across features during the training process. Effectively, based on the distribution of the training set, standard Z-score normalization is performed without distorting the physical relevance of the inputs.

Although the method has been adopted, it should be noted that this implementation does have certain limitations. The target data remain unnormalized, which results in potential issues depending on the expected scale of the output. If predicted outputs vary significantly, the optimization process may encounter difficulty in achieving desired accuracy and convergence, as a result of discrepancy in magnitudes. This effect is more pronounced in multiscale problems whereby outputs are expected

to represent physical variables with vastly different scales.

A depiction of the method is illustrated in Figure 4.3.

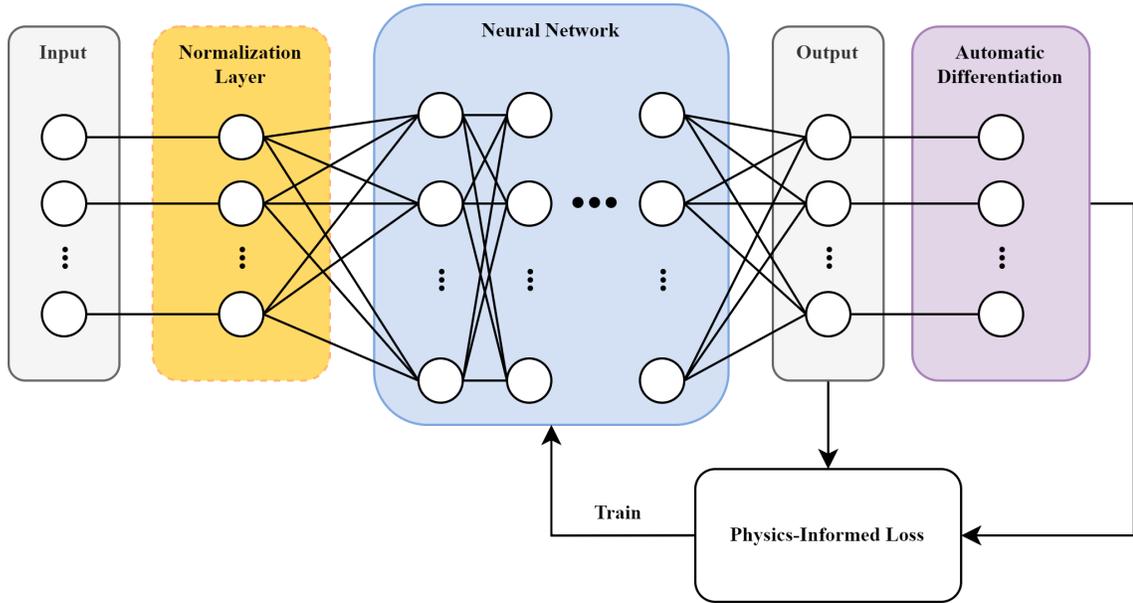


Figure 4.3: Depiction of the normalization layer employed to normalize inputs for processing by the NN, while maintaining the physical meaning of input data, as adapted from [111]

It is also noted that it is theoretically possible to normalize both inputs and outputs and subsequently propagate the variance and mean of said distributions through the network, as has been demonstrated in the work of Xu and colleagues [160]. However, the approach outlined proves unfeasible in the specific context of this implementation, due to the presence of latent, unobservable target variables. Latent variables present a challenge in the determination and utilization of mean and variance of latent distributions. Consequently, the lack of availability of these statistical measures prevents the effective normalization of outputs, thereby complicating the implementation of normalization techniques. It is hypothesized, based on prior works on the modified Bouc-Wen model [136, 82], that resultant outputs are of consistent scale,

and that the lack of output normalization does not pose significant issues. Nonetheless, this method’s efficacy is limited, and the efficient normalization techniques for PINNs remain a promising area of future exploration.

Approximation via Neural Network and Automatic Differentiation Approximate values of response and latent response are utilized to inform the training curriculum through physics-informed loss components. An iterative process is followed, adjusting the network parameters and physical parameters respectively to minimize aforementioned loss components.

A sequential initialization strategy is implemented consistent with the work of Liu and colleagues [81], and makes use of information gained from prior training iterations, ensuring continuity in the learning process and building upon previous knowledge progressively. The process of updating physical parameters based on previous batch predictions also contributes to the stability of the training process. More generally, the identification process maintains a connection between consecutive training steps, which can lead to more coherent and consistent parameter updates.

The optimization of both the NN parameters and the unknown structural parameters is performed with the Adaptive Moment Estimation (ADAM) optimizer, initially introduced by Kingma and Ba [62]. ADAM is an optimization algorithm which adaptively adjusts learning rate of parameters during training through estimates of the first and second moments of gradients. The optimizer adjusts the learning rate based on the average of the gradients (first moment) and the uncentered variance (second moment) [62].

Automatic Differentiation (AD) is conventionally utilized to compute the gradients of loss for a tested set of network parameters, during the backpropagation process

[93]. It is emphasized here, as in addition, AD is also employed to capture gradients of resultant predictions with respect to inputs, typically time, effectively calculating the derivative of certain physically relevant variables with respect to time. AD is a computational technique designed to efficiently and systematically compute the derivatives of functions and is widely used in deep learning, through applying the chain rule of calculus to each elementary operation within the computational graph of the NN, which itself is effectively a function composed of simpler operations. The process of the optimization process employing AD may be outlined as follows:

- **Forward Pass:** The forward pass involves computing the outputs of the NN, including output at each layer level. Condensing the representation of desired parameters (ie. weights, biases, physical parameters) as θ , the output of each layer may be represented as follows:

$$\begin{aligned}z^{(1)} &= \theta^{(1)}x, \\a^{(1)} &= \sigma(z^{(1)}), \\&\vdots \\z^{(L)} &= \theta^{(L)}a^{(L-1)}, \\ \hat{y} &= \sigma(z^{(L)}).\end{aligned}$$

- **Loss Function:** A loss \mathcal{L} is defined based on observations and deviations from known physics. Additional details on the derivation of the loss will be presented in the subsequent sub-section.

- **Backward Pass:** AD computes the derivatives of the output with respect to each input by propagating gradients backward through the network layer. Initially starting from the loss \mathcal{L} , the gradient of loss with respect to the output may be obtained:

$$\frac{\partial \mathcal{L}}{\partial \tilde{y}}$$

This is propagated backwards throughout the network. For the output layer, the gradient is:

$$\begin{aligned} \delta^{(L)} &= \frac{\partial \mathcal{L}}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial z^{(L)}} = \frac{\partial \mathcal{L}}{\partial \tilde{y}} \cdot \sigma'(z^{(L)}) \\ \frac{\partial \mathcal{L}}{\partial \theta^{(L)}} &= \frac{\partial \mathcal{L}}{\partial \tilde{y}} \frac{\partial \tilde{y}}{\partial z^{(L)}} \frac{\partial z^{(L)}}{\partial \theta^{(L)}} = \delta^{(L)} (a^{(L)})^T \end{aligned}$$

and for prior hidden layers l , where $l = \{L - 1, L - 2, \dots, 1\}$:

$$\begin{aligned} \delta^{(l)} &= (\theta^{(l+1)})^T \delta^{(l+1)} \cdot \sigma'(z^{(l)}) \\ \frac{\partial \mathcal{L}}{\partial \theta^{(l)}} &= \delta^{(l)} (a^{(l-1)})^T \end{aligned}$$

- **Update Parameters:** Relevant parameters are updated in the direction that minimizes the loss function. This iterative process is typically carried out using optimization algorithms like gradient descent or its variants:

$$\theta^{(l)} \leftarrow \theta^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \theta^{(l)}}$$

where σ is the activation function, and σ' is the derivative of the activation function. η represents the learning rate, and L is the number of layers in the network.

AD requires a single forward pass and backward pass through the network to compute the necessary derivatives with respect to prior operations in the network irrespective of input dimensionality. This efficiency is pertinent when dealing with high-dimensional inputs, where methods such as finite difference would be slow and computationally expensive.

Loss Function The above equations outlined are in alignment with the original formulation by Raissi [110], whereby the network is trained to minimize discrepancies between predicted time derivatives of solution space, with the calculated value of time derivatives utilizing governing differential equations:

The physics-based loss functions are defined, and subsequently minimized by the network. The loss function is defined for each sample point i of total samples taken N for the sequence:

$$\mathcal{L}_{observed} = \frac{1}{N} \sum_{i=1}^N [x_i - \tilde{x}_i]^2 + \frac{1}{N} \sum_{i=1}^N [\dot{x}_i - \dot{\tilde{x}}_i]^2 \quad (4.1.15)$$

$\mathcal{L}_{observed}$ consists of two primary components designed to guide the network towards accurate predictions. The first component, $\frac{1}{N} \sum_{i=1}^N [x_i - \tilde{x}_i]^2$, focuses the optimization algorithm on minimizing the squared differences between observed response values x_i and their corresponding predictions \tilde{x}_i . Complementing this, the second term, $\frac{1}{N} \sum_{i=1}^N [\dot{x}_i - \dot{\tilde{x}}_i]^2$, penalizes deviations in predicted derivatives \dot{x}_i from observed derivatives $\dot{\tilde{x}}_i$, and in general ensures that the network adheres to the observed forced response dynamics of the damper system. In addition to enforcing observed responses,

the ODE governing the behaviour is also enforced, represented as:

$$\mathcal{L}_z = \frac{1}{N} \sum_{i=1}^N \left[(\dot{\tilde{z}}_i) - \left(-\gamma |\dot{\tilde{x}}_i - \dot{\tilde{y}}_i| \tilde{z}_i |\tilde{z}_i|^{n-1} - \beta (\dot{\tilde{x}}_i - \dot{\tilde{y}}_i) |\tilde{z}_i|^n + A (\dot{\tilde{x}}_i - \dot{\tilde{y}}_i) \right) \right]^2, \quad (4.1.16)$$

$$\mathcal{L}_y = \frac{1}{N} \sum_{i=1}^N \left[(\dot{\tilde{y}}_i) - \left(\frac{1}{c_{0,i} + c_{1,i}} (\alpha_i \tilde{z} + k_0 (\tilde{x} - \tilde{y}) + c_{0,i} \dot{\tilde{x}}_i) \right) \right]^2. \quad (4.1.17)$$

The equation 4.1.16 presented defines the loss based on the explicitly defined equation governing rate of change of hysteretic displacement, denoted as \mathcal{L}_z . This component penalizes the discrepancy between the derivative of the hysteretic displacement and a reconstructed form of the differential equation that represents the same derivative. For each sample i in a sequence of N samples, the loss is calculated as the squared difference between $\dot{\tilde{z}}_i$ and the expression involving physical parameters and the predicted displacements \tilde{x}_i , \tilde{y}_i , and \tilde{z}_i , with hysteretic loop characteristics governed by the parameters γ , β , and A . This is in accordance with equation 2.2.10 as detailed in Section 2.

A similar process was followed for the loss function represented by the equation 4.1.17, for intermediary displacement, corresponding to the original equation 2.2.11. The function \mathcal{L}_y penalizes deviations between the derivative of intermediary displacement and the reconstructed differential equation that models this derivative. More specifically, the loss measures the squared difference between $\dot{\tilde{y}}_i$ and the terms involving parameters: α_i , $c_{0,i}$, $c_{1,i}$, k_0 , k_1 as well as the predicted variables. Further to this, an additional component was applied to enforce adherence to observations in

damping force:

$$\mathcal{L}_{response} = \frac{1}{N} \sum_{i=1}^N [F_{measured,i} - (\alpha_i \tilde{z}_i + c_{0,i} (\dot{\tilde{x}}_i - \dot{\tilde{y}}_i) + k_0 (\tilde{x}_i - \tilde{y}_i) + k_1 \tilde{x}_i)]^2, \quad (4.1.18)$$

where the unction $\mathcal{L}_{response}$ penalizes the residual between the measured force, $F_{measured,i}$, and the reconstructed predicted force.

The optimization process minimizes the weighted sum of component losses from various contributions and may be expressed as:

$$\begin{aligned} \Theta_{\text{phys}}^*, \Theta_{\text{net}}^* = \underset{\Theta_{\text{phys}}, \Theta_{\text{net}}}{\text{argmin}} & (\lambda_{observed} \mathcal{L}_{observed}(\theta_x, \theta_p) + \lambda_y \mathcal{L}_y(\theta_y, \theta_q) + \lambda_z \mathcal{L}_z(\theta_z, \theta_r) + \\ & \lambda_{response} \mathcal{L}_{response}(\theta_x, \theta_y, \theta_z, \theta_p, \theta_q, \theta_r)), \end{aligned} \quad (4.1.19)$$

where $\Theta_{\text{phys}} = \{\theta_x, \theta_y, \theta_z\}$ and $\Theta_{\text{net}} = \{\theta_p, \theta_q, \theta_r\}$ represents the parameters to be determined through the optimization algorithm. The weights λ_x , λ_y and λ_z allow for tuning the model to prioritize certain component losses, which serves to ensure that the network balances adherence to physical principles while fitting the data.

4.1.2 Hyperparameter optimization

Hyperparameter optimization is often resource-intensive in evaluating the objective function over the full range of specified parameters, involving training the model and calculating the metric for each set of hyperparameters [28]. This process becomes increasingly cumbersome with an increasing number of hyperparameters and computationally expensive models.

As a result, manual execution of this process becomes impractical. While grid search and random search offer some improvement over manual tuning by automating the train-predict-evaluate cycle within a predefined hyperparameter space, they still exhibit inefficiencies [8]. These methods cannot intelligently select the next set of hyperparameters based on past evaluations. Both grid and random search operate without taking advantage of knowledge gained from prior assessments, which leads to substantial time expenditure on evaluating sub-optimal hyperparameters.

Bayesian optimization approaches the optimization problem by formulating it as a probabilistic model [30], with typical implementations using a Gaussian Process (GP) or Tree-structured Parzen Estimator (TPE) model, to represent the unknown objective function [30]. The surrogate model gives the probabilistic estimation of the objective function and its uncertainty within the hyperparameter space.

In this implementation, the TPE is employed to estimate the objective function within the hyperparameter space. The TPE involves the identification of two density functions. Through this, a probabilistic model is constructed to guide the search for optimal hyperparameters as follows:

1. The TPE initiates by randomly selecting a subset of hyperparameters and subsequently arranging them according to their performance scores [152, 99]. Through the random sampling of hyperparameter sets from predefined priors, this approach provides an initial approximation of regions within the search space that yield proficient models.
2. Subsequent to the initial sampling from our priors, the hyperparameter set undergoes division into two distinct groups based on predefined quantiles. Specifically, a quantile threshold γ , where γ ranges from 0 to 1, is applied to partition

the hyperparameter search space. Combinations resulting in models performing within the top $\gamma * 100\%$ of all previously generated models form one distribution denoted as $l(x)$, while the remaining hyperparameter combinations constitute another distribution designated as $g(x)$ [9, 152, 99]. Thus for a set of hyperparameters θ the TPE defines the probability distribution utilizing probability densities $l(\theta)$ and $g(\theta)$:

$$p(\theta|\mathcal{L}) = \begin{cases} l(\theta) & \text{if } \mathcal{L} < \mathcal{L}^* \\ g(\theta) & \text{if } \mathcal{L} \geq \mathcal{L}^* \end{cases} \quad (4.1.20)$$

where \mathcal{L}^* represents the threshold loss corresponding to the quantile threshold.

3. Applying Parzen Estimators, the two groups, denoted as $g(x)$ and $l(x)$, undergo modelling into estimated densities using kernel density estimators. To guide the selection of the next hyperparameter configuration for evaluation, the Bayesian optimization process incorporates the Expected Improvement (EI) acquisition function, which evaluates the potential improvement over the current best-observed value and decides the next set of hyperparameters to evaluate. The identification of hyperparameters with the highest expected improvement is evaluated in accordance with the ratio of $g(x)$ to $l(x)$, as illustrated in the work of [9].

$$EI(x) \propto \left(\gamma + \frac{g(x)}{l(x)}(1 - \gamma) \right)^{-1} \quad (4.1.21)$$

whereby the point x is where the algorithm expects the highest improvement in the objective function.

4. Following the selection of each hyperparameter configuration, the chosen set of hyperparameters that maximizes the EI undergoes evaluation with its performance metrics with respect to the objective function recorded. The model is updated accordingly to the newly acquired observations, incorporating the latest performance data, an iterative process that continues for a predefined number of iterations or until the convergence criterion is met.

4.2 Results and Discussion

The results from employing the PINN approach outlined in Section 4.1 to estimate the physical parameters of a modified Bouc-Wen model are presented in this section. The established framework was implemented using *PyTorch* within a Python environment. Table 4.1 presents the parameters determined through the aforementioned processes.

From the above process, the best hyperparameters refined through this Bayesian optimization strategy with TPE are returned as the optimal configuration for the given problem. The hyperparameter optimization of the NNs yielded a series of adjustments across various parameters. Using the *Optuna* library, the hyperparameters are refined via Bayesian optimization to identify optimal configurations for each network. For the NNs employed, the learning rate was set to 0.003083, the number of layers to 4, and the number of neurons per layer to 116. The dropout rate was adjusted to 0.2995. Other parameters included a weight initialization method set to Xavier initialization [66]. The activation function was set to the hyperbolic tangent, which is typical to PINNs [110], and the optimizer type was configured as the ADAM optimizer. Each network was trained for a maximum of 500 epochs to allow sufficient iterations for the learning process, with the execution terminated prior in the case all

Parameter	Value	Units
α_a	9.6341e+01	$\frac{kN}{m}$
α_b	1.2555e+02	$\frac{kN}{m} \cdot V$
β	8.2089e+03	m^{-2}
γ	8.1997e+03	m^{-2}
A	2.3086e+01	—
c_{0a}	1.8059e+01	$\frac{kN \cdot s}{m}$
c_{0b}	1.7504e+01	$\frac{kN \cdot s}{m} \cdot V$
c_{1a}	9.8029e-01	$\frac{kN \cdot s}{m}$
c_{1b}	2.6244e+01	$\frac{kN \cdot s}{m} \cdot V$
k_0	9.7866e-01	$\frac{kN}{m}$
k_1	1.4645e-01	$\frac{kN}{m}$
n	1.9928e+00	—

Table 4.1: List of parameters and their corresponding values, as determined by the PINN parameter estimation algorithm outlined in Section 4.1.

parameters and composite loss have reached convergence ($\pm 1\%$ change). Weights for the composite loss function were set at $\lambda_{observed} = 1$, $\lambda_{response} = 0.001$, $\lambda_y = \lambda_z = 10$. The explanation for this is that the weights of the latent variables are prioritized due to lacking training data [82]. Furthermore, the weight of the response is set as such due to the scale of the output data being inconsistent with the rest of the loss values. Figure 4.4 demonstrates the total loss values of the number of trials run for Bayesian optimization.

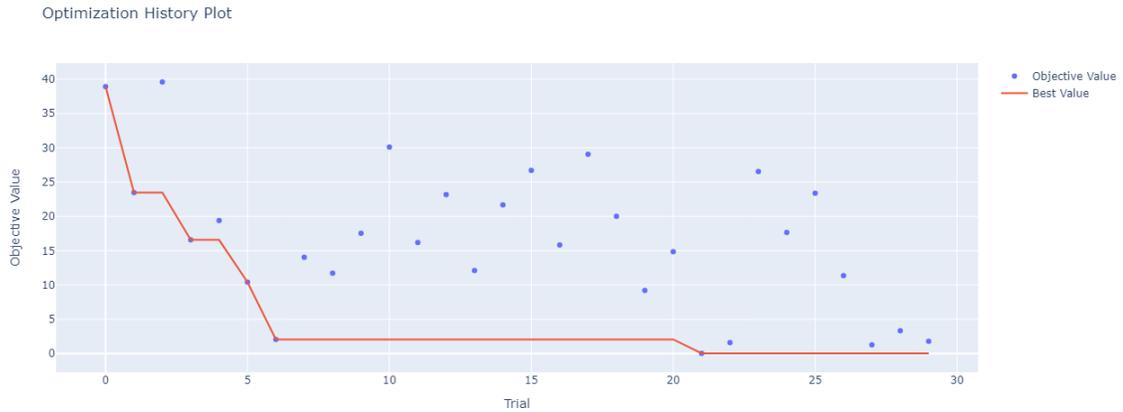


Figure 4.4: Hyperparameter optimization history utilizing Bayesian optimization.

In addition to NN hyperparameters, learning rates for added physical parameters are also fine-tuned. Custom-added parameters representing physical values require separate adjustments of their learning rates compared to the rest of the NN parameters, due to the nature of these parameters as representations of physical quantities. Parameters may exhibit different scales or importance, and it would thus be advantageous to mitigate the risk of overshadowing the influence of other NN components that might dominate training dynamics under uniform learning rate settings. The

initial learning rates for these parameters are summarized in Table 4.2

Parameter	Learning Rate	Parameter	Learning Rate
β	0.0671615	γ	0.0671615
$c_{0,a}$	0.000114872	$c_{0,b}$	2.63448e-05
$c_{1,a}$	0.000389324	$c_{1,b}$	0.0234330
α_a	0.000102989	α_b	1.92305e-05
k_0	5.03670e-05	k_1	7.96682e-05
A	0.000281767		

Table 4.2: Learning rates, as determined from hyperparameter optimization, tabulated for physical parameters.

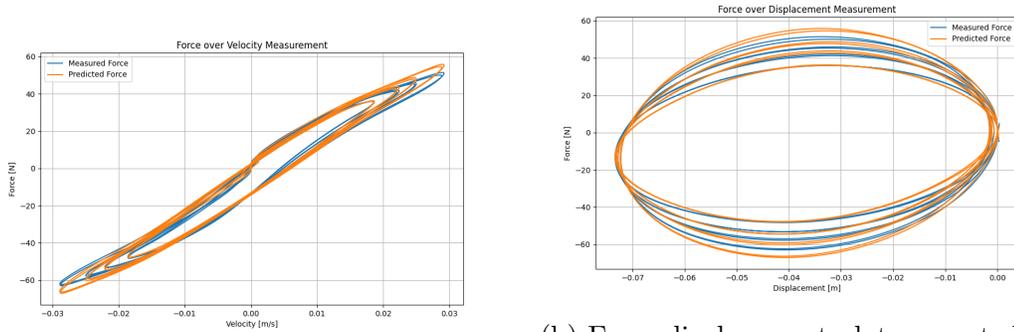
The validation process assesses how well the model generalizes to novel scenarios and unseen variations in experimental conditions; Testing the model’s ability to accurately predict force-time, force-velocity, and force-displacement relationships across a spectrum of applied voltages and applied forced excitations. Comparisons were drawn between the model’s predictions derived from the PINN framework and observations obtained from the experimental setup. Testing was performed on varying unseen data collected from the experimental setup, which comprises an applied voltage range of 0V to 4.003 and a velocity ranging from 0 cm/s to 10.23 cm/s for constant acceleration cyclical motion. The consistency between the PINN estimates and the experimental data is quantified by the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) values. The formula for calculating the error metrics based

on observed force F_i and predicted force \tilde{F}_i is given as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (F_i - \tilde{F}_i)^2} \quad (4.2.1)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{F_i - \tilde{F}_i}{F_i} \right| \times 100\% \quad (4.2.2)$$

A sample of collected data is plotted, visualizing the effect of increasing the speed of actuation and its effects, illustrated through the force-velocity and force-displacement hysteresis curves in Figure 4.5a, and 4.5b.



(a) Hysteresis loop generated by predicted physical parameters and observed values.

(b) Force-displacement plot generated by predicted physical parameters and observed values.

Figure 4.5: Hysteresis loop of measured behaviour. Plot generated via varying actuation speed from 0.5291 cm/s to 2.672 cm/s.

The adherence of the predicted results to the experimental values suggests that the model accurately is able to capture the non-linear characteristics, as well as the effects of hysteresis and latent variables of the MR damper system. In addition, the effects of applied voltage on MR dampers are studied in additional detail, as it is the primary methodology by which MR dampers may vary their damping response. To further validate the robustness of the estimated parameters, the force-time, force-velocity,

and force-displacement relations were plotted at varying applied voltages. These plots are shown in Figures 4.6 - 4.11, and compare the estimated values obtained from the PINN framework against the measured data from physical experimental setups across a variety of sampled tested voltages.

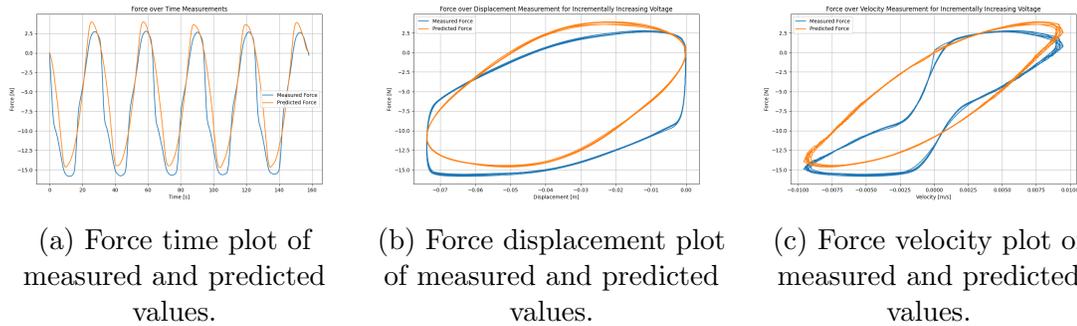


Figure 4.6: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 0 V.

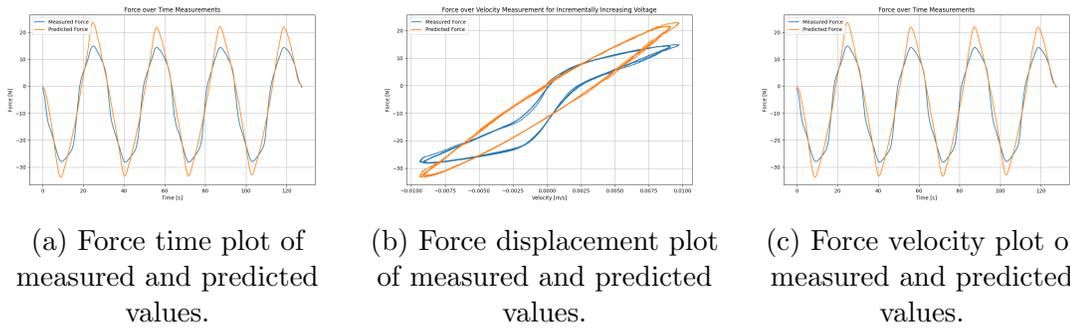


Figure 4.7: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 0.6678 V.

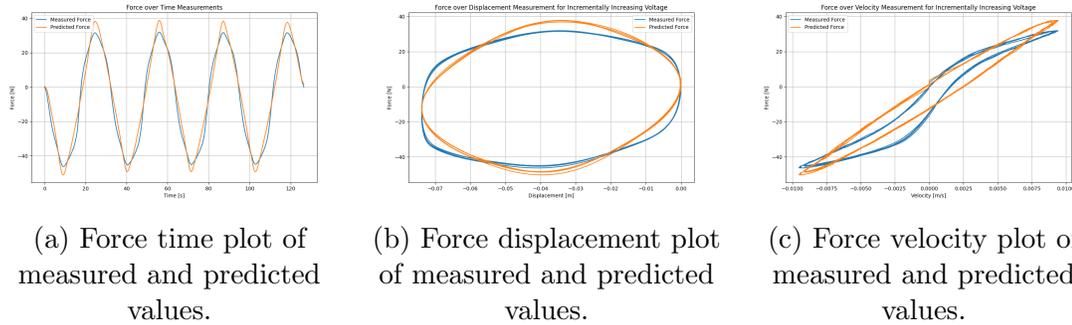


Figure 4.8: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 1.6342 V.

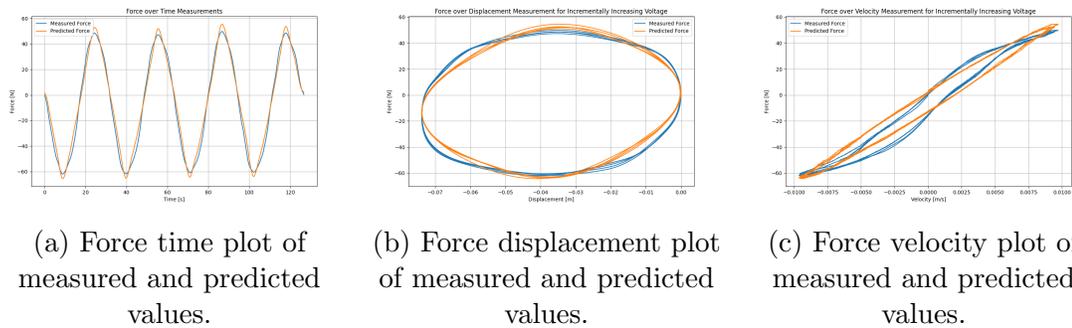


Figure 4.9: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 2.2703 V.

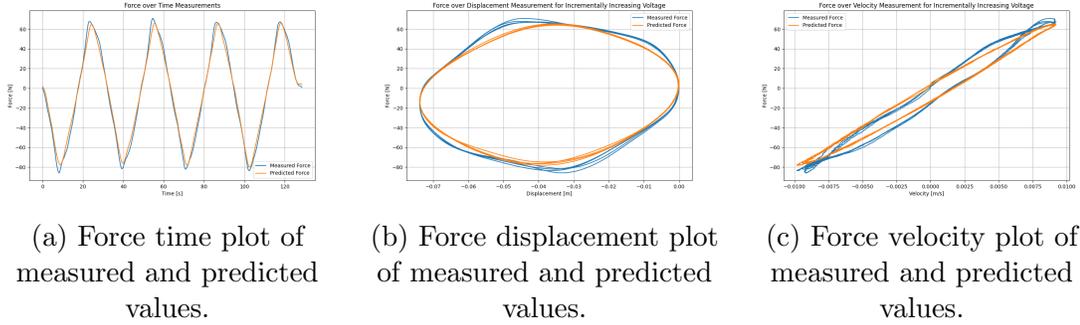


Figure 4.10: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 3.2046 V.

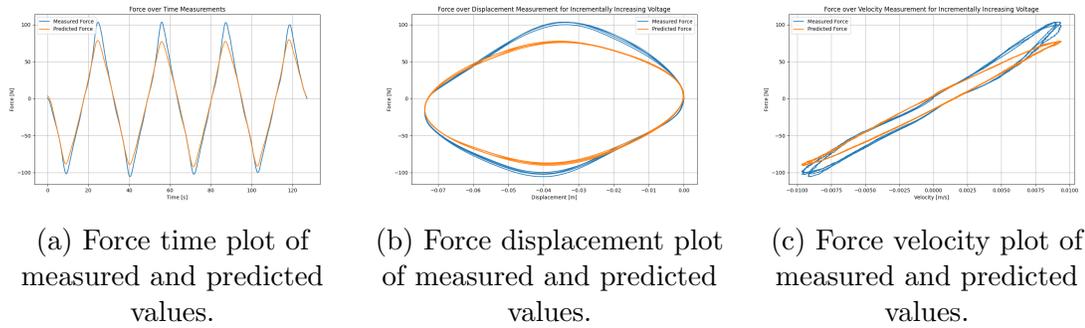


Figure 4.11: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at an applied voltage of 4.003 V.

The comparisons indicate a general agreement between the estimated values and the observed data, and to varying degrees, is evident across voltage levels tested. The results are tabulated for varying applied voltage levels in Table 4.3.

Applied Voltage [V]	RMSE [N]	MAPE [%]
0	2.4601	83.6179
0.1335	2.1637	76.6524
0.2671	2.3572	62.6252
0.3870	3.8462	67.1547
0.5342	2.7365	71.1235
0.6678	3.0275	63.1236
0.8013	2.8726	69.0563
0.8832	4.5472	43.9433
1.0684	3.1936	48.2712
1.1218	3.6295	31.9752
1.3355	4.4927	38.6336
1.4691	5.2365	33.2655
1.6342	4.7255	54.3958
1.7362	6.3927	29.8527
1.8697	6.1056	25.1946
2.0033	5.0275	28.9264
2.1368	6.3829	23.4808
2.2703	8.2628	19.0564
2.4039	7.9982	26.8462
2.5374	7.6292	27.2657
2.6711	9.2816	24.8185
2.8045	8.3452	22.3741
2.9378	7.6253	21.6938
3.0712	9.7643	23.5072
3.2046	9.9654	21.1348
3.3380	10.7523	19.5406
3.4713	9.1984	18.3841
3.6048	8.6745	22.7132
3.7382	11.0347	18.8519
3.8716	10.8765	20.1954
4.0030	9.5643	19.7382

Table 4.3: Measured MSE and MAPE values across varying applied voltages.

The predictions generated by the developed parameters generally aligned well with

the response observed from the experimental setup. However, the model exhibited varying RMSE and MAPE across the range of applied voltages. The general trend indicated an increase in RMSE with increased applied voltage, which is to be expected, due to the scaling effect increasing applied voltage had on the magnitude of force data. With an increase in applied voltage, the magnitude of resulting resistive damping forces also increases, resulting in a higher absolute value. On the other hand, MAPE showed a decreasing trend from lower applied voltages, before stabilizing around a percentage of 20% at an applied voltage of 2.1368, suggesting that the model’s relative accuracy improved at higher voltages, even though the absolute error increased. The behaviour is visualized in the Figure 4.12.

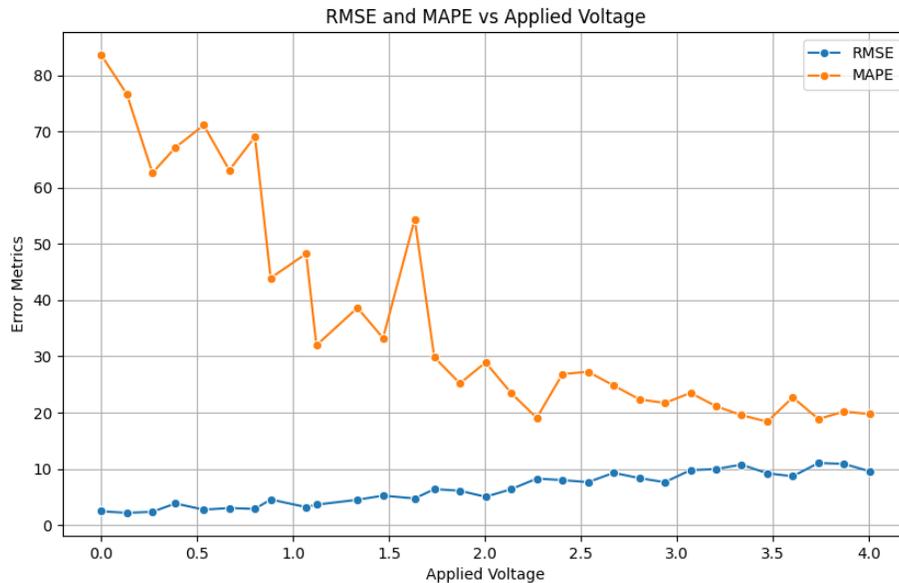


Figure 4.12: RMSE and MAPE with increasing applied voltage.

While the model demonstrated improvement in modelling responses at higher applied voltage, it has encountered issues in accurately capturing sudden jumps or

spikes in the data, as evidenced by plots depicted in Figures 4.6 to 4.11. There exist rapid changes in the resultant damping force, as observed in experimental force-time graphs during movements. These responses were less pronounced with the prediction of the model, leading to a lower maximum absolute value and a reduced range in the predictions, indicating that while the model performed reasonably well in capturing the overall trends in the dynamic behaviour, it falls short in modelling transient peaks and dynamic responses accurately at higher voltages.

The discrepancy observed in prediction accuracy at varying voltage inputs, particularly the significant disproportionate deviation in capturing hysteresis at low voltage inputs for MAPE, may be attributed to several factors inherent to both the physics of the system and the limitations of the implemented machine learning approach.

At lower applied voltage, the response of the MR damper is relatively weaker, making it more susceptible to external effects, with the effect emphasized at the extremities of the stroke length. While the force measurements have been zeroed to account for the initial offset force, the lower magnitude of response at the lower end of input increases vulnerability to external influences.

A significant factor contributing to these discrepancies is the effect of the accumulator within the MR damper. The force exerted by the diaphragm and compressed nitrogen gas, especially at low velocities, low applied voltages and extended stroke lengths examined in the study, becomes non-negligible. As explored in other, previous publications on the subject, the influence of the accumulator is non-negligible at lower speeds, applied voltage, and long stroke lengths [72, 75]. The interference outlined has been attributed to the fact that systems involving pneumatics often have inherent compliance or elasticity, due to the compressibility of the gas and the flexibility of the

system components. With lower magnitudes of applied excitations, these compliant elements absorb and transmit small forces which affect the piston's response profile. At a higher applied voltage, the overall damping forces and the dynamics of the system become more dominant, and this effect is not as obvious, evidenced by relatively lower MAPE values.

It is also evident that the model encounters challenges in capturing sudden shifts in force effects such as abrupt changes in acceleration across all applied voltage levels. A variety of factors may contribute; The model's training primarily focused on gradual transitions and continuous data. Sudden shifts in force, which may occur due to unexpected external factors or rapid changes in experimental settings (ie. such as brief delays in sending/receiving commands, jolts to the systems et cetera.) may not be accounted for in established physics, nor collocation points selected. Consequently, the model's ability to generalize and predict accurately under such dynamic conditions is compromised. The PINN framework, while robust in incorporating physical laws into its predictions, may require additional specific collocation point selection strategies to learn and adapt to rapid changes in regions of significant deviation due to circumstances not predicted by known physical models, as have been explored in the work of Lu and colleagues [84].

In experimental setups, measurements capture the transient spikes and peaks that arise from abrupt changes in voltage or mechanical conditions imposed by the experiment. A predictive model such as the model in this study smooths out these transient effects due to filtering mechanisms inherent in their formulation and required for numerical stability. In addition, the process of parameter estimation utilizing PINNs emphasizes fitting the average behaviour or cyclic trends rather than capturing

transient/extreme values seen in experimental data. This discrepancy primarily arises from the optimization criteria used during parameter estimation, which while there do exist components that minimize loss with respect to data points, are outweighed by the components that minimize the discrepancy in known ODEs with predictions or are simply not selected at locations of concern.

Due to collecting data from devices of varying baud rates, data points gathered during data collection are irregular in time, albeit with small time increments (>0.005 s) between them. As a result, this introduces a degree of variability in the model inputs and targets. Irregular sampling intervals lead to uneven distribution of data points across the force-time domain, resulting in slight gaps or clusters in the dataset, and a greater chance that data points are mismatched in terms of their positioning in the time domain. This phenomenon renders it difficult to discern rapid changes in force that occur between sparsely sampled points.

It is also important to note that the modified Bouc-Wen model, which serves as the basis of physics-based regularization in this implementation, does not perfectly capture the behaviours of MR dampers. The model has been shown to have inherent limitations with respect to representing the damper's response [116, 120, 148]. Due to the fixed-form nature of the parametric model itself, certain complexities may not be captured as well.

Despite these shortcomings, however, the use of PINNs has demonstrated its capabilities in capturing the overall dynamics of the MR damper across a range of operating conditions. In addition, the framework's flexibility allows for the adaptation of alternate systems for the characterization of their response. With altering neural network parameters, data from a degraded or faulty system may potentially be

accounted for through purely data-driven adjustments to the neural network parameters, while simultaneously freezing previously identified physical parameters. The flexibility of the PINN framework also enables its application to various types of differential equations and physical models beyond the Bouc-Wen model.

4.2.1 Limitations of methodology and recommendations

This section details some of the limitations associated with this implementation of PINNs.

The computational cost associated with PINNs is substantial due to the complexity of the deep learning architecture backbone and the potential high-dimensional parameter space involved. An issue that may be further exacerbated by the addition of physical parameters within the optimization process. As shown in the literature, despite the promise it has demonstrated, PINNs may not consistently outperform traditional numerical analysis methods such as FE analysis, especially so for well-established problems, in which said conventional methods have been extensively optimized [84].

The curse of dimensionality, as with conventional deep learning models, represents a challenge for PINNs, as the computational cost of training increases exponentially with the number of input dimensions. While PINNs offer a quick and pragmatic approach to learning dynamics from high-fidelity simulators that are expensive to run, this expedited learning process may compromise accuracy for computational efficiency. The complexity of deep learning models increases with the number of dimensions, drastically inflating the number of parameters and, consequently, the computational cost and data needed for training. High-dimensional ODEs exacerbate

these difficulties, as the network depth and architecture complexity must scale appropriately to adequately approximate certain responses, resulting in increased training times.

In addition, as alluded to in the prior sections, PINNs struggle with slow convergence and poor accuracy in dealing with multiscale problems, causing PINNs to fail to accurately capture the dynamics of all variables or parameters involved. This issue has been outlined in the works of many authors in the area. This has been attributed to the spectral bias inherent in NNs [105], which suggests that NNs are biased towards learning less complex functions, prioritizing low-complexity functions. This feature, which has been utilized by authors to explain the generalization ability of NNs, has also exacerbated the issue of convergence in a multiscale setting. It has been shown that even for simple problems, PINNs struggle to approximate the high-frequency features within the solutions [151].

The addition of a physical regularizer, depending on the problem being solved, may introduce additional degrees of complexity to the loss function overall. Current methods of optimization rely primarily on gradient descent and its variants, in which the network adjusts its parameters in steps toward the direction of minimal error with respect to loss. With the varying, potentially competing loss components, there exists added complexity of the loss function landscape, further complicating the process of optimizations through the introduction of local minima, for example. This aspect of physics-based regularization has been noted in the work of [65], whereby the characteristic increase in model complexity has been noted with the introduction of soft regularization terms.

The additional complexity hinders the practical application of PINNs in high-dimensional scenarios, with various authors in the field currently working to extend PINNs to be able to accommodate. Of note, a study by Yang and colleagues saw the implementation of a physics-informed Generative adversarial network (GAN) to address the challenges associated with high dimensionality in solving stochastic differential equations [162]. Their work has demonstrated effectiveness in managing problems up to 30 dimensions.

PINNs are flexible in the incorporation of physics within the learning process, their approximation of boundary conditions and solutions to PDEs may not always achieve the desired numerical accuracy. In numerous complex systems, the precise form of PDEs may be unknown, as well. In addition, the approximation quality heavily depends on the network architecture, the optimization process, and the methodology by which boundary conditions are enforced. While the integration of physical laws into the learning process is advantageous, it does not guarantee precise numerical accuracy.

In addition, through the physics-informed loss functions, physics-based loss terms act as a penalization for the network in the case of violations, however, they are not enforced as hard constraints. This proves an issue in hybrid loss functions involving penalization terms with respect to labelled data as inaccuracies in the data may cause the corresponding loss term to dominate within the hybrid loss function. To a lesser extent, with respect to physics-based regularization and PINNs in general, as the physical loss is not strictly enforced, physical violations or deviations from expected physical behaviours may still be produced by the network.

In addressing the issue of computational efficiency in PINNs, the exploration of

PINNs coupled with Random Projections should be highlighted. While the direct examples within the recent literature specifically highlighting PINNs integrated with Random Projections are sparse, the concept of Random Projections itself is well-established, and their application for dimensional reduction has been recognized for its capability to enhance computational efficiency when working with high-dimensional data. The integration may represent a practical reduction in computational complexity, utilizing the efficacy of Random Projections to manage the high dimensionality of data. The sparse random projections technique, as discussed by Li and colleagues in their work, illustrates the potential benefits of random projections in processing high-dimensional data to enhance computational efficiency with minimal loss of information[77]. Applied concurrently with the PINN, this suggests a promising direction for future research where the strengths of both methodologies are synergized. A study in this area by Fabiani and colleagues introduces a strategy for solving initial value problems of nonlinear ODEs and index-1 differential-algebraic equations, using random projections and focusing on estimating the weights from the hidden to the output layer using Newton iterations [26]. The proposed strategy by Fabiani and colleagues utilized a singular value decomposition for low-dimensional systems and sparse QR factorization with regularization for large-dimensional systems. The proposed strategy has been evaluated against various benchmark problems by the authors, and comparative analyses with traditional MATLAB solvers (ode15s, ode23s, and ode23t) demonstrate the strategy's potential as a viable alternative to conventional solvers [26].

Another avenue for addressing the uncertainty surrounding the form and presence

of governing differential equations could utilize the equation-free appr. The equation-free approach is used in analysis where equations describing the system's behaviour are difficult or impossible to derive. It utilizes a detailed microscopic simulation to gather information about the system's state and then applies the information on a macroscopic scale to guide the overall analysis and prediction. This methodology allows for bypassing the derivation of detailed macroscopic descriptions [58, 59]. This synergy is explored through the work of Roberts and colleagues, in their presentation of an Equation-Free Patch Scheme [115], which enables large-scale simulations to be conducted through computations on small, distinct microscale patches. The Equation-Free Patch Scheme was applied in the study for simulating a heterogeneous elastic beam, showcasing the potential for accurate and efficient multiscale simulation without macroscopic equations [115].

In addition, manifold learning techniques offer another direction for understanding the dynamics of systems beyond the reach of classical differential equation-based models. Manifold learning identifies the hidden structures within high-dimensional data, represented as a lower-dimensional manifold embedded within the higher-dimensional space. Manifold learning facilitates the reduction of complex dynamics into a more manageable, lower-dimensional representation. Through this, it enables visualization, dimensionality reduction, and feature extraction.

Various authors have attempted to integrate manifold learning for applications in the discovery of features in latent spaces. Authors such as Galaris integrated PIML with manifold learning in the context of lattice Boltzmann model simulations for PDEs [35]. Their study employs the parsimonious Diffusion Maps to identify the dimension of the manifold. This method is employed for feature selection over

the parameter space, providing a streamlined method to analyze numerical bifurcations in PDEs from lattice Boltzmann simulations [35]. This approach demonstrates the potential of combining advanced machine-learning techniques with computational physics to enhance the understanding and analysis of complex systems.

In other works, Burbulla and colleagues introduced a method that integrates geometric transformations within PINNs by incorporating a diffeomorphism, which is a mapping of a reference domain allowing for the robust adaptation to geometric variations including those on lower-dimensional manifolds [13]. The effectiveness of this approach is demonstrated across varying problems, including solving the Eikonal equation on an Archimedean spiral, addressing the Poisson problem on a surface manifold, simulating incompressible Stokes flow in a deformed tube, and performing shape optimization with the Laplace operator. Vaquero and colleagues extends this approach by visualizing the Hamilton-Jacobi PDE as an optimization problem, with solutions approximated using ML techniques [147]. The authors introduce a method for constructing Poisson integrators that preserve Poisson geometry on integrable Poisson manifolds. Graycyk and colleagues discuss a manifold-based autoencoder method in their work where the manifold latent space evolves based on the Ricci flow in a physics-informed setting for learning nonlinear dynamics in time [38]. Finally, Krishnanunni and colleagues introduced a manifold-regularized layerwise sparsifying training approach in the neural architecture adaptation domain, showcasing the method's efficacy in learning and iterating over models [64].

With parameter estimation and system identification in general, the robustness of the parameters estimated to variances within real-world applications is of interest

for the practical implementation of said strategies. Parameters developed in this implementation of the PINN for system identification in an MR damper experimental setup were identified and validated within a controlled environment in which external factors and perturbations to the system are minimized. In such conditions, the physical system is well-characterized by the model. Extending this implementation to real-world applications may introduce discrepancies between the predicted and observed behaviours of the system due to factors such as environmental variations, unmodeled dynamics, and measurement noise,

PINNs are advantageous in this regard, in that they can partially accommodate deviations through the inherent approximative capabilities of NNs. A potential point to accommodate deviations from the learned parameters may be to freeze identified physical parameters and adjust NN parameters according to observed residuals. In effect, the NN serves as a corrective mechanism, whereby NN parameters would serve to model the residuals between predicted and observed measurements. This is, in effect, reduced to a black-box model of said residuals, with the results being subject to inherent limitations of interpretability and transparency inherent to NN models, as discussed in prior sections on PINNs.

The incorporation of techniques to quantify uncertainties within predictions may also be advantageous. Uncertainty Quantification (UQ) offers a probabilistic assessment of the model's output predictions and estimated parameters. UQ allows for more informed application, such that the risks posed by deviations as a result of perturbations may be quantified. For instance, probabilistic methods such as Bayesian uncertainty quantification, where physical parameters and resulting predictions of outputs are treated as random variables with associated probability distributions.

The NN may be modified to predict mean response and associated uncertainties, with the range of possible outcomes being outputted as well.

Potential research areas may focus on adding adaptive capabilities to PINNs that are capable of on-line learning, and updating the physical parameters and neural network components as new data becomes available. This may be advantageous in scenarios whereby the operating state of the system changes, conditions such as wear or degradation of components. The implemented model continuously refines predictions with operational context altered, beneficial in fields such as structural health monitoring in which environmental conditions and system loads can vary unpredictably over time.

Transfer learning may be employed to enhance training and convergence speed when adapting to new environments. The model trained in one environment is fine-tuned in another, adapting the parameters to different operating conditions. With system knowledge from the controlled environment as a basis, the speed at which the model adapts and reaches a steady state in foreign environments is accelerated.

Popular in current implementations, ensemble methods are also advantageous in enhancing robustness. In this case, implementations may be PINNs trained under varying conditions of system operation, varying sets of collocation points or varying regions of the physical domain. The ensemble prediction is aggregated to produce a more robust estimate of the physical parameters. This would have the potential to increase resilience to real-world changes for more reliable predictions across a range of conditions.

Overall, while effective and pragmatic, PINNs are still limited by the aforementioned challenges, necessitating ongoing research and development to fully realize

their potential.

Chapter 5

Data Analysis and Modelling

This chapter is dedicated to the construction of a discretized model of state evolution within the MR damper utilizing the parameters identified in the prior chapter, to be used for various estimation and machine-learning tasks downstream. This analysis is relevant in achieving several academic objectives, including the exploration and validation of approaches within the field. The examination and validation of the model with experimental data will provide insights necessary for building a foundational model to be used to extend the application of novel methodologies in estimation, predictive modelling, machine learning, and other relevant research areas.

5.1 Mathematical Model

The modified Bouc-Wen model [136], as detailed in Chapter 2, is employed to represent the MR damper. This model is selected due to its well-established ability to accurately characterize the complex, nonlinear hysteresis behaviour exhibited by MR dampers under varying operational conditions.

In a system with mass m connected to the MR damper and input force $\mu(t)$, the damping force equation 5.1.1 forms the restorative force in the equilibrium condition:

$$m\ddot{x} + c_1\dot{y} + k_1x = \mu(t) \quad (5.1.1)$$

$$m\ddot{x} + \alpha z + c_0(\dot{x} - \dot{y}) + k_0(x - y) + k_1x = \mu(t) \quad (5.1.2)$$

$$\dot{z} = -\gamma |\dot{x} - \dot{y}| z |z|^{n-1} - \beta (\dot{x} - \dot{y}) |z|^n + A(\dot{x} - \dot{y}), \quad (5.1.3)$$

$$\dot{y} = \frac{1}{c_0 + c_1} (\alpha z + k_0(x - y) + c_0\dot{x}). \quad (5.1.4)$$

5.1.1 Discretized form

While the continuous model is essential for understanding the fundamental dynamics, practical applications often require a discrete representation due to the nature of sampled observations being discrete themselves [32]. Thus it is the objective to alter the continuous model into a format suitable for numerical analysis and real-time implementation. This section presents a discretized version of the modified Bouc-Wen state space model, through which, enables the application of various estimation techniques, such as the Kalman filter and its variants [39, 34, 106] and other recursive algorithms, facilitating their integration into broader systems for real-time monitoring and control.

As established in prior sections, the Bouc-Wen model is parametric, involving a set of parameters that characterize the system's behaviour. These parameters,

representing the physical properties of the system, have been identified using the PINN methodology outlined in the prior section: Section 4.1.

The model is discretized using the Euler method to obtain a discrete-time version [140, 32]. Here, T denotes the sampling interval, with k denoting the counter for discrete increments within the sequence. The Euler method approximates the continuous system dynamics by iteratively updating the state variables over discrete time steps defined by T .

The voltage-dependent parameters $\alpha[k]$, $c_0[k]$ and $c_1[k]$ are frequently employed in the evaluation of latent and observable variables within the modified Bouc-Wen equations, and may be calculated via input voltage values, $u[k]$, as:

$$\alpha[k] = \alpha_a + \alpha_b u[k], \quad (5.1.5)$$

$$c_0[k] = c_{0,a} + c_{0,b} u[k], \quad (5.1.6)$$

$$c_1[k] = c_{1,a} + c_{1,b} u[k], \quad (5.1.7)$$

The dynamics of a system may be described with both external and internal displacements, velocities, and hysteresis effects. The external displacement of the system, denoted as $x[k]$, is updated based on the current displacement and the external velocity $v_x[k]$.

$$x[k + 1] = x[k] + T \cdot v_x[k]. \quad (5.1.8)$$

The velocity of the external displacement, $v_x[k]$, is updated based on the current velocity, internal forces, and external influences. The equation is expressed as:

$$v_x[k + 1] = v_x[k] + T \cdot \frac{1}{m} (-c_1[k]v_y[k] - k_1x[k]). \quad (5.1.9)$$

The internal displacement, denoted as $y[k]$, is influenced by several factors, and is incremented per time step based on the derivative of internal displacement defined explicitly in equation 2.2.11.

$$y[k + 1] = y[k] + T \cdot \left(\frac{1}{c_0[k] + c_1[k]} (\alpha[k]z[k] + k_0(x[k] - y[k]) + c_0[k]v_x[k]) \right), \quad (5.1.10)$$

where the derivative of the internal displacement $v_y[k]$ may be abbreviated as:

$$v_y[k] = \frac{1}{c_0[k] + c_1[k]} (\alpha[k]z[k] + k_0(x[k] - y[k]) + c_0[k]v_x[k]). \quad (5.1.11)$$

Altogether, the equation representing the internal displacement may be abbreviated as:

$$y[k + 1] = y[k] + T \cdot v_y[k]. \quad (5.1.12)$$

The hysteresis component $z[k]$ represents the internal frictional and memory effects within the system. Of note, it is dependent upon the calculated value of v_y from equation 5.1.11. Its update rule captures the nonlinear and rate-dependent behaviour of the system:

$$z[k + 1] = z[k] + T \cdot \left(-\gamma |v_x[k] - v_y[k]| z[k] |z[k]|^{n-1} - \beta (v_x[k] - v_y[k]) |z[k]|^n + A (v_x[k] - v[k]) \right). \quad (5.1.13)$$

Similar to above the increment per time step, or the derivative of the hysteretic displacement $v_z[k]$ is previously described, and explicitly defined in equation 2.2.10

$$v_z[k] = -\gamma |v_x[k] - v_y[k]| z[k] |z[k]|^{n-1} - \beta (v_x[k] - v_y[k]) |z[k]|^n + A (v_x[k] - v[k]). \quad (5.1.14)$$

The equation representing the update of the hysteretic displacement may be abbreviated as:

$$z[k + 1] = z[k] + T \cdot v_z[k]. \quad (5.1.15)$$

Finally, the resultant damping force is calculated as a function of previously calculated latent variables and is modelled in accordance with equation 2.2.8, expressed explicitly as:

$$F[k + 1] = \alpha[k]z[k] + c_0[k] (\dot{x}[k] - \dot{y}[k]) + k_0 (x[k] - y[k]) + k_1 (x[k]). \quad (5.1.16)$$

In addressing measurement noise covariance, it is important to note that variables y and z , and by extension their derivatives with respect to time, are not physically measurable, as they do not correspond to any tangible values due to the phenomenological nature of the model. Consequently, they are excluded from the noise covariance matrix. Thus, in this case, the states of concern are: x , v_x , and F .

The system dynamics, being relatively well-known and predictable compared to the uncertainties in measurements, are assumed to have an error covariance an order of magnitude smaller, consistent with previous studies [75, 72]. The measurement

error covariance matrix is structured as follows:

$$Q = R \cdot 10^{-1} \quad (5.1.17)$$

$$R = \begin{bmatrix} 5.5134 \cdot 10^{-4} & 0 & 0 \\ 0 & 7.797 \cdot 10^{-4} & 0 \\ 0 & 0 & 22.41 \end{bmatrix} \quad (5.1.18)$$

The measurement noise covariance matrix R represents the uncertainties associated with the sensor readings, arising from various sources: including sensor inaccuracies, environmental factors, and other external influences. Factory testing quantified these uncertainties, allowing for the construction of covariance matrix R . The system noise covariance matrix, denoted by Q , was not directly measured during the testing process but was rather assumed to be an order of magnitude smaller than the measurement noise covariance matrix, R . This assumption is grounded in the standard practice that system noise typically exhibits lower variability compared to measurement noise in similar contexts, as demonstrated in the work of Lee [72].

5.2 Simulation

For future test cases involving condition monitoring, the modified Bouc-Wen model state space was constructed in three distinct phases. These phases represent different scenarios of the applied voltage to the MR damper experimental setup, specifically the default case, undervoltage fault, and overvoltage fault.

Parameter Name	Parameter (Nominal)	Parameter (Undervoltage)	Parameter (Overvoltage)	Units
α	4.7299e+01	2.2189e+01	7.2408e+01	$\frac{kN}{m}$
β	8.2089e+03	8.2089e+03	8.2089e+03	m^{-2}
γ	8.1997e+03	8.1997e+03	8.1997e+03	m^{-2}
A	2.3086e+01	2.3086e+01	2.3086e+01	—
c_0	2.3310e+01	1.9809e+01	2.6811e+01	$\frac{kN \cdot s}{m}$
c_1	8.8537e+00	3.6047e+00	1.4103e+01	$\frac{kN \cdot s}{m}$
k_0	9.7866e-01	9.7866e-01	9.7866e-01	$\frac{kN}{m}$
k_1	1.4646e-01	1.4646e-01	1.4646e-01	$\frac{kN}{m}$
n	1.9928e+00	1.9928e+00	1.9928e+00	—

Table 5.1: List of parameters and their corresponding values, as determined by the PINN parameter estimation algorithm outlined in Section 4.1.

For the first phase, representing the default, the MR damper operates under standard voltage conditions and characterizes the MR damper’s performance in the absence of voltage anomalies. The parameters associated with this phase reflect the typical operational state and act as a reference point by which other phases may be compared against. The undervoltage fault phase simulates conditions where the applied voltage is lower than the standard operating voltage, while the overvoltage fault phase explores the effects of an excessive voltage supply.

The parameters for each phase are listed below and serve as the basis for simulations using the above state space model.

From the parameters estimated in section 4.1, the force response is simulated in accordance. The models depicted in the figure illustrate the force-position hysteresis relationship of the MR damper at a velocity of 0.903 cm/s, which illustrates the

dynamic response of the system at varying phases: starting with the default state, followed by an undervoltage fault, and ending with overvoltage fault.

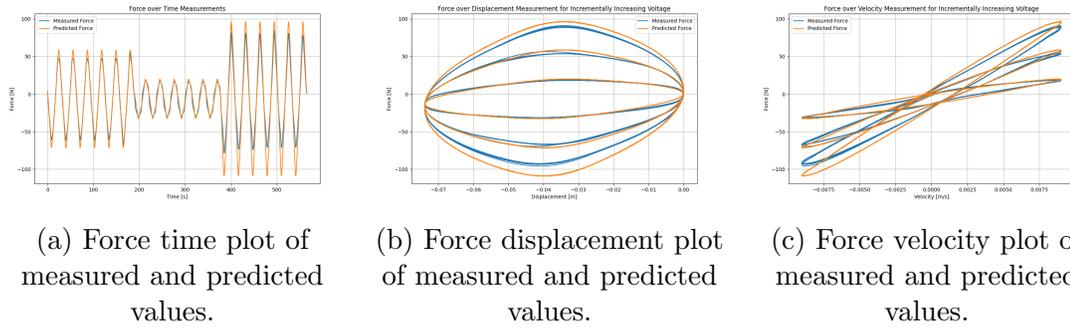


Figure 5.1: Overall plots of measured behaviour, contrasted with predicted force response utilizing parameters identified, consisting of plots of force response with respect to time, displacement, and velocity at varying applied voltages, representing the nominal state, undervoltage fault state, and overvoltage fault state.

The linear actuator drove the MR damper for a total of 565.48 seconds for 6 cycles in each phase, with a constant acceleration of 0.1354 cm/s during both the extension and retraction phases. The position and velocity profiles captured by the actuator encoder, as shown in the figures, provide a detailed account of the MR damper’s movement. Initially, a voltage of 2.2703 V was applied to the MR damper, representing normal operation. During this phase, the MR damper was allowed to fully extend and retract, establishing a baseline for its performance.

At 190.79 seconds, an undervoltage fault was introduced by decreasing the voltage to 0.6678 V. After completing a total of 6 full cycles of motion, an overvoltage fault was introduced at 378.98 seconds by increasing the applied voltage to 4.0030 V.

Throughout the test, three modes (normal, overvoltage, and undervoltage) were experienced, allowing for a range for analysis of the MR damper’s behaviour under varying applied voltage and fault detection conditions. The results from these tests

and simulations may be employed in estimation for determining the operational limits and for developing and applying strategies for condition monitoring and anomaly detection.

The characterization of the modified Bouc-Wen model and its application across different voltage scenarios provides a mode of analysis for the operational behaviour of MR dampers. Validation against measured values is performed and reinforces the model's fidelity in capturing the dynamics of the damper system under normal, overvoltage, and undervoltage conditions.

Looking ahead, this model may serve as a foundation for future research and development efforts aimed at demonstrating proof of concept for various novel machine learning and estimation techniques, such as the sliding innovation filter introduced in [33], the smooth variable structure filter [71, 74], the Reinforced Lattice Kalman Filter [107] and other innovations. In conjunction with the experimental setup, outlined in Chapter 3, the discretized model will serve as a testing setup for implementing the aforementioned strategies in various engineering and condition-monitoring applications, whereby real-time monitoring and fault detection methodologies involving the state space developed are employed for proactive maintenance in diverse operational environments [88, 14, 15].

Chapter 6

Conclusion

This study presented the development and implementation of PINNs, with application to a mechanical system. The study details the construction of an MR damper experimental setup for the collection of empirical data. Utilizing data gathered from the experimental setup, an implementation of PINNs for system identification and modelling was formulated for the MR damper.

The experimental setup constructed integrated the various devices required for actuation and sensing, synchronizing and storing collected data systematically. Data collection from various devices were unified through a custom implementation for data collection, featuring the collection, processing, and storage of readings through multi-threading operations within a central user interface. Data collected was stored in a locally hosted database in Structured Query Language. Stored data is employed in various downstream tasks, primarily serving to validate the effectiveness of novel strategies utilizing machine learning and estimation, such as the system identification and modelling procedures introduced within this research. In addition, the data collected supports future research endeavours in the domain as well.

An issue addressed in this study is the identification of model parameters in MR dampers, which involve latent variables. A system identification methodology was developed around the growing paradigm of PINNs, aimed at extending its capabilities in solving inverse problems to a system of ODEs with unobservable responses. This implementation demonstrates the feasibility of this approach, being the introduction of learning biases through the integration of established physics with the NN training process. Through validation, this methodology has been shown to have accurate estimation capabilities for physical parameters and is capable of reconstructing the observed forced response from inputs.

Based on the collected data and the identified parameters, a discrete state-space model of the MR damper was developed. The state-space model constructed provides a framework for describing the behaviour of the dynamic system in nominal and faulty operating conditions and sets a foundation for future implementations of novel estimation and machine-learning techniques.

Overall, the study’s findings highlight the potential of physics-informed neural networks in the system identification of complex systems with latent variables.

6.1 Future Work

Future research directions are aimed at exploring the further refinement of PINNs as well as extending their application to encompass a larger range of engineering systems.

While the PINN represents a pragmatic approach to the solution of inverse problems, they are still limited by various challenges involving high dimensional data,

inability to deal with multiscale data and sensitivity to how known physics is enforced, necessitating ongoing research. As detailed in the section 4.2.1, at its core, PINNs still suffer from the many limitations associated with conventional deep learning, which forms its backbone. Techniques such as manifold learning, equation-free approach, and random projections hold promise in further extending the capabilities of PINNs. An interesting avenue of research is the incorporation of the prominent transformer model, incorporating the attention mechanism, enabling effective handling of temporal dependencies inherent in physical systems. which may improve the performance of the system identification process.

Beyond its application to MR dampers, future research can extend the developed methodologies to analyze and model other mechanical systems, with the potential to develop methodologies applicable to a wider range of engineering applications.

Other directions for future work may be extending developed PIML methodologies condition monitoring techniques. As discussed in section 2, PINNs have already shown significant potential in various facets of condition monitoring, for example: estimating remaining useful life to anomaly detection, fault prognosis, and predictive maintenance.

Overall, these future efforts aim to build upon the foundational work presented in this study, driving the development of more sophisticated, efficient, and practical solutions for the analysis of dynamic systems and monitoring.

6.2 Summary of Contributions

The overall contributions of this work may be summarized as follows:

Experimental Setup Development and Data Collection

- Constructed the physical experimental setup with an MR damper.
- Developed a unified data collection program for a multi-device system.
- Ensured synchronization and storage of data through multi-threading.
- Implemented database for centralized access and analysis of collected data.

Application of PINN method

- Performed literature review, with a focus on the application of PINNs in real-world engineering processes.
- Developed methodologies for system identification of MR dampers, focusing on models with latent variables.
- Developed operation and workflow of PINNs
- Employed PINNs to solve the inverse problem, optimizing model and physical parameters simultaneously to circumvent issues with latent variables.

Validation and Model Development

- Validated the developed methodologies using data gathered from the experimental setup.
- Constructed a discrete state-space model of the MR damper based on identified parameters, setting up the foundation for validating techniques developed in future work.

Appendix A

Compilation of Literature Review Articles

Article Title	Citation	Description	Application
Microcrack Defect Quantification Using a Focusing High-Order SH Guided Wave EMAT: The Physics-Informed Deep Neural Network GuwNet	[141]	Quantification of microcrack defects with hybrid physics-informed architecture design based on various deep learning frameworks, regularized via network structure and hybrid feed-forward and back-propagation loss	structural health monitoring for detection of micro-crack defects

Article Title	Citation	Description	Application
Physics-informed turbulence intensity infusion: A new hybrid approach for marine current turbine rotor blade fault detection	[31]	Feature extraction via continuous wavelet transform from vibrational data. The classification was performed with a neural network, with physics-informed loss function to obtain turbulence intensity features	Anomalous behavior detection and fault diagnosis in turbine rotor blades
A physics-informed neural network for creep-fatigue life prediction of components at elevated temperatures	[173]	Neural network regularized via physics-informed loss function, penalizing the model for unrealistic predictions (negative or extreme values) of fatigue life	Structural health monitoring for creep-fatigue life in steel specimen

Article Title	Citation	Description	Application
Data-driven prognos- tics with low-fidelity physical informa- tion for digital twin: physics-informed neu- ral network	[61]	Physics-informed loss func- tion penalizing deviations from expected values, de- termined by low-fidelity physical model	Structural health monitoring for crack propagation
Long-term fatigue es- timation on offshore wind turbines inter- face loads through loss function physics- guided learning of neural networks	[23]	Features selected through recursive feature elimi- nation from sensors and monitoring data. Estima- tion of fatigue via neural network regularized by novel physics-informed loss function, reflective of pri- ority given to long-term estimation	Structural health monitoring for wind turbines fatigue life

Article Title	Citation	Description	Application
Physics-informed meta-learning for machining tool wear prediction	[79]	Parameters of dynamic relationships governing tool wear used to establish input space for individual models at different stages of degradation via cross physics-data fusion. Meta-learning model is employed to learn the experiences of ML models and optimized via physics-informed loss	Tool life predictions
A physics-informed deep learning framework for inversion and surrogate modelling in solid mechanics	[41]	Physics-informed neural network for solving differential equations governing linear elasticity and non-linear von Mises elastoplasticity	Elastostatics modelling in solid mechanics

Article Title	Citation	Description	Application
Identification of Material Parameters from Full-Field Displacement Data Using Physics-Informed Neural Networks	[4]	Material parameter estimation via solution of momentum equation and governing equations of linear elasticity	Structural health monitoring
Inferring vortex induced vibrations of flexible cylinders using physics-informed neural networks	[60]	Approximation of the linear beam-string equations via PINN for simulation of a cylindrical structure in uniform flow	Structural health monitoring
Physics-Informed Machine Learning and Uncertainty Quantification for Mechanics of Heterogeneous Materials	[10]	Solution of PDE governing momentum balance and constitutive equations of elasticity, optimized via physics-informed loss function penalizing deviations from PDE and boundary conditions	Surrogate modeling of elastic deformations

Article Title	Citation	Description	Application
Simulation of guided waves for structural health monitoring using physics-informed neural networks	[113]	Solving PDEs governing wave propagation with PINNs, regularized by physics-informed loss function based on deviations from PDEs and boundary conditions	Structural health monitoring in aerospace structures
A physically consistent framework for fatigue life prediction using probabilistic physics-informed neural network	[176]	Probabilistic PINN optimized via hybrid loss function based on fatigue life distributions with respect to stress experienced	State of health monitoring and fatigue life estimation
A robust physics-informed neural network approach for predicting structural instability	[86]	Feed-forward PINN optimized based on deviation from data, instability information, and boundary conditions	Structural health monitoring via estimation of structural instability

Article Title	Citation	Description	Application
Machine Fault Classification using Hamiltonian Neural Networks	[127]	PINN encoding the laws of Hamiltonian mechanics to learn operating state of system from vibrational data, machinery state identification using network parameters as features	Machinery fault diagnosis for rotating machinery
Physics-informed machine learning for surrogate modeling of wind pressure and optimization of pressure sensor placement	[178]	Finite element based computational fluid dynamics model for the generation of input features, PINN employed for the solution to Navier–Stokes equations of incompressible flows, with Dirichlet and Neumann boundary conditions	Structural health monitoring in buildings

Article Title	Citation	Description	Application
Physics informed neural network for health monitoring of an air preheater	[48]	Stacked PINNs for solving non-denationalized governing equations for heat transfer between the fluids and metal interface, regularized by physics-informed loss function based on deviation from PDEs, boundary and interface conditions	Condition monitoring and health monitoring in air heating system
Robust Regression with Highly Corrupted Data via Physics Informed Neural Networks	[103]	Feed-forward PINN based on the least absolute deviation method to reconstruct PDE solutions and parameters from highly corrupt sensor data	Corrupt data and parameter reconstruction

Article Title	Citation	Description	Application
A generic physics-informed neural network-based framework for reliability assessment of multi-state systems	[177]	Feed-Forward PINN regularized by deviations from ODE of system state transition and initial conditions. Individual element of the loss parse through projecting conflicting gradients to establish continuous latent function for reliability assessment	Reliability assessment
Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling	[172]	Physics-Informed Loss (Dynamic System with Ground Excitation)	Structural Health Monitoring
A physics-informed deep learning approach for bearing fault detection	[128]	Physics-Informed Loss (Deviation from Physics-Based Threshold Model Penalized)	Machinery fault detection and diagnosis in bearings

Article Title	Citation	Description	Application
Physics-guided deep neural network for structural damage identification	[46]	CNN employed as feature extraction for both the physics and data domain. The network was regularized in accordance to labelled data as well as the objective of minimizing feature discrepancy between domains	Structural health monitoring in bridge structures
Bridge damage identification under the moving vehicle loads based on the method of physics-guided deep neural networks	[165]	Physics-informed loss function for feature fusion between the physics-based numerical model and data-driven model)	Structural health monitoring in bridge structures

Article Title	Citation	Description	Application
A physics-informed convolutional neural network with custom loss functions for porosity prediction in laser metal deposition	[90]	Physics-informed CNN with loss function penalizing deviations from ideal simulated parameters)	Process monitoring in additive manufacturing for porosity buildup)
Physics-Informed Learning for High Impedance Faults Detection	[78]	Physics-informed convolutional autoencoder, with physics-informed regularization based on elliptical relation characteristics of voltage and current plots	Fault detection in power grids
Physics-informed deep learning for signal compression and reconstruction of big data in industrial condition monitoring	[119]	Physics-informed convolutional autoencoder, featuring loss term incorporating auto-correlation and Fast Fourier Transform metrics	Data compression for collected monitoring signatures in machinery fault detection and diagnosis

Article Title	Citation	Description	Application
Physics guided neural network for machining tool wear prediction	[150]	Cross physics-data fusion for the integration of physical parameters within model input. Physics-informed loss function employed to enforce relationship between tool degradation with respect to operation progress	Condition monitoring for tool wear
A Novel Physics-Informed Framework for Real-Time Adaptive Monitoring of Offshore Structures	[81]	Employed a physics-informed RNN for solution to governing equations of eigensystem, representative of the modal identification process	Structural health monitoring

Article Title	Citation	Description	Application
Physics-Informed LSTM hyperparameters selection for gearbox fault detection	[19]	Maximization of Mahalanobis distance between healthy state and established physics-informed fault state for LSTM optimization process	Machinery fault diagnosis in gear boxes

Table A.1: Literature Compiled for physics-guided or physics-informed regularisation technique employed

Appendix B

Components and Specifications

This appendix section specifies the technical and operating parameters of the physical components detailed in Section 3, which describes the experimental setup. Here, details on the exact technical specifications for each component used in the experiment are provided.

B.0.1 RD-8041-1 MR Damper

Technical Specifications for the LORD RD-8041-1 MR Damper are provided by LORD Corporation, available at [20]. Typical and electrical properties are summarized in Tables B.1 and B.2 respectively.

Properties	Value
Stroke length	74 [mm]
Extended Length	248 [mm]
Body Diameter	42.1 [mm] max
Shaft Diameter	10 [mm]
Tensile Strength	8896 [N] max
Peak to Peak Damper Forces 5 cm/sec at 1 A	> 2447 [N]
Peak to Peak Damper Forces 5 cm/sec at 1 A	< 667 [N]
Operating Temperature	71 [°C] max

Table B.1: Typical properties of the LORD RD-8041-1 MR damper, as specified by [20].

Properties	Value
Input Current: Continuous for 30s	1 [A] max
Input Current: Intermittent	2 [A] max
Input Voltage	12 [V]
Resistance at ambient temperature	5 [Ω]
Resistance at maximum operating Temperature (71 [°C])	7 [Ω] max

Table B.2: Electrical properties of the LORD RD-8041-1 MR damper, as specified by [20].

B.0.2 RAS1-500S-S Force Sensor

Properties	Value
Operating Force	226.80 [kgf]
Safe Overload	340.2 [kgf], 680.4 [kgf] (Ultimate Overload)
Accuracy	± 0.02 %
Rated Excitation	5-10 [V]
Full Scale Output	3 mv/V ± 10 %
Input Impedance	$385 \pm 30 \Omega$
Output Impedance	$350 \pm 5 \Omega$
Insulation	$> 5,000 M\Omega$
Creep, 1 hour	± 0.05 %
Operating Temperature Range	$350 \pm 5 \Omega$
Compensated Temperature Range	$350 \pm 5 \Omega$
Output Impedance	$350 \pm 5 \Omega$
Output Impedance	$350 \pm 5 \Omega$

Table B.3: Properties of the RAS1-500S-S Force Sensor, as specified by [124].

B.0.3 Ultramotion A1 Linear Actuator

Properties	Value
Force (peak)	1512 [<i>N</i>] max
Force (continuous)	756 [<i>N</i>] max
Speed	356 [<i>mm/s</i>] max
Stroke length	76.2 [<i>mm</i>]
Resolution	3.1 [μ <i>m</i>]
Operating Voltage	8 to 36 [<i>VDC</i>]
Operating Temperature	-40 [$^{\circ}$ <i>C</i>] to 71 [$^{\circ}$ <i>C</i>]

Table B.4: properties of the Ultramotion A1 linear actuator, as specified by [92].

Bibliography

- [1] M. Abdul Aziz, S. M. Mohtasim, and R. Ahammed. State-of-the-art recent developments of large magnetorheological (mr) dampers: a review. *Korea-Australia Rheology Journal*, 34(2):105–136, 2022.
- [2] N. Alsadi, W. Hilal, O. Surucu, A. Giuliano, S. A. Gadsden, J. Yawney, and M. A. AlShabi. Neural network training loss optimization utilizing the sliding innovation filter. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, volume 12113, pages 577–589. SPIE, 2022.
- [3] W. Ang, W. Li, and H. Du. Experimental and modeling approach of a mr damper performance under harmonic loading. *Journal of the Institution of Engineers*, 44(4):1–14, 2004.
- [4] D. Anton and H. Wessels. Identification of material parameters from full-field displacement data using physics-informed neural networks. *Researchgate Preprint*, 2021.
- [5] E. A. Antonelo, E. Camponogara, L. O. Seman, J. P. Jordanou, E. R. de Souza, and J. F. Hübner. Physics-informed neural nets for control of dynamical systems. *Neurocomputing*, 579:127419, 2024.

- [6] A. Ashfak, A. Saheed, K. A. Rasheed, and J. A. Jaleel. Design, fabrication and evaluation of mr damper. *International Journal of Aerospace and Mechanical Engineering*, 1:27–33, 2011.
- [7] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43, 2018.
- [8] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.
- [9] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.
- [10] B. Bharadwaja, M. A. Nabian, B. Sharma, S. Choudhry, and A. Alankar. Physics-informed machine learning and uncertainty quantification for mechanics of heterogeneous materials. *Integrating Materials and Manufacturing Innovation*, pages 1–21, 2022.
- [11] R. Bouc. A mathematical model for hysteresis. *Acta Acustica united with Acustica*, 24(1):16–25, 1971.
- [12] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- [13] S. Burbulla. Physics-informed neural networks for transformed geometries and manifolds. *arXiv preprint arXiv:2311.15940*, 2023.

- [14] R. Bustos, S. A. Gadsden, M. Biglarbegian, M. AlShabi, and S. Mahmud. Battery state of health estimation using the sliding interacting multiple model strategy. *Energies*, 17(2):536, 2024.
- [15] Q. Butler, Y. Ziada, D. Stephenson, and S. Andrew Gadsden. Condition monitoring of machine tool feed drives: A review. *Journal of Manufacturing Science and Engineering*, 144(10):100802, 2022.
- [16] Q. Butler, B. Sicard, W. Hilal, S. A. Gadsden, and Y. Ziada. Nonlinear filtering using the double exponential transformation. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXXIII*, volume 13057, pages 16–28. SPIE, 2024.
- [17] R. H. Carver and K.-C. Tai. *Modern multithreading: implementing, testing, and debugging multithreaded Java and C++/Pthreads/Win32 programs*. John Wiley & Sons, 2005.
- [18] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express*, 28(8):11618–11633, 2020.
- [19] Y. Chen, M. Rao, K. Feng, and M. J. Zuo. Physics-informed lstm hyperparameters selection for gearbox fault detection. *Mechanical Systems and Signal Processing*, 171:108907, 2022.
- [20] L. Corporation. Lord-MAR — shoplordmr.com. <https://www.shoplordmr.com/mr-products/rd-8041-1-mr-damper-long-stroke>. [Accessed 05-06-2024].

- [21] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.
- [22] P. R. Dahl. Solid friction damping of mechanical vibrations. *AIAA journal*, 14(12):1675–1682, 1976.
- [23] F. De Santos, P. D'Antuono, K. Robbelein, N. Noppe, W. Weijtjens, and C. Devriendt. Long-term fatigue estimation on offshore wind turbines interface loads through loss function physics-guided learning of neural networks. *Renewable Energy*, 205:461–474, 2023.
- [24] C. C. De Wit, H. Olsson, K. J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on automatic control*, 40(3):419–425, 1995.
- [25] A. Dominguez, R. Sedaghati, and I. Stiharu. A new dynamic hysteresis model for magnetorheological dampers. *Smart materials and structures*, 15(5):1179, 2006.
- [26] G. Fabiani, E. Galaris, L. Russo, and C. Siettos. Parsimonious physics-informed random projection neural networks for initial value problems of odes and index-1 daes. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(4), 2023.
- [27] S. A. Faroughi, N. M. Pawar, C. Fernandes, M. Raissi, S. Das, N. K. Kalantari, and S. Kouros Mahjour. Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid

- mechanics. *Journal of Computing and Information Science in Engineering*, 24(4):040802, 2024.
- [28] M. Feurer and F. Hutter. Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, pages 3–33, 2019.
- [29] E. Forbes. *Learning Concurrency in Python*. Packt Publishing Ltd, 2017.
- [30] P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.
- [31] B. Freeman, Y. Tang, Y. Huang, and J. VanZwieten. Physics-informed turbulence intensity infusion: A new hybrid approach for marine current turbine rotor blade fault detection. *Ocean Engineering*, 254:111299, 2022.
- [32] B. Friedland. *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.
- [33] S. A. Gadsden and M. Al-Shabi. The sliding innovation filter. *IEEE Access*, 8:96129–96138, 2020.
- [34] S. A. Gadsden, S. Habibi, and T. Kirubarajan. Kalman and smooth variable structure filters for robust estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):1038–1050, 2014.
- [35] E. Galaris, G. Fabiani, I. Gallos, I. Kevrekidis, and C. Siettos. Numerical bifurcation analysis of pdes from lattice boltzmann model simulations: a parsimonious machine learning approach. *Journal of Scientific Computing*, 92(2):34, 2022.

- [36] D. Gamota and F. Filisko. Dynamic mechanical studies of electrorheological materials: moderate frequencies. *Journal of rheology*, 35(3):399–425, 1991.
- [37] A. Giuliano, S. A. Gadsden, W. Hilal, and J. Yawney. Convolutional variational autoencoders for secure lossy image compression in remote sensing. In *Sensors and Systems for Space Applications XVII*, volume 13062, pages 124–134. SPIE, 2024.
- [38] A. Gracyk. Ricci flow-guided autoencoders in learning time-dependent dynamics. *arXiv preprint arXiv:2401.14591*, 2024.
- [39] M. S. Grewal and A. P. Andrews. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.
- [40] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.
- [41] E. Haghghat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379: 113741, 2021.
- [42] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [43] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [44] W. Hu and N. M. Wereley. Hybrid magnetorheological fluid–elastomeric lag

- dampers for helicopter stability augmentation. *Smart Materials and Structures*, 17(4):045021, 2008.
- [45] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao. Normalization techniques in training dnns: Methodology, analysis and application. *IEEE transactions on pattern analysis and machine intelligence*, 45(8):10173–10196, 2023.
- [46] Z. Huang, X. Yin, and Y. Liu. Physics-guided deep neural network for structural damage identification. *Ocean Engineering*, 260:112073, 2022.
- [47] F. Ikhouane and S. J. Dyke. Modeling and identification of a shear mode magnetorheological damper. *Smart Materials and Structures*, 16(3):605, 2007.
- [48] V. Jadhav, A. Deodhar, A. Gupta, and V. Runkana. Physics informed neural network for health monitoring of an air preheater. In *PHM Society European Conference*, volume 7, pages 219–230, 2022.
- [49] A. D. Jagtap, Z. Mao, N. Adams, and G. E. Karniadakis. Physics-informed neural networks for inverse problems in supersonic flows. *Journal of Computational Physics*, 466:111402, 2022.
- [50] L. M. Jansen and S. J. Dyke. Semiactive control strategies for mr dampers: comparative study. *Journal of engineering mechanics*, 126(8):795–803, 2000.
- [51] X. Jia, J. Willard, A. Karpatne, J. Read, J. Zwart, M. Steinbach, and V. Kumar. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 558–566. SIAM, 2019.

- [52] R. Jiménez and L. Álvarez-Icaza. Lugre friction model for a magnetorheological damper. *Structural Control and Health Monitoring*, 12(1):91–116, 2005.
- [53] R. Jimnez and L. Alvarez. Real time identification of structures with magnetorheological dampers. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 1, pages 1017–1022. IEEE, 2002.
- [54] G. M. Kamath, N. M. Wereley, and M. R. Jolly. Characterization of magnetorheological helicopter lag dampers. *Journal of the American Helicopter Society*, 44(3):234–248, 1999.
- [55] J. Karandikar, T. Schmitz, and S. Smith. Physics-guided logistic classification for tool life modeling and process parameter optimization in machining. *Journal of Manufacturing Systems*, 59:522–534, 2021.
- [56] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [57] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.
- [58] I. G. Kevrekidis and G. Samaey. Equation-free multiscale computation: Algorithms and applications. *Annual review of physical chemistry*, 60:321–344, 2009.

- [59] I. G. Kevrekidis, C. W. Gear, and G. Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal*, 50(7):1346–1355, 2004.
- [60] E. Kharazmi, D. Fan, Z. Wang, and M. S. Triantafyllou. Inferring vortex induced vibrations of flexible cylinders using physics-informed neural networks. *Journal of Fluids and Structures*, 107:103367, 2021.
- [61] S. Kim, J.-H. Choi, and N. H. Kim. Data-driven prognostics with low-fidelity physical information for digital twin: physics-informed neural network. *Structural and Multidisciplinary Optimization*, 65(9):255, 2022.
- [62] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [63] L. Korad Technology CO. KD SERIES_DONGGUAN KORAD TECHNOLOGY CO., LTD..Programmable DC Power Supplies and Electronic Loads — koradtechnology.com. <https://www.koradtechnology.com/product/84.html#>. [Accessed 07-06-2024].
- [64] C. Krishnanunni and T. Bui-Thanh. Layerwise sparsifying training and sequential learning strategy for neural architecture adaptation. *arXiv preprint arXiv:2211.06860*, 2022.
- [65] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

- [66] S. K. Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017.
- [67] N. Kwok, Q. Ha, T. Nguyen, J. Li, and B. Samali. A novel hysteretic model for magnetorheological fluid dampers and parameter identification using particle swarm optimization. *Sensors and Actuators A: Physical*, 132(2):441–451, 2006.
- [68] N. Kwok, Q. Ha, M. Nguyen, J. Li, and B. Samali. Bouc–wen model parameter identification for a mr fluid damper using computationally efficient ga. *ISA transactions*, 46(2):167–179, 2007.
- [69] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.
- [70] A. Lee and S. A. Gadsden. Application and comparison of nonlinear estimation strategies on a magnetorheological hysteretic force model.
- [71] A. Lee, S. A. Gadsden, and S. A. Wilkerson. An adaptive smooth variable structure filter based on the static multiple model strategy. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII*, volume 11018, pages 408–417. SPIE, 2019.
- [72] A. S. Lee. *Sliding innovation filtering: Theory and applications*. PhD thesis, University of Guelph, 2021.

- [73] A. S. Lee, S. A. Gadsden, and M. Al-Shabi. Application of nonlinear estimation strategies on a magnetorheological suspension system with skyhook control. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–6. IEEE, 2020.
- [74] A. S. Lee, S. A. Gadsden, S. A. Wilkerson, and M. AlShabi. An adaptive formulation of the smooth variable structure filter based on static multiple models. *Int. J. Robot. Autom.*, 38:1–10, 2023.
- [75] A. S. Lee, Y. Wu, S. A. Gadsden, and M. AlShabi. Interacting multiple model estimators for fault detection in a magnetorheological damper. *Sensors*, 24(1):251, 2023.
- [76] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, 2020.
- [77] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296, 2006.
- [78] W. Li and D. Deka. Physics-informed learning for high impedance faults detection. In *2021 IEEE Madrid PowerTech*, pages 1–6. IEEE, 2021.
- [79] Y. Li, J. Wang, Z. Huang, and R. X. Gao. Physics-informed meta learning for machining tool wear prediction. *Journal of Manufacturing Systems*, 62:17–27, 2022.
- [80] P. Lischinsky, C. Canudas-de Wit, and G. Morel. Friction compensation for an industrial hydraulic robot. *IEEE Control Systems Magazine*, 19(1):25–32, 1999.

- [81] F. Liu, Q. Yu, H. Song, X. Li, L. Liu, and D. Liu. A novel physics-informed framework for real-time adaptive monitoring of offshore structures. *Available at SSRN 4349124*, 2023.
- [82] T. Liu and H. Meidani. Physics-informed neural networks for system identification of structural systems with a multiphysics damping model. *Journal of Engineering Mechanics*, 149(10):04023079, 2023.
- [83] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.
- [84] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM review*, 63(1):208–228, 2021.
- [85] A. L’heureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz. Machine learning with big data: Challenges and approaches. *Ieee Access*, 5:7776–7797, 2017.
- [86] H. T. Mai, T. T. Truong, J. Kang, D. D. Mai, and J. Lee. A robust physics-informed neural network approach for predicting structural instability. *Finite Elements in Analysis and Design*, 216:103893, 2023.
- [87] N. Matloff and F. Hsu. Tutorial on threads programming with python. *University of California*, 2007.
- [88] A. McCafferty-Leroux, W. Hilal, S. A. Gadsden, and M. Al-Shabi. Adaptive sif-kf estimation for fault detection in attitude control experiments. In *Sensors*

- and Systems for Space Applications XVII*, volume 13062, pages 181–195. SPIE, 2024.
- [89] A. McCafferty-Leroux, A. Newton, and S. A. Gadsden. Parameter estimation and control of an automatic balancing system for cubesat research and applications. In *Sensors and Systems for Space Applications XVII*, volume 13062, pages 196–214. SPIE, 2024.
- [90] E. McGowan, V. Gawade, and W. Guo. A physics-informed convolutional neural network with custom loss functions for porosity prediction in laser metal deposition. *Sensors*, 22(2):494, 2022.
- [91] C. Meng, S. Seo, D. Cao, S. Griesemer, and Y. Liu. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*, 2022.
- [92] U. Motion. Servo cylinder configurator; ultra motion — ultramotion.com. <https://www.ultramotion.com/servo-cylinder-configurator/>. [Accessed 07-06-2024].
- [93] K. P. Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.
- [94] M. Nemirovsky and D. Tullsen. *Multithreading architecture*. Springer Nature, 2022.
- [95] B. Nguyen. *Design, Development, and Experimental Validation of a Magnetorheological Damper Test Bench*. PhD thesis, University of Guelph, 2022.

- [96] A. Occhiuzzi, M. Spizzuoco, and G. Serino. Experimental analysis of magnetorheological dampers for structural control. *Smart Materials and Structures*, 12(5):703, 2003.
- [97] H.-U. Oh and J. Onoda. An experimental study of a semiactive magnetorheological fluid variable damper for vibration suppression of truss structures. *Smart Materials and Structures*, 11(1):156, 2002.
- [98] E. Oware, S. Moysey, and T. Khan. Physically based regularization of hydrogeophysical inverse problems for improved imaging of process-driven systems. *Water Resources Research*, 49(10):6238–6247, 2013.
- [99] Y. Ozaki, Y. Tanigaki, S. Watanabe, and M. Onishi. Multiobjective tree-structured parzen estimator for computationally expensive optimization problems. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 533–541, 2020.
- [100] L. Pagnier and M. Chertkov. Physics-informed graphical neural network for parameter & state estimations in power systems. *arXiv preprint arXiv:2102.06349*, 2021.
- [101] L. Pang, G. Kamath, and N. Wereley. Analysis and testing of a linear stroke magnetorheological damper. In *39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, page 2040, 1998.
- [102] S. Patro and K. K. Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

- [103] W. Peng, W. Yao, W. Zhou, X. Zhang, and W. Yao. Robust regression with highly corrupted data via physics informed neural networks. *arXiv preprint arXiv:2210.10646*, 2022.
- [104] W. Peng, W. Zhou, X. Zhang, W. Yao, and Z. Liu. Rang: A residual-based adaptive node generation method for physics-informed neural networks. *arXiv preprint arXiv:2205.01051*, 2022.
- [105] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.
- [106] A. Rahimnejad, S. A. Gadsden, and M. Al-Shabi. Lattice kalman filters. *IEEE Signal Processing Letters*, 28:1355–1359, 2021.
- [107] A. Rahimnejad, J. Enayati, L. Vanfretti, S. A. Gadsden, and M. AlShabi. Reinforced lattice kalman filters: A robust nonlinear estimation strategy. *IEEE Open Journal of Signal Processing*, 2023.
- [108] M. Rahman, Z. C. Ong, S. Julai, M. M. Ferdous, and R. Ahamed. A review of advances in magnetorheological dampers: their design optimization and applications. *J. Zhejiang Univ. Sci. A*, 18(12):991–1010, 2017.
- [109] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.
- [110] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems

- involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [111] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.
- [112] R. B. Randall. *Vibration-based condition monitoring: industrial, automotive and aerospace applications*. John Wiley & Sons, 2021.
- [113] M. Rautela, M. RAUT, and S. Gopalakrishnan. Simulation of guided waves for structural health monitoring using physics-informed neural networks. *STRUCTURAL HEALTH MONITORING 2021*, 2021.
- [114] S. J. Raymond and D. B. Camarillo. Applying physics-based loss functions to neural networks for improved generalizability in mechanics problems. *arXiv preprint arXiv:2105.00075*, 2021.
- [115] A. J. Roberts, T. Tran-Duc, J. E. Bunder, and Y. Kevrekidis. Accurate and efficient multiscale simulation of a heterogeneous elastic beam via computation on small sparse patches. *ANZIAM Journal*, 64:C161–C177, 2022.
- [116] A. Rossi, F. Orsini, A. Scorza, F. Botta, N. P. Belfiore, and S. A. Sciuto. A review on parametric dynamic models of magnetorheological dampers and their characterization methods. In *Actuators*, volume 7, page 16. MDPI, 2018.
- [117] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

- [118] P. Ruhnau, A. Stahl, and C. Schnörr. Variational estimation of experimental fluid flows with physics-based spatio-temporal regularization. *Measurement Science and Technology*, 18(3):755, 2007.
- [119] M. Russell and P. Wang. Physics-informed deep learning for signal compression and reconstruction of big data in industrial condition monitoring. *Mechanical Systems and Signal Processing*, 168:108709, 2022.
- [120] İ. Şahin, T. Engin, and Ş. Çeşmeci. Comparison of some existing parametric models for magnetorheological fluid dampers. *Smart materials and structures*, 19(3):035012, 2010.
- [121] C. Sakai, H. Ohmori, and A. Sano. Modeling of mr damper with hysteresis for adaptive vibration control. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 4, pages 3840–3845. IEEE, 2003.
- [122] F. M. Salem. *Recurrent Neural Networks*. Springer, 2022.
- [123] A. M. Schafer and H. G. Zimmermann. Recurrent neural networks are universal approximators. In *Artificial Neural Networks–ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part I 16*, pages 632–640. Springer, 2006.
- [124] L. Sensors. RAS1 — loadstarsensors.com. <https://www.loadstarsensors.com/ras1.html>. [Accessed 07-06-2024].
- [125] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver. Review of tool condition

- monitoring in machining and opportunities for deep learning. *The International Journal of Advanced Manufacturing Technology*, 109(3):953–974, 2020.
- [126] I. H. Shames. *Elastic and inelastic stress analysis*. CRC Press, 1997.
- [127] J. Shen, J. Chowdhury, S. Banerjee, and G. Terejanu. Machine fault classification using hamiltonian neural networks. *arXiv preprint arXiv:2301.02243*, 2023.
- [128] S. Shen, H. Lu, M. Sadoughi, C. Hu, V. Nemani, A. Thelen, K. Webster, M. Darr, J. Sidon, and S. Kenny. A physics-informed deep learning approach for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 103:104295, 2021.
- [129] B. Sicard, Q. Butler, P. Korsieb, Y. Wu, Y. Ziada, and S. A. Gadsden. Design considerations for building an iot enabled digital twin machine tool sub-system. In *2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, pages 1–5. IEEE, 2023.
- [130] B. Sicard, Q. Butler, P. Korsieb, Y. Wu, Y. Ziada, and S. A. Gadsden. Iiodt: Industrial internet of digital twins for hierarchical asset management in manufacturing. In *2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, pages 1–5. IEEE, 2023.
- [131] B. Sicard, Q. Butler, Y. Wu, S. Abdolahi, Y. Ziada, and S. A. Gadsden. Generating synthetic data for data-driven solutions via a digital twin for condition monitoring in machine tools. In *Synthetic Data for Artificial Intelligence and*

- Machine Learning: Tools, Techniques, and Applications II*, volume 13035, page 1303503. SPIE, 2024.
- [132] B. Sicard, A. McCafferty-Leroux, R. Appuhamy, A. Newton, P. Kosierb, and S. A. Gadsden. Aerospace digital twins: examining applications of digital twin technology to unmanned aerial vehicles and satellites. In *Autonomous Systems: Sensors, Processing, and Security for Ground, Air, Sea, and Space Vehicles and Infrastructure 2024*, volume 13052, pages 57–66. SPIE, 2024.
- [133] B. S. Sicard, Q. Butler, Y. Ziada, and S. A. Gadsden. Cognitive dynamic digital twin: enhancements for digital twin platforms based on human cognition. In *Big Data V: Learning, Analytics, and Applications*, volume 12522, pages 48–63. SPIE, 2023.
- [134] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [135] S. Soltane, S. Montassar, O. Ben Mekki, and R. El Fatmi. A hysteretic bingham model for mr dampers to control cable vibrations. *Journal of Mechanics of Materials and Structures*, 10(2):195–206, 2015.
- [136] B. Spencer Jr, S. Dyke, M. Sain, and J. Carlson. Phenomenological model for magnetorheological dampers. *Journal of engineering mechanics*, 123(3):230–238, 1997.
- [137] J. Sproston and A. El-Wahed. Applications of electro-rheological fluids in vibration control: a survey. *Smart Materials and Structures*, 5(4):464, 1996.

- [138] R. Stanway, J. Sproston, and N. Stevens. Non-linear identification of an electro-rheological vibration damper. *IFAC Proceedings Volumes*, 18(5):195–200, 1985.
- [139] R. Stanway, J. Sproston, and N. Stevens. Non-linear modelling of an electro-rheological vibration damper. *Journal of Electrostatics*, 20(2):167–184, 1987.
- [140] H. J. Stetter et al. *Analysis of discretization methods for ordinary differential equations*, volume 23. Springer, 1973.
- [141] H. Sun, L. Peng, J. Lin, S. Wang, W. Zhao, and S. Huang. Microcrack defect quantification using a focusing high-order sh guided wave emat: The physics-informed deep neural network gwnet. *IEEE Transactions on Industrial Informatics*, 18(5):3235–3247, 2021.
- [142] O. Surucu, S. A. Gadsden, and J. Yawney. Condition monitoring using machine learning: A review of theory, applications, and recent advances. *Expert Systems with Applications*, 221:119738, 2023.
- [143] D. N. Tanyu, J. Ning, T. Freudenberg, N. Heilenkötter, A. Rademacher, U. Iben, and P. Maass. Deep learning methods for partial differential equations and related parameter identification problems. *Inverse Problems*, 39(10):103001, 2023.
- [144] A. Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.
- [145] A. M. Tartakovsky, C. O. Marrero, P. Perdikaris, G. D. Tartakovsky, and

- D. Barajas-Solano. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research*, 56(5):e2019WR026731, 2020.
- [146] T. Terasawa, C. Sakai, H. Ohmori, and A. Sano. Adaptive identification of mr damper for vibration control. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 3, pages 2297–2303. IEEE, 2004.
- [147] M. Vaquero, D. M. de Diego, and J. Cortés. Designing poisson integrators through machine learning. *arXiv preprint arXiv:2403.20139*, 2024.
- [148] D. Wang and W. H. Liao. Magnetorheological fluid dampers: a review of parametric modelling. *Smart materials and structures*, 20(2):023001, 2011.
- [149] E. R. Wang, X. Q. Ma, S. Rakhela, and C. Su. Modelling the hysteretic characteristics of a magnetorheological fluid damper. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 217(7): 537–550, 2003.
- [150] J. Wang, Y. Li, R. Zhao, and R. X. Gao. Physics guided neural network for machining tool wear prediction. *Journal of Manufacturing Systems*, 57:298–310, 2020.
- [151] S. Wang, H. Wang, and P. Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.

- [152] S. Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*, 2023.
- [153] Y.-K. Wen. Method for random vibration of hysteretic systems. *Journal of the engineering mechanics division*, 102(2):249–263, 1976.
- [154] N. M. Wereley, L. Pang, and G. M. Kamath. Idealized hysteresis modeling of electrorheological and magnetorheological dampers. *Journal of Intelligent Material Systems and Structures*, 9(8):642–649, 1998.
- [155] N. M. Wereley, G. M. Kamath, and V. Madhavan. Hysteresis modeling of semi-active magnetorheological helicopter dampers. *Journal of intelligent material systems and structures*, 10(8):624–633, 1999.
- [156] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403: 115671, 2023.
- [157] Y. Wu, B. Sicard, and S. A. Gadsden. Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring. *Expert Systems with Applications*, page 124678, 2024.
- [158] Y. Wu, B. Sicard, and S. A. Gadsden. A review of physics-informed machine learning methods with applications to condition monitoring and anomaly detection. *arXiv preprint arXiv:2401.11860*, 2024.

- [159] S. Xu, Z. Sun, R. Huang, D. Guo, G. Yang, and S. Ju. A practical approach to flow field reconstruction with sparse or incomplete data through physics informed neural network. *Acta Mechanica Sinica*, 39(3):1–15, 2023.
- [160] S. Xu, C. Yan, Z. Sun, R. Huang, D. Guo, and G. Yang. On the preprocessing of physics-informed neural networks: How to better utilize data in fluid mechanics. *arXiv preprint arXiv:2403.19923*, 2024.
- [161] G. Yang, B. Spencer Jr, J. Carlson, and M. Sain. Large-scale mr fluid dampers: modeling and dynamic performance considerations. *Engineering structures*, 24(3):309–323, 2002.
- [162] L. Yang, D. Zhang, and G. E. Karniadakis. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1):A292–A317, 2020.
- [163] L. Yang, X. Meng, and G. E. Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.
- [164] G. Yao, F. Yap, G. Chen, W. Li, and S. Yeo. Mr damper and its application for semi-active control of vehicle suspension system. *Mechatronics*, 12(7):963–973, 2002.
- [165] X. Yin, Z. Huang, and Y. Liu. Bridge damage identification under the moving vehicle loads based on the method of physics-guided deep neural networks. *Mechanical Systems and Signal Processing*, 190:110123, 2023.

- [166] Y. Yin, V. Le Guen, J. Dona, E. de Bézenac, I. Ayed, N. Thome, and P. Galinari. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021 (12):124012, 2021.
- [167] A. Yoosefdoost, S. M. Tahsien, S. A. Gadsden, W. D. Lubitz, and M. Kaviani. Artificial neural networks and extended kalman filter for easy-to-implement runoff estimation models. In *Canadian Society of Civil Engineering Annual Conference*, pages 1071–1099. Springer, 2022.
- [168] R. Yu and R. Wang. Learning dynamical systems from data: An introduction to physics-guided deep learning. *Proceedings of the National Academy of Sciences*, 121(27):e2311808121, 2024.
- [169] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.
- [170] G. Zaccane. *Python parallel programming cookbook*. Packt Publishing Ltd, 2015.
- [171] W. Zhai, D. Tao, and Y. Bao. Parameter estimation and modeling of nonlinear dynamical systems based on runge–kutta physics-informed neural network. *Nonlinear Dynamics*, 111(22):21117–21130, 2023.
- [172] R. Zhang, Y. Liu, and H. Sun. Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, 215:110704, 2020.

- [173] X.-C. Zhang, J.-G. Gong, and F.-Z. Xuan. A physics-informed neural network for creep-fatigue life prediction of components at elevated temperatures. *Engineering Fracture Mechanics*, 258:108130, 2021.
- [174] C. Zhou and R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.
- [175] Q. Zhou, S. R. Nielsen, and W. Qu. Semi-active control of three-dimensional vibrations of an inclined sag cable with magnetorheological dampers. *Journal of sound and vibration*, 296(1-2):1–22, 2006.
- [176] T. Zhou, S. Jiang, T. Han, S.-P. Zhu, and Y. Cai. A physically consistent framework for fatigue life prediction using probabilistic physics-informed neural network. *International Journal of Fatigue*, 166:107234, 2023.
- [177] T. Zhou, X. Zhang, E. L. Droguett, and A. Mosleh. A generic physics-informed neural network-based framework for reliability assessment of multi-state systems. *Reliability Engineering & System Safety*, 229:108835, 2023.
- [178] Q. Zhu, Z. Zhao, and J. Yan. Physics-informed machine learning for surrogate modeling of wind pressure and optimization of pressure sensor placement. *Computational Mechanics*, pages 1–11, 2022.
- [179] X. Zhu, X. Jing, and L. Cheng. Magnetorheological fluid dampers: A review on structure design and analysis. *Journal of intelligent material systems and structures*, 23(8):839–873, 2012.