Application of MR Fluid Actuators Using Machine Learning

B.Tech. Project Report

Submitted by

Animesh Kumar Singh (B20ME012)

Garvit Meena (B20ME033)

Under the Supervision

Of

Prof. B. Ravindra,

Department of Mechanical Engineering

Indian Institute of Technology, Jodhpur

November, 2023

CONTENTS

Acknowledgements

We would like to express our deepest gratitude to **Prof. B. Ravindra**, Department of Mechanical Engineering, **Indian Institute of Technology Jodhpur**, for his guidance and support throughout our B.Tech. project. The opportunity we had with the Indian Institute of Technology, Jodhpur, has been invaluable for our learning and professional development. Therefore, we consider ourselves very fortunate individuals, as we were provided the opportunity to be part of it. We will strive to utilize the knowledge and skills we have gained in the best possible way.

Abstract

This research project focuses on enhancing **Magnetorheological (MR) fluid actuators** for vehicle suspension systems by integrating **machine learning** methodologies. The key innovation lies in advanced contour analysis for data extraction, with a primary emphasis on **Support Vector Regression (SVR)** and comparative analysis with Neural Networks (and other applicable models).

The study commences by employing advanced contour analysis to **extract** detailed **data** from experimental graphs, forming a comprehensive dataset for analysis. The precision of this data extraction process is pivotal for reliable device performance analysis, ensuring a nuanced understanding of MR fluid actuators' behavior.

SVR, a machine learning model tailored for continuous data, takes center stage in accurately modeling the intricate relationships between input parameters and actuator responses. Its capability to handle dynamic data proves crucial in predicting the actuators' behavior under varying conditions, offering high accuracy in capturing nuanced responses.

The research methodology extends to include other machine learning models, such as Neural Networks, providing a comparative analysis to underscore SVR's efficacy in the specific context of MR fluid actuators. The objective is to identify the most effective machine learning approach for optimizing actuator performance in vehicle suspension systems.

Machine learning insights are seamlessly integrated into a **Simulink** model, acting as a virtual testing ground to bridge theoretical predictions with real-world applications. This allows for the translation of machine learning predictions into practical insights, facilitating the development and optimization of MR fluid actuators for use in vehicle suspension systems.

The application of intelligent control systems aims to achieve optimal ride comfort, stability, and handling in vehicles. This research contributes to the automotive industry by showcasing the adaptability of MR fluid actuators and the precision of machine learning in crafting intelligent suspension systems. Emphasizing the data extraction process underscores the importance of accurate and detailed information for reliable device performance analysis. In conclusion, the synergistic integration of MR fluid actuators, machine learning methodologies, and Simulink modeling represents a significant advancement in designing and optimizing vehicle suspension systems within a concise framework.

---

1. Introduction

Magnetorheological (MR) fluid is a special type of smart material that can alter its flow and mechanical properties in response to changes in an applied magnetic field. The fluid consists of a base oil mixed with tiny iron particles and additives like surfactants. When a magnetic field is applied, the iron particles align and form a chain structure within the fluid, creating a solid-like phase.

The strength of this chain structure is directly proportional to the intensity of the magnetic field. This unique property allows for precise control over the fluid's behavior. The force required to break the chain structure serves as the driving force for MR fluid-based actuators.

The MR fluid can switch between a liquid phase and a solid-like phase reversibly based on the presence or absence of the magnetic field. Different operational modes of MR actuators include flow, shear, and squeeze modes. In the flow mode, for instance, the MR fluid occupies the space between two stationary surfaces, and its rheological properties change based on the applied magnetic field, enabling control over the actuator's movement.

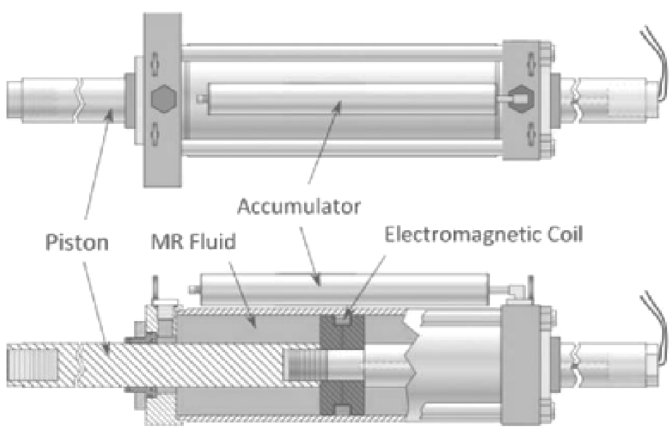


Fig. 1 : Actual MR Actuator

Ref: Image Reference

In the flow mode of magnetorheological (MR) fluid-based actuators, such as dampers, two plates are present, and flow occurs due to a pressure difference between the two ends. A magnetic field is applied perpendicular to the flow. Dampers are a common type of actuator that makes use of the flow mode. The flow mode is characterized by the MR fluid changing from a liquid phase to a solid-like phase based on the intensity of the applied magnetic field, allowing for controlled movement.

In the shear mode, the MR fluid is positioned between a stationary plate and a sliding plate. A magnetic field is applied perpendicular to the movement of the plate. For the plate to move, the chain structure formed in the MR fluid must break, generating an actuating force. Brakes and clutches are examples of actuators that employ the shear mode. The shear mode enables precise control over the movement of sliding plates.

In squeeze mode, the MR fluid occupies the space between two moving plates. A magnetic field is applied in the same direction as the plates' movement, and flow occurs perpendicular to the plates' direction. The squeeze mode is suitable for applications where a small displacement is needed but a substantial actuating force is required. It is commonly used in dampers or mounts.

Actuators utilizing MR fluid, including dampers, brakes, clutches, and mounts, offer several advantages. Despite the need to design a magnetic core for generating a magnetic field, these actuators boast fast response times (less than 1 ms), low power consumption (less than 5 W for a vehicle damper), and a simple structural design. One distinctive advantage is the failsafe feature inherent in MR actuators, without requiring additional devices. This makes MR actuators attractive for various industries, such as automotive engineering, aerospace engineering, manufacturing engineering, and civil engineering.

The fundamental principle underlying MR fluids lies in their composition, typically consisting of a base oil, magnetizable micro-scale iron particles, and surfactant additives. When subjected to an external magnetic field, these iron particles align to form chain structures within the fluid, inducing a transition from a liquid to a solid-like phase. The strength of these structures is directly correlated with the intensity of the applied magnetic field, allowing for a continuum of mechanical responses. This inherent versatility has led to the widespread incorporation of MR fluid actuators in engineering systems, ranging from adaptive vehicle suspension to civil engineering dampers.

While MR fluid actuators offer unique advantages such as rapid response times, low power consumption, and built-in fail-safe features, optimizing their performance requires a nuanced understanding of the intricate relationship between magnetic field variations and resulting mechanical behaviors. Traditional control methods often struggle to capture the complexity of this relationship, prompting the exploration of advanced computational techniques, with a particular emphasis on machine learning, to enhance control precision.

1.1 Motivation:

This project aims to optimize Magnetorheological (MR) fluid actuators using Support Vector Regression (SVR) and machine learning. Motivated by the need for enhanced precision, adaptability, and control, we bridge experimental data and simulations, unlocking the full potential of MR fluid technology for efficient applications in engineering systems.

1.2 Problem Statement:

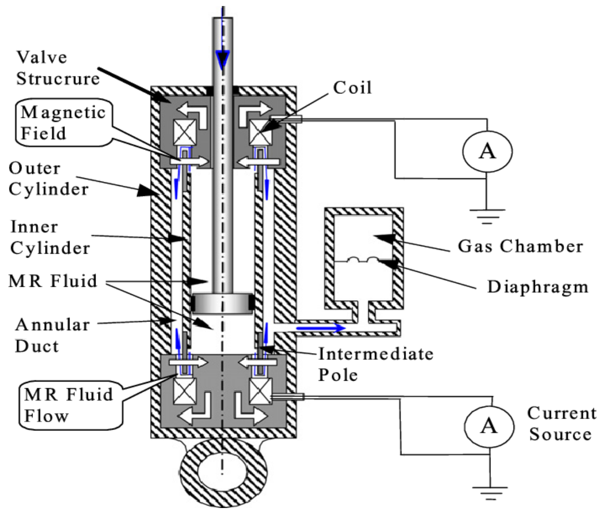
The optimization of Magnetorheological (MR) fluid actuators poses a significant challenge due to the complex relationship between magnetic field variations and resulting mechanical behaviors. Traditional control methods often fall short of capturing the nuanced intricacies of this relationship, hindering the full realization of the potential benefits offered by MR fluid technology. The lack of precision in control mechanisms limits the adaptability of MR fluid actuators in engineering systems, particularly in applications such as adaptive vehicle suspension and civil engineering dampers.

The problem statement revolves around the need for a more sophisticated and precise control approach to unlock the full potential of MR fluid actuators. Current control methods struggle to provide the level of adaptability and efficiency required for optimal performance. The goal is to address this gap by leveraging machine learning methodologies, with a specific focus on Support Vector Regression (SVR), to extract meaningful insights from experimental data. This approach aims to enhance the control precision, adaptability, and overall performance of MR fluid actuators, making them more effective and versatile in a range of engineering applications. The project seeks to overcome the limitations of traditional control methods and establish a foundation for the seamless integration of MR fluid actuators into advanced engineering systems.

1.3 Objective:

The primary objective of this research is to leverage machine learning methodologies, specifically Support Vector Regression (SVR), to extract meaningful insights from experimental data obtained through rigorous testing of MR fluid actuators. Contour analysis techniques are employed to distill crucial features from experimental graphs, forming a curated dataset. The SVR model, chosen for its aptitude for handling continuous data and intricate relationships, is then trained and fine-tuned to accurately model the behavior of MR fluid actuators.

The integration of SVR into a Simulink model of an MR fluid actuator serves as a virtual testbed for the real-time application of machine-learned control strategies. By bridging the gap between experimental data and simulated control systems, this research seeks to enhance the precision and adaptability of MR fluid actuators through the symbiotic fusion of machine learning and smart material technology.



Ref: Google Images

Fig. 2: Details of MR fluid Damper/Actuator

---

2. Work done

Initial Work Done:

Embarking on a journey to enhance the performance of Magnetorheological (MR) fluid actuators, our research faced a pivotal challenge - the absence of direct experimental data and the requisite apparatus. Undeterred, our focus shifted to a pioneering approach: the extraction and augmentation of crucial data directly from experimental graphs. This initiative became the cornerstone of our investigation, necessitating the application of machine learning techniques to transmute graphical information into a comprehensive and scalable dataset.

MR fluid actuators, renowned for their adaptive response to varying magnetic fields, constitute a vital component in modern engineering systems. The precision of these actuators relies on a nuanced understanding of the relationships between input parameters such as velocity, frequency, current, and the resulting output force. In the absence of direct experimental data, our mission became twofold: first, to extract meaningful data points from graphical representations of these parameters, and second, to employ machine learning methodologies for data augmentation and predictive modeling.

The absence of direct experimental data prompted the development of a robust strategy leveraging machine learning to extract, transform, and generate a rich dataset from graphical representations. The graphical data, encompassing velocity, frequency, current, and output force, became the primary source material for our analytical endeavors. This innovative approach not only overcome the limitations of lacking experimental apparatus but also opened avenues for harnessing machine learning to extrapolate trends, patterns, and relationships latent within the graphical depictions.

In this research endeavor, the extraction and augmentation of data from experimental graphs signify a departure from conventional methodologies, underscoring our commitment to innovation and adaptability. The forthcoming sections will delve into the intricacies of our methodology, where machine learning algorithms play a pivotal role in transforming visual representations into a comprehensive and scalable dataset. This unique approach not only addresses the challenges posed by the absence of direct experimental data but also positions our research at the vanguard of pioneering methodologies within the field of smart materials and machine learning integration.

2.1 Data Extraction from Experimental Graphs Using Computer Vision:

(i). Data Extraction:

In this section, we aim to extract and compare hysteresis loops obtained from both the numerical model and experimental data. The hysteresis loop serves as a critical indicator for understanding the behavior of the system under study. By juxtaposing results from the computational model with real-world experimental data, we can gain valuable insights into the accuracy and reliability of the simulation.

By systematically extracting and comparing hysteresis loops, we aim to validate the numerical model and enhance our understanding of the system's dynamic behavior.

Step 1: Reading the graphical images:

The provided Python script uses the Pillow and Matplotlib libraries to open and display an image specified by a given path, facilitating the initial visual inspection in the context of data extraction.

Code:

```
# Import necessary libraries
from PIL import Image
import matplotlib.pyplot as plt
# Path to the image
image_path = "path_to_graphs_image"
# Open the image using PIL
image = Image.open(image_path)
```

So, the graphs that we are using for the data extraction are:

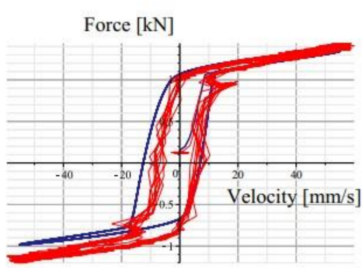


Fig. 3(a)

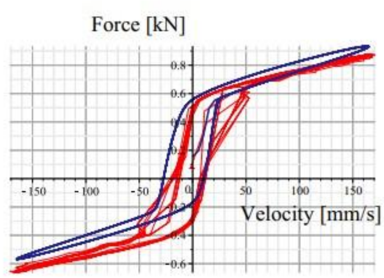


Fig. 3(b)

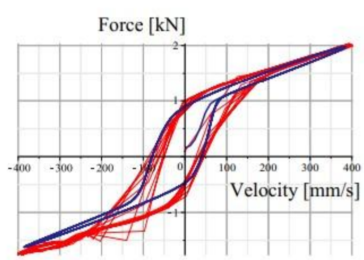


Fig. 3(c)

The report includes a comparative analysis of hysteresis loops derived from both the numerical model and experimental data under different conditions: (**Fig. 3(a)**) frequency (f) at 0.5 Hz and current (I) at 1 A; (**Fig. 3(b)**) frequency at 1.5 Hz and current at 0.3 A; and (**Fig. 3(c)**) frequency at 3.5 Hz and current at 0.6 A.

Step 2: Separating the graphical features by OpenCV:

OpenCV, the Open Source Computer Vision library, is a powerful tool extensively utilized in computer vision and image processing applications. In the context of separation, OpenCV is instrumental in reading and converting images, performing color space transformations, and implementing advanced image processing techniques such as color thresholding and bitwise operations. Its versatility makes it well-suited for a broad range of tasks, from basic operations like loading and displaying images to more complex computer vision tasks such as image segmentation and feature extraction.

Code:

```
# Reading and converting the image using OpenCV
image = cv2.imread("path_to_the_graphs_image")
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# Define color thresholds for red and blue
lower_red, upper_red = np.array([100, 0, 0]), np.array([255, 100, 100])
lower_blue, upper_blue = np.array([0, 0, 100]), np.array([100, 100, 255])
# Create masks for red and blue portions
red_mask, blue_mask = cv2.inRange(image_rgb, lower_red, upper_red), cv2.inRange(image_rgb, lower_blue, upper_blue)
# Apply masks to the original image
red_segmented, blue_segmented = cv2.bitwise_and(image_rgb, image_rgb, mask=red_mask), cv2.bitwise_and(image_rgb, image_rgb, mask=blue_mask)
```

The provided code segment utilizes the OpenCV library for image processing. It begins by reading an image from a specified path and converting it from the default BGR color space to RGB. Subsequently, color thresholds are defined for both red and blue components, establishing the acceptable ranges in the RGB color space. Binary masks are created by applying these color thresholds to the original RGB image. These masks serve to isolate the red and blue portions of the image. The final result is achieved by applying the masks to the original image through bitwise operations, yielding segmented images highlighting the red and blue components. So, in the next step we will work upon the red-segmented image which is our area of interest.

Step 3: Contour Analysis and Point Extraction in Red-Segmented Image using OpenCV:

The Python script employs OpenCV and NumPy for contour detection and point analysis in a red-segmented image. After loading the image, contours are identified, and the largest one, assumed to represent a significant curve, is selected. Points along this curve are interpolated to enhance density, and a manageable subset is randomly sampled. The coordinates of these points are then transformed into a desired coordinate system and saved to a CSV file. The script concludes by visualizing the selected points on the original image using Matplotlib. This comprehensive process enables the extraction and analysis of specific curve features, providing valuable insights into the image's structural characteristics.

Code:

```
# Find contours in the image

contours, _ = cv2.findContours(red_segmented, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE

# Choose the largest contour (assuming it's the curve you're interested in)

largest_contour = max(contours, key=cv2.contourArea)

# Get points along the largest contour

curve_points = [tuple(point[0]) for point in largest_contour]

# Interpolate points along the curve (to get more data in the dataset)

interpolated_curve_points = []

for i in range(len(curve_points) - 1):

    x1, y1 = curve_points[i]

    x2, y2 = curve_points[i + 1]

        num_interpolated_points = int(np.sqrt((x2 - x1)**2 + (y2 - y1)**2)) #Interpolation
formula

    x_interpolated = np.linspace(x1, x2, num_interpolated_points)

    y_interpolated = np.linspace(y1, y2, num_interpolated_points)

    interpolated_curve_points.extend(zip(y_interpolated, x_interpolated))
```

```
# Sampling Points
num_points = 10,000
if len(interpolated_curve_points) <= num_points:
    selected_points = interpolated_curve_points
else:
        selected_indices = np.random.choice(len(interpolated_curve_points), num_points, replace=False)
    selected_points = [interpolated_curve_points[i] for i in selected_indices]
# Transform coordinates to the desired coordinate system according to the coordinates of the graph
transformed_points = []
for point in selected_points:
    y, x = point
    new_x = (x / red_segmented.shape[1]) * (Xmax + (-Xmin)) - Xmin
    new_y = (y / red_segmented.shape[0]) * (Ymax + (-Ymin)) - Ymin
    transformed_points.append((new_y, new_x))
```

Step 4: To extract the data inside the contour:

In the analysis process, an impediment was encountered as the OpenCV approach struggled to reliably identify data points within the contour due to sparse data points near the middle of the curve. Consequently, a pragmatic solution was adopted by resorting to manual data inclusion using the online software WebPlotDigitizer. This approach allowed for the meticulous extraction of data points from the respective graphs. To align the extracted data with real-world coordinates, a transformation was applied to convert image coordinates to the desired coordinate system, ensuring accuracy and reliability in subsequent analyses

2.2 Machine Learning Implementation

(i). Feature Selection:

The relevant features for modeling the MR fluid actuator behavior are identified. In this context, key parameters such as velocity, frequency, and current serve as inputs, while the force exerted by the actuator acts as the target variable. The dataset is structured to include these features, forming the basis for the subsequent training and testing phases.

The feature selection process in the context of extracting data from experimental graphs involves a systematic approach to identify and choose parameters that significantly influence the behavior of the MR fluid actuator. The following steps elaborate on how features were selected:

- **Identification of Influential Parameters:**

  Features that exhibit distinct variations and are likely to impact the actuator's performance are identified. In this specific case, the parameters chosen are frequency, velocity, and current. The selection is based on the observed trends and variations in the experimental graphs.

```
# Features selected based on graph analysis
features_selected = ['frequency', 'velocity', 'current']
```

- **Relevance to Actuator Behavior:**

  The selected features are deemed relevant as they capture essential aspects of the actuator's dynamics. Frequency represents the oscillation rate, velocity signifies the speed of movement, and current relates to the electrical input.

- **Correlation Analysis:**

  A correlation analysis may be performed to quantify the relationships between selected features and the output force of the MR fluid actuator. This statistical analysis helps validate the chosen features' significance.

- **Fluid Dynamics Knowledge Integration:**

  Understanding the physical principles governing the MR fluid actuator aids in identifying parameters crucial to its operation.

- **Validation through Machine Learning Models:**

  The final validation of feature selection occurs through the integration of selected parameters into machine learning models. The chosen features should contribute meaningfully to the model's ability to predict the output force of the MR fluid actuator.

(ii). Data Splitting:

The dataset is divided into training and testing sets to facilitate the training and evaluation of machine learning models. The common practice of an 80-20 split ratio is adopted, ensuring a robust assessment of model performance on unseen data.

```
# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

(iii). Data Standardization:

To enhance the convergence and performance of machine learning models, feature scaling is applied. The features are standardized using techniques such as StandardScaler, ensuring uniform scaling and preventing certain features from dominating the learning process.

```
# Standardize the feature data

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
```

(iv). Support Vector Regression (SVR) Model:

The project emphasizes the application of Support Vector Regression (SVR) due to its efficacy in handling continuous data and capturing intricate patterns in complex datasets. A hyperparameter tuning process is implemented using GridSearchCV to optimize SVR parameters, including the choice of kernel, regularization parameter (C), and epsilon.

```python
# Hyperparameter tuning using GridSearchCV
param_grid = {
    'kernel': ['rbf'],
    'C': [10],
    'epsilon': [0.1]
}
grid_search = GridSearchCV(SVR(), param_grid, cv=5, scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)
best_svr = grid_search.best_estimator_
```

The performance of the SVR model is evaluated using metrics such as Mean Squared Error (MSE) and R-squared (R2) on the test data. These metrics provide insights into the accuracy and predictive capabilities of the model.

```python
# Evaluate the SVR model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

→ Grid Search:

GridSearchCV is employed in the context of Support Vector Regression (SVR) to systematically fine-tune the model's hyperparameters. In machine learning, selecting the right hyperparameters is crucial for optimizing model performance. GridSearchCV automates this process by exhaustively searching through a predefined grid of hyperparameter values, evaluating each combination through cross-validation. In the case of SVR, hyperparameters such as the choice of kernel, regularization parameter (C), and epsilon for the loss function significantly impact the model's ability to capture underlying patterns in the data. GridSearchCV ensures a comprehensive exploration of these hyperparameter combinations, allowing for the identification of the configuration that yields the best predictive performance. This systematic approach eliminates the need for manual trial and error, streamlining the model development process, and enhancing the overall efficiency and effectiveness of the SVR model for predicting magnetorheological fluid damper behavior.

Define Parameter Grid:

Specify a grid of hyperparameters that you want to explore. In the case of SVR, common hyperparameters include the choice of kernel ('linear', 'rbf', etc.), regularization parameter 'C', and epsilon for the epsilon-insensitive loss function.

```
param_grid = {
    'kernel': ['linear', 'rbf'],
    'C': [0.1, 1, 10],
    'epsilon': [0.1, 0.01, 0.001]
}
```

Instantiate GridSearchCV:

Create a GridSearchCV object, specifying the SVR model, the parameter grid, the number of cross-validation folds (cv), and the scoring metric. In this example, 'neg_mean_squared_error' is used as the scoring metric.

```
grid_search = GridSearchCV(svr, param_grid, cv=5, scoring='neg_mean_squared_error')
```

Fit GridSearch to Training Data:

```
grid_search.fit(X_train, y_train)
```

Following the identification of the optimal parameters through GridSearchCV, we proceeded to implement the SVR model. While various machine learning models, including random forest, neural networks, and linear regression, were tested, the emphasis was placed on Support Vector Regression (SVR) based on the guidance from our professor. The decision to focus on SVR aligns with the unique characteristics of the magnetorheological fluid damper system. The SVR model, renowned for its ability to handle complex relationships in data, particularly suits the intricate dynamics associated with magnetorheological fluid behavior. Furthermore, to ensure a deep understanding of the model's workings and tailor it precisely to our application, we took the initiative to build the SVR model from scratch.

→ About Models Implemented:

- **Support Vector Regression (SVR):**
  1) Details: Explored Radial Basis Function (RBF) and other kernels via grid search.
  2) Hyperparameters: Tuned C (regularization), epsilon (tube width), kernel type, and gamma.
  3) Rationale: SVR chosen for its ability to handle non-linear relationships in magnetorheological fluid dynamics.

- **Random Forest:**
  1) Details: Ensemble learning model.
  2) Hyperparameters: Tuned tree count, depth, leaf size, etc.
  3) Rationale: Suitable for capturing non-linear patterns in data, serving as a benchmark against SVR.

- **Neural Networks:**
  1) Details: Deep learning model with multiple layers.
  2) Hyperparameters: Tuned layers, neurons, activation functions, learning rate, etc.
  3) Rationale: Explored for its ability to model complex relationships in fluid behavior, providing an alternative to SVR.

- **Linear Regression:**
  1) Details: Simple linear model.
  2) Hyperparameters: No tuning as it's a parametric model.
  3) Rationale: Used as a baseline for simplicity and interpretability in result comparison.

- **Building SVR from Scratch:**
  1) Details: Implemented SVR without pre-existing libraries.
  2) Customization: Tailored structure and parameters for magnetorheological fluid damper data.
  3) Outcome: Limited success led to a decision not to pursue further due to suboptimal results.

Having explored various models, our focus narrowed down to support Support Vector Regression (SVR) due to its favorable performance. We utilized GridSearchCV for hyperparameter tuning, optimizing the SVR model for accurate predictions. Subsequently, we employed the trained SVR model to predict force values based on given parameters such as frequency, velocity, and current. The comparison of these predictions with values obtained from a Simulink model is crucial, as it serves as a validation step for aligning our machine learning predictions with a simulated physical model. This comparison aids in understanding the SVR model's proximity to the expected behavior of the magnetorheological fluid damper under different operational conditions, contributing valuable insights for potential applications in vehicle suspension systems.

Simulink Integration:

The optimized SVR model is seamlessly integrated into Simulink models, establishing a crucial bridge between machine learning predictions and real-world applications in vehicle suspension systems. This step ensures that the theoretical insights derived from machine learning are translated into actionable and practical solutions.

Results Visualization:

The final step involves visualizing the predicted vs. actual force values. Scatter plots provide a clear representation of the model's accuracy in capturing the dynamic behavior of MR fluid actuators. This visualization serves as a comprehensive and intuitive means of assessing the model's performance.

```
# Visualize the predicted vs. actual values

plt.scatter(y_test, y_pred)

plt.xlabel("Actual Force")

plt.ylabel("Predicted Force")

plt.title("Actual vs. Predicted Force (SVR)")

plt.show()
```

This detailed methodology underscores the systematic and rigorous approach adopted in the exploration of MR fluid actuators, from data extraction to model evaluation and real-world application. The integration of SVR, a robust machine learning algorithm, and Simulink models positions this research at the forefront of advancements in the optimization of vehicle suspension systems.

→ Added new features:

To increase accuracy, we have added these features:

```
X = data[['velocity', 'frequency', 'current', 'Area', 'Width ', 'Height','Perimeter', 'Aspect Ratio', 'Circularity']]
```

We initially extracted the underlined features and applied Support Vector Regression (SVR). However, the expected increase in accuracy did not materialize. In an attempt to improve results, we explored an alternative approach by assigning different weights to the features. Despite these efforts, the outcomes remained consistent, and we did not achieve the anticipated increase in accuracy. Ultimately, the implementation was undertaken with the primary objective of improving accuracy, but unfortunately, the desired results were not attained.

2.3 Theoretical/Experimental work

MR Damper:

For positive values of the shear rate, , the total stress is given by

$$\tau = \tau_{y\,(field)} + \eta\dot{\gamma}$$

In this model, for nonzero piston velocities, , the force generated by the device given by:

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

The forces on either side of the rigid bar are equivalent; therefore,

$$c_1\dot{y} = \alpha z + k_0(x - y) + c_0(\dot{x} - \dot{y})$$

The total force generated by the system is then found by summing the forces in the upper and lower sections of the system:

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

the total force can also be written as:

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

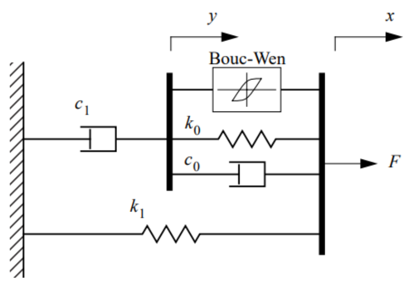


Fig. 4: Bouc-Wen MR DAMPER MODEL

Utilizing these equations and other fluid dynamics concepts we build the simulink model of MR DAMPER.

In our research, we focused on the utilization of existing data obtained from a research paper that conducted dynamic tests on a magnetorheological fluid damper (MRD1003-5) produced by the LORD company. The original experiment aimed to characterize the behavior of the MR damper under various dynamic conditions.

The experimental setup involved subjecting the damper to harmonic and linearly varying kinematic excitations with different displacement amplitudes and velocities. The tests were conducted using an MTS hydraulic testing system, recording force, displacement, piston velocity, and damper housing temperature. The resulting force–velocity hysteresis loops, observed under diverse combinations of amplitude and frequency, were a crucial outcome of the experiment.

In our work, we extracted valuable information and insights from the graphical representation of these hysteresis loops presented in the research paper. Rather than conducting the physical experiments ourselves, we focused on the mathematical modeling and analysis of the MR damper using machine learning techniques, such as Support Vector Regression (SVR), applied to the extracted data. Our objective was to develop a predictive model for the force output based on input parameters like velocity, frequency, and current, ultimately exploring potential applications in vehicle suspension systems.

3. Results and Discussion

3.1 Results of Data Extraction:

(i). <u>Results of the graphical separation of features (red and blue) using OpenCV:</u>

→ For Graph 1 (f=0.5 Hz, I=1 A):

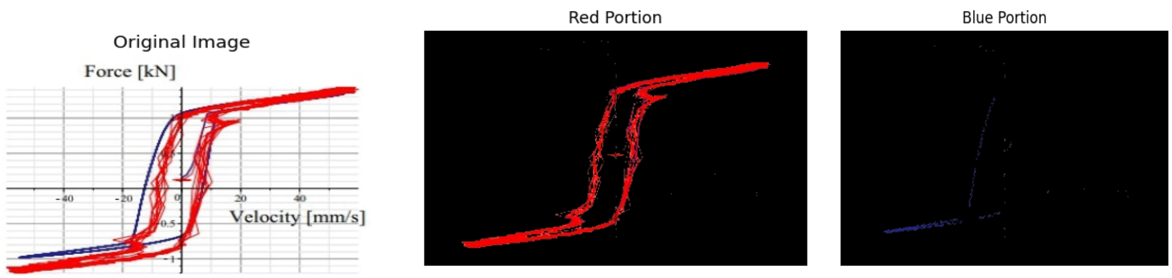


Fig. 5(a): Red and Blue portions separated from the original graph (f=0.5 Hz, I=1 A).

→ For Graph 2 (f=1.5 Hz, I=0.3 A):

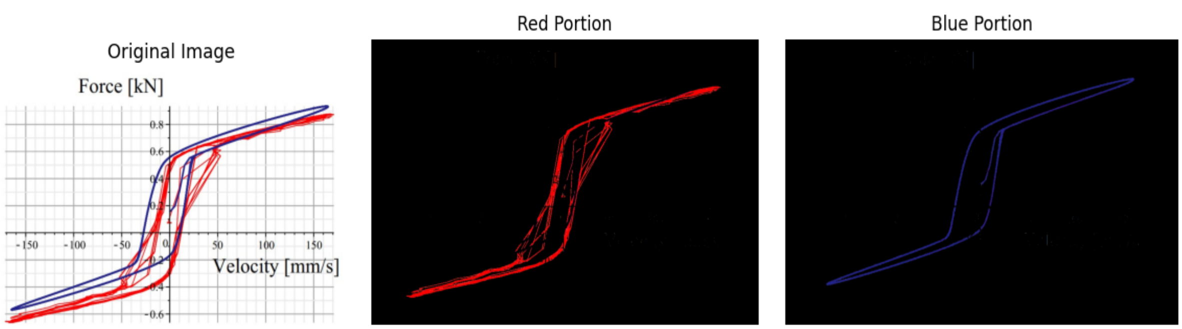


Fig. 5(b): Red and Blue portions separated from the original graph (f=1.5 Hz, I=0.3 A).

→ For Graph 3 (f=3.5 Hz, I=0.6 A):

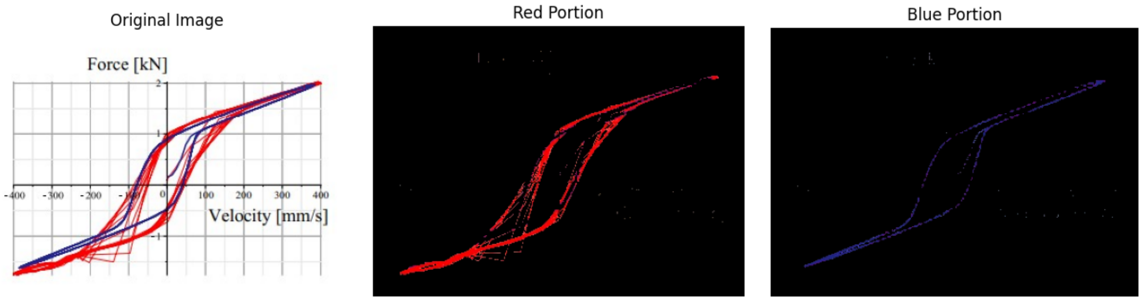


Fig. 5(c): Red and Blue portions separated from the original graph (f=3.5 Hz, I=0.6 A).

<u>(ii). Result of each graphs contoured plot image:</u>

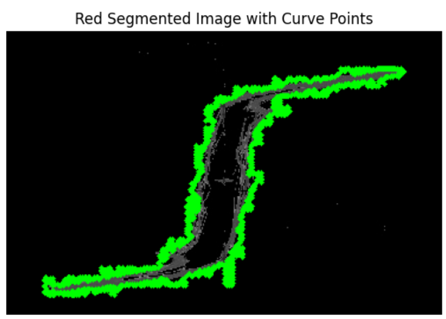


Fig. 6(a): Data points at the perimeter of the graph (**f=0.5 Hz, I=1 A)**

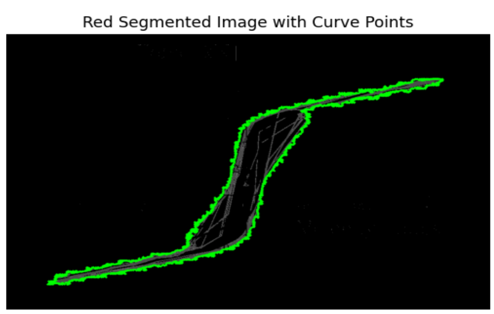


Fig. 6(b): Data points at the perimeter of the graph (**f=1.5 Hz, I=0.3 A)**

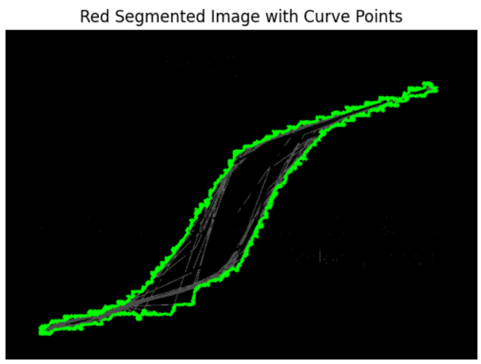


Fig. 6(c): Data points at the perimeter of the graph (**f=3.5 Hz, I=0.6 A)**

(iii). Results of the data combined after the data extraction:

Link to the data: Combined_Data

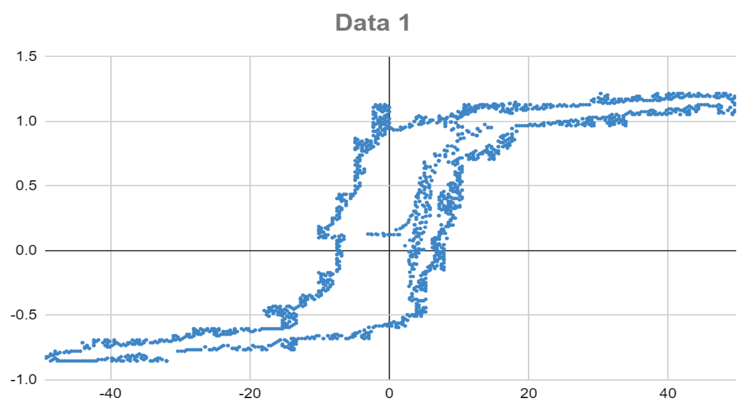


Fig. 7(a): Graph obtained by plotting data extracted for graph (f=0.5 Hz, I=1 A)

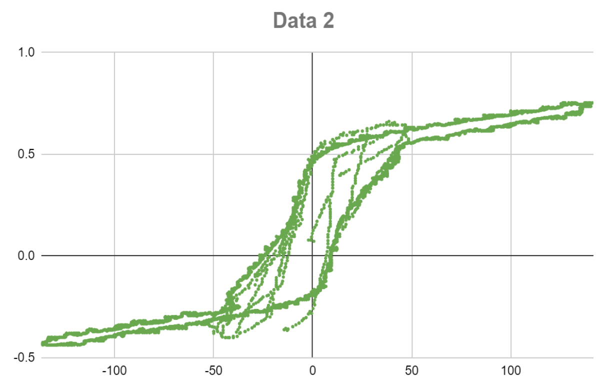


Fig. 7(b): Graph obtained by plotting data extracted for graph (f=1.5 Hz, I=0.3 A)

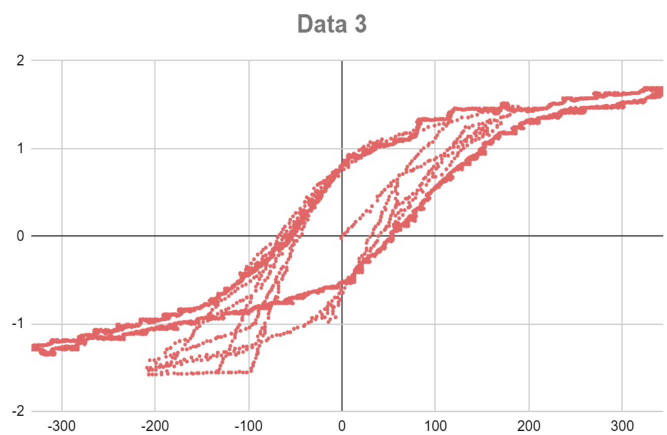


Fig. 7(c): Graph obtained by plotting data extracted for graph (f=3.5 Hz, I=0.6 A)

Fig. 7(d): Plot obtained by combining the above three respective graphs

3.2 Results of Machine Learning:

(i). SVR with Hyperparameters ('C': 10, 'epsilon': 0.1, 'kernel': 'rbf'):

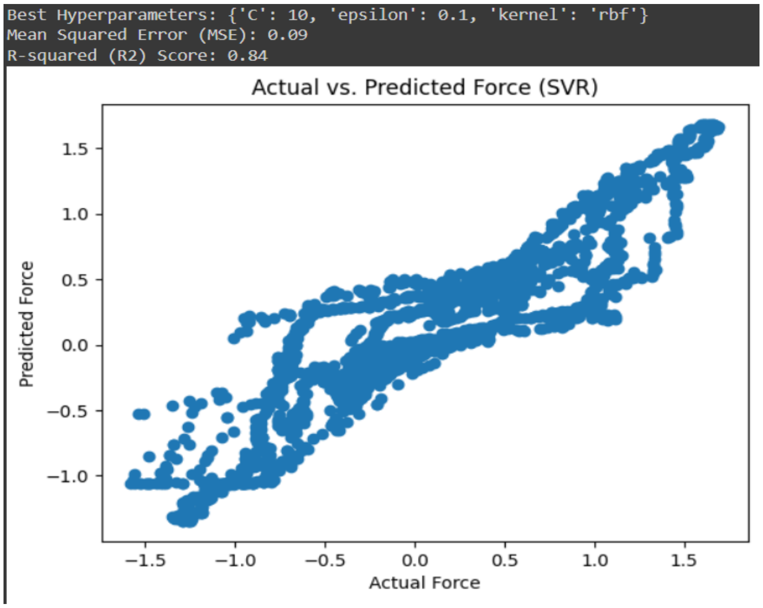


Fig. 8 (i): SVR accuracy = 0.84 with hyperparameters ('C': 10, 'epsilon': 0.1, 'kernel': 'rbf')

(ii). SVR with Hyperparameters ('C': 10, 'epsilon': 0.1, 'gamma': 500, 'kernel': 'rbf')

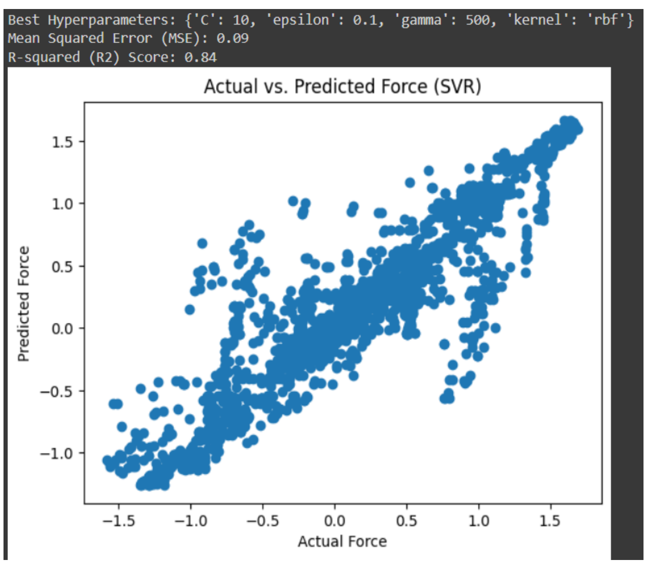


Fig. 8 (ii): SVR accuracy = 0.84 with hyperparameters ('C': 10, 'epsilon': 0.1, 'gamma': 500, 'kernel': 'rbf')

(iii). Actual v/s Predicted Force Graph in SVR:

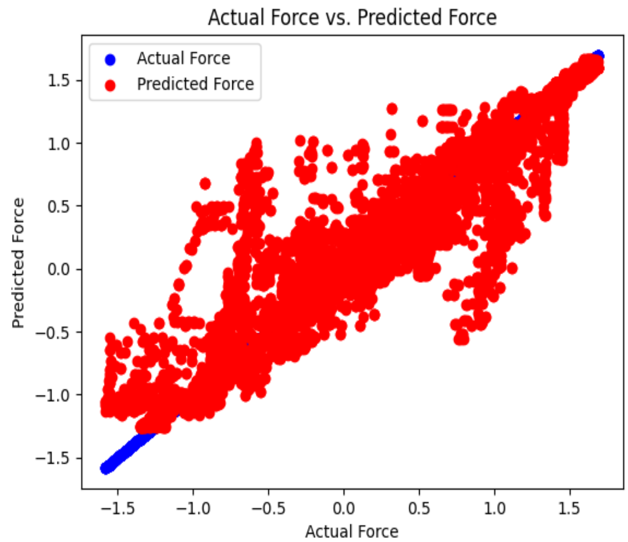


Fig. 8(iii): Actual Force v/s Predicted Force by implementing SVR

(iv). Actual v/s Predicted Force (SVR) and Residuals v/s Predicted Force (SVR):

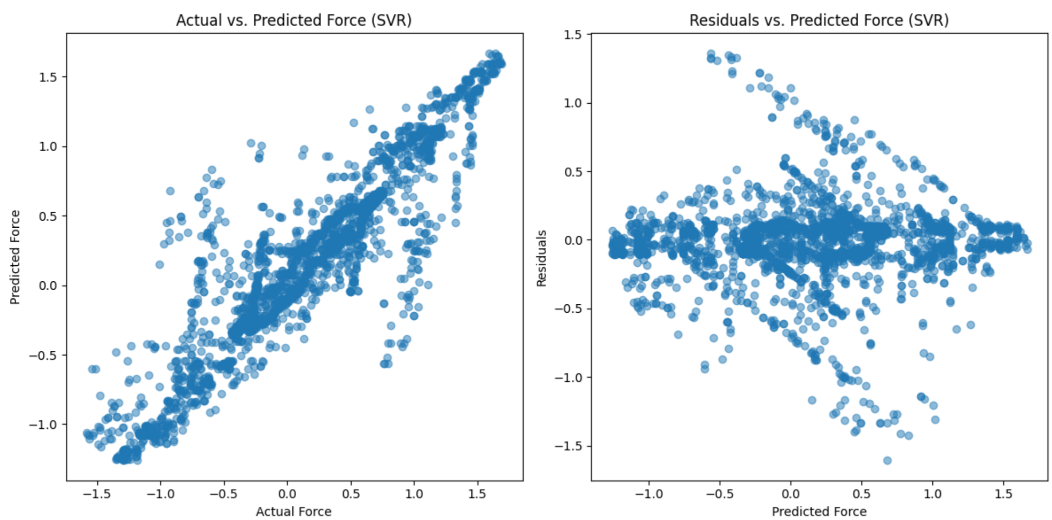


Fig. 8(iv): Actual v/s Predicted Force (SVR) and Residuals v/s Predicted Force (SVR)

(v). Results of SVR with parameters ('frequency', 'current', 'velocity'):

```
Best Hyperparameters: {'C': 10, 'epsilon': 0.1, 'gamma': 500, 'kernel': 'rbf'}
Mean Squared Error (MSE): 0.09
Root Mean Squared Error (RMSE): 0.29
Mean Absolute Error (MAE): 0.18
R-squared (R2) Score: 0.84
Explained Variance Score: 0.84
```

Fig. 8(v): Final Results of SVR Model that we have used

(vi). Result of the predicted force code which gives the predicted result based on SVR:

```
Enter Frequency: 0.5
Enter Velocity: -31.95652174
Enter Current: 1
Predicted Force: -0.75
```

Fig. 8(vi): By taking user inputs of features our SVR Model predicts the force and here is an example.

(vii). Result of Random Forest Model:

```
Best Hyperparameters: {'max_depth': 1000, 'min_samples_split': 10, 'n_estimators': 100}
Mean Squared Error (MSE): 0.07
R-squared (R2) Score: 0.87
```

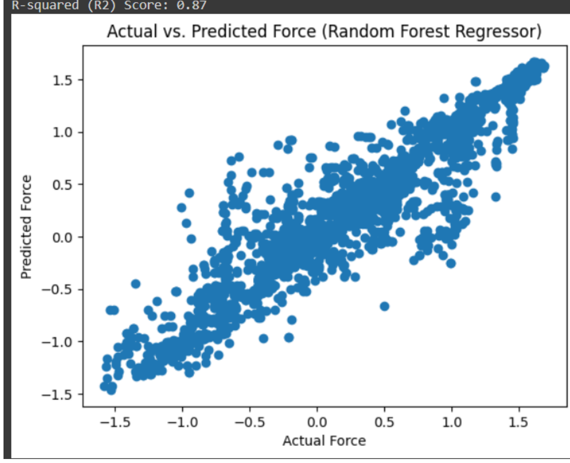


Fig. 8(vii): Results obtained by applying Random Forest with accuracy of 0.87

(viii). Results of Neural Network:

```
82/82 [==============================] - 0s 1ms/step - loss: 0.0822
Mean Squared Error (MSE): 0.08
82/82 [==============================] - 0s 942us/step
R-squared (R2) Score: 0.85
```

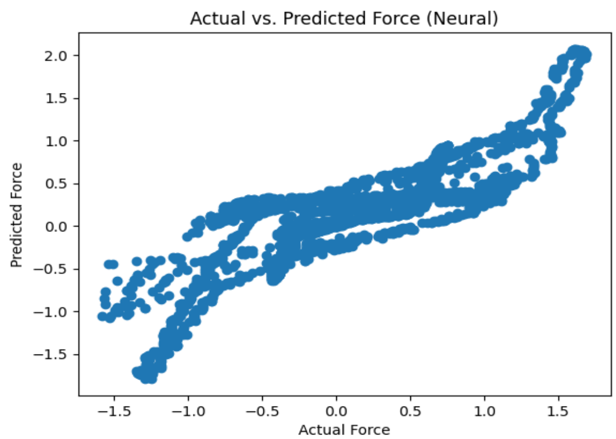


Fig. 8(viii): Both images show the results obtained by Neural Networks with accuracy of 0.85

(ix). Results of Linear Regression:

```
Mean Squared Error (MSE): 0.14
R-squared (R2) Score: 0.74
```

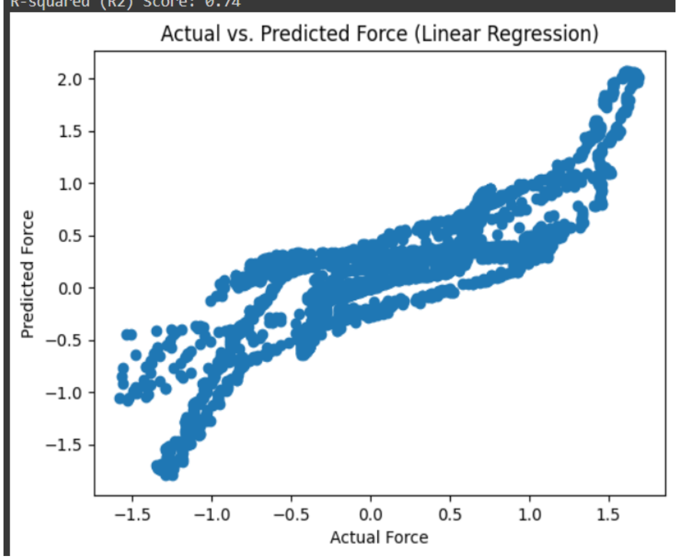
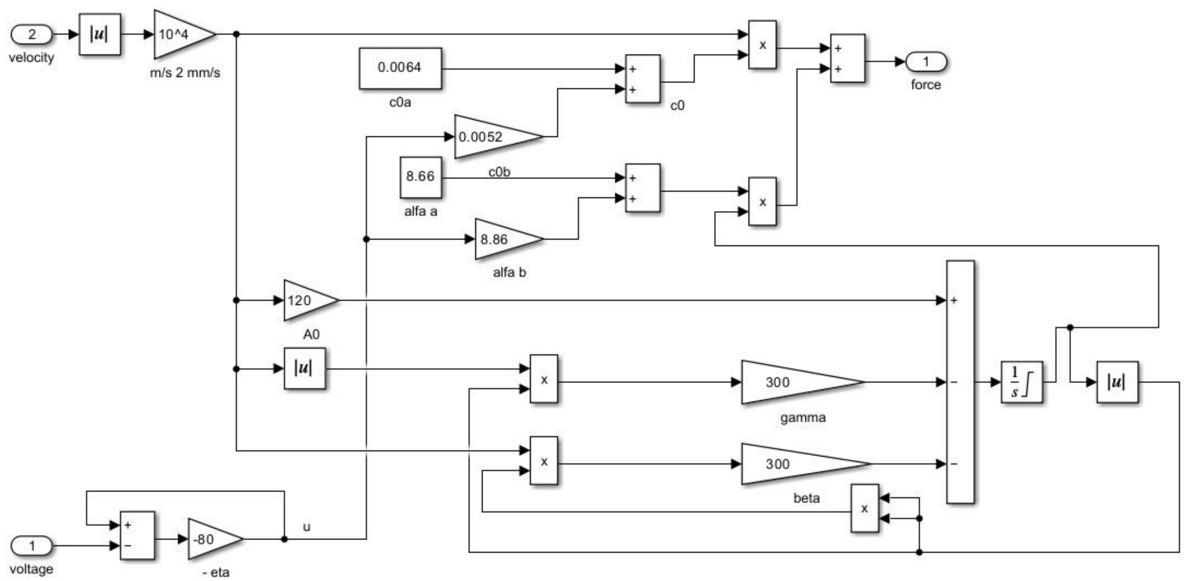


Fig. 8(ix): Results obtained by Linear Regression with the lowest accuracy of 0.74

3.3 Results of Simulink Model:

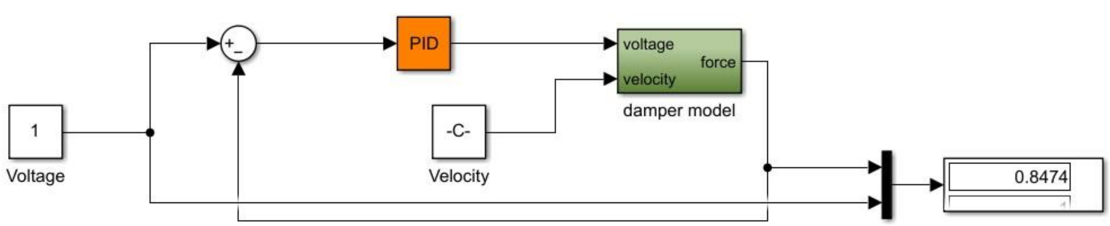
We have run our model for Velocity = 31.95652174, Frequency = 0.5, Current = 1

(i). MR Damper Model:



(ii). Force Predictor using MR damper Model:

Here, we have assumed Resistance = 1. So, Voltage = 1 (as I = 1).



So, it is giving output as 0.8474 which is nearby to the actual force value i.e. 0.853 at velocity = 31.95652174

4. Conclusion

In conclusion, our research project represents a significant stride in the enhancement of Magnetorheological (MR) fluid actuators for vehicle suspension systems through the integration of machine learning methodologies. The utilization of Support Vector Regression (SVR) stands out as a pivotal element in achieving accurate predictions of MR damper forces, contributing to the improvement of suspension system performance without relying on actual data.

Our project began by implementing advanced contour analysis to extract detailed information from experimental graphs, forming a robust dataset crucial for analysis. The precision of this data extraction process played a key role in ensuring reliable device performance analysis, providing a nuanced understanding of MR fluid actuators' behavior.

SVR, as a machine learning model tailored for continuous data, played a central role in accurately modeling the intricate relationships between input parameters and actuator responses. Its unique capability to handle dynamic data proved crucial in predicting the behavior of MR fluid actuators under varying conditions, offering high accuracy in capturing nuanced responses. The positive results obtained through SVR implementation emphasize its efficacy in the specific context of MR fluid actuators, showcasing its ability to predict forces and contribute to the optimization of suspension systems.

Furthermore, our research methodology extended to include a comparative analysis with other machine learning models, such as Neural Networks. This comparative approach underscored SVR's effectiveness and highlighted its superiority in crafting intelligent suspension systems for vehicle applications. The objective was to identify the most suitable machine learning approach for optimizing actuator performance.

The seamless integration of machine learning insights into a Simulink model acted as a virtual testing ground, bridging theoretical predictions with real-world applications. This integration allowed for the translation of SVR predictions into practical insights, facilitating the development and optimization of MR fluid actuators for use in vehicle suspension systems.

By predicting forces without relying on actual data, our project contributes to the advancement of intelligent control systems, aiming to achieve optimal ride comfort, stability, and handling in vehicles. The adaptability of MR fluid actuators and the precision of machine learning showcased in our research underscore the potential for crafting intelligent suspension systems.

In conclusion, the synergistic integration of MR fluid actuators, SVR modeling, and machine learning methodologies represents a significant advancement in designing and optimizing vehicle suspension systems. The success of the SVR model in predicting forces without actual data underscores its relevance and effectiveness in enhancing the performance of MR damper actuators for improved suspension systems.

Applications in Vehicle Dynamics and Suspension:

Our research project, integrating machine learning methodologies, particularly Support Vector Regression (SVR), enhances Magnetorheological (MR) fluid actuators for vehicle suspension systems. Through advanced contour analysis and SVR, we achieve precise modeling of complex relationships between input parameters and actuator responses, optimizing ride comfort and stability. Notably, our predictive model allows estimating MR damper forces without actual data, offering adaptability to varying conditions. The integration into a Simulink model provides a virtual testing ground, bridging theory with application, contributing to the development of intelligent control systems. Overall, our project showcases the potential for SVR and machine learning in crafting responsive, efficient, and adaptive suspension systems, making impactful contributions to the automotive industry.

---

5. Future Work:

Future work in this research project could explore the refinement and expansion of machine learning models beyond Support Vector Regression (SVR), incorporating emerging techniques or hybrid models for even greater predictive accuracy. Further investigation into real-time implementation and validation of the developed intelligent suspension systems in physical vehicles could provide valuable insights into practical performance. Additionally, considering the integration of sensor data and feedback mechanisms could enhance the adaptability of the system to dynamic driving conditions. Exploring the scalability of the proposed approach to diverse vehicle types and conducting long-term durability assessments would contribute to the broader applicability and robustness of the developed intelligent suspension system. Finally, collaborative efforts with industry partners for real-world testing and validation could facilitate the transition of this research from a theoretical framework to a practical solution, fostering advancements in the field of vehicle dynamics and suspension systems.

---

6. Appendix:

A.1 Github Link:

https://github.com/garvitcreeper/MAGNETORHEOLOGICAL-MR-FLUID-ACTUATORS-USING-MACHINE-LEARNING

All the code and packages are provided on our github repository.

---

7. References:

1. Graczykowski, C., & Pawłowski, P. (2017). Exact physical model of magnetorheological damper. *Applied Mathematical Modelling*, 47(2017), 400-424.

2. K. D. Saharuddin, M. H. M. Ariff*, K. Mohmad, I. Bahiuddin, Ubaidillah, S. A. Mazlan, N. Nazmi, and A. Y. A. Fatah Prediction Model of Magnetorheological (MR) Fluid Damper Hysteresis Loop using Extreme Learning Machine Algorithm.

3. https://in.mathworks.com/matlabcentral/fileexchange/51131-gaot-ecm-seismic-vibration-case-study?s_tid=srchtitle_support_results_1_mr%20damper

4. Simulation of Active Force Control Using MR Damper in Semi Active Seat Suspension System R Rosli et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1062 012005

5. PERFORMANCE TEST AND MATHEMATICAL MODEL SIMULATION OF MR DAMPER C. Wu 1 , Y.C. Lin 2 and D.S. Hsu 3

6. Applications of Magnetorheological Fluid Actuator to Multi-DOF Systems: State-of-the-Art from 2015 to 2021 Jong-Seok Oh 1,2 , Jung Woo Sohn 3,* and Seung-Bok Choi 4,5,*
