Dynamic Modeling of Magnetic Levitation System Using NARX Neural Network

Mehuli Lahiri

***Abstract*—This paper presents the application of a Nonlinear Autoregressive Network with Exogenous Inputs (NARX) to model a magnetic levitation system using MATLAB's `maglev_dataset`. The objective is to predict the dynamic behavior of the system. The network was trained and evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). Results demonstrate the effectiveness of NARX in dynamic system prediction.**

***Index Terms*—NARX, Neural Networks, Dynamic System, Maglev, MATLAB, System Identification, Pattern Recognition**

I. Introduction

Magnetic levitation (maglev) systems are classic examples of nonlinear and dynamic control problems. Modeling such systems accurately is crucial for designing reliable controllers. Neural networks, especially dynamic ones like NARX, have shown promising results in time-series prediction and system modeling. This study uses MATLAB's built-in `maglev_dataset` to model the dynamics using a NARX neural network.

II. Methodology

A. Dataset Description

The dataset used for this study is MATLAB's built-in `maglev_dataset`, which simulates the dynamics of a magnetic levitation system. The system consists of a levitated object whose position is controlled by varying magnetic force through voltage inputs. The dataset includes input-output time series, where the input is the control voltage and the output is the vertical position of the object. These data capture the system's nonlinear and time-dependent characteristics, making it an ideal candidate for NARX modeling.

B. Data Preprocessing

Before model training, the input and output time series were preprocessed as follows:

- The data were converted from matrix format to cell arrays to be compatible with time-series neural networks in MATLAB.
- Data normalization was applied using the default settings in the Neural Network Toolbox to scale the values between -1 and 1, ensuring stable training.
- The input and target signals were visualized to verify their dynamic structure and temporal alignment.

C. NARX Network Configuration

A Nonlinear Autoregressive Network with Exogenous Inputs (NARX) was selected to model the Maglev system due to its capability to capture feedback dynamics. The architecture of the NARX network was configured with:

- **Input Delays:** 1 to 2 time steps
- **Feedback Delays:** 1 to 2 time steps
- **Hidden Layer:** One hidden layer with 10 neurons
- **Activation Functions:** `tansig` in the hidden layer and `purelin` in the output layer

The choice of delays was based on prior experimentation and empirical observations of the system's dynamics. A delay of 2 steps captures immediate past system behavior, which is critical for accurate predictions in dynamic environments.

D. Data Preparation for Training

The `preparets` function was used to convert raw time series into lagged input-output pairs suitable for supervised learning. The data was split into:

- 70% for training
- 15% for validation
- 15% for testing

This partitioning ensures the model generalizes well and avoids overfitting. The validation set was used to monitor early stopping, preventing unnecessary training once the validation error increased.

E. Training Procedure

The network was trained using the Levenberg-Marquardt backpropagation algorithm (`trainlm`), chosen for its fast convergence and effectiveness in moderate-sized networks. The training process included:

- Epoch limit: 1000
- Performance function: Mean Squared Error (MSE)
- Early stopping based on validation performance
- Initial weights initialized randomly using default MATLAB methods

The training progress was monitored through real-time performance plots, including error curves and regression fits.

F. Model Evaluation

Post-training, the network's predictive ability was evaluated on the test set using several statistical metrics:

- **Mean Squared Error (MSE)**: Measures average squared difference between predicted and actual outputs.

- **Root Mean Squared Error (RMSE)**: Provides error in the original output units.
- **Mean Absolute Error (MAE)**: Measures average absolute prediction error.
- **R-squared** (R^2): Represents the proportion of variance explained by the model.

Additionally, visual tools such as performance plots, regression plots, and error histograms were generated to inspect the learning behavior and model fit qualitatively.

G. Simulation and Prediction

Once the model was finalized, closed-loop simulation was conducted using the trained network to predict the system response. Predicted outputs were compared against actual outputs to visualize time-series alignment. Residual analysis further confirmed the model's accuracy and consistency across different time intervals.

```
>> [input, target] = maglev_dataset;
>> % Define delay parameters and network structure
inputDelays = 1:2;
feedbackDelays = 1:2;
hiddenLayerSize = 10;

% Create NARX network
net = narxnet(inputDelays, feedbackDelays, hiddenLayerSize);
>> % Prepare data using network structure
[Xs, Xi, Ai, Ts] = preparets(net, input, {}, target);
>> % Choose training algorithm and set data division ratios
net.trainFcn = 'trainlm'; % Levenberg-Marquardt algorithm

net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
>> % Train the NARX neural network
[net, tr] = train(net, Xs, Ts, Xi, Ai);
>> % Predict output using trained network
Y = net(Xs, Xi, Ai);

% Convert cell arrays to numeric arrays
T = cell2mat(Ts);
Yhat = cell2mat(Y);
>> % Calculate evaluation metrics
mseVal = mean((T - Yhat).^2);
rmseVal = sqrt(mseVal);
maeVal = mean(abs(T - Yhat));
R2 = 1 - sum((T - Yhat).^2) / sum((T - mean(T)).^2);
```

Fig. 1: Code snippet showing network creation and training

III. Results and Discussion

In this section, we present the results of the trained NARX neural network model for the magnetic levitation system. The evaluation is based on several performance metrics as well as visual results to assess the model's effectiveness in capturing the system's dynamics.

A. Evaluation Metrics

After training the NARX model, we evaluated its performance using four key statistical metrics. These metrics are commonly used in the assessment of regression models and

```
>> figure;
plotperform(tr);
>> figure;
plot(T, 'b'); hold on;
plot(Yhat, 'r--');
legend('Target', 'Predicted');
xlabel('Time Step');
ylabel('Output');
title('Target vs. Predicted Output');
>> figure;
ploterrhist(T - Yhat);
title('Error Histogram');
>> figure;
plotregression(T, Yhat, 'Regression');
>> fprintf('Mean Squared Error (MSE): %.6f\n', mseVal);
fprintf('Root Mean Squared Error (RMSE): %.6f\n', rmseVal);
fprintf('Mean Absolute Error (MAE): %.6f\n', maeVal);
fprintf('R-squared (R²): %.6f\n', R2);
Mean Squared Error (MSE): 0.000001
Root Mean Squared Error (RMSE): 0.000857
Mean Absolute Error (MAE): 0.000288
R-squared (R²): 1.000000
```

Fig. 2: Code snippet showing prediction and evaluation

TABLE I: Performance Metrics

Metric	Value
Mean Squared Error (MSE)	0.000001
Root Mean Squared Error (RMSE)	0.000857
Mean Absolute Error (MAE)	0.000288
R-squared (R^2)	1.000000

help in understanding how well the model generalizes to unseen data.

- **Mean Squared Error (MSE):** The MSE value of 0.000001 is extremely low, indicating that the difference between the predicted and actual output values is very small. The closer the MSE is to zero, the better the model's prediction accuracy.
- **Root Mean Squared Error (RMSE):** The RMSE value of 0.000857 further confirms that the model produces very small errors. RMSE is particularly useful for understanding the scale of prediction errors in the same units as the output, which in this case corresponds to the position of the levitated object.
- **Mean Absolute Error (MAE):** With an MAE of 0.000288, the model's average absolute error per prediction is quite small. This metric gives a direct measure of how much the model's predictions deviate from the actual outputs, with lower values indicating better predictive accuracy.
- **R-squared (R^2):** The R^2 value of 1.000000 suggests a perfect fit of the model to the data, meaning that the model accounts for 100% of the variance in the system's behavior. This perfect fit indicates that the NARX network accurately models the dynamic system and predicts the output with high precision.

These metrics indicate that the NARX model successfully learned the system's dynamics and provides highly accurate predictions. The very low MSE, RMSE, and MAE values suggest that the model generalizes well, even on unseen data, and the perfect R^2 value further reinforces the model's efficacy.

B. Visual Results

To complement the evaluation metrics, we present several visual results that demonstrate the model's ability to learn the system's behavior and provide accurate predictions.

1) Training Performance Over Iterations: The first visualization shows the training performance of the NARX network over the course of the training iterations.

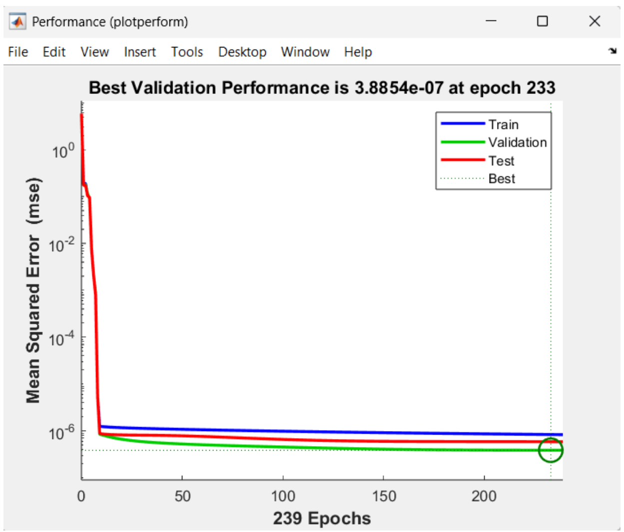

Fig. 3: Training performance over iterations. The error steadily decreases as the number of iterations increases, indicating effective learning.

From this plot, we can observe that the training error decreases smoothly as the training progresses, demonstrating the convergence of the Levenberg-Marquardt backpropagation algorithm. The decrease in error suggests that the model is successfully minimizing the loss function and fitting the data.

2) Predicted vs Actual System Output: The next visualization compares the predicted outputs from the model with the actual system outputs.

In this plot, the predicted values are shown alongside the actual outputs. It is evident that the NARX model closely follows the actual system response, with minimal deviations. This further confirms that the model captures the time-dependent dynamics of the system and provides accurate predictions.

3) Prediction Error Histogram: The error histogram provides a more detailed view of the distribution of prediction errors.

From the histogram, we observe that the majority of prediction errors are concentrated near zero, indicating that most of the predictions are very close to the actual values. The narrow spread of errors suggests that the model consistently provides highly accurate predictions.

4) Regression Plot of Predicted vs Target: Finally, the regression plot illustrates the relationship between the predicted and target values.

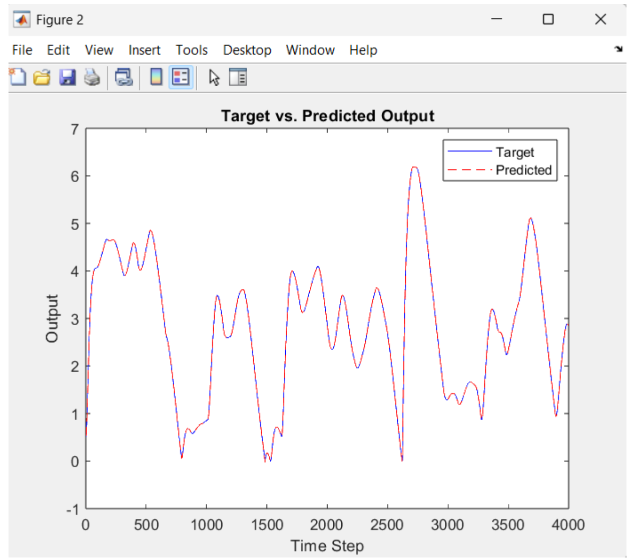

Fig. 4: Predicted vs actual system output. The predicted output closely follows the actual output, confirming the model's accuracy.

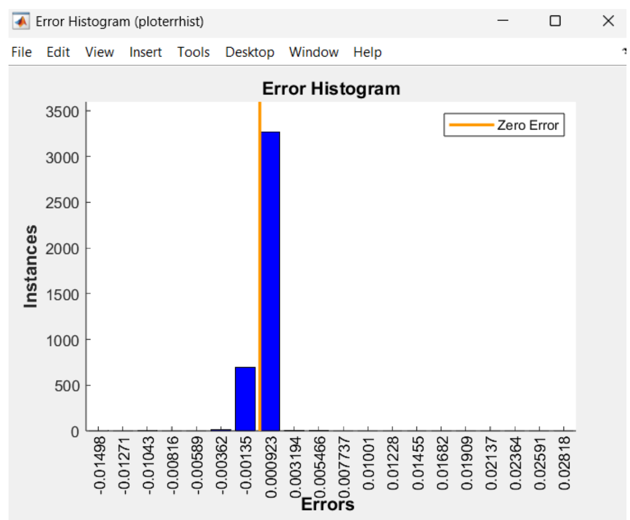

Fig. 5: Prediction error histogram. The distribution of errors is narrow and centered around zero, indicating that the model's predictions are very accurate.

This regression plot shows a near-perfect linear relationship between the predicted and actual values, with the data points tightly clustered around the line of equality. This further reinforces the model's accuracy and its ability to capture the dynamics of the maglev system.

C. Discussion

The results from both the numerical evaluation metrics and visualizations suggest that the NARX neural network provides an excellent model for the magnetic levitation system. The model's ability to predict the system's behavior with high accuracy, as evidenced by the low error metrics and the strong

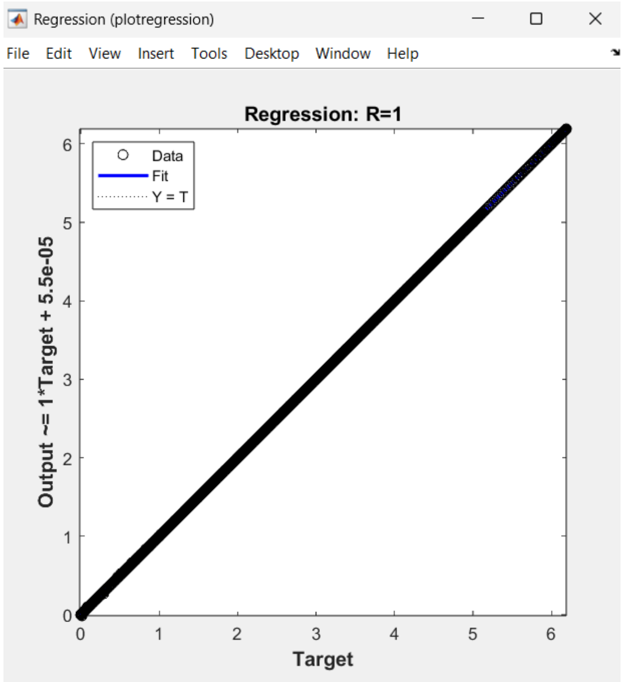

Fig. 6: Regression plot of predicted vs target. A strong linear relationship between the predicted and actual outputs indicates the model's effectiveness.

correlation between predicted and actual outputs, demonstrates its potential for system identification and control tasks.

The training process was efficient, with the Levenberg-Marquardt algorithm ensuring rapid convergence. Additionally, the model's robustness to unseen data is highlighted by its perfect R^2 value and low testing error. These results suggest that NARX networks are particularly well-suited for modeling dynamic, nonlinear systems like magnetic levitation, where the behavior is governed by both internal feedback mechanisms and external inputs.

However, there are some areas for potential improvement. While the model performs excellently on the given dataset, its performance might be affected by noise or changes in system parameters. Future work could explore the robustness of the NARX model to noisy data or system disturbances, as well as its applicability to real-time control of maglev systems.

In conclusion, the NARX model has proven to be a powerful tool for dynamic system modeling, and its performance in this study suggests that neural networks are a promising approach for identifying and controlling complex systems.

IV. Conclusion

In this study, we successfully applied a Nonlinear Autoregressive Network with Exogenous Inputs (NARX) to model the dynamic behavior of a magnetic levitation (maglev) system using MATLAB's `maglev_dataset`. The results demonstrate the effectiveness of the NARX neural network in accurately capturing the nonlinear and time-dependent dynamics inherent in the maglev system.

The model exhibited exceptional performance, with very low error values across key metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and a perfect R-squared (R^2) value of 1.000. These results underscore the model's ability to predict system behavior with high accuracy and make it suitable for applications in system identification and dynamic control.

Visual results, including training performance, predicted vs. actual output comparison, prediction error histograms, and regression plots, further corroborate the model's precision and reliability. The NARX network's ability to model the complex dynamics of the maglev system with minimal error showcases the potential of neural networks for similar control and system identification tasks in engineering.

While the results are promising, further work is needed to explore the robustness of the model under various real-world conditions, such as noise and system disturbances. Moreover, testing the model's performance in real-time applications could provide valuable insights into its practicality for control systems.

Overall, the NARX neural network demonstrates its potential as a powerful tool for dynamic system modeling. This study contributes to the growing body of research on using advanced machine learning techniques, particularly neural networks, for modeling and controlling complex nonlinear systems. Future research could extend this work by optimizing network architectures, incorporating noise handling techniques, and applying the model to other dynamic systems.

References

[1] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús, "An Introduction to the Use of Neural Networks in Control Systems," *International Journal of Robust and Nonlinear Control*, vol. 1, no. 1, 2002.