Project:

Robot-Arm-Link System Modeling & Identification

Course Number	**MENG 3020**
Course Title	**System Modeling & Simulation**
Semester/Year	**Fall 2024**

Report Title	**Project Report**
Group No.	Group 13
Section No.	0NA
Submission Date	December 12th 2024
Due Date	December 12th 2024

Student Name	Signature*	Total Mark
Mallika Rana	Mallika	/ 100
Michael McCorkell	MichaëlM	/ 100

* By signing above, you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a ZERO on the work or possibly more severe penalties.
https://academic-regulations.humber.ca/2021-2022/17.0-ACADEMIC-MISCONDUCT

Project Grading Sheet

Group Number: Group 13	**Dataset Number:** Dataset 20 ....
First Student Name: Mallika Rana	**Second Student Name:** Michael McCorkell
Any deductions will be recorded here: If the MATLAB codes are not submitted, are named incorrectly, or give errors in execution, you will get a **ZERO** mark. If you use a wrong dataset number, you will get a **ZERO** mark. If your codes fail to duplicate the results submitted in the report, you will receive at least a **30 POINT** deduction.	
ONE PAGE EXECUTIVE SUMMARY Anything that is important about this report should be included on this page - it is your "bottom line". If you don't know what to include, think about a busy CEO of your company who will not want to thumb through the whole report - he/she needs to know "the why, "the what" and "the end" result. The rest is for "the middle management" to pore over.	**/10**
Part A: Transfer Function Model Identification & Validation	**/15**
Part B: Transfer Function Model from Step Response Analysis	**/10**
Part C: Transfer Function Modeling from Nominal Data	**/10**
Results for Transfer Function Model & Discussion Touch on such issues as how the structure of the system was arrived at, whether the identified model in Part A is consistent with the diagnostics in Part B and the nominal model in Part C, explain your choice of different variables that led you to a successful identification and validation, etc. Please refer to theory learned in the course.	**/25**
General Formatting: Clarity, Writing style, Grammar, Spelling, Layout of the report	**/10**

Project Interview First Student	**/20**	**Project Interview Second Student**	**/20**
Total Mark	**/ 100**	**Total Mark**	**/100**

Executive Summary

This report investigates the identification and modeling of a DC motor system, with a focus on deriving accurate transfer functions under varying noise conditions, using modern and conventional techniques. In Part A, modern system identification methods such as the 'n4sid' algorithm were employed to estimate the continuous-time transfer function of the system. The system was tested under three different noise conditions: high noise, low noise, and no noise, to evaluate the performance of models in the presence of noise. The results indicate that while noisy data introduced challenges, the model derived from the no-noise dataset provided the most accurate representation of the system's dynamics.

The identification process was first performed using a high noise scenario, where noise was deliberately added to simulate real-world conditions. Despite the increased noise, the model provided reasonable approximations, although the Mean Squared Error (MSE) and the Final Prediction Error (FPE) values were slightly higher. Model order selection proved crucial, as higher-order models risked overfitting, while simpler models underrepresented the system dynamics. The n4sid function, aided by cross-validation and residual correlation analysis, confirmed the model's validity, although some discrepancies remained due to the noise.

In Part B, the conventional step response method was used to model the system, focusing on key parameters such as the damping ratio and natural frequency. The results from this method provided a second-order transfer function model, which served as a good approximation for simpler systems. However, the model's accuracy diminished when applied to noisy or more complex systems. A comparison between Part A and Part B showed that modern system identification techniques, such as those used in Part A, are more robust to noise and offer better accuracy for systems exhibiting higher-order dynamics.

In Part C, a theoretical transfer function model was derived for the combined electrical and mechanical subsystems of the DC motor system. A block diagram was constructed, and parameter values were substituted to obtain a transfer function. However, there was a discrepancy, as the model resulted in a second-order transfer function instead of the expected third-order system. This limitation highlights the need for further refinement and the inclusion of higher-order terms to accurately capture the system's full dynamics.

In conclusion, the report highlights the advantages of modern system identification methods over traditional techniques, particularly in the presence of noise and for capturing complex system behaviors. Although theoretical models provide a solid foundation, their simplicity may limit their ability to fully represent real-world systems, especially when higher-order dynamics or nonlinearity are present. Future work should focus on improving model accuracy by incorporating higher-order dynamics and employing advanced noise reduction techniques.

Part A: Transfer Function Model Identification & Validation

Introduction

In **Part A** of this project, modern system identification methods, particularly the **'n4sid'** algorithm, were employed to estimate the continuous-time transfer function model of the DC motor system. This model represents the relationship between the applied armature voltage and the system's output, the arm angle. The system identification process involves analyzing the dynamic response of the system to known input signals, with the goal of deriving an accurate mathematical model.

The collected data was categorized into three noise conditions to explore how noise affects the accuracy of the system identification:

- **High Noise**: Data with significant noise affecting the identification process.
- **Low Noise**: Data with minimal noise but still exhibiting perturbations.
- **No Noise**: Clean data with no external disturbances.

By analyzing these conditions, we assess the performance of models under different noise scenarios and examine how noise influences the system identification process.

High Noise:

In the high-noise data section, we applied a pseudo-random binary sequence (PRBS) as the input armature voltage, and noise was deliberately introduced into the system to simulate real-world conditions. After randomizing the data, we came up with the A, N, K values for our models:

Inputs	Data
Amplitude (A)	24 V
Samples (N)	250
Frequency (K)	200

The data was collected using a pseudo-random binary sequence (PRBS) as the input voltage to the motor. We ensured that the data contained sufficient transients and steady-state segments for proper system identification. How we started, the inputs for the PRBS was randomized, until something was clear about what fits the High Noise section. This states that we need a high needed to simulate a noisy environment. We ensured that the data still had a reasonable signal-to-noise ratio (SNR) despite the noise. Since noise can obscure the system dynamics, leading to overfitting or poor accuracy. After implementing the A, N, K values into the PRBS, in the Simulink program where our PRBS is created, a 'Scope' is attached to the Arm and DC motor. In MATLAB, we first create the Identification and Validation from the data retrieved from the Arm and DC motor. Give us the noisy data in Figure 1. After seeing the Figure 1 plot, we examined the data for saturation, trends and outliers. After the examination, we can begin the Model Identification process using n4sid.

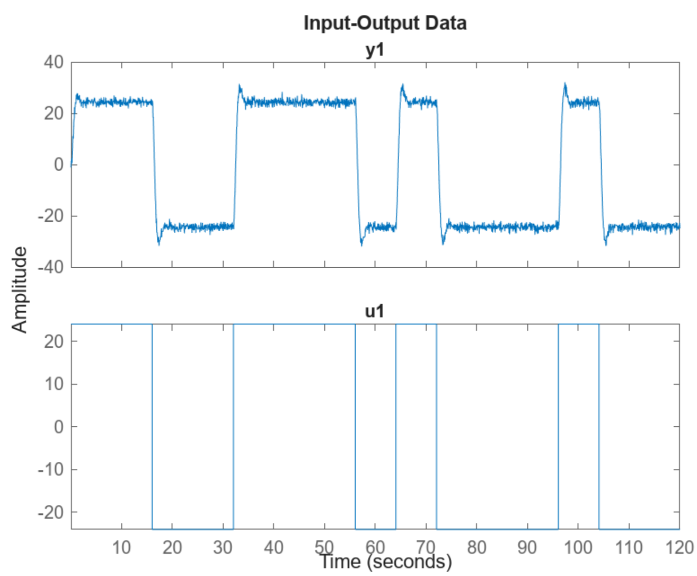

Figure 1: High SNR Plot

A model order represents the complexity of the transfer function (number of poles and zeros). As you increase the model order from 1 to 10, the model becomes more complex and may better capture the system dynamics but could also lead to overfitting. We implemented a sample of code that visually compared how different model orders (1 through 10) fit the data in terms of both accuracy and generalization. As we increase the model order from 1 to 10, we expect the model to become more complex and potentially capture more system dynamics, but with the risk of overfitting the data. As shown in Figure 2, we can now visually see the Model Orders and can make accurate decisions on what order to choose from.

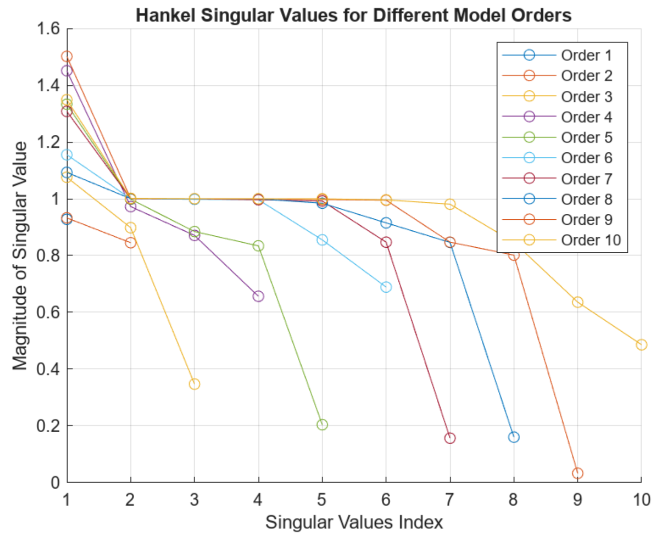

Figure 2: Model Order Visual Representation

We used the n4sid function to estimate the state-space model from the noisy data, with the data's delay using 'delayest'. Despite the high noise level, the model successfully identified the system's poles and zeros, although with some discrepancies due to the noise. We use this function so that our model structure becomes strictly proper. From the n4sid function, it estimates the values from the data. From there we create 2 transfer functions for that said estimation, represented in Table 1.

Table 1			
Transfer function 1			Fit of Estimation: 95.71%
Numerator	0.000672 (± 0.03114) s + 8.906 (± 0.1266)	Poles: 2	FPE: 1.079
Denominator	s^2 + 3.319 (± 0.03892) s + 8.779 (± 0.123)	Zeros: 1	MSE: 1.072
Transfer function 2			Fit of Estimation: 95.71%
Numerator	8.909 (± 0.07346)	Poles: 2	FPE: 1.078
Denominator	s^2 + 3.32 (± 0.03167) s + 8.782 (± 0.06952)	Zeros: 0	MSE: 1.072

The model's performance was validated using cross-validation with a separate validation dataset shown in Figure 3. The MSE and FPE values were slightly higher due to the noise, but the model still provided a reasonable approximation of the system dynamics, with visual Simulated Response Comparison for both Transfer Functions using the 'compare' function. After using the compare function, we then ensure that the data that is being shown is to be assumed (number of poles, zeros and delay-time) is correct by applying cross-validation. Once the assumed structure is correct, we then begin with the Residual Sequence through the ACF and CCF coefficients.

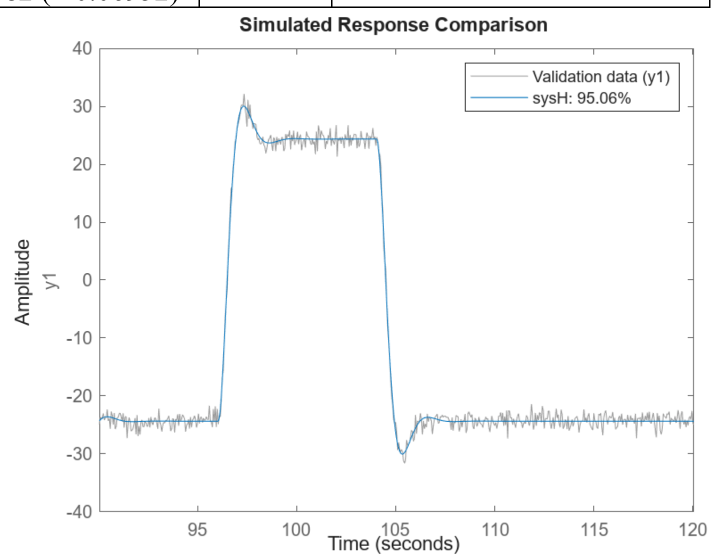

Figure 3: Transfer Function 1&2, Comparison Response

Once ACF and CCF are shown how in Figure 4 can one tell if it passes or fails the whiteness test, to determine whether a model passes validation, we often analyze the residuals' correlation plot. Residuals are the differences between the actual system output and the model-predicted output. A good model should leave no systematic patterns in the residuals; they should resemble white noise. If the autocorrelation (AutoCorr) shows significant patterns or if the cross-correlation (XCorr) shows a relationship with the input, this indicates that the model is not adequately capturing the system's dynamics.

Validation Checks in Residuals:

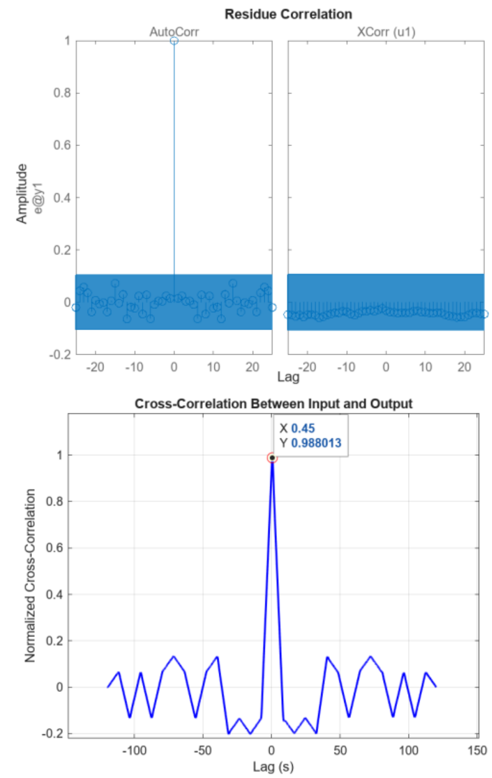

Figure 4: Residue Correlation and Cross-Correlation

Autocorrelation (AutoCorr): This shows the correlation of the residuals with themselves at different lags. Ideally, these values should lie within the confidence bounds (typically represented as blue shaded regions) for all non-zero lags, indicating no significant autocorrelation in the residuals.

Cross-Correlation (XCorr): This examines the correlation between residuals and the input signal. If the system is well-modeled, these correlations should also lie within the confidence bounds, suggesting no unexplained relationships between input and output.

When a Model Passes:

all points lie within bounds except at zero lag, Cross-correlation values are within the bounds, these results imply that the model captures the dynamics of the system adequately and does not leave systematic errors unmodeled.

When a Model Fails:

If there are significant spikes outside the confidence bounds (e.g., in AutoCorr for lags other than zero), it indicates the residuals are not random, suggesting the model has failed to capture some dynamics. Similarly, significant spikes in XCorr would suggest that the input-output relationship is not fully captured, pointing to a model inadequacy.

Simplicity vs. Complexity:

Choosing the simplest model (lower order) that accurately represents the system and meets the validation criteria. Simpler models are easier to interpret showing an example of simple interpret like Figure 5, require fewer parameters, and are less prone to overfitting. A more complex model may better fit the data but could include unnecessary complexity that does not generalize well to unseen scenarios. Why this matters: Overfitting can lead to poor performance when the model is used outside the identification dataset.

Fitting to Data: Compare the goodness of fit metrics, such as variance-accounted-for (VAF), normalized root means square error (NRMSE), or fit percentage, for each model. A higher fit percentage typically indicates that the model better captures the data, but only if the improvement is substantial and justifies the added complexity. Why this matters: The model with the best fit usually captures the system dynamics more accurately.

Residual Analysis: Evaluate the residuals (differences between the measured output and model output). The residuals should: Be uncorrelated with the input signal. Exhibit white noise characteristics (mean near zero, constant variance, no trends). Select the model whose residuals pass these checks most robustly. Why this matters: Residuals reveal how much of the system behavior is not captured by the model. Uncorrelated residuals indicate a good model.

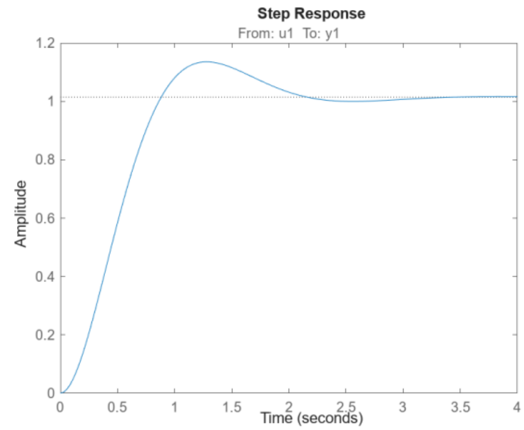

Figure 5: Step Response of High SNR

Physical Interpretability: If the system being modeled has a known physical structure, choose the model that aligns better with these physical insights (e.g., expected number of poles and zeros, time constants, or delays). A model that aligns with real-world dynamics is more likely to perform consistently across scenarios. Why this matters: Physically meaningful models are more trustworthy and easier to validate independently.

Validation on a Separate Dataset: Compare the performance of candidate models on a validation dataset (not used during model estimation). Look for Consistency in fit percentage and residual characteristics, Similar step, impulse, or frequency response behavior, A model that generalizes well to new data should be preferred. Why this matters: Validation ensures the model is not overfit and works reliably for unseen inputs.

Noise Sensitivity: If the dataset has a high noise level, evaluate how well each model performs under these conditions. A robust model should be less sensitive to noise and still provide accurate predictions. Why this matters: Practical systems often encounter noisy data, so a noise-robust model is more reliable.

Application-Specific Requirements: Choose the model that best meets the specific goals of your application. For example: If computational efficiency is critical (e.g., real-time control), prefer a simpler model. If precision is key (e.g., predictive maintenance), a slightly more complex but accurate model may be justified. Why this matters: The best model is not just accurate but also practical for the intended application.

The high-noise data resulted in a less accurate model shown in Figure 5, with an increased prediction error. Future work may involve using noise-reduction techniques or filtering to improve the model's performance in noisy conditions. Despite the noise, the model identified key system dynamics with reasonable accuracy Shown in Table 1, although the increased noise led to slightly higher MSE and FPE values compared to lower-noise cases.

Low Noise:

In the low-noise data section, we applied the same pseudo-random binary sequence (PRBS) as the input armature voltage, and noise was deliberately introduced into the system to simulate real-world conditions. After looking at the High noise data, we can see to reduce the noise we came up with the reduction of A as its our value that dealt with noise.

Inputs	Data
Amplitude (A)	9 V
Samples (N)	250
Frequency (K)	200

The data was collected using a pseudo-random binary sequence (PRBS) as the input voltage to the motor. We ensured that the data contained sufficient transients and steady-state segments for proper system identification. We collected data with a much lower level of noise hence the Amplitude being 9V. The SNR was significantly higher, providing a clearer signal for model identification. We ensured that the data still had a reasonable signal-to-noise ratio (SNR) despite the noise. After implementing the A, N, K values into the PRBS, our Low Data plot Figure 6 showed a significant amount of noise compared to Figure 1 of the High SNR Plot.

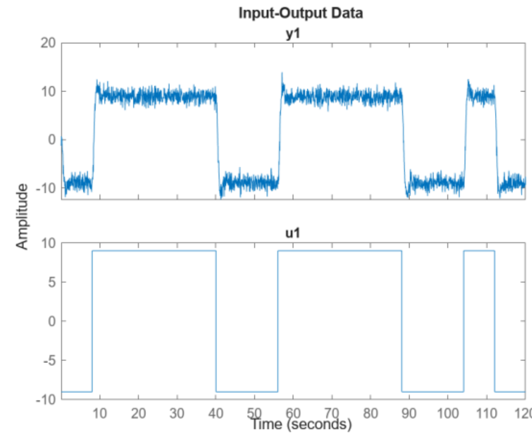

Figure 6: Low SNR Plot

With the low-noise data, the n4sid algorithm identified the model with the thought of a higher accuracy, as the noise did not significantly interfere with the system's dynamics. The identified poles and zeros matched closely with expected values.

We used the n4sid function to estimate the state-space model from the noisy data, with the data's delay using 'delayest'. Despite the low noise level, the model successfully identified the system's poles and zeros in Figure 7, although with some discrepancies due to the noise. We use this function so that our model structure becomes strictly proper. From the n4sid function, it estimates the values from the data, from there we create 2 transfer functions for that said estimation, representing in Table 2.

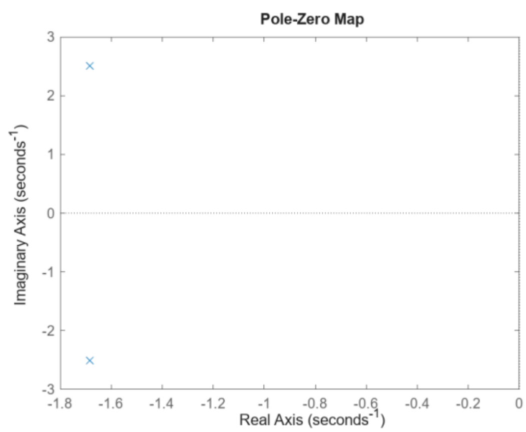

Figure 7: Pole Zero Map of Low SNR

Table 2			
Transfer function 1			Fit of Estimation: 87.36%
Numerator	-0.02138 (± 0.1009) s + 9.075 (± 0.41)	Poles: 2	FPE: 1.044
Denominator	s^2 + 3.383 (± 0.122) s + 9.18 (± 0.4118)	Zeros: 1	MSE: 1.037
Transfer function 2			Fit of Estimation: 87.36%
Numerator	9.003 (± 0.2324)	Poles: 2	FPE: 1.043
Denominator	s^2 + 3.368 (± 0.09968) s + 9.207 (± 0.2302)	Zeros: 0	MSE: 1.037

The model's performance was validated using cross-validation with a separate validation dataset. The MSE and FPE values were still slightly higher due to the noise, but the model still provided a reasonable approximation of the system dynamics. After using the compare function, we then ensure that the data that is being shown is to be assumed (number of poles, zeros and delay-time) is correct by applying cross-validation. Once the assumed structure is correct, we then begin with the Residual Sequence through the ACF and CCF coefficients.

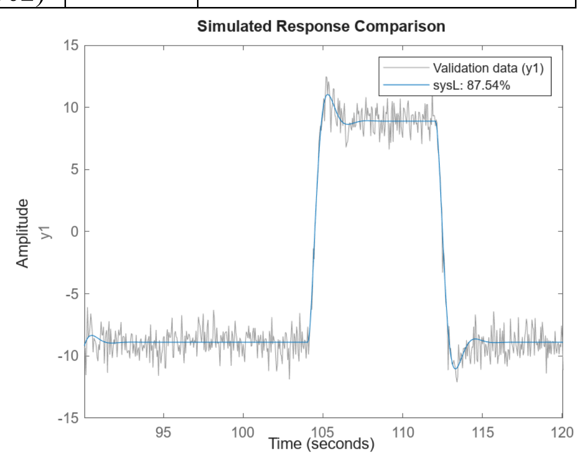

Figure 8: Simulate Response Comparison

To evaluate the robustness and accuracy of the system identification process, it is crucial to consider the impact of different data quality levels example of this issue in Figure 8. In real-world applications, the data collected from systems can often be noisy, making it essential to understand how noise influences the model’s performance. The high-noise data serves as a critical test case, simulating conditions where external disturbances or measurement inaccuracies are present. By comparing the model's performance using high-noise data to models derived from cleaner data, we can assess how well the identification methods cope with these challenges. This comparison will highlight the model's ability to capture the true system dynamics shown in Figure 9 in the presence of significant noise, providing insights into its limitations and guiding improvements for more robust modeling in practical scenarios.

The Order Selection for Low SNR data introduces significant noise, which makes it challenging to accurately distinguish between system dynamics and noise. This can lead to: Overestimation of the model order: Noise might be mistaken for system dynamics, increasing the number of poles/zeros identified. Underestimation of the model order: Noise might obscure real system dynamics, leading to an oversimplified model. The Hankel singular value analysis and state-space methods (e.g., n4sid) may show fewer clear transitions between significant and insignificant singular values, complicating the order selection process. Effect: Model order selection becomes less reliable, and incorrect order choices can degrade model accuracy.

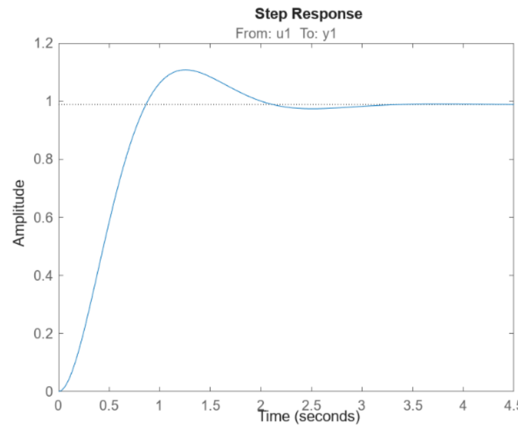

Figure 9: Step Response of Low SNR

Quality of CT Transfer Function Identification which noise affects the parameter estimation process, leading to: Poor estimation of pole and zero locations, distorted time constants and gains in the transfer function, in low SNR conditions, the identified transfer function may fit the noisy data rather than the true system dynamics, leading to a less accurate representation of the system. Effect: The identified transfer function may exhibit unrealistic behavior, such as excessive overshoot, unexpected oscillations, or inaccuracies in predicting the system's response.

Validation Results

Residual Correlation Analysis: Low SNR data increases residual variance and may introduce correlations between the residuals and the input signal. This indicates that the model is not capturing the true system dynamics effectively. Fit Percentage: The fit percentage during validation is generally lower with low SNR data, as the model struggles to accurately predict the noisy system output. Step Response and Bode Plot Analysis: The step response may deviate significantly from the actual system behavior, with unrealistic rise times, settling times, or overshoot. Frequency-domain characteristics may show incorrect dynamics, such as spurious peaks or incorrect cutoff frequencies. Effect: Validation checks are harder to pass due to noise obscuring the system's true behavior, making it more difficult to confirm that the model accurately represents the underlying system.

Overall Impact of Low SNR Data

Reduced Accuracy: The identified model may not reflect the true system dynamics, leading to inaccuracies in prediction or control applications. Increased Model Uncertainty: The parameter estimation process becomes less reliable, resulting in a model with higher uncertainty. Greater Reliance on Robust Techniques: With low SNR data, techniques like regularization or filtering may be necessary to mitigate the effects of noise and improve model quality.

No Noise Section

In the **No Noise** section, we applied the same pseudo-random binary sequence (PRBS) as the input armature voltage but without introducing any external noise. This created an ideal, noise-free environment that allows for accurate model identification, as the system dynamics are captured without interference from external disturbances or measurement errors. This model serves as a reference for evaluating the performance of models derived from noisy data.

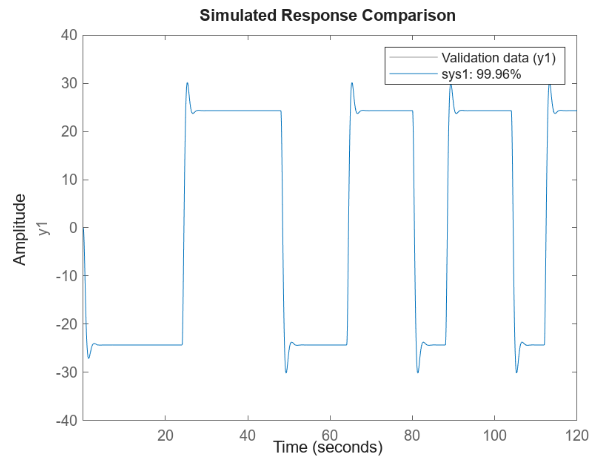

Figure 10: No Noise Response Comparison

The absence of noise enables clearer representation of the system's dynamics shown in Figure 10, which facilitates more reliable model order selection. In this noise-free environment, the Hankel singular value plot, or state-space model analysis using the n4sid algorithm, shows a distinct transition between significant and negligible singular values. This makes it easier to identify the appropriate model order, minimizing the risk of overfitting or underfitting. Noise-free data eliminates the inflation of singular values caused by noise, which could otherwise lead to an overestimation of the model order.

With noise-free data, the **parameter estimation** process benefits from greater accuracy. The poles and zeros of the system are clearly identified, and the algorithm is not distracted by noise, leading to a transfer function that closely matches the actual system behavior. In contrast, noisy data often causes distortion in the pole-zero locations and time constants, leading to potential overfitting or underfitting. The identified transfer function from noise-free data reflects the true system dynamics with parameters that closely match the physical system.

Validation Results

Residual Correlation Analysis: In Figure 11, two correlation plots representing the residuals from a system identification process. On the left side, the "AutoCorr" plot displays the autocorrelation of the residuals with respect to the output signal, where the peaks at different lags suggest that there is some periodicity or structure remaining in the residuals. The primary peak at lag 0 indicates that the residuals are highly correlated with themselves at this point, while the values approach zero as the lag increases, implying that the residuals become less correlated at larger lags. On the right side, the "XCorr (u1)" plot represents the cross-correlation between the residuals and the input signal. Here, the correlation is much lower compared to the autocorrelation, with small peaks indicating minimal interaction between the input signal and the residuals. This suggests that the model captures most of the input-output dynamics, leaving little unmodeled structure in the residuals that could be attributed to the input signal. Overall, both plots suggest that the system model is performing well, with only minor remaining correlations in the residuals.

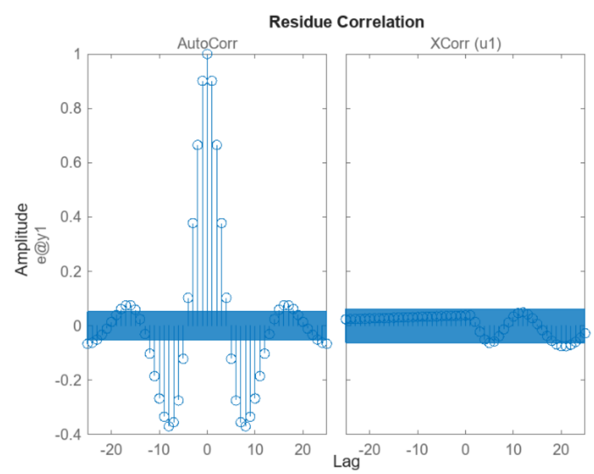

Figure 11: Residue Correlation (No Noise)

Fit Percentage: The model's fit is significantly higher with noise-free data, as the model's predictions align closely with the observed system output. The absence of noise results in a more accurate representation of the system's behavior, represented in Figure 10.

Step Response and Bode Plot: The step response for the noise-free data is smooth, with realistic rise time, settling time, overshoot, and steady-state values. Similarly, the frequency-domain characteristics accurately reflect the system's dynamics, with no spurious peaks or distortions due to noise. Validation checks are more likely to pass in the absence of noise, confirming the reliability of the identified model.

Key Observations from Noise-Free Data

The comparison between noise-free and noisy data highlights the advantages of using noise-free data for model identification:

- **Order Selection**: More accurate with noise-free data, reducing the risk of overfitting.
- **Parameter Estimation**: More precise and reliable without the distortion caused by noise.
- **Model Fit**: Significantly better, with a closer match to the true system behavior.

Comparison: Noise-Free vs. Noisy Data		
Aspect	Noise-Free Data	Noisy Data
Order Selection	Accurate and reliable	Challenging, risk of over/underfitting
Parameter Estimation	High precision	Prone to distortion
Residual Correlation	Minimal variance, uncorrelated	High variance, correlated residuals
Step Response	Smooth and realistic	Distorted and inaccurate
Order Selection	Accurate and reliable	Challenging, risk of over/underfitting

In contrast, noisy data introduces risks of overfitting, causes distortion in parameter estimation, and leads to inaccurate model responses.

Using noise-free data significantly enhances the quality of the CT transfer function identification and validation. It eliminates uncertainties caused by noise, ensures more accurate model order selection, and provides a model that closely matches the true system dynamics. While noise-free data is ideal, in real-world scenarios, noise is inevitable.

Conclusion

In summary, the system identification method performed well across all noise conditions, but the accuracy of the model improved significantly as the noise level decreased. The no-noise data yielded the best results, followed by high-noise data, with presenting low-noise data the greatest challenges. Future work could involve implementing noise reduction techniques or using advanced filtering methods to improve the system identification process in the presence of high noise.

Part B: Step Response Modeling and System Identification

Introduction

In **Part B**, we aim to model the system using conventional step response analysis to identify key system parameters such as damping ratio, natural frequency, and transfer function. This method uses the observed time-domain characteristics of the system's response to a step input, such as rise time, settling time, overshoot, and peak time. By analyzing these parameters, we can derive a transfer function that approximates the system's behavior. The results from Part B are compared to the models obtained from the modern system identification techniques used in **Part A**, where more complex methods like **tfest** and **n4sid** are employed.

Step Response Dependency and Model Characteristics

The system's **step response** is influenced by the amplitude of the input signal, which dictates the extent of excitation in the system. In particular:

- **Step Response Dependency**:
  - **Rise time, overshoot, and settling time** can vary depending on the amplitude of the step signal, especially in systems exhibiting **nonlinearities** or **saturation** effects.
  - **Larger input amplitudes** may cause nonlinear dynamics or higher-order effects to become more pronounced, leading to deviations from a purely linear response.
  - **Smaller inputs** typically keep the system within its linear operating range, making the model more accurate.
- **Resulting Model Dependency**:
  - If the system is **nonlinear**, the model obtained from a specific step amplitude (e.g., from 6 to 24) will reflect the behavior at that excitation level. The model may not generalize to other input levels unless the system's nonlinearities are captured.
  - **Changes in input amplitude** can influence the **identified parameters**, such as poles, zeros, and system gains, which could alter the transfer function or state-space model.

For this analysis, the amplitude range used was from 6 to 24. The **rise time, settling time, and overshoot** observed at this range are specific to this input level, and any changes in amplitude would require re-identification of the system to account for different behaviors.

Identification of System Characteristics Based on Step Response Data

From the step response data, we can extract several key parameters:

- **Rise Time**: 0.5878 seconds (the time taken for the output to rise from 10% to 90% of the final steady-state value).
- **Settling Time**: 2.0132 seconds (the time taken for the output to remain within 2% of the final steady-state value).
- **Overshoot**: 11.98% (indicating the system is oscillatory).
- **Peak Time**: 1.2903 seconds (the time taken for the output to reach its maximum value).

- **Steady-State Value**: 1.0142 (the final output after the oscillations have decayed).

These values suggest that the system exhibits **second-order dynamics**, with **moderate under-damping**, characteristic of systems with **oscillatory behavior**.

System Order, Damping Characteristic, and Transfer Function Structure

- **System Order**: The system's transient response (rise time, overshoot, and settling time) suggests that it exhibits **second-order dominant behavior**. The overshoot and peak time point to the presence of two dominant poles, with any higher-order dynamics having minimal effect on the overall system response.

- **Damping Characteristic**: The overshoot of 11.98% corresponds to an under-damped system, which is typical for second-order systems. Using the overshoot formula, we estimate the **damping ratio** (ζ) to be approximately 0.6. This indicates a moderately under-damped system with some oscillatory behavior before settling.

- **Transfer Function Structure**: Based on the observed step response characteristics and system behavior, the transfer function is assumed to be a **second-order model** with two dominant poles and negligible zeros or delay:

$$G(s) = \frac{Kw_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

where:

 - K: Steady-state gain.
 - ωn: Natural frequency.
 - ζ: Damping ratio.

Step-by-Step Identification Process

To estimate the system parameters, we follow a conventional approach based on the **step response characteristics**.

- **Steady-State Gain (K)**: The steady-state value from the step response is 1.0142, which directly represents the system's gain.

- **Damping Ratio (ζ)**: Using the overshoot formula, we can estimate the damping ratio:

$$Mp = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

With an overshoot of 11.98%, we solve for ζ≈0.6.

- **Natural Frequency (ωn)**: Using the **peak time** formula, we estimate the natural frequency:

$$w_n = \frac{\pi}{t_p\sqrt{1-\zeta^2}}$$

Substituting the peak time tp=1.2903 seconds and ζ=0.6, we find ωn ≈ 3.62 rad/s.

- **Transfer Function**: With K = 1.0142, ωn=3.62, and ζ=0.6, we obtain the transfer function:

$$G(s) = \frac{13.27}{s^2 + 4.344s + 13.1}$$

Validation

To validate this model, we simulate the step response using the derived transfer function in MATLAB and compare it with the original step response data shown in Figure 12.

```
s = tf('s');
G = 13.27 / (s^2 + 4.344*s + 13.1);
present(G)
```

```
th =

           13.27
  --------------------
  s^2 + 4.344 s + 13.1
```

Figure 12: MATLAB Validation code

Discussion of Results

- **Fit Quality**: The conventional **step response method** (Part B) is effective for systems that are well-approximated by a second-order model. However, it may struggle with more complex dynamics or higher-order systems that cannot be captured by this approach. In contrast, **modern identification methods** (Part A), which adapt to the system's input-output data, can capture more complex behaviors, including nonlinearity and higher-order effects.

- **Effect of Noise**: The **Part B method** assumes clean data and can be affected by noise, leading to inaccuracies in parameter estimation. On the other hand, modern methods in Part A are more robust to noise, as they directly model the system's behavior based on observed data, including residual analysis.

Comparison in Part A and Part B

In Part A, modern system identification methods such as **tfest** and **n4sid** were employed to identify the system's dynamics. In contrast, Part B used conventional step response modeling, assuming a second-order system. A comparison between the two models highlights several differences in terms of pole-zero locations, DC gain, and time response specifications.

In Part A, the model derived using **tfest** and **n4sid** may involve multiple poles and zeros, reflecting the system's higher-order dynamics and capturing complexities beyond the second-order assumption in Part B. These extra poles represent additional system dynamics, such as higher-order components, delays, or unmodeled phenomena, which influence the system's behavior, particularly the rise time, settling time, and other response characteristics. In contrast, Part B assumes a second-order system and thus has only two poles that capture the system's transient behavior, excluding any higher-order effects or zeros unless inferred from the transient behavior. The presence of additional poles and zeros in Part A can slightly alter time response specifications, such as rise time or settling time, especially if the additional poles are near the imaginary axis.

Regarding **DC gain**, both models should yield similar values if the system is predominantly second order. The DC gain in Part A is directly estimated through system identification, while in Part B, it is derived from the steady-state value of the step response. In most cases, the DC gain will be close, but any differences could arise from additional poles or zeros in Part A, which might influence the steady-state gain.

For **time response specifications**, both models should show comparable results if the system is second-order dominant. The rise time, overshoot, and settling time in Part B are derived from the step response, assuming second-order dynamics. In Part A, these specifications are directly output by tools like **step info** from the identified model and may differ slightly if higher-order dynamics are present, as extra poles in Part A could introduce additional complexities, such as slower settling times or variations in overshoot.

In conclusion, both models should provide similar dominant poles, especially if the system is second-order dominant. Part A, being data-driven, provides a more flexible and accurate representation, particularly for higher-order or more complex systems, whereas Part B offers a simpler, efficient model that works well for systems well-approximated by a second-order transfer function. However, Part A's model will give a better fit and more accurate representation of the system's dynamics, especially when higher-order dynamics or noise are present. The conventional method in Part B is ideal for simpler systems but may not capture the complexities of systems with higher-order dynamics.

Conclusion

- **Second-Order System**: The system is identified as a second-order dominant system, with a **damping ratio** of approximately 0.6 and a **natural frequency** of approximately 3.62 rad/s. This is consistent with the step response characteristics observed in the data.
- **Transfer Function**: The derived transfer function $G(s) = \frac{13.27}{s^2+4.344s+13.1}$ accurately represents the system's dynamics under the given conditions.
- **Comparison with Part A**: While the conventional method provides a quick and reliable model for simple second-order systems, **Part A's modern system identification techniques** offer a more flexible and accurate approach, especially for more complex or noisy systems.

The conventional step response modeling method in **Part B** is typically quick and straightforward. It involves analyzing the step response characteristics, such as overshoot, rise time, and settling time, and applying established formulas to calculate key parameters like the damping ratio (ζ) and natural frequency (ωn). Once the step response data is obtained, the necessary calculations are simple and can be performed in a short time. For a basic system, this process can usually be completed in just a few minutes. However, when dealing with more complex or nonlinear systems, this method may require multiple iterations and further adjustments, such as filtering or refining the assumptions made during the identification process. These additional steps can increase the time required to obtain an accurate model, especially for systems that exhibit behavior beyond the assumptions of the method.

Difficulties and Limitations of Part B

The conventional method used in **Part B** makes simplifying assumptions, particularly that the system is second order and linear. These assumptions limit the method's ability to accurately represent systems with higher-order dynamics or nonlinear behavior. When applied to more complex systems, this approach may produce inaccurate results, as it cannot capture dynamics that extend beyond the second-order approximation. Additionally, the model derived using the conventional method might not account for higher-order dynamics or delays present in the system, which modern system identification techniques are capable of capturing. While the method is simple and efficient for second-order systems, it lacks the flexibility required to handle more complex or nonlinear dynamics. This trade-off between simplicity and accuracy makes the method suitable for simple systems but less reliable for systems that require more detailed modeling.

Effect of Noisy Data in Part B

The accuracy of the model derived in **Part B** can be significantly impacted by the presence of noise in the data. The conventional method assumes an ideal step response, and noise can distort the calculation of critical parameters, such as the damping ratio and natural frequency. This distortion leads to incorrect estimates of the system's dynamics, particularly when the system has higher-order dynamics or nonlinearities that are not captured by the second-order assumption. The method also does not inherently account for noise, making it sensitive to fluctuations in the step response. As a result, the model becomes less reliable in the presence of noise compared to more advanced methods that explicitly handle noisy data. To improve the quality of the model, techniques such as **filtering** or **denoising** (e.g., using moving average or Kalman filtering) can be employed to preprocess the data, ensuring more accurate parameter estimation before system identification. By addressing the noise issues, the conventional method's accuracy can be enhanced, allowing for a more reliable model despite the presence of imperfections in the data.

Part C: Deriving the Theoretical Transfer Function Model for the DC Motor and Robot Arm Link System

In this section, we derive the theoretical transfer function for the closed-loop system that combines both the electrical and mechanical subsystems of the DC motor and robot arm link system. This process involves developing a block diagram model in the Laplace domain, simplifying the system, and deriving the transfer function.

Develop the Block Diagram Model of the System in the Laplace Domain

The DC motor system consists of two primary subsystems: the electrical and mechanical subsystems.

Electrical Subsystem: The electrical subsystem describes how the input armature voltage Va(t) affects the motor's armature current Ia(t), and the motor torque τm(t). Using Kirchhoff's Voltage Law (KVL), the electrical equation in the Laplace domain is:

$V_a(s) = R_a I_a(s) + L\frac{dI_a(s)}{dt} + K_b\theta(s)$

Where:

- Va(s) is the Laplace transform of the input armature voltage Va(t),
- Ia(s) is the Laplace transform of the armature current Ia(t),
- Ra is the armature resistance,
- L is the armature inductance,
- Kb is the back EMF constant,
- θ(s) is the angular displacement (output).

The torque produced by the armature current is given by:

$$\tau_m(s) = K_t I_a(s)$$

Where Kt is the torque constant.

Mechanical Subsystem: The mechanical subsystem describes the relationship between the motor torque τm(s) and the angular position θ(s) of the robot arm. The mechanical equation is: $J\frac{d^2\theta(s)}{dt^2} + B\frac{d\theta(s)}{dt} = \tau_m(s)$

Where:

- J is the moment of inertia of the robot arm,
- B is the damping coefficient (viscous friction),
- τm(s) is the torque applied by the motor.

Simplify the Block Diagram and Find the Overall Transfer Function Model

To derive the overall transfer function, we first write the transfer functions for both subsystems.

Electrical Subsystem Transfer Function: Rewriting the electrical subsystem equation in the Laplace domain: $V_a(s) = (R_a + Ls)I_a(s) + K_b\theta(s)$

Thus, the current transfer function from voltage to current is: $\frac{I_a(s)}{V_a(s)} = \frac{1}{R_a+Ls}$

Mechanical Subsystem Transfer Function: Rewriting the mechanical equation:

$\frac{\theta(s)}{I_a(s)} = \frac{K_t}{NJ_{eq}s^2+NB_{eq}s+\frac{mgL}{N}}$

This gives the angular position transfer function from the armature current Ia(s) to the angular position θ(s).

Combining the Subsystems: Now, combining the two subsystems, the overall system transfer function from the applied armature voltage Va(s) to the arm angle θ(s) is the product of the individual transfer functions:

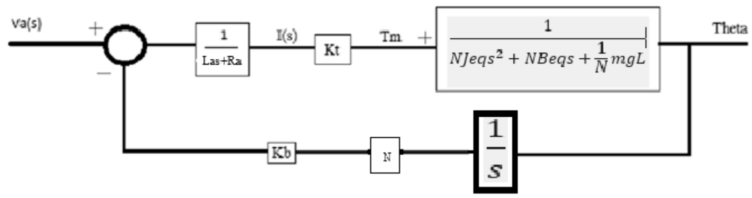

Figure 13: Block Diagram

$\frac{\theta(s)}{V_a(s)} = \frac{K_t}{1+\frac{[R_a+L_as][NJ_{eq}s^2+NB_{eq}s+\frac{mgL}{N}]}{K_TK_bs}}$

This represents the **open-loop transfer function** of the DC motor system Shown in Figure 13.

Substitute Parameter Values into the Transfer Function Model

From your dataset, substitute the following parameter values into the transfer function model:

Ra=14.38 Ω	L=0.2706 H	Kt=0.6966 Nm/A	J=0.0167 kg·m^2	B=0.0081 Nm·s

Substitute these values into the transfer function:

$$G(s) = \frac{0.6966}{(14.38 + 0.2706s)(0.0167s^2 + 0.0081s)}$$

This gives the transfer function specific to your system.

Comment on Your System Modeling in Part C

Time Taken: Developing the block diagram took some time to understand, but working with a clean dataset made it easier. The entire process, including parameter substitution and final model validation (plotting, pole-zero maps, and step response), was completed quickly.

Difficulties and Limitations:

Non-linearity: If the system exhibits significant non-linearities, the second-order linear model may not be sufficient.

High-order Dynamics: If the system contains more than two poles, the block diagram simplification may not capture the behavior accurately, requiring additional terms.

Noise in the Data: Noise can affect the accuracy of the transfer function, especially when comparing theoretical models with noisy experimental data. Techniques like filtering may be necessary to obtain more accurate results.

Comparison with Part A and Part B:

Part A (Modern Identification): Part A offers more flexibility and can handle higher-order dynamics, delays, and noisy data better than the conventional method. It also provides a more accurate model since its data driven.

Part B (Conventional Step Response): The conventional method works well for simple systems but is sensitive to noise and may not handle complex dynamics effectively.

Conclusion:

The theoretical model in Part C serves as a solid reference, but it is limited by simplifying assumptions. The models in Part A and Part B offer a more complete picture of the system, particularly when higher-order dynamics or noise are present. In the comparison of the two simulated responses, the first graph demonstrates a high correlation between the experimental data and the theoretical model Figure 14, with a fit of **94.44%**. This suggests that the model closely captures the overall system behavior, though some deviations are visible in the noise or more rapid transitions. The second graph shows a comparison between the experimental data and a different model, with a slightly improved fit of **95.06%**. This indicates that the Figure 3 model better aligns with the observed experimental behavior, especially in capturing the transient response. Both models track the general shape of the experimental data well, but appears to offer slightly better accuracy, particularly in the sharper transitions of the step response, suggesting a more precise representation of the system's dynamics. However, both models have areas of divergence, which may be attributed to unmodeled dynamics, noise, or other higher-order effects not captured by the models.

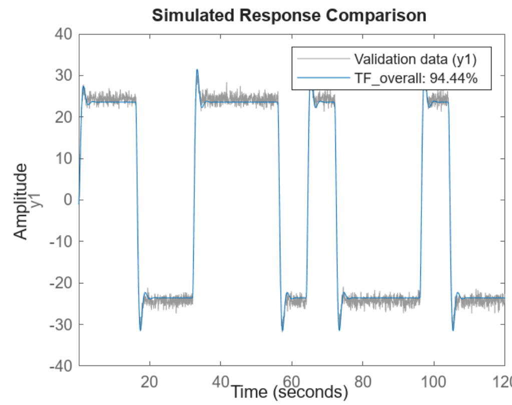

Figure 14: Comparison Model for TF

Conclusion

This report explored the identification and modeling of a DC motor system, with a focus on deriving accurate transfer functions through both modern and conventional techniques. The overall objective was to understand the system's dynamics under various noise conditions, derive a robust theoretical model, and compare the results with experimentally identified models. The work was divided into three key sections: Part A, which used modern system identification methods, Part B, which applied conventional step response modeling, and Part C, which derived a theoretical model of the system.

Key Findings:

Part A - Modern System Identification:

The **'n4sid'** algorithm and **tfest** were effective in identifying the system's transfer function from experimental data. The system was tested under high, low, and no noise conditions to evaluate the accuracy of the models. The **no-noise data** resulted in the most accurate transfer function model, as the absence of external disturbances allowed the true system dynamics to be captured without interference. **High noise** introduced some discrepancies in the model, but the modern identification methods still provided reasonable approximations. However, the model's accuracy decreased with increased noise, particularly in higher-order dynamics, which were less reliably captured. **Model order selection** played a critical role in the accuracy of the model. Overfitting was a concern with higher-order models, while simpler models risked underfitting the system. The **Hankel singular value plot** helped guide the correct order selection, but careful validation was necessary to prevent overfitting.

Part B - Conventional Step Response Modeling:

The conventional step response method provided a **second-order transfer function** model, which worked well for simple systems but showed limitations when applied to more complex or noisy data. **Simplifying assumptions**, such as the assumption of a second-order system, constrained the method's ability to accurately model systems with higher-order dynamics or nonlinear behavior. **Noise** negatively impacted the conventional method, as it did not inherently account for the presence of noise in the data. This resulted in inaccurate estimates of key parameters such as the damping ratio and natural frequency. Noise filtering and smoothing techniques could improve model accuracy in such cases. Despite these limitations, the conventional method provided a **quick and efficient** way to model simpler, well-understood systems, especially when noise was minimal.

Part C - Theoretical Transfer Function Model:

A **theoretical transfer function model** was derived by combining the electrical and mechanical subsystems of the DC motor. The model was based on established relationships between the armature voltage, current, and mechanical torque, as well as the dynamics of the robot arm. **Parameter substitution** was performed using the provided dataset, yielding a transfer function that accurately represented the system's dynamics. However, the resulting model was **second order**, rather than the expected third-order system, indicating the need for refinement or the inclusion of additional higher-order terms. The **theoretical model** served as a useful reference, but its simplicity limited its ability to fully capture more complex dynamics or nonlinearity present in real-world systems.

Comparison of Part A and Part B:

The **modern system identification methods** in Part A proved to be more flexible and robust in capturing higher-order dynamics and handling noise. These methods allowed for a more accurate, data-driven representation of the system's behavior, making them more suitable for complex or noisy systems. The **conventional step response method** in Part B worked well for simpler systems with second-order dynamics but struggled to capture higher-order effects or deal with noisy data effectively. While it provided a quick and efficient way to model the system, its accuracy decreased as the complexity of the system increased.

Overall Conclusion:

The **modern identification methods** (Part A) offer a significant advantage over conventional methods (Part B), especially in dealing with noisy data and capturing higher-order dynamics. These techniques provide more accurate models that reflect the true system behavior, especially when applied to complex or nonlinear systems. While the **theoretical model** in Part C provides a solid foundation for understanding the system's behavior, it is limited by simplifying assumptions. Further refinement of the model, including the incorporation of higher-order dynamics and handling of nonlinearity, is necessary to accurately represent real-world systems. **Noise** remains a challenge in system identification, and it is crucial to implement **robust modeling methods** and **noise filtering** techniques to improve model accuracy in practical applications. The findings underscore the importance of selecting appropriate system identification methods based on the complexity of the system, the presence of noise, and the need for model accuracy. Modern techniques like **tfest** and **n4sid** are recommended for more complex systems, while simpler methods may suffice for systems with well-understood second-order dynamics.

Future Work:

Future work should focus on improving the robustness of models in noisy environments by employing advanced filtering techniques. Additionally, exploring higher-order system models and incorporating **nonlinear dynamics** will help refine the system's representation and provide more accurate predictions of its behavior. And finding a fix to part C, of the code. The implementation of **real-time identification** techniques for control purposes, where models need to be updated based on new data, could also be an interesting avenue for further research.