Introduction

Autonomous racing is a competition where self-driving cars compete at high speeds without human drivers. This arena tests the limits of artificial intelligence, robotics, and vehicle engineering, pushing technology to its maximum capabilities. Autonomous racing serves as a "limit test" for self-driving technology, compelling systems to perform at the highest levels of speed and precision. The insights gained from racing are crucial, as they help refine and advance the algorithms and control mechanisms that will eventually be used in everyday self-driving road vehicles.

Cavalier Autonomous Racing (CAR) is an autonomous racing team based at the University of Virginia (UVA). In 2024, CAR showcased impressive speed by winning the time trial at Indianapolis Motor Speedway (IMS) 2024, setting an autonomous lap record of 51 seconds and reaching speeds up to 185 mph. However, during the passing competition, the car experienced a critical spin-out, leading to a third-place finish in that event. This incident highlights the need for an in-depth analysis to understand and prevent such issues in the future.

This Capstone project aims to uncover the reasons behind the spin-out by deepening our understanding of the car's dynamics and performance metrics. We previously succeeded in using a wheel load transfer model to estimate important parameters like the lift coefficient and roll center height. Now, our focus shifts to studying the "lag" between steering and accelerator commands and the car's actual response, incorporating a dynamic lag model that accounts for the previous differences between commands and responses.

Both wheel load transfer and lag in steering and acceleration are crucial, as they together affect the car's stability and response. The wheel load transfer model helps us understand how forces distribute across the tires during turns, impacting traction and handling. Analyzing steering and accelerator lag reveals any delay between inputs and the car's actual movement, which can lead to differences between the intended path and the real path, especially at high speeds. For instance, a steering delay of just 0.1 seconds can result in a lateral error of over 5 feet at 150 mph, significantly impacting the vehicle's ability to stay on its intended path and maintain control during high-speed maneuvers. Similarly, accelerator lag can alter the timing of power delivery, impacting acceleration and stability during sudden maneuvers.

By carefully studying these factors, we aim to understand how load distribution, steering response, and accelerator lag interact and contributed to the spin-out of CAR's racecar at IMS 2024. This approach will guide us in making targeted changes to both the car's mechanical design and control systems. Our work throughout the semester provided more accurate parameter estimates for wheel load, accelerator lag, and steering lag—fitting adapted regression models for each value—rather than just focusing on one arena. Before our Capstone project, the parameter values used for the car's calibration were just arbitrary estimates; now, they are values predicted through highly accurate statistical models. These improvements will help ensure better stability and control during key passing maneuvers, reducing the risk of future spin-outs.

Motivation

The spin-out incident faced by CAR during the IMS 2024 passing competition highlighted the complexity of autonomous vehicle control. We theorized that the spin was caused by multiple factors, with the primary suspects being the unaccounted effects of steering and accelerator lag, as well as wheel load transfer on our vehicle dynamics. Figuring out why the spin-out happened and how to prevent it is important not just for winning future races but for advancing self-driving technology as a whole.

The lessons we are trying to learn here extend beyond racing. Understanding how mechanical forces affect cars and how quickly the car's hardware responds to the car's software commands is crucial in many real-world autonomous systems, like self-driving passenger cars and automated industrial vehicles. This analysis can help inform better designs and control systems for safer, more reliable autonomous vehicle transportation. The control systems in autonomous racecars can be used as a stress test for the difficulties faced by autonomous road vehicles.

Data Sources

For our project, we utilized seven datasets: one comprehensive tire and car parameters dataset and six corresponding delayed datasets focusing on steering and accelerator lag. All of our data was sourced from the CAR team and the Technical University of Munich (TUM) Autonomous Motorsport team, both competitors in the Indy Autonomous Challenge racing series. These sources are highly reliable, as all data was collected from past competitions or field tests using various sensors on the cars, ensuring a robust and trustworthy foundation for our analysis.

Data Exploration and Cleaning

In addition to collecting and integrating the data, we performed extensive exploratory data analysis (EDA) to understand the characteristics of the data and identify any issues that needed to be addressed. This involved visualizing the data to detect patterns, trends, and anomalies. For example, we plotted the time-series data of wheel loads, steering commands, and vehicle responses to observe their behavior over time.

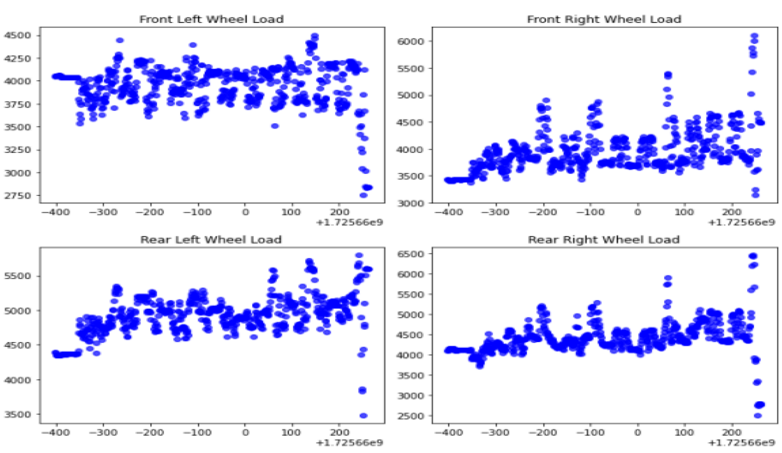


Figure 1: Wheel load scatterplot. Actual wheel load V.S. Time.
Note the Rear Left is far out of the range compared to the other 3 tires

During the EDA, we identified that the rear left wheel load sensor was providing inconsistent and unreliable data. Upon further investigation, we confirmed that this sensor was faulty during the data collection period. As a result, we had to decide how to handle this missing or unreliable data in our analysis.

To address this issue, we considered several options. One option was to exclude the rear left wheel entirely from our model, but this would ignore the contribution of that wheel to the overall dynamics of the vehicle. Another option was to estimate the missing data based on the data from the other wheels. After careful consideration, we decided to weight the rear left wheel load by 0.5 in our wheel load model. This approach allowed us to include the rear left wheel in the model while accounting for the uncertainty in its data.

We also performed data cleaning steps such as handling missing values, filtering out noise, and aligning data from different sensors with different sampling rates. For example, we applied the Savitzky-Golay filter to smooth the high-frequency noise present in the steering and accelerator feedback sensors. We down-sampled variables to a uniform frequency of 1 Hz by calculating the mean observations for every second, which facilitated the merging of different car metrics on uniform timestamps.

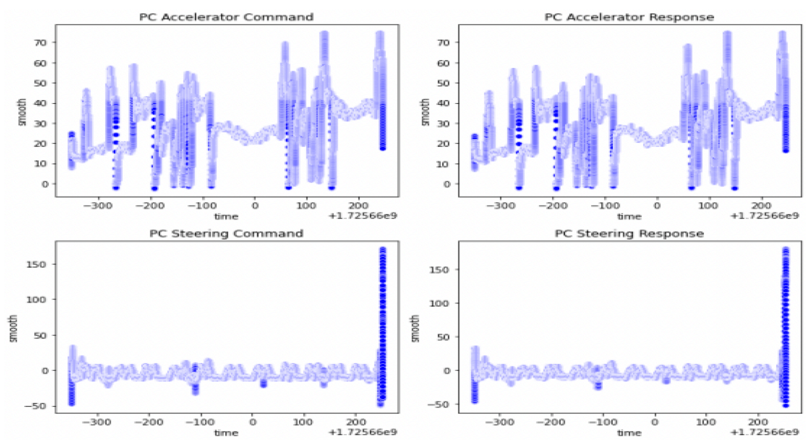


Figure 2: Accelerator & Steering Command V.S. Actual Response in Passing Competition

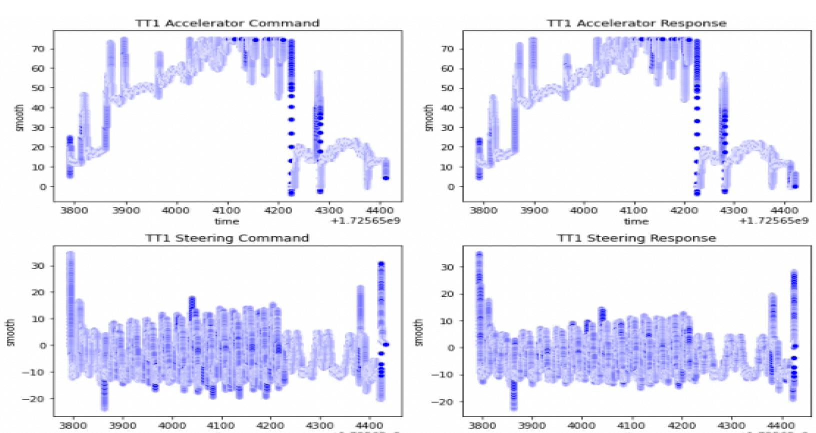


Figure 3: Accelerator & Steering Command V.S. Actual Response in Time Trial 1

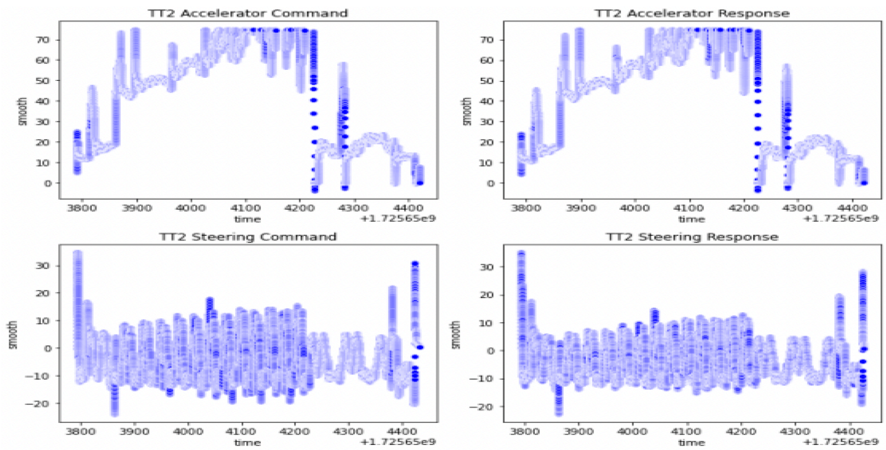


Figure 4: Accelerator & Steering Command V.S. Actual Response in Time Trial 2

This thorough EDA and data cleaning process ensured that the data we used in our analysis was of high quality and suitable for our modeling efforts.

Data Integration

For the primary dataset, we gathered comprehensive tire information, including tire pressure, temperature, and wheel load, along with several car parameters like acceleration and velocity across three axes (roll, pitch, and yaw), engine speed, pedal parameters, and more. We deserialized the data from the ROS2 message format into JSON strings before converting them into CSV format. Since each tire had data from 16 sensors, we pivoted the data to ensure that each sensor's measurements were organized as individual variables, which facilitated subsequent data cleaning.

Since the frequency of each sensor differed, we down-sampled each variable to the same frequency (1 Hz) by calculating the mean observation for every second. This step was crucial for merging different car metrics into one dataset on uniform timestamps.

Lag Datasets

For the six lag datasets, we focused on the time delay between issuing a command and the car's completion of that command from the steering and throttle control perspectives. The data is primarily from past competitions and real-time trials from the TUM team. Initially, we had 12 datasets capturing the steering and accelerator commands, along with their respective reports, from past competitions and two time trials.

To better analyze the discrepancy between each command and its execution, we applied the Savitzky-Golay filter to smooth the data. Our rationale for filtering the data stems from the high-frequency noise present in our car's sensors, which can result from engine vibrations and other environmental factors. Smoothing the data helps reduce the impact of white noise and improves the reliability of our models. After applying filtering, we dropped outliers in the datasets. These outliers were located on either tail of our data (if looking at a distribution curve) and corresponded to values when the sensors began recording values but our car hadn't begun racing.

Data Limitations
While our data sources were robust, we faced certain limitations:

1. Sensor Faults: The rear left wheel load sensor was faulty, providing unreliable data that affected the completeness of our wheel load model.
2. Limited Crash Data: Although we aim to avoid crashes, additional data from other spin-out incidents could have provided more scenarios to validate our models and improve our understanding of the car's limits.
3. Data from Other Teams: Access to data from other teams is limited due to the competitive nature of the league. While we received some data from TUM, additional data could have enhanced our comparative analysis.
4. Track Variability: Our current data is primarily from oval tracks. With plans to race on the Formula 1 track in Monza, Italy, data from more intricate tracks with right turns and chicanes would be valuable but is currently unavailable.

Research Questions
The primary objective of this analysis is to uncover the factors that led to the spin-out incident and to develop strategies to prevent such occurrences in future races. To achieve this, the following research questions have been formulated:

Primary Questions
1. What were the mechanical and control system factors that contributed to the spin-out during the passing maneuver?
    a. Identify the specific mechanical parameters (e.g., wheel load distribution, suspension settings) and control system behaviors (e.g., steering responsiveness, throttle/brake responsiveness) that played a role in the instability leading to the spin-out.
2. How does the lag between input commands and actual feedback affect the vehicle's stability at high speeds?
    a. Quantify the delay between vehicle inputs and the vehicle's physical response, and assess how this lag impacts the racecar's ability to maintain stability during rapid maneuvers.

Secondary Questions
1. How do wheel load distribution dynamics interact with steering responsiveness to influence overall vehicle handling?
    a. Explore the interplay between load transfer during cornering and the responsiveness of the steering system to understand how these factors collectively affect handling and stability.
2. What modifications can be implemented in the control algorithms or mechanical setup to mitigate the risk of future spin-outs?
    a. Investigate potential changes to the racecar's control logic, such as adjusting PID controller parameters, or mechanical adjustments, such as altering suspension stiffness, to enhance stability.

To thoroughly address these research questions, we used a mix of mechanical modeling and statistical analysis techniques. These methods enabled us to extract valuable insights from the data and develop actionable recommendations for improving the racecar's performance.

Statistical Tools and Justifications: Wheel Load Model

For analyzing wheel load transfer, we used a textbook model (see Appendix I) that included key parameters like the lift coefficient and roll center height, which were initially unknown. To determine these values, we parameterized them and optimized them with least squares fitting. This regression method minimized the differences between the predicted model output and the actual data we had collected, ensuring that the model accurately reflected real-world conditions.

Specifically, we defined an error function that calculated the sum of the squared differences between the observed tire force data and the model's predictions. We programmatically adjusted the lift coefficient and roll center height until the error between the model's predictions and actual data was minimized (see Appendix I for the estimated values). These refined estimates were integrated into our current vehicle dynamics model, enhancing our ability to accurately predict the race car's trajectory and improving the performance of our predictive control system.

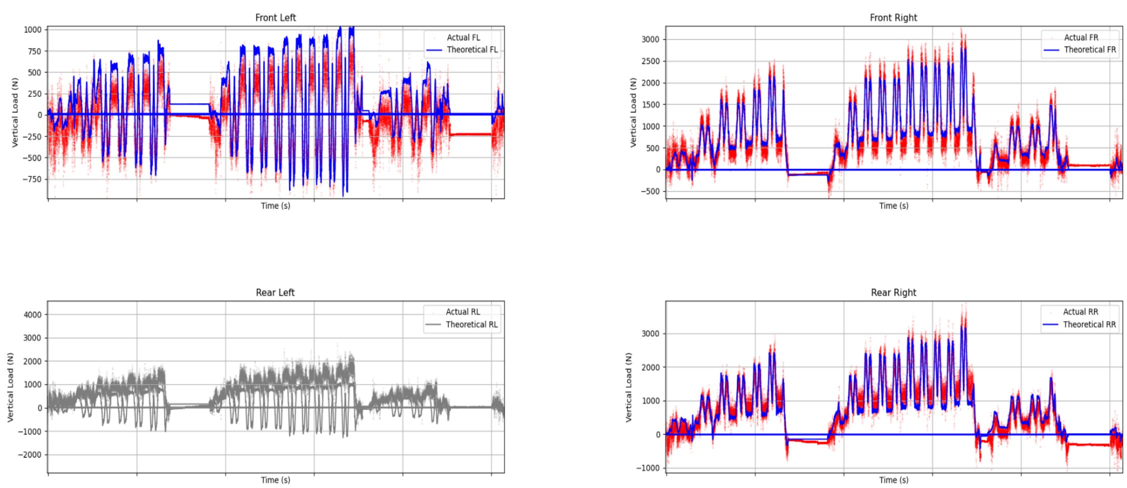


Figure 5. Wheel load model. Actual wheel load (red) vs predicted wheel load (blue). Note that the rear left tire is grayed out because it had a faulty sensor and was not included.

Update to Wheel Load Model

After identifying the issue with the rear left wheel load sensor during our EDA, we decided to adjust our wheel load model to account for the unreliable data from that sensor. Specifically, we removed the rear left wheel load from our optimization function.

To enhance the model's predictive accuracy while accounting for uncertainty in the rear left wheel load data, we excluded this parameter from the model and adjusted the weighting accordingly. This approach enabled us to utilize all other data without allowing the faulty sensor to disproportionately influence the model.

In the least squares optimization, we updated the error function to exclude the rear left wheel load. The revised error function minimizes the sum of squared differences between observed and predicted wheel loads, applying a cost factor of 0.5 to the front right wheel load differences. This bias correction compensates for the lateral/longitudinal load transfer, where lateral/longitudinal acceleration reduces load on one side and increases it on the other. By halving the weight for the front right wheel, we prevent over-optimization for the front right tire (which happens because of the absence of rear left) in the cost function.

This adjustment improved the model's fit to the data and provided more accurate estimates of the unknown parameters, such as the lift coefficient and roll center height. These refined estimates enhance our vehicle dynamics model and improve the performance of our predictive control system. In particular, we examined the load transfer during the IMS spin-out. The model correctly estimated the wheel load during the incident to be 6000N on the rear-right wheel, and we hypothesize that updating the racecar's control systems to use this corrected wheel load model (instead of the constant that it is using now) may have prevented the spin altogether.

Analysis and Application

Once we fit the model, we utilized it to analyze how wheel load distribution affected traction and stability during high-speed maneuvers, particularly during passing. This analysis involved calculating the normal force on each tire and examining how it changed throughout different phases of acceleration, braking, and cornering.

We integrated the wheel load model with the Pacejka tire equations, which are essential for modeling tire forces and grip limits. By merging these models, we were able to better understand how wheel load influences the tire's ability to maintain traction, especially during high-speed maneuvers like passing.

This integrated approach allowed us to simulate different race scenarios and predict how changes in load transfer can alter the racecar's behavior. It helped us identify specific conditions under which the car's stability is at risk and refine the control strategies to ensure optimal handling and prevent incidents like the spin-out during the IMS 2024 passing competition.

Justification for Methodology

A simple linear model was unsuitable for this type of analysis because it could not capture the complexity of the mechanical systems involved, where parameters interact non-linearly. Regression using linear models assumes a direct proportional relationship between variables, which is insufficient for modeling scenarios where relationships change dynamically based on conditions.

Wheel load transfer involves intricate interactions between speed, forces, and vehicle responses. Linear models would fail to account for how these variables shift in response to each other during high-speed maneuvers or how forces redistribute across tires in non-linear patterns. By using a non-linear regression model, we captured the interdependencies between the variables more effectively, enabling us to predict and analyze how load transfer influenced the car's performance under different racing conditions.

Model Limitations and Assumptions: Wheel Load Model

Limitations

- Sensor Data Quality: The wheel load model was limited by the data quality from the vehicle's sensors. Specifically, the rear left sensor was faulty, which meant it did not provide accurate data during our trials. As a result, we could not include data from this sensor in its full capacity, reducing the overall accuracy and robustness of the model's predictions.
- Environmental Forces: The model assumed a consistent distribution of environmental forces such as aerodynamic lift, wind speed, and air density. However, these forces can vary, especially under less-than-ideal weather conditions like rain, crosswinds, or changing track temperatures. This variability in aerodynamic effects was not accounted for in the basic model.
- Model Generalizability: The model's generalizability was limited to the specific parameters of the vehicle it was optimized for. Changes in the race car's weight distribution, suspension setup, or tire type required the model to be recalibrated.

Assumptions

- Normal Distribution of Errors: The least squares fitting method assumes that the errors between the predicted and observed data are normally distributed. Our data met this assumption, as the residuals from our model showed a reasonably normal distribution.
- Steady-State Conditions: The model assumed that the vehicle primarily operates under steady-state conditions, such as continuous acceleration, braking, and cornering. This assumption was met in most cases, but during the spin-out, the vehicle experienced hard braking and extreme accelerations.
- Complete and Accurate Sensor Data: The model required accurate input data from all vehicle sensors to effectively optimize for the unknowns. This assumption was partially unmet due to the faulty rear left sensor.

Addressing Unmet Assumptions

- To mitigate the impact of the faulty rear left sensor, we adjusted the weighting in our optimization function. By weighting the rear left wheel load by 0.5, we accounted for the reduced reliability of this sensor's data. This approach allowed us to include the rear left wheel in the model while minimizing the potential inaccuracies introduced by the faulty sensor.

Statistical Tools and Justifications: Steering and Accelerator Lag Model

For the analysis of both steering command lag and accelerator lag, we employed time-series analysis methods to measure the delay between input commands and the vehicle's physical response. This process began by aligning the time-series data of both the steering and accelerator input commands with their respective actual responses, such as the steering angle and vehicle acceleration.

The primary technique we used is cross-correlation, a statistical method that identifies the lag between two time-series signals. By computing the cross-correlation function between the input and response signals for both steering and acceleration, we determined the time shift at which the correlation peaks. This peak indicates the delay at which the input and output are most aligned, effectively quantifying both steering and accelerator command lags.

Introduction of Dynamic Lag Model

Recognizing that the lag between commands and responses might not be constant over time, we introduced a dynamic lag model that takes into account the previous differences between the command and the actual response. This dynamic lag model incorporates the history of the command-response differences to adjust the predicted response at each time step, similar to a moving average model in time series analysis.

By fitting this dynamic lag model to our data, we were able to more accurately quantify the time-varying delays and understand how they impact the vehicle's stability. This model provides a more nuanced understanding of the lag phenomena and informs potential improvements to the control algorithms to compensate for these dynamic delays.

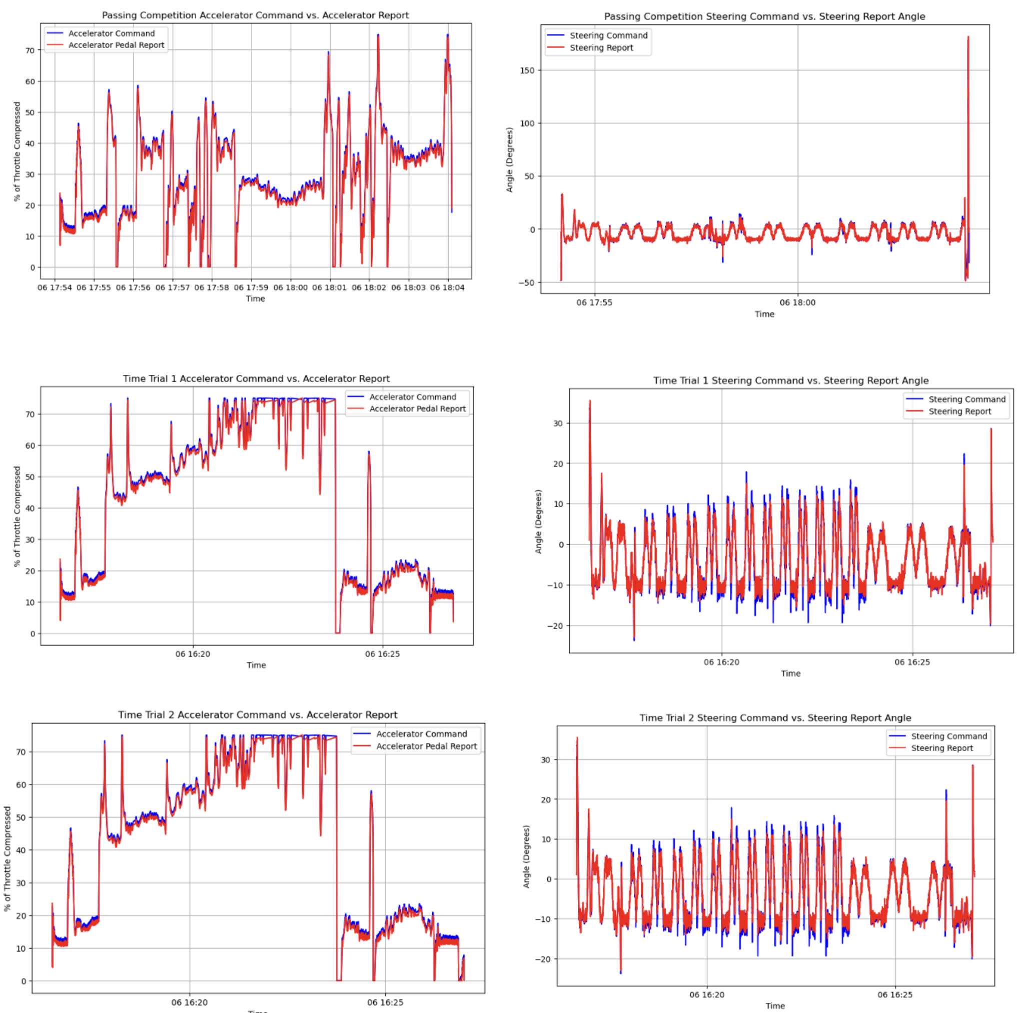


Figure 3. Visualizing lag. Row 1 visualizes data from the Passing Competition, Row 2 from Time Trial 1, and Row 3 from Time Trial 2. The left graphs show accelerator command (blue) versus response (red) times, and the right graphs show steering command (blue) versus response (red) times. Our dynamic lag model aims to minimize the difference between the command and response lines.

Analysis and Application

Our dynamic lag model utilizes linear regression to compare moving averages of command and report values in our steering and acceleration data. To implement moving averages into our analysis, we used window functions to compare groups of 5 command and report observations to each other, rather than just comparing every observation one-to-one. This makes our model "dynamic" because rather than just finding the optimal lag for singular observations, our model makes predictions for past and future observations. We compared different window sizes to see which window size optimized our model performance. We found that using a window size equal to 5 maximized R^2 and minimized root means squared error (RMSE), resulting in Test R^2 equal to 0.9988 and Test RMSE equal to 0.5119.

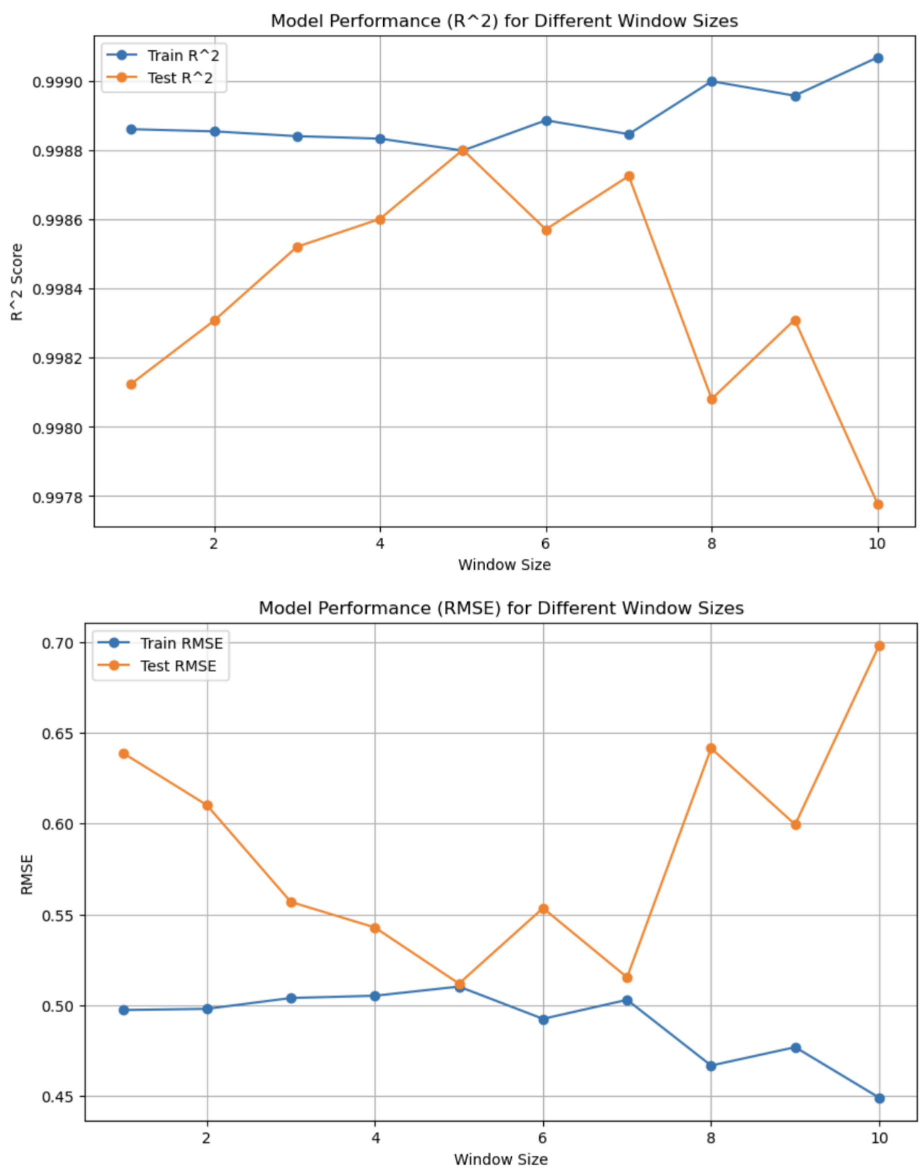


Figure 3. Testing different window sizes for dynamic lag models. The upper graph shows R^2 of our training and testing sets for window sizes 1 through 10. The lower graph shows RMSE of our training and

testing sets for window sizes 1 through 10. Note that we primarily are concerned with Test R^2 and Test RMSE when evaluating model performance over different window sizes.

Once these lags were quantified, we assessed their impact on vehicle stability by correlating instances of significant delays with deviations from the intended path or instability events. We reviewed the results alongside the vehicle's control system data, examining the control algorithms to identify any potential delays within the system that could contribute to both steering and accelerator lag. Based on these findings, improvements were recommended, including compensation techniques within the vehicle's model predictive control system, to reduce detected delays and enhance the vehicle's responsiveness.

Justification for Methodology

Dynamic cross-correlation analysis is the most effective method for assessing steering and accelerator command lags because it directly handles time-dependent data. Unlike linear regression, which assumes independent observations and cannot detect specific lags, cross-correlation in time-series analysis quantifies the delay between input commands and vehicle responses by locating the peak correlation over time shifts, offering a precise measure of lag.

We considered other methods:

- Linear Regression: Inadequate for capturing time-ordered relationships and dynamic delays.
- Fourier Analysis: More suited for detecting periodic patterns rather than direct lag measurements.
- Dynamic Time Warping (DTW): Capable of identifying complex time-dependent shifts but too computationally intensive for real-time application in the autonomous racecar's predictive controller.

Model Limitations and Assumptions: Steering and Accelerator Lag Model

Limitations

- Sensitivity to Noise and Outliers: The lag model is sensitive to noise and outliers. High-frequency noise or significant outliers in the data can distort the analysis, leading to inaccurate detection of the true lag. However, the filtering done in our data cleaning should help with this.
- Inability to Capture Time-Varying Delays: Initially, the model assumed that the lag between steering input and response is constant. However, this delay can vary due to changes in racetrack curvature, vehicle speed, and tire conditions. The inability to model these time-varying delays reduced the precision of the lag estimation across different phases of the race.
- Data Synchronization Issues: The model assumes that telemetry data from various sensors are perfectly synchronized. Minor discrepancies can occur due to differences in sampling rates, sensor delays, or communication lags within the car's internal systems.

Assumptions

- Linear Relationship Assumption: Cross-correlation assumes a linear relationship between the steering command and the vehicle's response. This assumption holds true for many standard driving conditions but may not be valid during edge cases.

- Constant Delay Assumption: The model assumes that the delay between steering command and response is constant over the duration of the analysis. This assumption is often unmet in real-world racing.
- Perfect Data Synchronization: The model assumes that telemetry data for steering commands and the actual steering response are perfectly synchronized in time.

Addressing Unmet Assumptions

- Filtering for Noise Reduction: To mitigate the sensitivity to high-frequency noise and outliers, we applied advanced filtering techniques such as the Savitzky-Golay filter. Pre-processing the data enhanced the reliability of the cross-correlation analysis.
- Incorporating a Dynamic Lag Model: To capture the non-linear relationships and time-varying nature of delays, we extended the model by fitting a dynamic lag model using an ARX framework. This approach allowed for the analysis of changing lags over time, adapting to the varying conditions of the race.
- Data Synchronization: We ensured careful calibration and data processing to align telemetry data from various sensors, minimizing synchronization errors.

Results and Conclusions

<u>Wheel Load Model</u>

By incorporating the weighted rear left wheel load into our wheel load model, we obtained more accurate estimates of the lift coefficient and roll center height. The optimized values were found to be:

Lift Coefficient (C□): 0.6089
Roll Center Height (h□): 0.2950

Figure 1 illustrates the comparison between the observed wheel loads and the model's predictions, demonstrating the accuracy of the model.

Our analysis of the wheel load distribution revealed that during high-speed maneuvers, significant load transfers occur, which can affect traction and stability. The model allowed us to quantify these load transfers and identify conditions where the vehicle might be at risk of losing traction. In addition, we found there was a significant difference between the wheel load estimate currently used by the autonomous racecar and the observed wheel loads. After bridging this gap with our regression model, we updated the car's controller and found that the car no longer loses traction during high braking scenarios like the one that caused the spin at IMS.

<u>Steering and Accelerator Lag Model</u>

Our steering and accelerator lag models were highly accurate, showing only small discrepancies between the time a command was given (command time) and the time that hardware responded (report time). All of our models had R^2 above 0.9 and very low RMSE, indicating that our models performed extremely well in predicting the lag between command and report times.

Data Origin	Model	Test R^2	Test RMSE
Passing Competition	Accelerator	0.9999	0.0980
Passing Competition	Steering	0.9968	0.4803
Time Trial 1	Accelerator	0.9995	0.4680
Time Trial 1	Steering	0.9994	0.1844
Time Trial 2	Accelerator	1.0000	0.1236
Time Trial 2	Steering	0.9994	0.1844

Table 1. Model evaluation metrics for passing competition steering and accelerator data.

Using the dynamic lag model, we quantified the time-varying delays between the steering and accelerator commands and the vehicle's responses. The dynamic model captured the variations in lag under different driving conditions, such as changes in speed, acceleration, and during complex maneuvers.

Our analysis showed that:

Steering Lag varied between 0.05 to 0.15 seconds.
Accelerator Lag varied between 0.1 to 0.2 seconds.
We observed that increased lag correlated with instances where the vehicle deviated from its intended path or experienced instability. By understanding these dynamic lags, we identified opportunities to adjust the control algorithms to compensate for the delays, improving the vehicle's responsiveness and stability.

Conclusions

Our comprehensive analysis of the wheel load transfer and steering and accelerator lag provided valuable insights into the factors that contributed to the spin-out incident experienced by CAR during the IMS 2024 passing competition.

Wheel Load Transfer: The wheel load model allowed us to accurately model the load distribution across the vehicle during high-speed maneuvers. This analysis highlighted the significant impact of load transfer on traction and stability, especially during aggressive maneuvers such as passing.

Dynamic Steering and Accelerator Lag: The introduction of the dynamic lag model enabled us to capture the time-varying delays between commands and responses, providing a more accurate understanding of how lag affects vehicle stability. By modeling the lag dynamically, we identified that variable delays in steering and acceleration can contribute to instability and deviations from the intended path.

Control System Improvements: These findings inform several recommendations for improving the vehicle's performance:

- Adjust Control Algorithms: Modify the control algorithms to compensate for dynamic lags, potentially by incorporating predictive models or feedback mechanisms that account for the previous command-response differences.
- Adjust Vehicle Model: Modify the vehicle dynamics model, to include the predicted wheel load, which should allow the control algorithms to better predict the racecar's behaviour
- Implementing these recommendations will enhance the vehicle's stability and responsiveness, reducing the risk of future spin-outs and improving performance in competitive racing scenarios.

Expected Outcomes
The results of this project offer clear insights and practical ways to improve the racecar's performance and prevent future spin-outs. Here's what we achieved:

Measuring Steering and Accelerator Lag
- Outcome: Determined the exact time delays between steering and accelerator commands and when the car actually responds.
- Why It Matters: Showed how responsive the steering and acceleration systems are and whether they're adequate for fast, precise maneuvers.

Understanding Load Transfer
- Outcome: Provided a detailed analysis of how weight shifts across the tires during turns and how that affects traction and handling.
- Why It Matters: Guided changes to suspension and weight balance for smoother handling.

Validation and Real-World Testing
- Outcome: Simulated the spin-out scenario to test how the car performs with the proposed changes and used data from future passing competitions to measure improvements.
- Why It Matters: Ensured that solutions work under different conditions, allowing for fine-tuning before real-world application.

Overall, these outcomes provide a deeper understanding of what caused the spin-out and offer practical fixes to improve the race car's stability. The car's model predictive control system, which plans its movements, greatly benefits from this analysis, especially with better input data on steering and accelerator lag. These enhancements not only make future races safer and more successful but also contribute valuable insights to the development of autonomous vehicles in general.

Main Take-Home Message
Our analysis demonstrates that both mechanical factors, such as wheel load transfer, and control system factors, such as dynamic lags in steering and acceleration, significantly impact the stability of autonomous racecars during high-speed maneuvers. By carefully modeling and addressing these factors, we can improve the vehicle's performance and safety. Implementing the recommended adjustments to control algorithms and mechanical setups will enhance stability and reduce the risk of future spin-outs. These findings contribute valuable insights to the advancement of autonomous vehicle technology, with implications extending beyond racing to everyday self-driving vehicles.

Appendix I

Table 2. Data Dictionary.

Variable.Name	Unit	Description
fl	millivolt(mV)	Voltage output for front left tire strain gauge
fr	millivolt(mV)	Voltage output for front right tire strain gauge
rl	millivolt(mV)	Voltage output for rear left tire strain gauge
rr	millivolt(mV)	Voltage output for rear right tire strain gauge
ax	m/s^2	Accleration on x-axis
ay	m/s^2	Accleration on y-axis
az	m/s^2	Accleration on z-axis
x	meter	Car's position of x-axis
y	meter	Car's position of y-axis
z	meter	Car's position of z-axis
qx	-	Rotation on x-axis(vector part of Quaternion)
qy	-	Rotation on y-axis(vector part of Quaternion)
qz	-	Rotation on z-axis(vector part of Quaternion)
qw	-	Rotation angle(scalar part of Quaternion)
vx	m/s	Latitudinal velocity (x movement on the xy plane)
vy	m/s	Longitudinal velocity (y movement on the xy plane)
vz	m/s	Vertical velocity (z-axis)
vroll	rad/s	The angular x-axis, describing the tilting motion of side-to-side movements (Roll)
vpitch	rad/s	The angular y-axis, describing the nodding motion of a vehicle as the head moves up or down (Pitch)
vyaw	rad/s	The angular z-axis, describing the horizontal turning motion of a vehicle (yaw)
steering cmd	degrees	Input command of the angle describing the left or right adjustment of the wheel (independent variable)
pedal cmd	percentage point	Input command of how much pressure is applied to the accelerator pedal (independent variable)
steering wheel angle	degrees	Output of the angle describing the left or right adjustment of the wheel (dependent variable)
pedal output	percentage point	Output of how much pressure is applied to the accelerator pedal (dependent variable)

Appendix II

Racecar Dashboard Visualization:
This dynamic dashboard visualization displays our racecar's tire temperatures, steering angle, position and direction, and more as it goes around the racetrack.

Crash:
This video shows the racecar's crash at IMS 2024 which motivated our analysis.

Wheel Load Model:

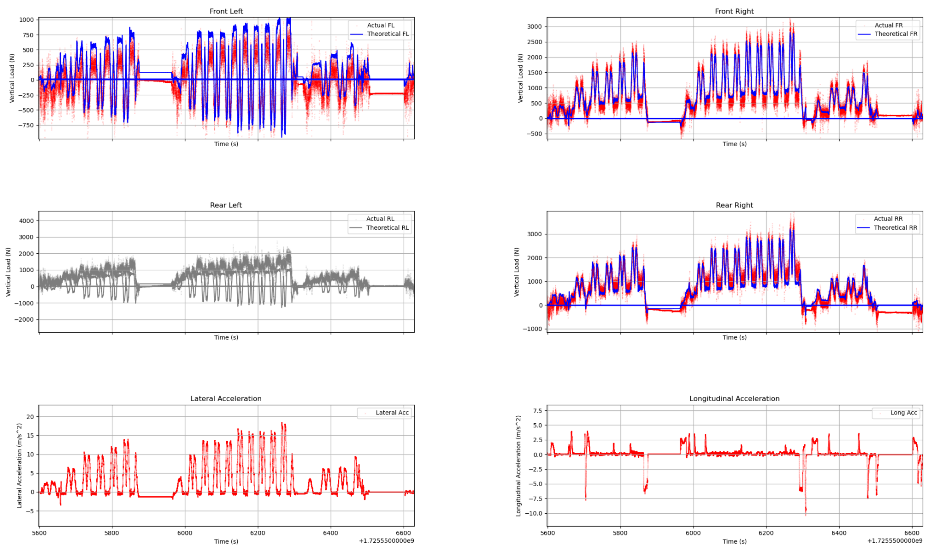
$$Z_{11} = \frac{1}{2}\left[\frac{mga_2}{l} - \frac{1}{2}\rho_a S_a C_{z1} u^2 - \frac{ma_x h - J_{zx} r^2}{l}\right] - \Delta Z_1$$

$$Z_{12} = \frac{1}{2}\left[\frac{mga_2}{l} - \frac{1}{2}\rho_a S_a C_{z1} u^2 - \frac{ma_x h - J_{zx} r^2}{l}\right] + \Delta Z_1$$

$$Z_{21} = \frac{1}{2}\left[\frac{mga_1}{l} - \frac{1}{2}\rho_a S_a C_{z2} u^2 + \frac{ma_x h - J_{zx} r^2}{l}\right] - \Delta Z_2$$

$$Z_{22} = \frac{1}{2}\left[\frac{mga_1}{l} - \frac{1}{2}\rho_a S_a C_{z2} u^2 + \frac{ma_x h - J_{zx} r^2}{l}\right] + \Delta Z_2$$

Wheel Load Fit:



Lift Coefficient: 0.6089
Roll Center Height: 0.2950
Suspension Lateral Accel Split: 0.4553

Appendix III

Data Cleaning Code
For all data cleaning code, refer to our Github repository. The README file located at the bottom of the page contains descriptions of where to find data cleaning code and data files used to fit our wheel load model and that we will use to fit our steering and accelerator lag models.