8-2010

Fuzzy Logic Approach to Stability Control

Jeffery Anderson
Clemson University, jeffera@clemson.edu

FUZZY LOGIC APPROACH TO VEHICLE STABILITY CONTROL

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

by
Jeffery R. Anderson
August 2010

Accepted by:
Dr. E. Harry Law, Committee Chair
Dr. John Ziegert
Dr. Beshahwired Ayalew

Abstract

Traditional Electronic Stability Control (ESC) for automobiles is usually accomplished through the use of estimated vehicle dynamics from simplified models. Starting with the conventional two degree-of-freedom vehicle model, one can estimate the vehicle states from the driver steering input. From this estimate, vehicle sideslip angle can be found and this is generally used with a threshold value to initiate a control action. The input/output relationship of the model depends heavily on the accuracy of the parameters used and various means to correct model inaccuracies. Specifically, these models depend on the tire cornering stiffness which is prone to change with age and loading of the tires. Moreover, not all consumers will replace the original equipment (OE) tires with the same ones. Vehicle response is also directly related to coefficient of friction between the tire and road which varies with road and tire conditions. These issues may result in the degradation of the effectiveness of the ESC system. At the very least, they may require extensive tuning of the control algorithms.

This thesis proposes a different method for estimating the instability of a vehicle. It is solely based on measurable vehicle dynamic response characteristics including lateral acceleration, yaw rate, speed, and driver steering input. These signals are appropriately conditioned and evaluated with fuzzy logic to determine the degree of instability present. When the ”degree of instability” passes a certain threshold, the appropriate control action is applied to the vehicle in the form of differential yaw braking. Using only the measured response of the vehicle alleviates the problem of degraded performance when vehicle parameters change.

Finally, ten case studies of different vehicles, configurations, environments, driver models, and maneuvers are tested with the same ESC strategy to examine the concept of stability control without estimation. Four very different vehicles ranging from a sports car to a sport utility vehicle (SUV) in multiple configurations including degraded rear tires and different loading conditions are used in evaluating the proposed ESC. These vehicles and configurations are subjected to multiple maneuvers including a double lane change and a fishhook maneuver with tire-to-road conditions such as split mu and low mu to simulate slippery road conditions. The main result of this research is the evolution of a new ESC concept where performance is not based on a vehicle model with set parameters that lose effectiveness in estimating the vehicle dynamic states when the vehicle changes. Instead, the algorithm relies only on the current measurable dynamic states of the vehicle to preserve stability.

Acknowledgements

First, I would like to thank my advisor Dr. Harry Law for his time and dedication to our project. His years of knowledge and wisdom were invaluable to this work and I am grateful for his advice, guidance, and motivation throughout the process of earning my degree. I would also like to thank Dr. John Ziegert and Dr. Beshahwired Ayalew for serving on my committee as well as being exceptional professors in my course work.

Finally, I would like to take this opportunity to thank my parents for their love and support in helping me to achieve my goals. I can never thank them enough for the role they have played in my life and am forever grateful.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Electronic Stability Control is a very heavily researched topic with widely understood benefits to the safety of transportation. The Department of Transportation estimates thirty-four percent of single-vehicle passenger car crashes and fifty-nine percent of single-vehicle Sport Utility Vehicles (SUVs) crashes could be prevented with the implementation stability control; moreover, it will be federally mandated that in the model year 2012, all light vehicles will be required to have ESC [1]. These estimations suggest the dramatic potential to save lives which is why this topic is of such importance.

Traditionally, stability control is done by using estimators and observers to estimate vehicle dynamics based on driver inputs to the vehicle. The traditional two degree-of-freedom bicycle model is the model mainly used as the estimator in ESC algorithms. This model allows the engineer to estimate the vehicle sideslip angle of the vehicle which is the key parameter in determining the stability. Once the vehicle reaches a certain threshold of sideslip, corrective control action is then applied generally in the form of differentially braking the wheels. Other methods of control are also used including torque vectoring on an all wheel drive vehicle where the engine torque is appropriately redistributed as not to induce instability.

1.1 Research Motivation

As mentioned before, the traditional ESC strategy depends on the use of linear observers as seen below in Figure 1.1. This strategy commonly uses the two degree of freedom bicycle model as the observer to estimate the vehicle states based on the steering input to the model ($\mathbf{x_d}$). It is a closed loop observer that uses measured vehicle dynamics ($\mathbf{y}$) and a state estimator to turn those sensor traces into estimates of the actual vehicle states (x) and is then compared to the estimated traces from the vehicle model. A controller acts on these states and commands differential braking to the appropriate wheel to prevent the vehicle from spinning-out.

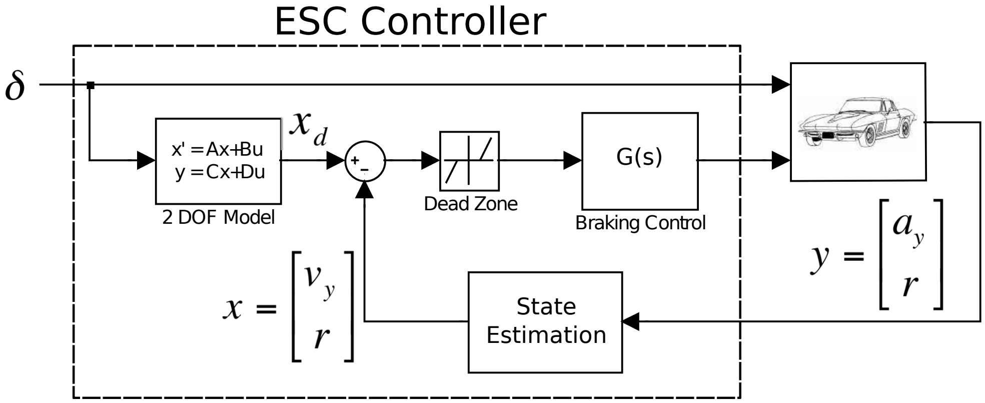


Figure 1.1: Traditional ESC Strategy Referenced from [2].

The two degree of freedom model is the essential problem in this approach; it heavily depends on several assumptions and accurate vehicle parameters to work correctly. Figure 1.2 represents the vehicle model as the two front tires and two rear tires combined. This model incorperates several assumptions including no lateral weight transfer and no longitudinal acceleration; thus, its degrees of freedom are yaw and lateral velocity. Forces and moments can be summed about the center of gravity and put in state space form as seen in Equation 1.1 [3]. Note that the C_α terms are the front and rear cornering stiffnesses as seen in Figure 1.3. C_α is the initial slope of the lateral force versus slip angle curve.

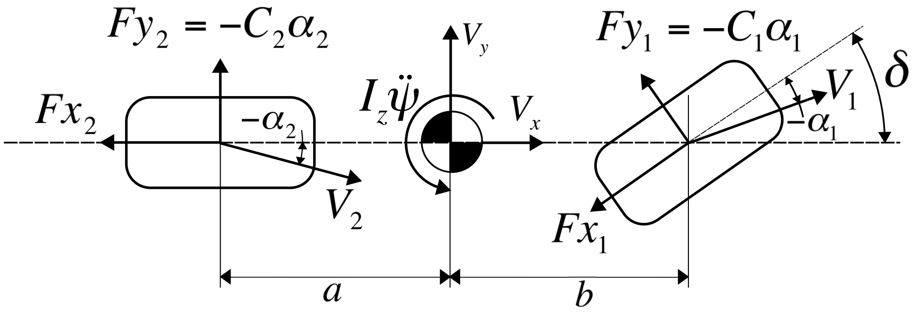


Figure 1.2: Two Degree-of-Freedom Model Referenced from [2].

$$\begin{bmatrix}\dot{V}_y\\ \dot{r}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} \frac{-2g}{WV_x}\left(C\alpha_f + C\alpha_r\right) & -V_x - \frac{2g}{WV_x}\left(C\alpha_f a - C\alpha_r b\right) & 0\\ \frac{-2}{I_z V_x}\left(C\alpha_f a - C\alpha_r b\right) & \frac{-2}{I_z V_x}\left(C\alpha_f a^2 + C\alpha_r b^2\right) & 0 \\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}V_x\\ r\\ \psi\end{bmatrix} + \begin{bmatrix}\frac{2g}{W}C\alpha_f\\ \frac{2a}{I_z}C\alpha_f\\ 0\end{bmatrix}\delta(t) \qquad (1.1)$$

1.1.1 Dependence of Cornering Stiffness C_α

As seen in Figure 1.3, the cornering stiffness of a tire is defined as the initial slope of the lateral force versus slip angle curve. This means that the vehicle model is only valid where C_α accurately describes the lateral force capability of the tire or the linear regime of the tire which is only valid to approximately 0.3 G. Cornering stiffness is also heavily dependent on tire vertical tire loading which is illustrated with the tire data in Figure 1.4.

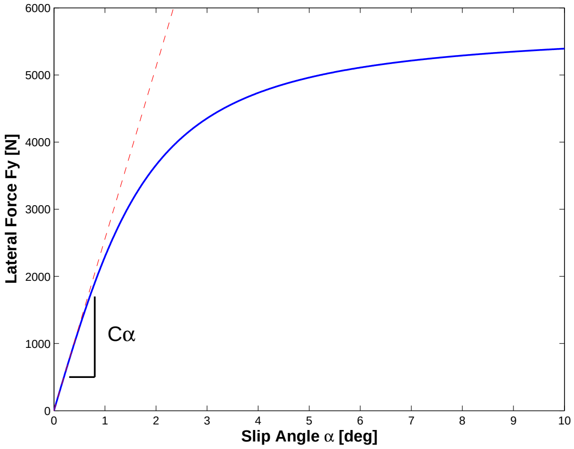


Figure 1.3: Cornering Stiffness C_α.

1.1.2 Dependence of Handling on Coefficient of Friction μ

In the two degree of freedom model, the coefficient of friction between the tire and the road is assumed such that that the tires do not saturate. This is especially problematic in low μ situations where the tires easily saturate and induce instability. There have been several strides to make this approach more robust by estimating coefficient of friction such as [2], but, this is another estimated parameter in the system and adds complexity to the ESC strategy.

1.2 Problem Statement

The goal of this research to is find another method of inferring stability of the vehicle based on its measurable dynamic response. The work of both Buddy Fey and Sunder Vaduri is heavily used in the research and can be seen in [5] and [6] respectively. Fey laid the ground work for this research in the form of using dynamic traces in race car data to indicate oversteer/understeer. In his book, he discusses the trends and combinations of signals that indicates these events. Vaduri took this one step further by using fuzzy logic along with Fey's approach to automate the process of data analysis to pinpoint understeer/oversteer events. This work was done in the form of the development of an expert system for the analysis of track test data. This ESC research takes it one step further by indicating oversteer in real-time and uses that information to actuate differential braking and prevent the vehicle from going unstable.

By using real measurable parameters as suggested by Fey and Vaduri, one can estimate the amount of oversteer present at any point in time and use that to prevent a vehicle from ”spinning-out.” This approach to stability control removes the necessity of an observer and its inherent problems as described above. By removing estimation, the ESC algorithm is simplified to signal conditioning and fuzzy logic.

1.3 Thesis Overview

This chapter discusses the background involved in this research including the reasons for a different approach from traditional stability control. Also discussed are the background sources for the research that will be presented in future chapters. Chapter 2 will discuss in more detail the trends and combination of signals that will indicate oversteer along with an overview of fuzzy logic. Chapter 3 will outline the specific algorithm used while Chapter 4 will outline a specific example to show how the fuzzy ESC algorithm reacts to a normal maneuver and stabilizes a vehicle. Chapter 5 will present a comprehensive set of results from ten case studies. Finally, conclusions and future work recommendations will be made in Chapter 6.

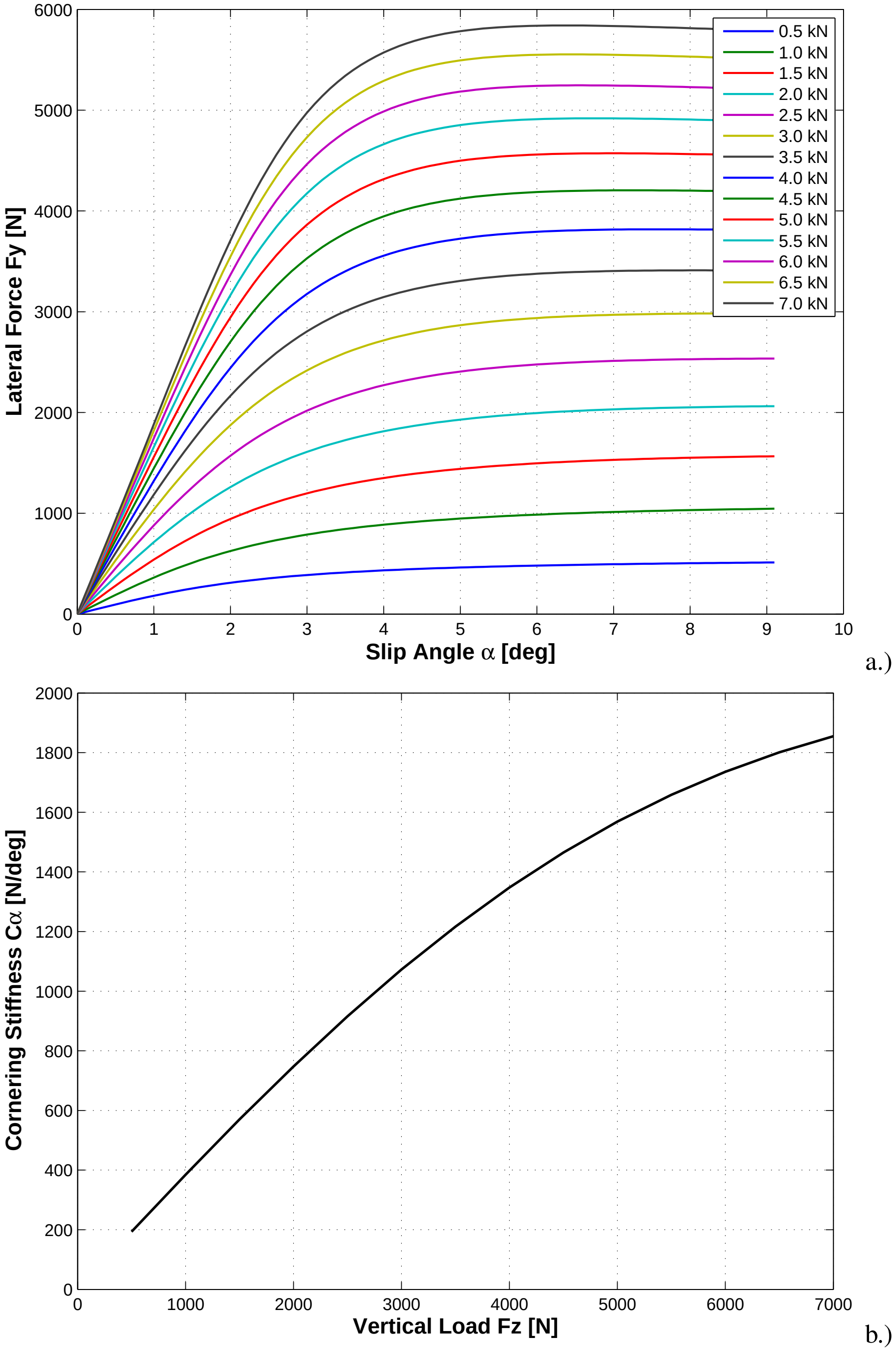


Figure 1.4: Cornering Stiffness Dependance a.) Lateral Force Dependance on Slip Angle α and Vertical Loading b.) Cornering Stiffness Dependance on Vertical Loading F_z [4].

Chapter 2

BACKGROUND

2.1 Indicators of Oversteer

In this section, the driver input and the response characteristics (i.e., lateral acceleration, yaw rate, etc.) and their relationships that are used to determine vehicle instability will be discussed. Fey describes oversteer very accurately with the following statement: ”It would be convenient if a single pattern would stand up, stomp the floor, and bellow ’oversteer’ or ’understeer.’ No such luck. Instead, the graphs usually provide a small catalog of symptoms that vary with the severity of the imbalance” [5]. For this research, steering wheel angle, lateral acceleration, and yaw rate are used to indicate an oversteer event. In the following sections, these will be examined in detail.

2.1.1 Steering Indicators

Steering can give a good indicator of a driver’s reaction to an imbalance in the vehicle. In the case of oversteer the driver will tend to reduce the steering wheel angle to prevent the vehicle from spinning-out. This can be seen in Figure 2.1, where the nominal BMW Mini in a double lane change (DLC) maneuver on a low friction surface is plotted. As the driver made the second turn (3.5 s), the tires saturated which produced a plateau effect on lateral acceleration. Moreover, the driver starts to counter-steer (5.0 s) to prevent the vehicle from spinning. At the same time, the vehicle sideslip angle is over 5 deg which is quite large and indicates that the driver barely maintained control of the vehicle. This illustrates how the driver’s reaction can be analyzed to determine an oversteer situation.

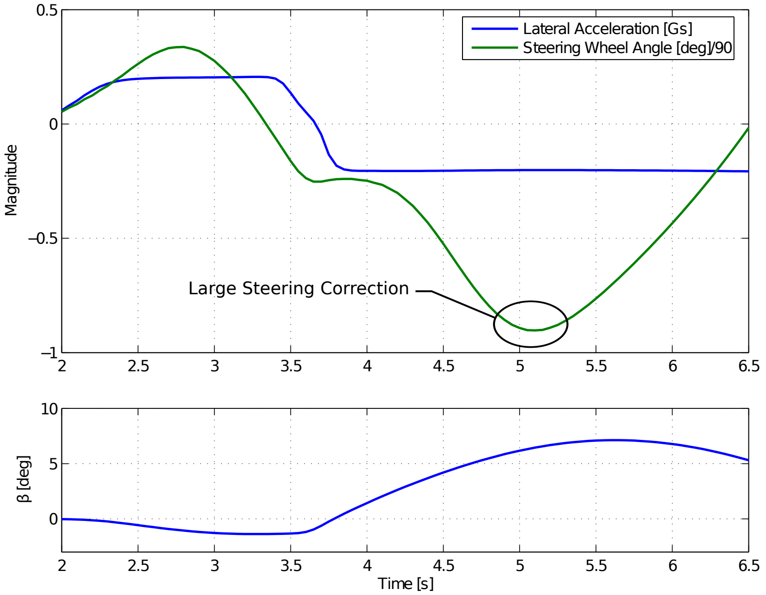


Figure 2.1: Steering Reaction to Large Imbalance - Nominal BMW Mini DLC Maneuver on $\mu = 0.2$ Surface ($V_x = 95$ kph). Note: Steering Wheel Angle is Normalized.

Taking this one step further, the raw data will be initially low-pass filtered at 3.5 Hz. This is the approximate bandwidth of an average driver and this removes high frequency content from other sources such as vehicle interference (i.e, noise and vibrations). It should be noted since this is a real-time algorithm, a first order filter is used to prevent as much lag as possible in the data conditioning step. The data is low-pass filtered again at 0.5 Hz. The remaining low frequency content represents the activity of a ”balanced” driver. It eliminates much of the driver’s corrections to the imbalances and preserves the general steering inputs. The difference between these two filtered sets represents the driver’s reaction to the imbalances in the vehicle [6]. For the same example as above (the low μ, Mini DLC), the filtered signal traces for steering wheel angle are seen in Figure 2.2. A large correction is indicated at 3.5 s and 5 s. Both indicate the driver’s reaction to imbalance. However, considering the vehicle sideslip trace, it is evident that the 5 s indicator is much more important than the 3.5 s indicator because of the magnitude of the vehicle sideslip angle. From this example, it is clear that steering wheel angle can be used to indicate oversteer; however, it must be considered together with the other indicating signals.

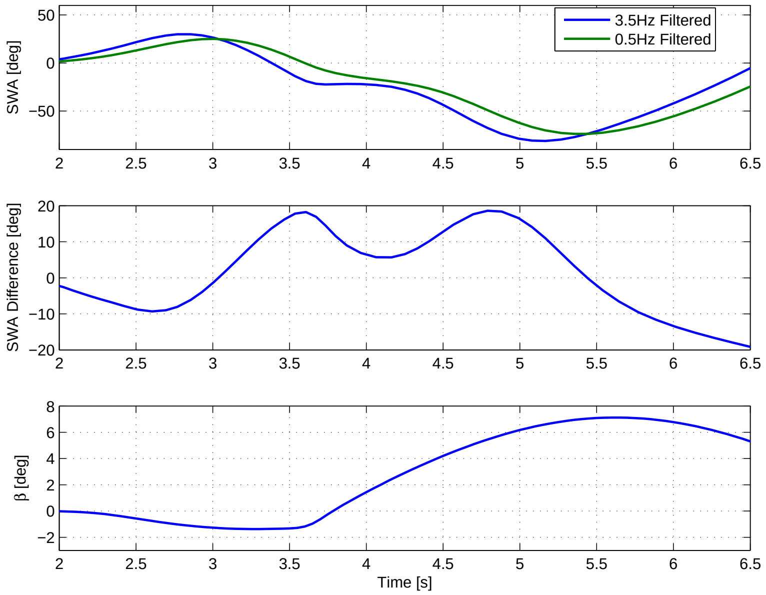


Figure 2.2: Filtered Steering Traces - Nominal BMW Mini DLC Maneuver on $\mu = 0.2$ Surface ($V_x = 95$ kph).

2.1.2 Lateral Acceleration

Lateral acceleration is another very good indicator of an unbalanced vehicle. According to Fey, ”an oversteering chassis frequently produces a rough lateral G graph. Its instability creates dips in the graph as the chassis loses and regains grip. Confirmation lies in the steering graph” [5]. Therefore, when there is a dip in lateral acceleration corresponding with a large steering correction as described above, there is likely an oversteer event.

Like the steering wheel angle traces, the lateral acceleration traces will be double filtered. The 0.5 Hz lowpass signal represents a balanced vehicle and the 3.5 Hz low-passed signal shows much more of the vehicle’s response while filtering inherent noise and unwanted data from other sources. The difference in these two traces represents the vehicle’s imbalance. Figure 2.3 shows the nominal BMW Mini in a DLC maneuver. The two filtered acceleration signals, the difference in these acceleration signals, and the vehicle side slip angle are plotted. At 4.7 and 7 s there are concurrent spikes in both the lateral acceleration difference and the vehicle sideslip that suggest oversteer. However, at 2 s, the spike in lateral acceleration difference does not correspond to a large sideslip angle. Therefore, lateral acceleration alone cannot indicate oversteer.

Lateral acceleration is highly sensitive to road imperfections and bumps can also produce dips in the signal. ”By themselves, lateral G’s are probably the most difficult visual tool for oversteer diagnosis” [5].

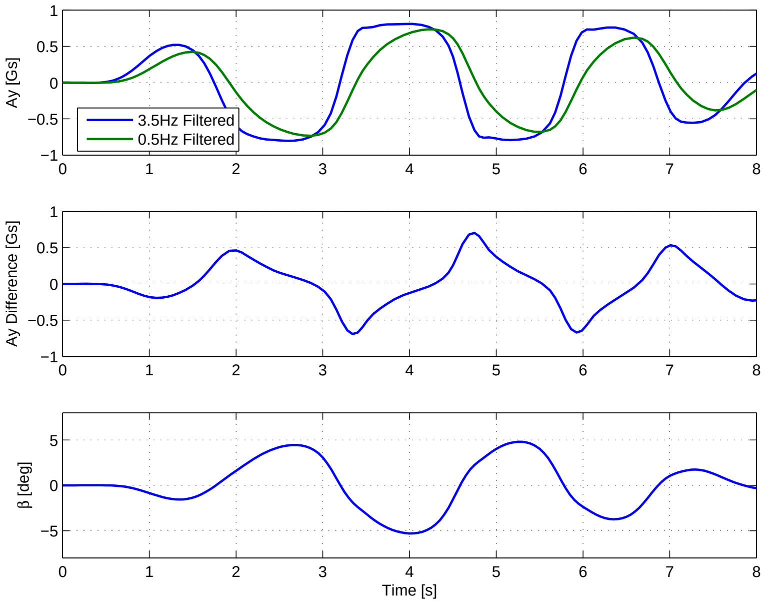


Figure 2.3: Filtered Lateral Acceleration Traces - Nominal BMW Mini DLC Maneuver ($V_x = 185$ kph, $\mu = 0.85$).

2.1.3 Combined Lateral Acceleration and Steering Wheel Angle

In the previous sections, it was shown that a spike in both steering wheel angle difference concurrent with a spike in lateral acceleration difference is a good indicator of oversteer. The nominal BMW Mini on a DLC will be used to illustrate this as seen in Figure 2.4. Here, the absolute differences in the heavily and lightly filtered signals of both lateral acceleration and steering wheel angle are presented along with the absolute vehicle sideslip angle. An oversteer event is indicated when both lateral acceleration difference and steering wheel angle difference produces a reasonably large magnitude at the same time and there is a concurrent spike in the vehicle sideslip angle. This is the basis for the fuzzy logic algorithm that will be presented in the next chapter.

2.1.4 Yaw Rate

The final signal that is used in this ESC algorithm is yaw rate. This signal is a physical check for the rest of the system. If yaw rate is too large, the vehicle is spinning-out. The previous indicator involving steering wheel angle and lateral acceleration depends on the driver's reaction to the imbalance. It is clear in Figure 2.5 that after 12 s of the run, the vehicle is spinning-out because of the excessively high yaw rate ($\dot{\psi} > 60$ deg/s) and extremely high sideslip ($\beta > 20$ deg). This signal combined with the steering difference

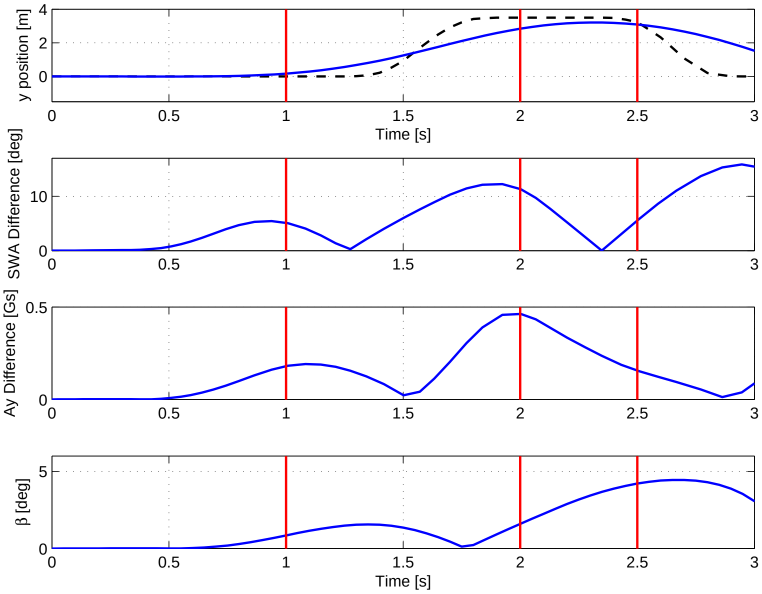


Figure 2.4: Steering and Lateral Acceleration Indicators of Oversteer - DLC with Nominal BMW Mini. Note: Steering Difference is Normalized for Plotting.

and lateral acceleration difference will be the three signals examined in the fuzzy logic to determine a level of oversteer present in the vehicle at each instant.

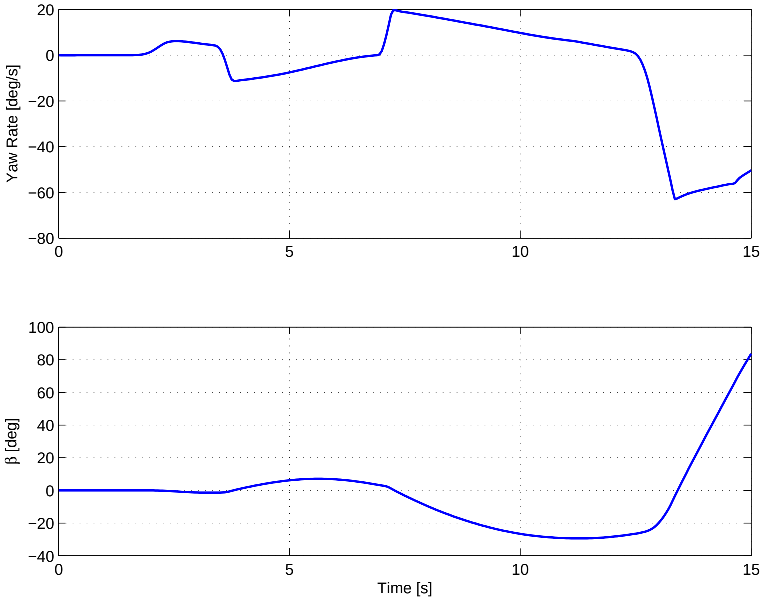


Figure 2.5: Yaw Rate Indicator of Oversteer - Nominal BMW Mini Low Mu DLC ($V_x = 95$ kph, $\mu = 0.2$).

2.2 Fuzzy Logic

Fuzzy logic is extensively used in this research to determine the level of oversteer present from the aforementioned signals. A brief description of the workings of fuzzy logic is given here and for an excellent reference, the reader should refer to Appendix A in Vaduri [6]. ”Fuzzy logic, which can be viewed as an extension of classical logical systems, provides an effective conceptual framework for dealing with the problem of knowledge representation in an environment of uncertainty and imprecision” [7]. It ”is almost synonymous with the theory of fuzzy sets, a theory which relates to the classes of objects with unsharp boundaries in which membership is a matter of degree” [8]. In other words, Fuzzy Logic breaks from the conventional 1 or 0 (*True* or *False*) logic to a multivalued logic to give a degree of an output (i.e. level of oversteer) from the inputs. This logic uses membership functions to apply linguistic operators to the variables and a verbal set of if-then rules to achieve an output. Fuzzy logic follows the following steps which will be discussed in more detail in the following sections [6].

1. Fuzzify the Inputs - Membership Functions
2. Apply the Fuzzy Operator - *AND* or *OR*
3. Apply the Implication Operator - *THEN*
4. Aggregate the Output - Evaluate Each Rule and Sum Results
5. Defuzzify the Aggregate - Return Degree of Output from the Inputs

2.2.1 Fuzzify the Inputs

In this step each input needs to be placed in a fuzzy domain which deals with linguistic operators. In other words, the fuzzy logic toolbox of MATLAB determines how the level of input fits in each membership function. For example, Figure 2.6 shows yaw rate has three membership functions to determine the level of yaw rate: low, medium, and high. The fuzzy logic tool box takes the current level of yaw rate (AV_z) and evaluates how it fits into each membership function. For instance, if the current level of yaw rate is 15 deg/s, the fuzzy logic would evaluate the low yaw rate membership as 0.2, the moderate yaw rate as 0.55, and the high yaw rate as 0 as illustrated in Figure 2.6.

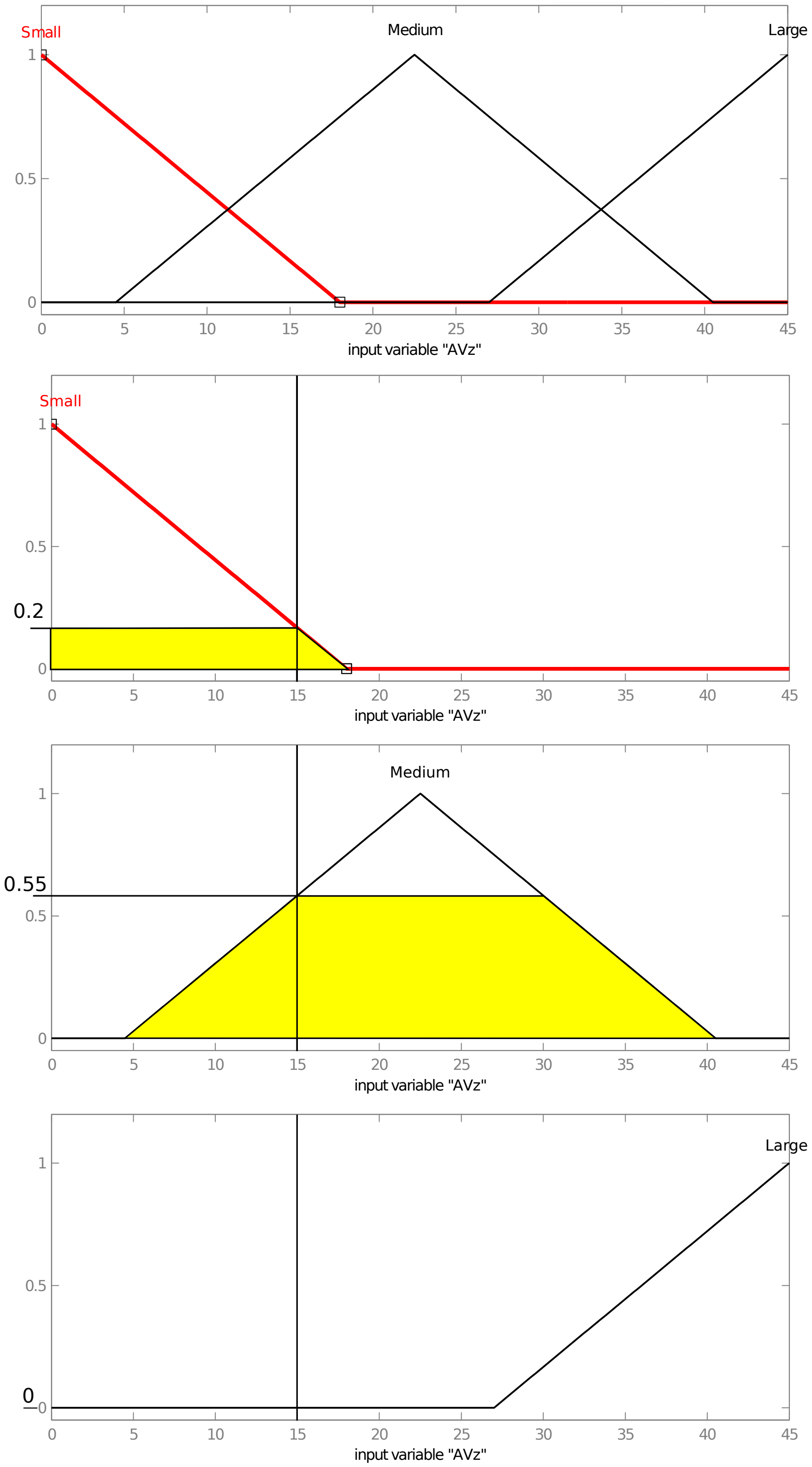


Figure 2.6: Yaw Rate (AVz) Membership Functions, deg/s.

2.2.2 Applying the Fuzzy Operator

After each input has been fuzzified, the fuzzy operator can be applied. This is a simple *AND* or *OR* operator that is used when dealing with multi-input rules. For example, it was previously discussed that when the difference in steering is large *AND* the difference in lateral acceleration is large, then, the level of oversteer is high. The differences represent the imbalance present in the vehicle and the driver's reaction to the imbalance. The difference refers to the difference in the lightly filtered (3.5 Hz) signal and the heavily filtered (0.5 Hz) signal. This step deals with applying the *AND* operator. In fuzzy logic, *AND* represents the minimum of two sets while *OR* represents the maximum as seen in Figure 2.7.

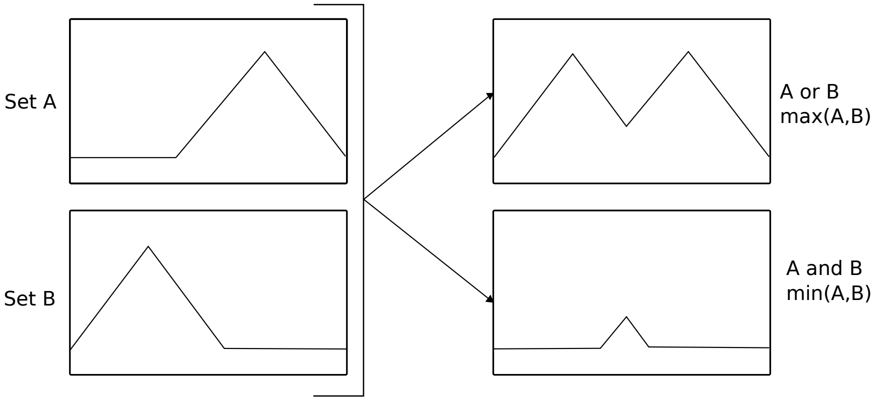


Figure 2.7: Fuzzy Operators Referenced from [6].

2.2.3 Applying the Implication Operator

This section deals with the *THEN* part of an if-then statement. In the previous section the *IF* part is evaluated for each variable and combination of variables in a specific rule. Next, the *THEN* needs to map the input of the if-then to the output. For example, *IF* yaw rate is moderate, *THEN* the level of oversteer is moderate. Using the same 15 deg/s yaw rate as in the previous example, the fuzzified input was rated a 0.55 for its moderate membership function as seen in Figure 2.6. Now, this 0.55 will get mapped to the output variable's moderate membership function at 0.55. Figure 2.8 shows the output variable's moderate oversteer membership function at 0.55. This area in the Moderate Oversteer membership function is the output for this specific rule. Each rule is evaluated like this to give an area of output for the rule.

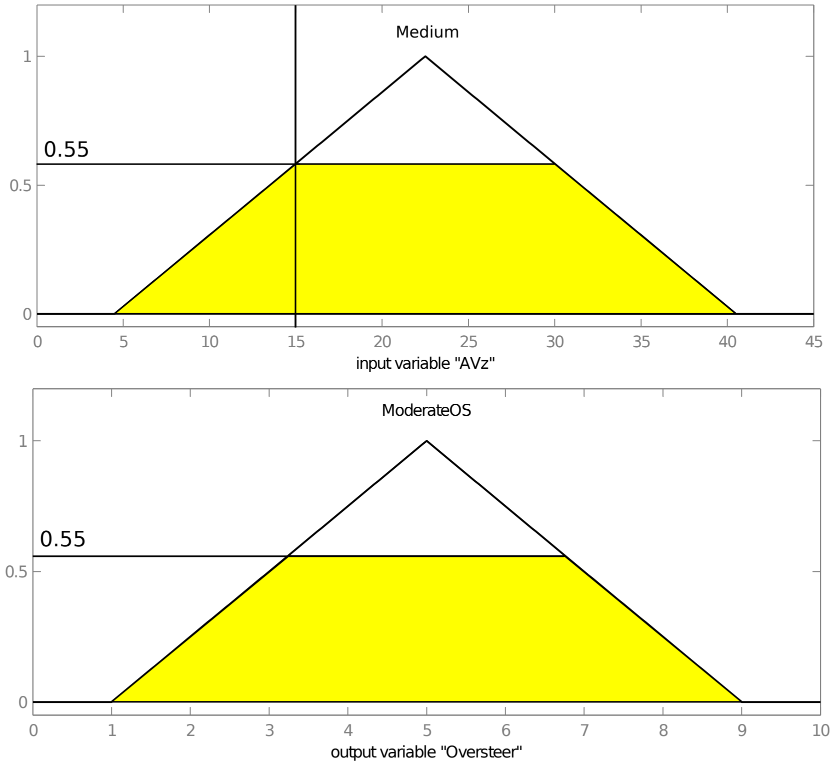


Figure 2.8: Yaw Rate (AVz) Membership Evaluated for Moderate Oversteer, deg/s.

2.2.4 Aggregate the Output for Each Rule

In the next step of the fuzzy process, each rule is evaluated to give an output. For example, the oversteer algorithm discussed in the next chapter has six rules to be evaluated. Each rule will have an output mapped to the oversteer variable. As in the previous example, the output of the rule, ”if moderate yaw rate, then moderate oversteer,” yields Figure 2.8. This is done for each specific rule in the algorithm resulting in a set of outputs for the fuzzy logic. In this step, the outputs of each rule are joined together into a single output by summing their results (summing the output areas of the individual rules). This is illustrated in Figure 2.9 at Step 4.

2.2.5 Defuzzify the Aggregate

Finally, a single aggregated output exists that covers the entire space of the output variable. In this step, a single number will be returned from this aggregated output. There are several defuzzification methods including the following: centroid, bisector, smallest, middle, and maximum. In this research, centroid defuzzification is the method employed. The x-value of the centroid (of the aggregated membership function or summed output area) is defined as the defuzzified output. This is is also illustrated in Figure 2.9 at Step 5.

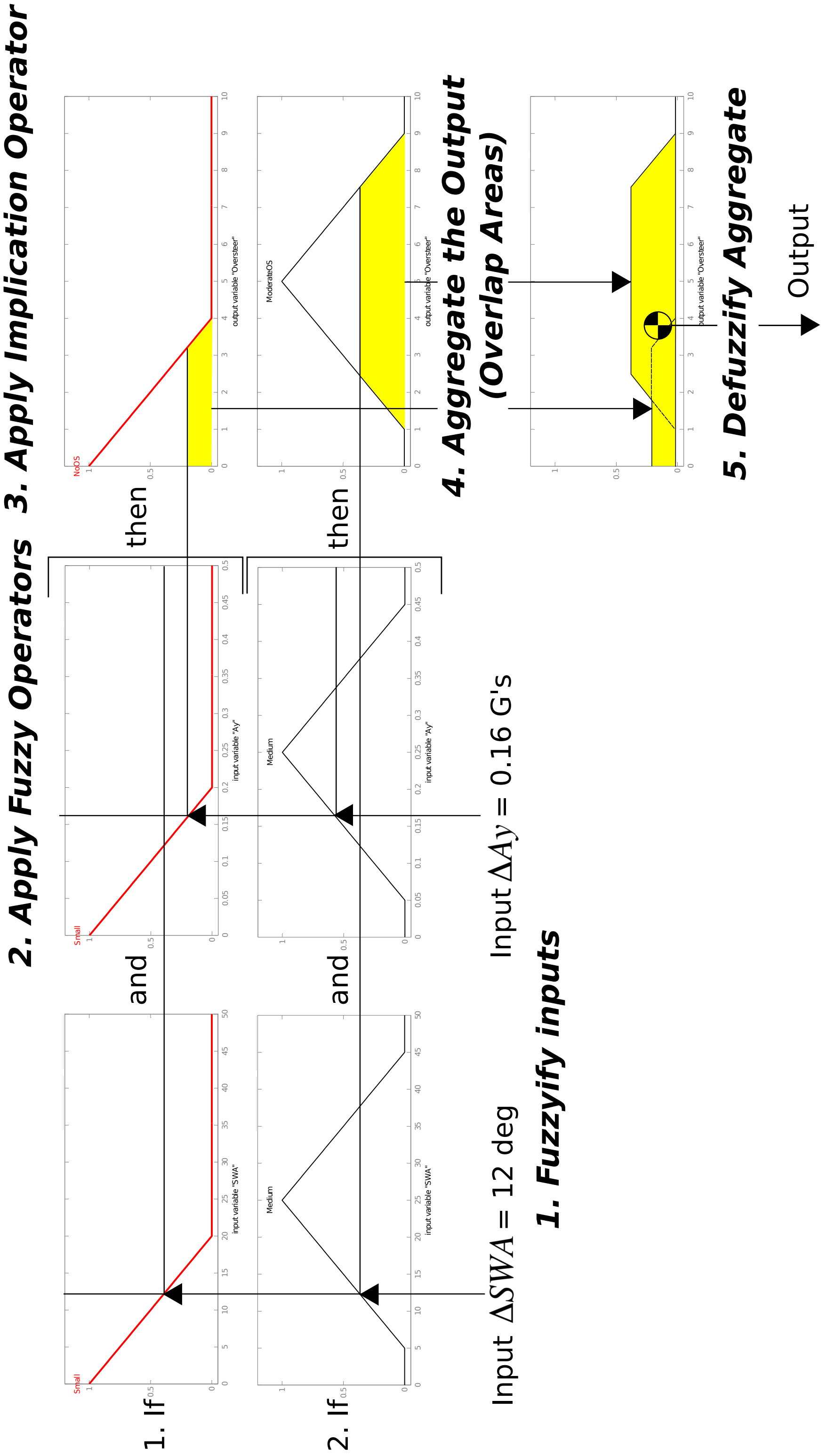


Figure 2.9: Fuzzy Logic Process as Seen in [6].

Chapter 3

FUZZY LOGIC ESC STRATEGY

3.1 Programming Environment

This research utilized a high fidelity vehicle model in CarSim that ran in parallel with the Simulink implementation of the stability control algorithm. These programs both run at a fixed time step. At each time step, CarSim will solve for the vehicle dynamic states and pass variables of interest to Simulink. At that time step, the Simulink program will run and output control inputs (corrective brake torque) back into CarSim where the process will be repeated. This programming environment allows the use of traditional Simulink control programming while still using a very user-friendly and accurate vehicle dynamics solver with over 50 degrees of freedom [9].

3.2 Import/Export Variables to ESC Strategy

From our vehicle model in CarSim, variables are passed in and out of the Simulink workspace as shown in Figure 3.1. This figure shows the function in the Simulink block diagram where variables are passed in and out of CarSim. On the left, inputs (wheel brake torques, driver model throttle, and driver model road wheel angle if used) to CarSim from the control strategy are shown. On the right of the CarSim S-Function are the output variables that Simulink imports from CarSim. A summary of the parameters used in the ESC strategy are shown in Table 3.1. As discussed in the previous chapter, lateral acceleration, steering wheel angle, and yaw rate will be used in the fuzzy logic structure to determine a level of oversteer. After a level of oversteer has been determined, the correct control action (brake torque) is applied to induce a yaw moment which will stabilize the vehicle.

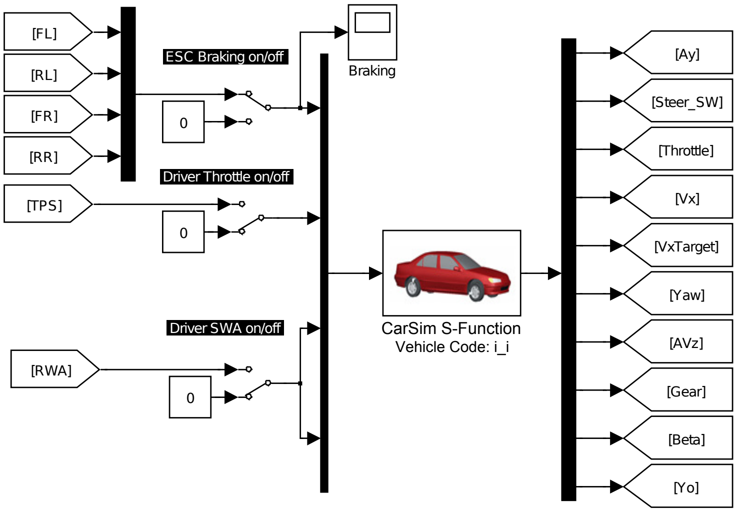


Figure 3.1: CarSim Input/Output Parameters in Simulink.

Parameters Imported from CarSim:

Variable Name	Symbol	Units
Ay	Lateral Acceleration	G's
Steer_SW	Steering Wheel Angle	deg
Vx	Longitudinal Velocity	kph
AVz	Yaw Rate	deg/s

Parameters Exported to CarSim:

Variable Name	Symbol	Units
FL	Front Left Brake Torque	$N \cdot m$
RL	Rear Left Brake Torque	$N \cdot m$
FR	Front Right Brake Torque	$N \cdot m$
RR	Rear Right Brake Torque	$N \cdot m$

Table 3.1: Summary of CarSim Import/Export Parameters.

3.3 Data Conditioning

After the parameters of interest are imported to the Simulink workspace, they must be appropriately conditioned as seen in Figure 3.2. Steering wheel angle and lateral acceleration are passed through a first order, lowpass Butterworth filter set at 3.5 Hz (approximately the bandwidth of an average driver [6]). Next, these signals are further lowpass filtered at 0.5 Hz. Signals with a bandwidth of 0.5 Hz are assumed to represent a balanced vehicle or driver [10]. These low bandwidth signals show general trends in vehicle and driver behavior. The differences between the heavily filtered (0.5 Hz) and the lightly filtered (3.5 Hz) traces represent the imbalances in the vehicle and the driver's reaction to the imbalance [10]. The differences of lateral acceleration and steering wheel angle are passed to the fuzzy logic control structure along with the absolute yaw rate signal trace which has been heavily filtered (at 0.5 Hz) to determine a level of oversteer.

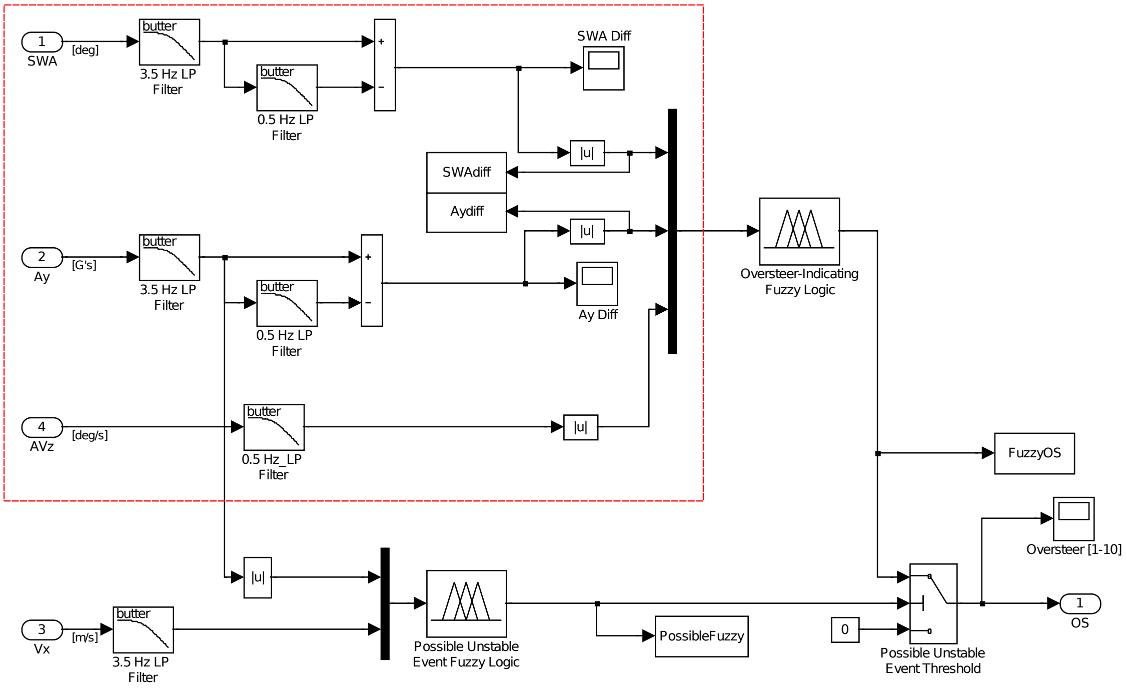


Figure 3.2: Simulink Data Conditioning (In Red Dashed Box).

3.4 Fuzzy Logic Strategy

Now that the signal traces have been appropriately conditioned, the fuzzy logic strategy can be examined (the conditioned signals are the outputs of the red-dashed block in Figure 3.2). The fuzzy logic structure that receives the outputs of the red-dashed box in Figure 3.2 is a three input, one output structure which uses a centroid defuzzification method to evaluate the level of oversteer.

3.4.1 Membership Functions

As discussed in Chapter 2, fuzzy logic requires defining membership functions to determine the level of each input signal. For this algorithm, large, medium, and small triangular membership functions are defined for each input parameter (difference in lateral acceleration, difference in steering wheel angle, and absolute yaw rate). The output parameter, i.e., the level of oversteer is defined from 0-10.

In this research, the values for the ranges of each membership function were chosen by examining the vehicle dynamic response through the double lane change maneuver. It should be noted that the nominal BMW Mini was the vehicle used to tune the algorithm (limits of the membership functions) and this single algorithm was used in all case studies presented in Chapter 5. The input and output membership functions can be seen in Figure 3.3, and are summarized in Table 3.2.

Each of these membership functions can be tuned to increase performance of the ESC strategy for a particular vehicle; however, the goal of this ESC strategy was to work on multiple vehicles in multiple configurations. For instance, in a sports car versus an SUV, it would be expected that the sports car could produce a higher acceptable yaw rate and the larger differences in lateral acceleration and steering. Consequently, the input ranges could be tuned to higher maximum values which would be more appropriate for this vehicle.

Variable Name	Units	Total Range	Small Membership	Medium Membership	Large Membership
Inputs:					
Steering Imbalance	deg	0-50	0-20	5-45	30-50
Lateral Acceleration Imbalance	G's	0-0.5	0-0.2	0.05-0.45	0.3-0.5
Absolute Yaw Rate	deg/s	0-45	0-17.5	4-41	27.5-45
Outputs:					
Level of Oversteer	_	0-10	0-4	1-9	6-10

Table 3.2: Summary of Membership Function Values for Oversteer-Indicating Fuzzy Logic Structure

3.4.2 Rules

Now that the inputs have been fuzzified from their membership functions, a set of rules must be applied to the inputs to find the appropriate output. The linguistic operators from the membership functions must be ordered in such a way to determine the correct level of oversteer. For example, if the imbalance in steering is large *and* the imbalance in lateral acceleration is large, *then* it is heavy oversteer. For this research, the *and* operator is used exclusively for rules containing two sets of inputs. In fuzzy logic, *and* will represent the minimum of each set included while *or* would represent the maximum. The following is the set of rules used to determine the level of oversteer present, and are summarized in Figure 3.4 where the input parameters are mapped to a level of output. Since lateral acceleration and steering wheel angle are combined in each rule, they are plotted together to show their contribution to the oversteer number. Yaw rate is independent of the other two variables and is plotted alone to show its contribution to the output.

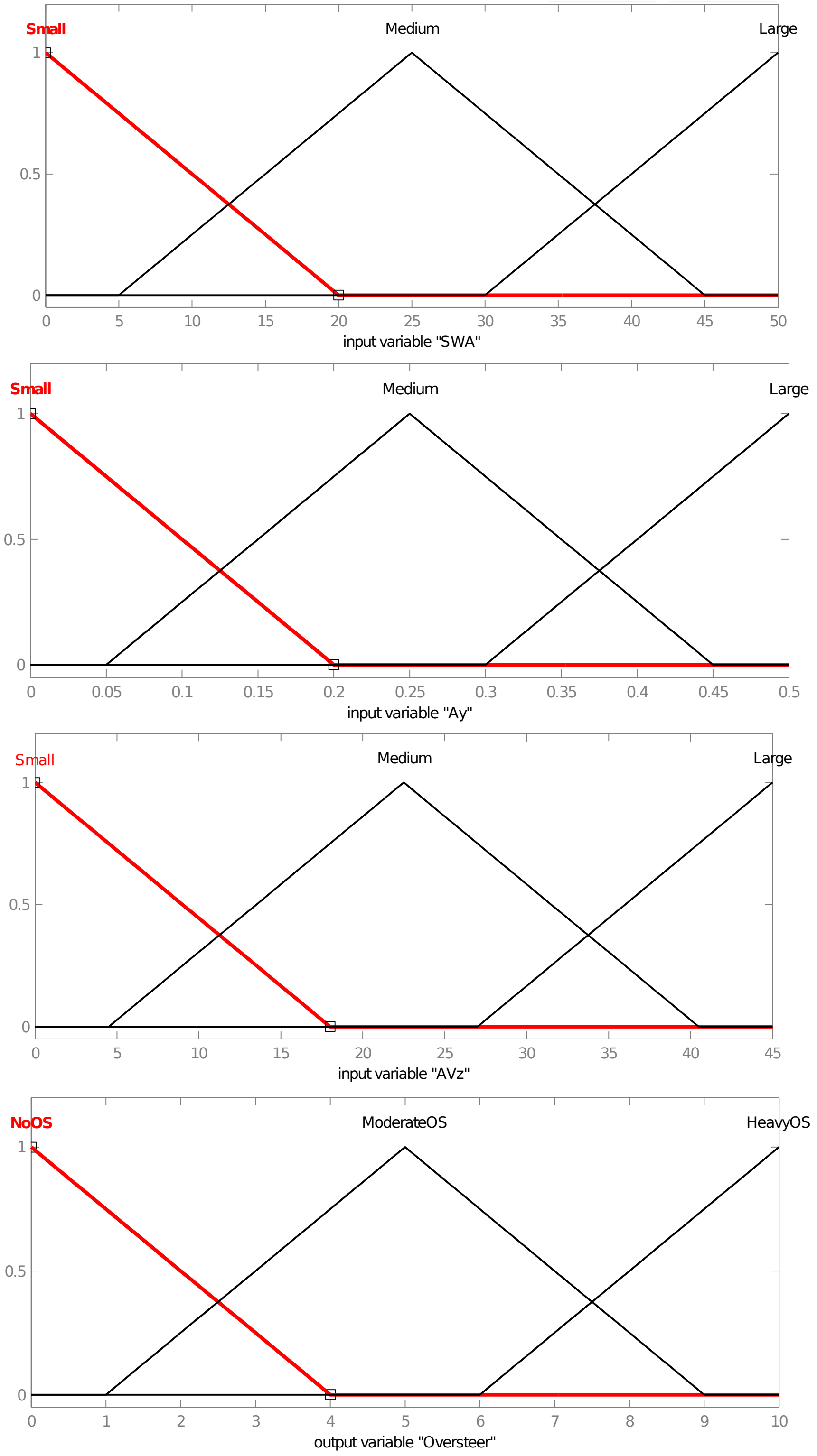


Figure 3.3: Oversteer Fuzzy Logic Membership Functions.

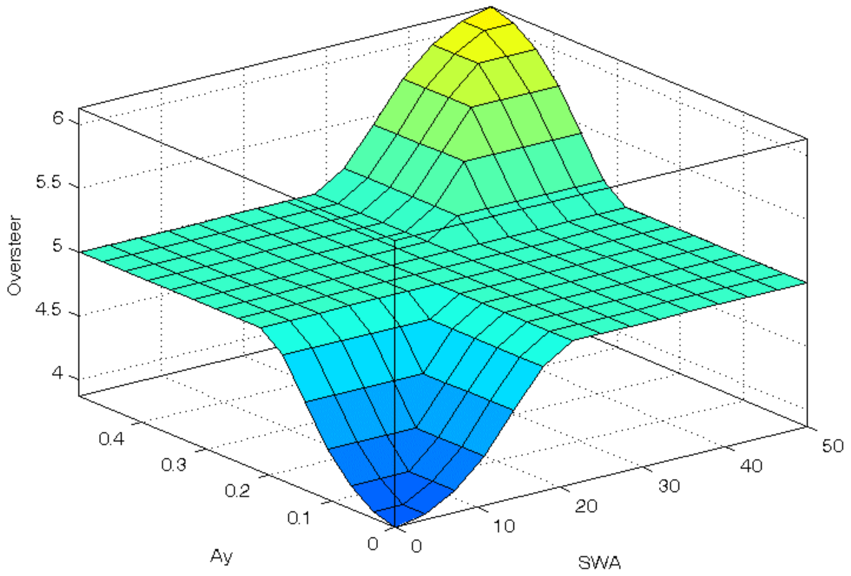
- If SWA is Small *and* Ay is Small *then* Oversteer is No Oversteer
- If SWA is Medium *and* Ay is Medium *then* Oversteer is Moderate Oversteer
- If SWA is Large *and* Ay is Large *then* Oversteer is Heavy Oversteer
- If AVz (i.e., yaw rate) is Small *then* Oversteer is No Oversteer
- If AVz is Medium *then* Oversteer is Moderate Oversteer
- If AVz is Large *then* Oversteer is Heavy Oversteer

3.4.3 Compiling and Defuzzification

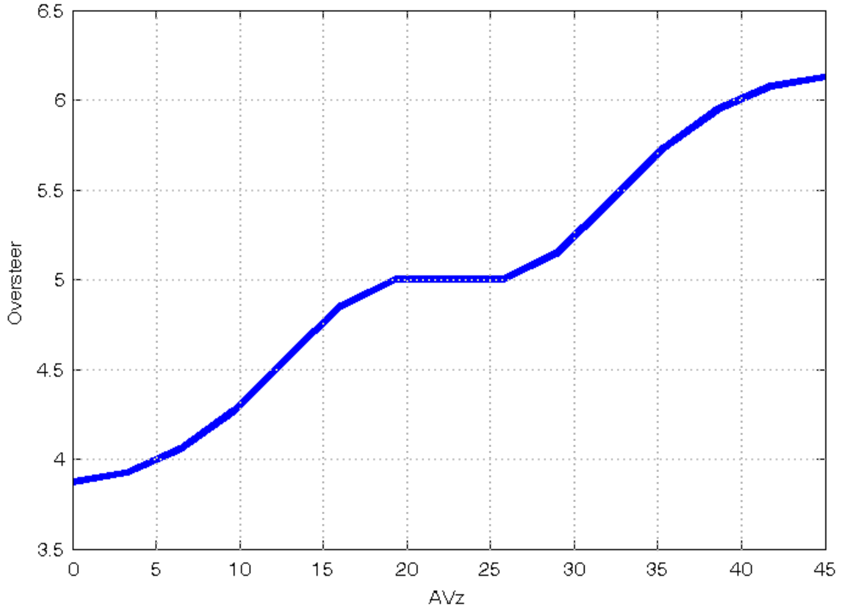
At each time-step, the fuzzified inputs are evaluated with the programmed set of rules one rule at a time. For example, the first rule looks how the difference in steering wheel angle and difference in lateral acceleration fit into their small membership functions. It determines the minimum of the two in their respective membership functions and finds the corresponding level of oversteer in the No Oversteer membership function. This level results in an area in the oversteer domain and is the outcome of the first rule. Each consecutive rule is evaluated in this manner to give an area in the oversteer domain. Next, the sum of the areas calculated by each rule is computed resulting in a final overall area in the oversteer domain. Finally, the centroid of this area is found and the level of oversteer is defined as the x-value of the centroid. An example is presented in Chapter 4 to clarify this process. Once a level of oversteer is found, control action can be applied from the indicated level.

3.5 Thresholds

From the three dynamic traces (Ay, SWA, and AVz) a level of oversteer can be selected; however, it can lead to incorrect levels of oversteer in certain cases. For example, at low speeds the driver will tend to steer much more than at high speed; this can lead to a false indication of oversteer at parking lot speeds. Therefore, a strategy must be employed to only initiate a control action after a certain speed is attained. Also, when the oversteer-indicating fuzzy logic applies corrective braking, the vehicle dynamic response at the next time step is evaluated and usually gives a much lower level of oversteer as the correction helped stabilize the vehicle. This leads to the brakes being prematurely released. For this problem, a hold algorithm is employed



a.) Oversteer Number Versus Difference in Lateral Acceleration, A_y [G's] and Difference in Steering Wheel Angle, SWA [deg]. Reference Yaw Rate (AVz) = 22.5 deg/s.



b.) Oversteer Number Versus Absolute Yaw Rate, AV_z [deg/s]. Reference Difference in Lateral Acceleration (A_y) = 0.25 G and Reference Difference in Steering Wheel Angle (SWA) = 25 deg.

Figure 3.4: Oversteer Rule Summary.

to hold the current level of oversteer for some time or until the oversteer level increases. In this ESC strategy 5 s was chosen as the hold time based on the nominal BMW Mini DLC; however, it should be noted that the level of oversteer fluctuates enough that ESC performance is not very sensitive to this parameter. Generally, 5 s of holding is rarely reached unless the vehicle is in a severe spin.

3.5.1 Unstable Event Threshold

To alleviate this problem of low speed control, a separate fuzzy logic structure using longitudinal speed (Vx) and lateral acceleration (Ay) to determine the level of a possible unstable event is used. As in the oversteer-indicating fuzzy logic structure, membership functions need to be defined for both sets of inputs to transform the values of each into linguistic operators. These can be seen in Figure 3.5 and are summarized in Table 3.3. In this case, the output has been divided into four membership functions to achieve better definition of the possible unstable event.

Input	Units	Total Range	Small Membership	Medium Membership	Large Membership
Lateral Acceleration, Ay	G's	0-1.1	0-0.44	0.1-1	1.7-1.1
Longitudinal Velocity, Vx	kph	0-125	0-50	13-113	75-125

Output	Total Range	Stable	Moderately Stable	Moderately Unstable	Unstable
Possible Unstable	0-10	0-3.3	0-6.667	3.333-10	6.667-10

Table 3.3: Summary of Membership Function Values for Indicating a Possible Unstable Event.

Now that the membership functions have been defined for all of the input and output variables, rules must be applied to evaluate the linguistic operators. The following are the nine rules used to determine a possible unstable event. These are summarized in the surface plot of Figure 3.6.

- If Ay is Small *and* Vx is Slow *then* Event is Stable
- If Ay is Medium *and* Vx is Slow *then* Event is Stable
- If Ay is Large *and* Vx is Slow *then* Event is Stable
- If Ay is Small *and* Vx is Medium *then* Event is Stable
- If Ay is Medium *and* Vx is Medium *then* Event is Moderately Stable
- If Ay is Large *and* Vx is Medium *then* Event is Moderately Unstable

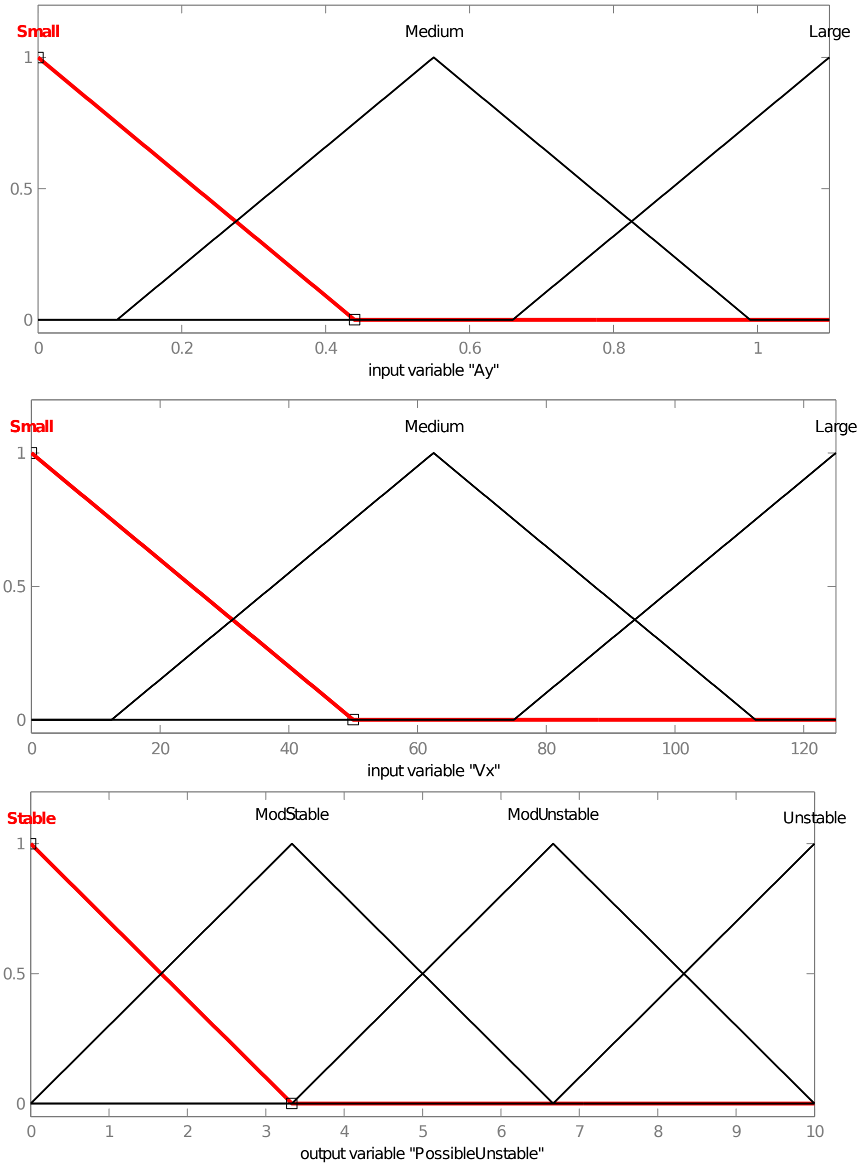


Figure 3.5: Fuzzy Logic Membership Functions for Indicating a Possible Unstable Event.

- If Ay is Small *and* Vx is Fast *then* Event is Stable
- If Ay is Medium *and* Vx is Fast *then* Event is Moderately Unstable
- If Ay is Large *and* Vx is Fast *then* Event is Unstable

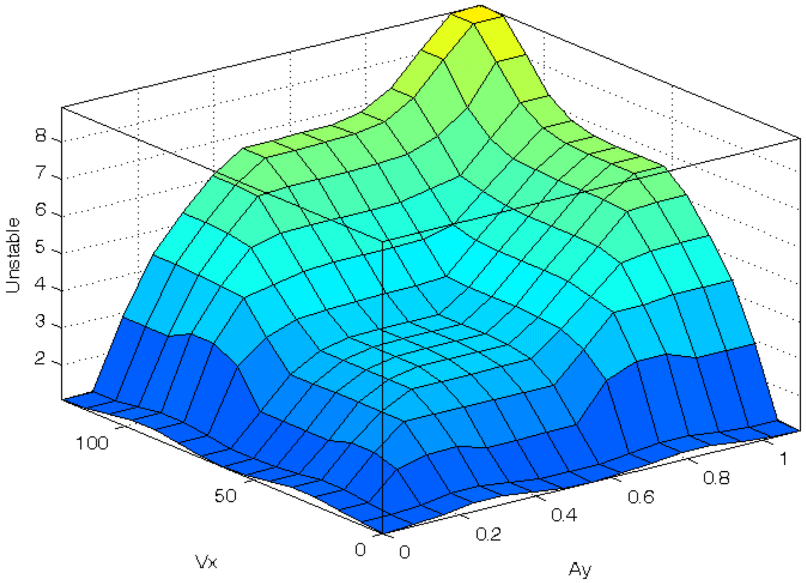


Figure 3.6: Rules to Evaluate a Possible Unstable Event.

When the possible unstable event reaches a threshold of 4 (based on tuning during the DLC maneuver), it allows the oversteer fuzzy logic structure to pass its level of oversteer. This is shown in Figure 3.7 which is the same as Figure 3.2 but emphasizes the threshold portion determined by the fuzzy logic for a possible unstable event. In other words, when the possible unstable event number is less than 4, a zero is indicated for the current level of oversteer. When the unstable event is greater than 4, the oversteer-indicating fuzzy logic's number is then used for the current level of oversteer. The possible unstable event fuzzy logic and its threshold is shown in Figure 3.7 and its result is discussed in detail in the example presented in Chapter 4.

3.5.2 Oversteer Hold

As described before and as seen in Figure 3.9, a simple hold algorithm is employed in Simulink to hold the current level of oversteer for a set amount of time or until the level of oversteer is increased.

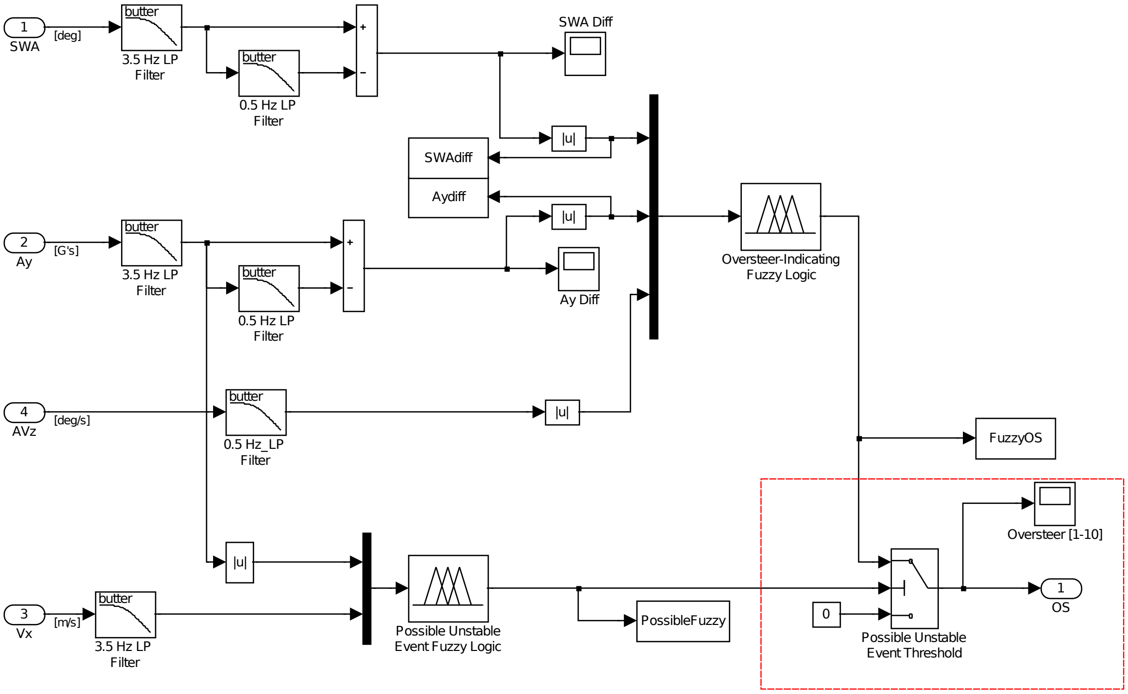


Figure 3.7: Simulink Threshold (Shown in Red Dashed Line) for a Possible Unstable.

The main reason for this algorithm is to prevent the oversteer level from prematurely dropping after the control action. For example when a large enough oversteer level is indicated and control action is initiated, a much lower oversteer value is usually calculated before the vehicle is stabilized and the control action is prematurely released. The Simulink block that performs the hold can be seen in Figure 3.9. In this block, the derivative of the oversteer signal is used to determine how the signal is changing (increasing/decreasing). If the signal starts to decreasing, it will sample and hold the current level of oversteer and the time at which this occurs. From this, the amount of time that the signal is held can be calculated from the time that the signal was held and the current time of the simulation. From here, an embedded MATLAB function is employed to determine which signal to pass (current oversteer, or held oversteer). The MATLAB function starts by examining whether the oversteer signal is increasing or decreasing. If it is increasing, then it will pass the oversteer number at that time step. If the oversteer number is decreasing, it will look at how long the signal has been held (from the previous sample and hold block). If it is less than the programmed hold time of 5 s, it will pass the held oversteer value. If the time has exceeded 5 s, the current oversteer value is passed. In other words, the oversteer hold algorithm will maintain local maxima as the current level of oversteer unless the value has been held for too long. Both the oversteer hold and the possible unstable event are further discussed in Chapter 4. Figure 3.8 shows the result of the oversteer hold block seen in Figure 3.9. It can be seen when the signal stops increasing, the oversteer value is held until the signal increases again (2.5, 2.7, 3.2, 3.9, and 4.3 s). The output, seen as a red dashed line, shows how the oversteer number (solid blue trace) is altered.

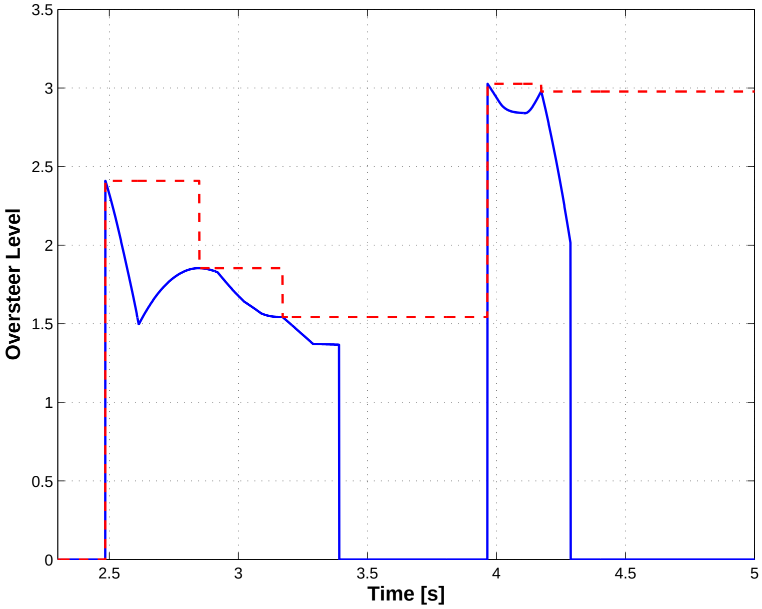


Figure 3.8: Result of Oversteer Hold Block.

3.6 Control Action

Now that the final number for oversteer has been calculated, corrective braking can be determined and applied to the vehicle. For this research, the oversteer level (0-10) was broken into three levels: no correction, moderate correction, and heavy correction. When the oversteer level was between 0 and 2, there will be no corrective action. When oversteer was indicated between 2 and 5.5, a moderate control strategy is employed. Finally, if the oversteer number is greater than 5.5, heavy correction is applied.

3.6.1 No Corrective Action

As described, if the oversteer number calculated by the ESC algorithm is less than 2, no control action will be applied to any wheel as the vehicle is currently stable.

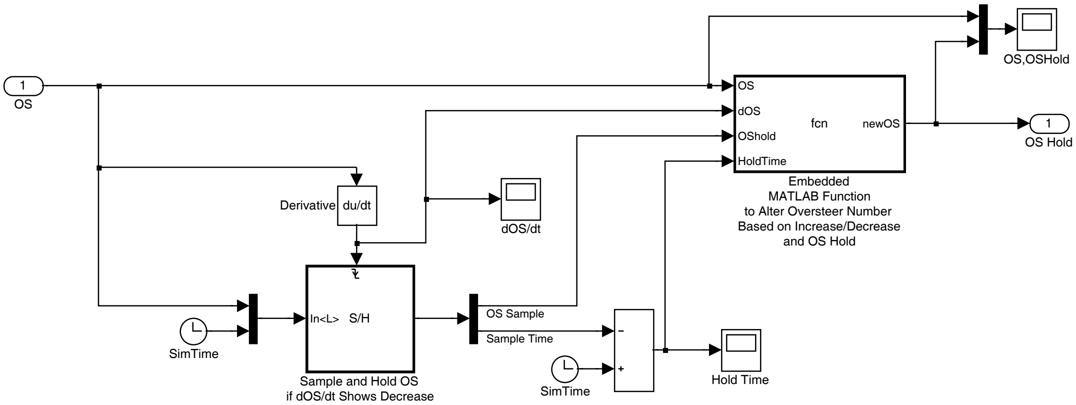


Figure 3.9: Simulink Block to Hold Current Oversteer Value.

3.6.2 Moderate Correction

In this case braking proportional to the yaw acceleration is used to slow/stop the vehicle spin. Starting with the free body diagram in Figure 3.10 where the brake force is applied to the left front wheel and summing moments leads to Equation 3.1. Braking can be applied to the front left or right depending on the direction of the spin. This is determined in the fuzzy algorithm from the derivative of the yaw rate signal i.e., the yaw acceleration. The sign of the vehicle yaw angle reveals the direction of the spin and simple if/then statements are used to apply the braking force to the correct wheel (front left or front right).

$$I_z\ddot{\psi} = -F_b\tfrac{T_f}{2} \tag{3.1}$$

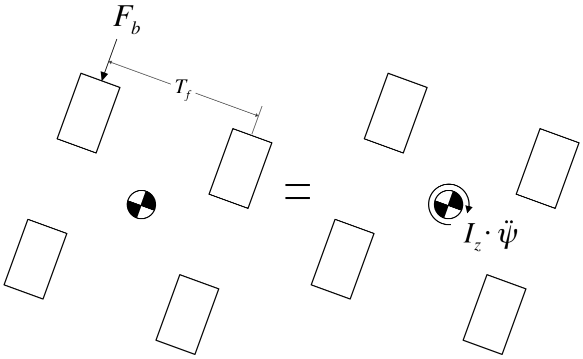


Figure 3.10: Free Body Diagram Illustrating Braking Proportional to Yaw Acceleration.

The braking torque is equal to the brake force times the effective rolling radius of the tire (Equation 3.2) and is proportional to the yaw acceleration.

$$M_z = F_b R_{eff} = R_{eff}\left(-\frac{2I_z\ddot{\psi}}{T_f}\right) = \frac{2I_z R_{eff}}{T_f}\ddot{\psi} = C\ddot{\psi} \tag{3.2}$$

The constant (C) was used in all ten case studies and proved to work well. These studies contained a wide variety of vehicles of different weights and properties and the constant C developed using the BMW Mini worked well for all simulations.

3.6.3 Heavy Correction

The heavy correction is defined when the level of oversteer is greater than 5.5. In this case a ”large” braking torque is applied to the correct wheel to apply as much of a yaw moment as possible. This is the worst case if the controller reaches a very high oversteer level and as much control as possible needs to be applied to prevent a spin-out. Therefore, the braked wheel must produce as much longitudinal force as possible to induce as much of a yaw moment as possible to cancel the spin. There is enough braking commanded with heavy correction to saturate the tires.

Chapter 4

EXAMPLE

4.1 Introduction

In this chapter, a detailed outline of how the fuzzy ESC strategy reacts to a typical maneuver and prevents the vehicle from spinning-out is presented. For this example, the BMW Mini vehicle model is used with degraded rear tires to simulate a worse case driving scenario that results in a situation where a typically understeering (with nominal tires) car experiences oversteer. The entire lateral force versus slip angle and loading surface is degraded by 30 % as seen in Figure 4.1. The Mini with degraded rear tires will complete a double lane change (DLC) maneuver (Figure 4.2) under normal driving conditions ($\mu = 0.85$) at 100 kph. In this example, the vehicle is given an initial velocity and coasts through the lane change maneuver. This way, there is as much longitudinal weight transfer forward as possible and the back tires are ”light,” simulating a worse case scenario under normal driving conditions. When the back tires are light, the result is the tire will not be able to generate as much lateral force as a heavily loaded tire. This can also be seen in Figure 4.1 which shows how lateral force generation is affected by normal loading. Since the degraded back tires cannot generate as much lateral force as a normally loaded tire, they are prone to saturate. If the back tires saturate before the front, the vehicle will become unstable and tend to spin-out.

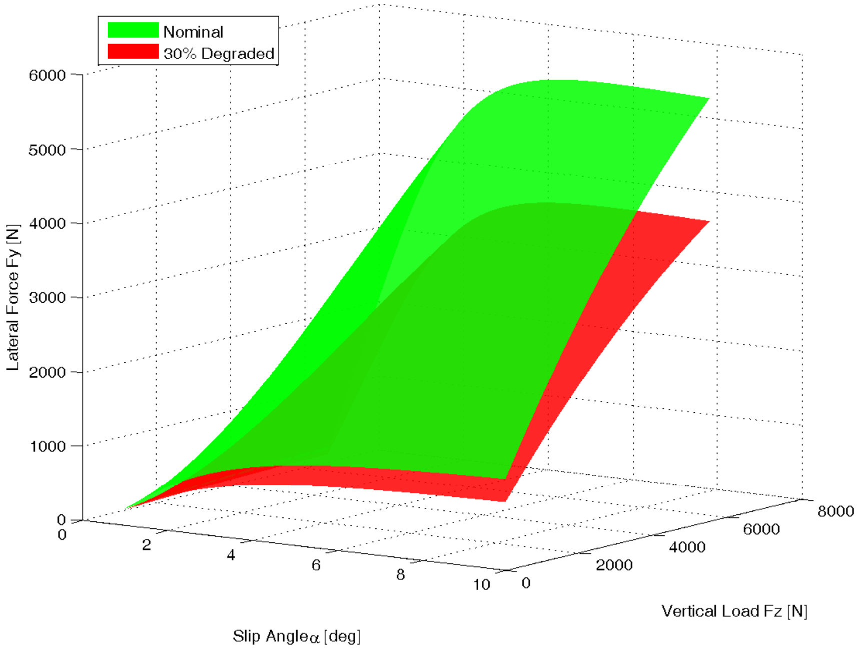


Figure 4.1: Rear Tire Lateral Force Curves for Nominal and 30 % Degraded.

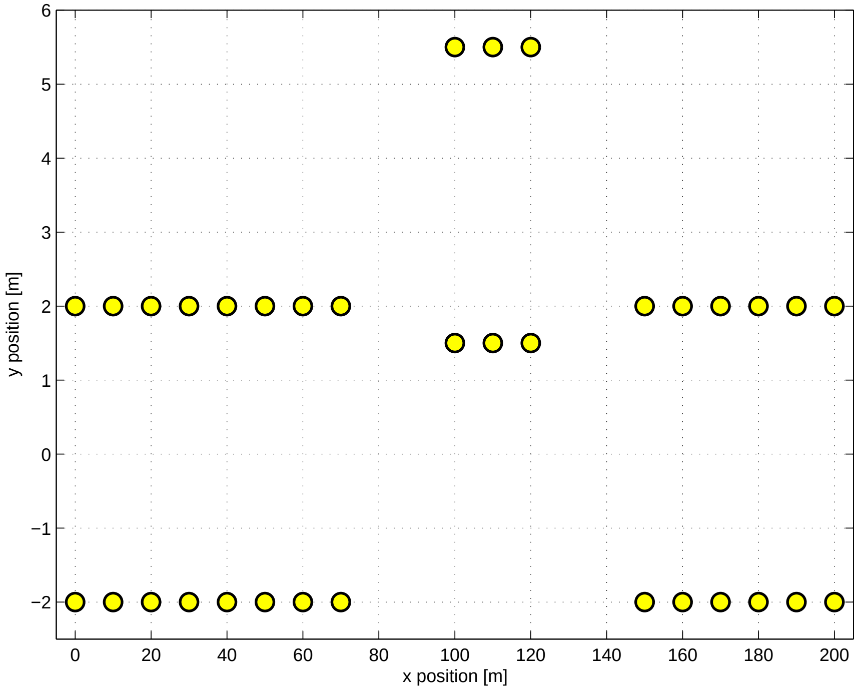


Figure 4.2: Cones Outlining Double Lane Change Course.

4.2 Normal Trajectory and Time Histories for Stable Double Lane Change at 100 kph

First, the traces for a vehicle traversing the DLC at a reasonable operating speed, 100 kph, will be examined. Figure 4.3 shows the trajectory of the degraded BMW Mini without ESC through this lane change maneuver. From the trajectory plot, one can see minimal overshoot and a very successful lane change maneuver.

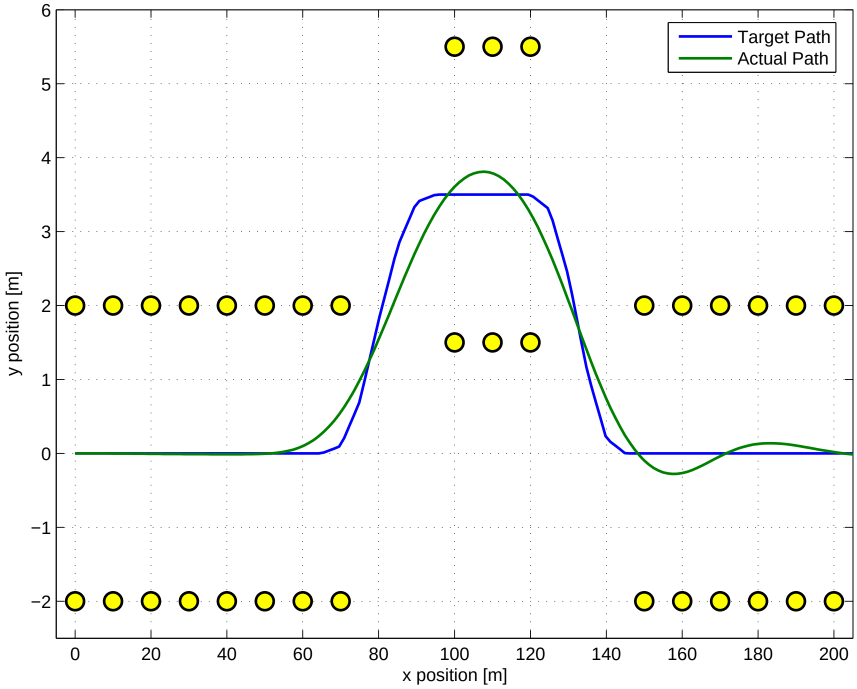


Figure 4.3: Trajectory for Degraded Mini Traversing the Double Lane Change at 100kph no ESC.

Examining the time histories in Figure 4.4 shows that the vehicle is very stable and remains under 3 deg of vehicle slip angle, β. Table 4.1 shows the maximum absolute value of each response parameter which proves the vehicle remained stable through the maneuver.

4.3 Increased Speed Double Lane Change at 105 kph

Now, the initial speed will be increased to 105 kph. Figure 4.5 shows the difference in trajectories from the 100 kph case. The increased velocity causes the rear tires to saturate, thus causing the vehicle to spin out. The time histories in Figure 4.6 show the differences between a successful DLC maneuver and one that

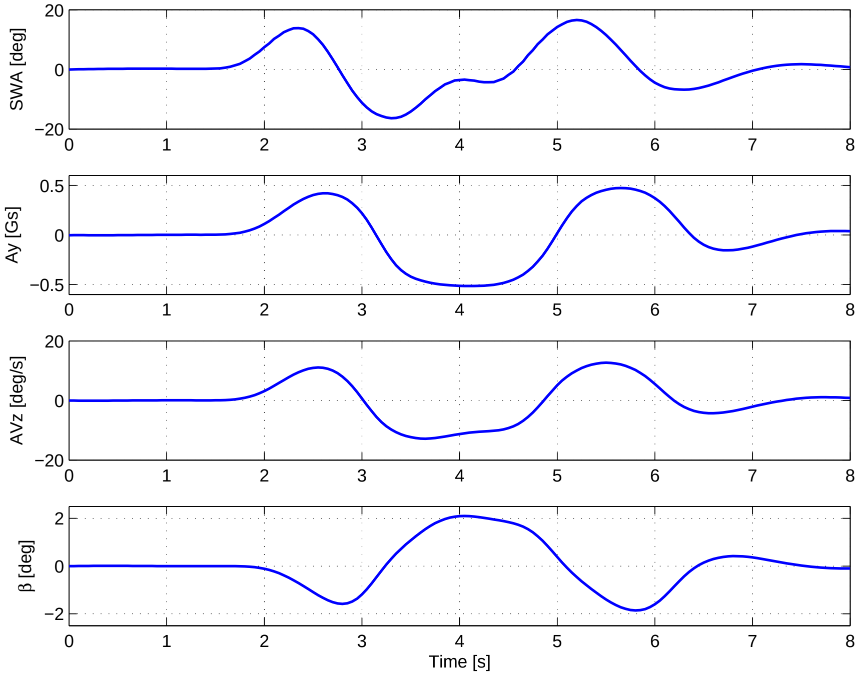


Figure 4.4: Vehicle Dynamic Traces for 100 kph DLC no ESC. Steering Wheel Angle (SWA), Lateral Acceleration (Ay), Yaw Rate (AVz), and Vehicle Sideslip Angle (β).

will result in the vehicle becoming unstable. The maximum values are summarized in Table 4.1. From the vehicle sideslip angle graph, it is clear that the vehicle is spinning out of control after 5.25 s when the sideslip angle sharply increases past 5 deg off the chart. Yaw rate produces a staggering increase at this time as well. Right a 5 s, the driver suddenly changes direction of steering (quickly counter-steers) which is how the driver typically reacts to a spin. Lateral acceleration also shows a slide; after 5 s, the graph plateaus showing the vehicle cannot produce any more lateral acceleration than it is currently. When looking at the four vehicle dynamic traces together, it is clear that at 105 kph, the vehicle spins out. The fuzzy logic will be cueing on differences in lateral acceleration, difference in steering, and yaw rate and are presented here in Figure 4.7 as well. These differences and yaw rate are much higher for the unstable. This shows how the oversteer number is much higher for the unstable case. The way these values are calculated will be discussed in next sections.

Variable Name	100 kph	105 kph	Units
Steering Wheel Angle, SWA	16.6	360	deg
Lateral Acceleration, Ay	0.51	0.74	G’s
Yaw Rate, AVz	12.8	87.9	deg/s
Vehicle Slip Angle, β	2.1	152.1	deg

Table 4.1: Maximum Values of Degraded Mini DLC at 100kph.

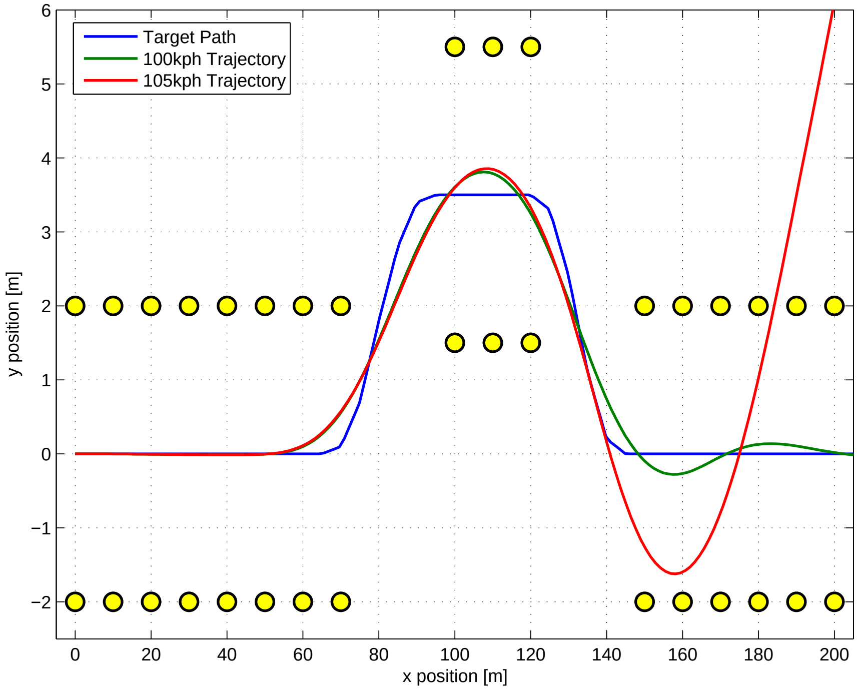


Figure 4.5: Trajectories for Degraded Mini Traversing the DLC at 100 kph and 105 kph no ESC.

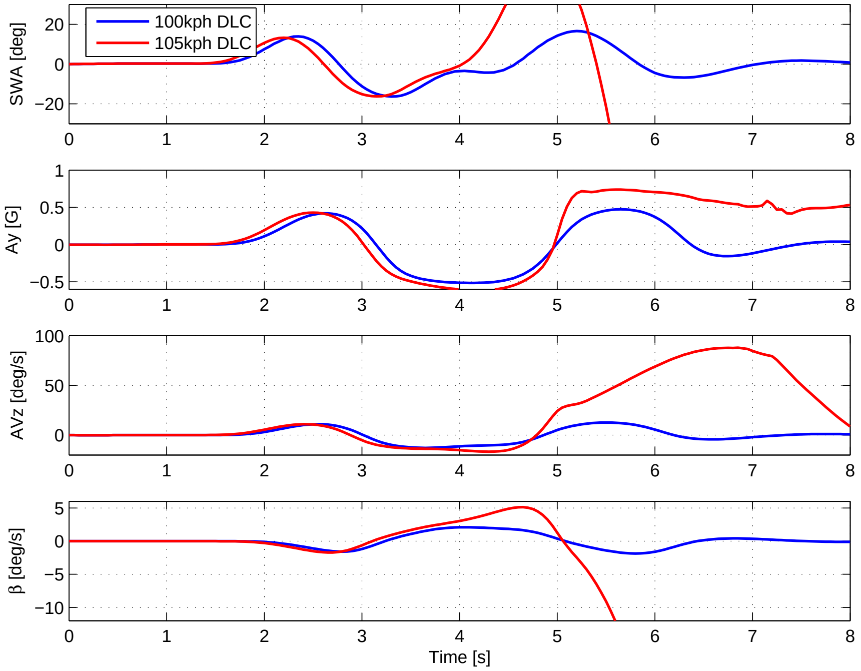


Figure 4.6: Vehicle Dynamic Traces for 100 kph and 105 kph DLC no ESC.

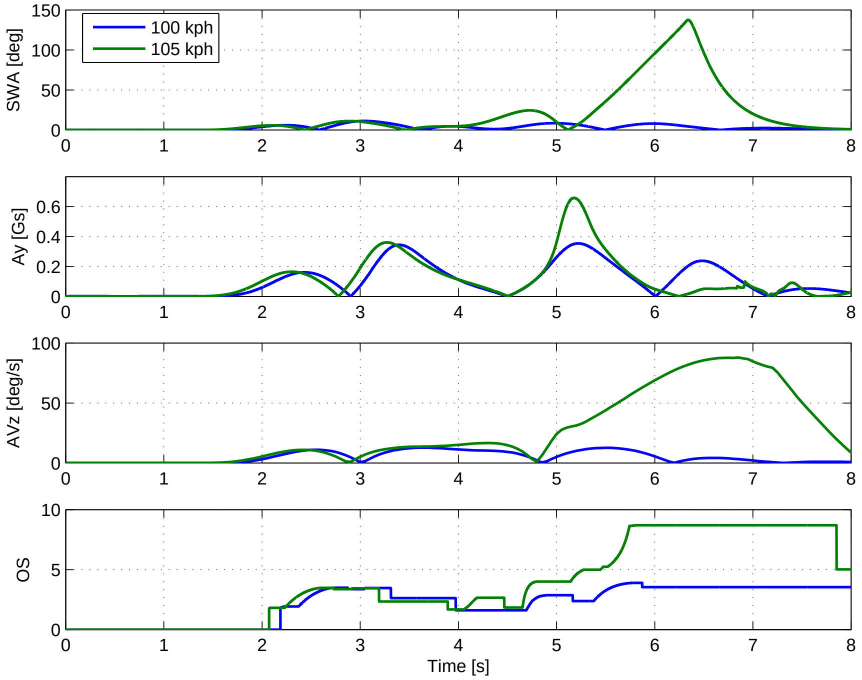


Figure 4.7: Vehicle Dynamic Traces for 100 kph and 105 kph DLC no ESC.

4.4 Indicating Oversteer

This section will outline how the oversteer-indicating fuzzy logic controller reacts to the vehicle response. As described in Chapter 2, the signals for steering and lateral acceleration are lowpass filtered at 3.5 Hz and again at 0.5 Hz. The differences in these filtered signals represent the imbalance in the vehicle. Yaw rate is the other parameter used and will be plotted as well for both the 100 and 105 kph cases. When both the difference in lateral acceleration and difference in steering are large, or the yaw rate is ”high,” the ESC will command braking to the correct wheel and balance the yaw moment. The signals that the fuzzy logic structure act on can be seen in Figure 4.8.

4.5 Evaluation of Rules for Time Step A

For the input signals as seen in Figure 4.8, an output of the level of oversteer will be calculated. To illustrate this, the six fuzzy logic rules discussed in the previous chapter will be evaluated at time step A seen in Figure 4.8 for the 105 kph DLC. These values are summarized in Table 4.2 for both time steps A and B.

First, the three signals at time step A are imported into the MATLAB fuzzy controller. These signals fit into the programed workspace as illustrated in Figure 4.9 which breaks the range of a signal down into individual membership functions to show a low, medium, and high range of values for each variables. Figure 4.9 shows at time step A, the difference in steering wheel angle is 2.18 deg and fits into the small membership function but not the medium or high membership function. The difference in lateral acceleration (0.133 G) fits into both the small and medium membership functions, but not the large. Absolute yaw rate (8.3 deg/s) fits into its small and medium membership function but not the large. Each specific rule takes a signal or combination of signals and examines how the signal fits into certain membership functions to determine the output for that specific rule.

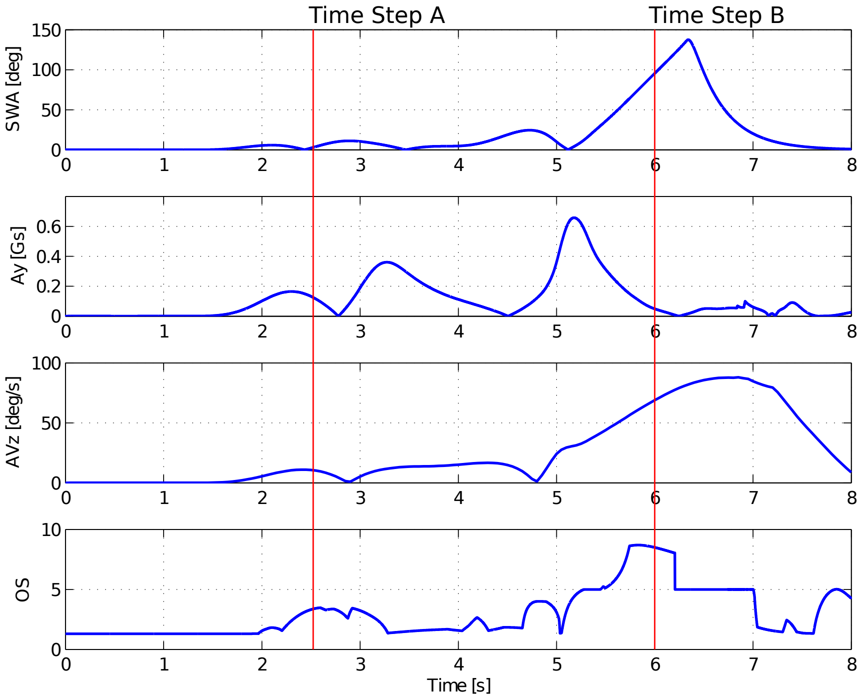


Figure 4.8: Inputs and Output of Fuzzy Logic Oversteer Indicator 105 kph DLC No ESC. Traces are Absolute Difference in Steering Wheel Angle (SWA), Absolute Difference in Lateral Acceleration (Ay), and Absolute Yaw Rate (AVz).

Variable Name	Time Step A $t = 2.5s$	Time Step B $t = 6s$	Units
Steering Wheel Angle Difference	2.18	50	deg
Lateral Acceleration Difference	0.133	0.05	G's
Yaw Rate	8.3	45	deg/s
Oversteer Number	3.32	8.7	

Table 4.2: Input and Output Values for Fuzzy Logic Oversteer Indicator.

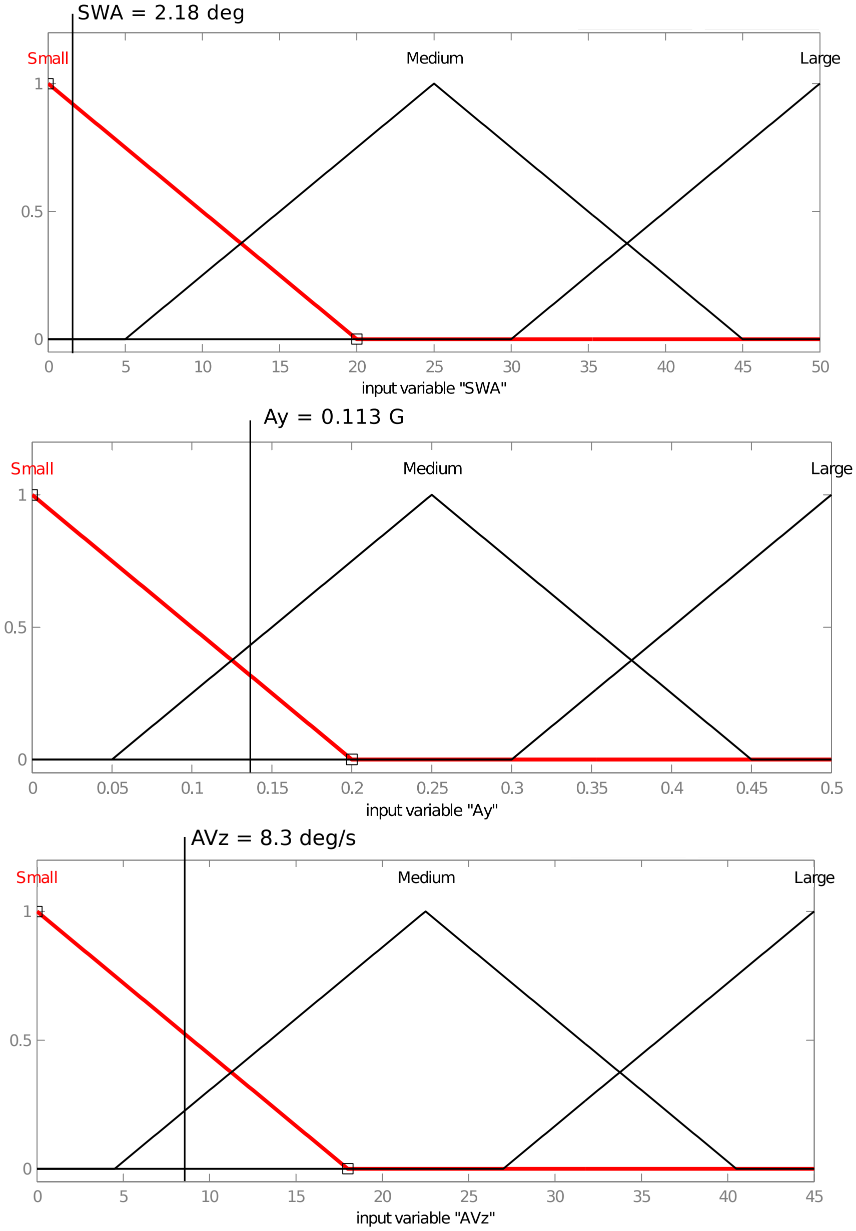


Figure 4.9: Inputs to Fuzzy Logic Control and Membership Functions for Inputs.

4.5.1 Rule 1: If SWA is Small *and* Ay is Small *then* Oversteer is No Oversteer

This rule looks at how both the differences in lateral acceleration and steering wheel angle fit into their small membership functions and is illustrated in Figure 4.10. Steering wheel angle difference (2.18 deg) produces a very good fit into its small membership function (0.9). On the other hand, lateral acceleration difference (0.133 G) produces a lower fit to its small membership function (0.4). Since this rule uses an *and* operator, the minimum of the two values, 0.9 and 0.4 will be used to evaluate the no oversteer membership function. The the no oversteer membership function is evaluated at 0.4 to give an area as the result of the first rule.

4.5.2 Rule 2: If SWA is Medium *and* Ay is Medium *then* Oversteer is Moderate Oversteer

Figure 4.11 illustrates this rule where medium difference in steering wheel angle and lateral acceleration will be examined. The difference in steering wheel angle (2.18 deg) is too low to produce any output in the medium membership function so it will be evaluated as 0 in the fuzzy domain. The difference in lateral acceleration (0.133 G) produces 0.4 in its medium membership function; however, since the *and* operator is used again, the minimum of 0 and 0.4 is evaluated as 0. The result is no area in the moderate oversteer membership function.

4.5.3 Rule 3: If SWA is Large *and* Ay is Large *then* Oversteer is Heavy Oversteer

In this rule (Figure 4.12), both the difference in steering wheel angle (2.18 deg) and the difference in lateral acceleration (0.133 G) are too small to produce any output for their respective large membership functions and are both evaluated as 0 in the fuzzy domain. The result of these inputs is 0 and is evaluated in the heavy oversteer membership function producing no area for this rule.

4.5.4 Rules 4-6: Yaw Rate

A recap of the final three rules rules for yaw rate is presented below and can be seen in Figures 4.13-4.15. For Rule 4, the yaw rate (8.3 deg/s) produces a 0.5 in its small membership function and is evaluated at 0.5 for the no oversteer membership function. For Rule 5, the yaw rate (8.3 deg/s) produces an output of 0.35 in its medium membership function and 0.35 is evaluated as the area for the moderate oversteer membership

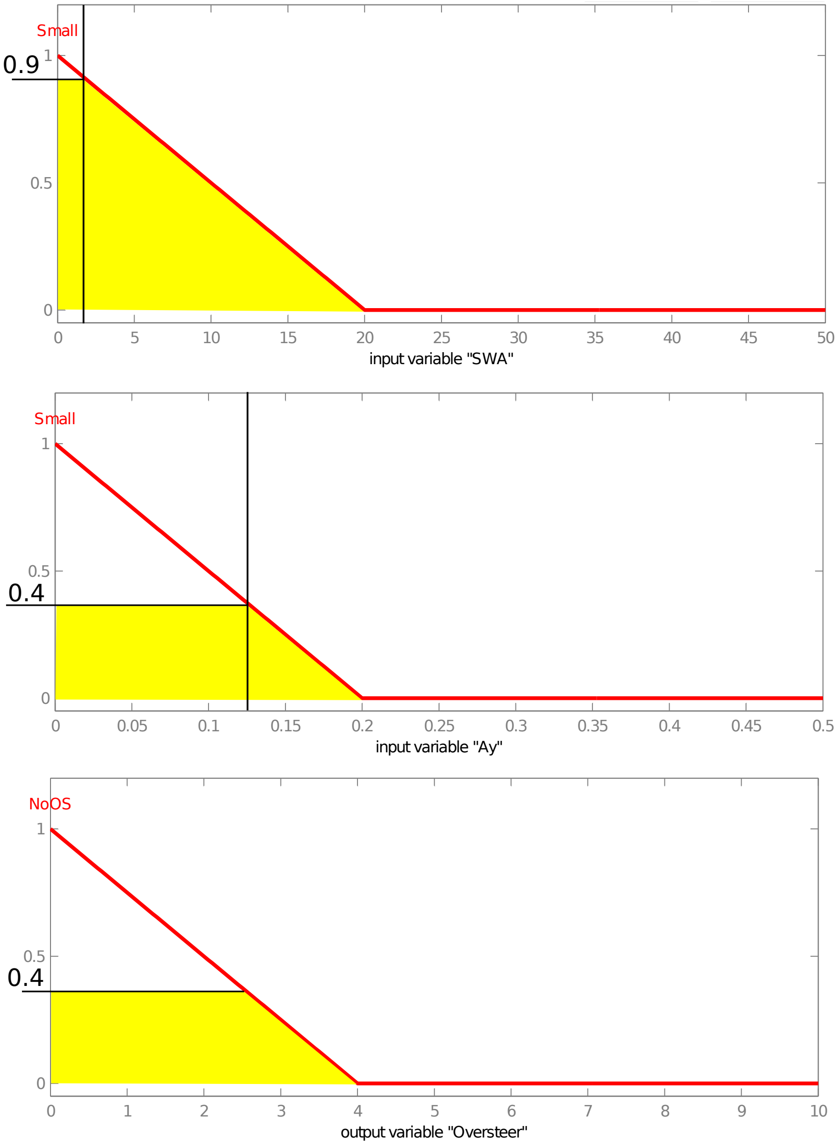


Figure 4.10: Rule 1: Low Difference in Steering Wheel Angle and Low Difference in Lateral Acceleration Evaluated for No Oversteer.

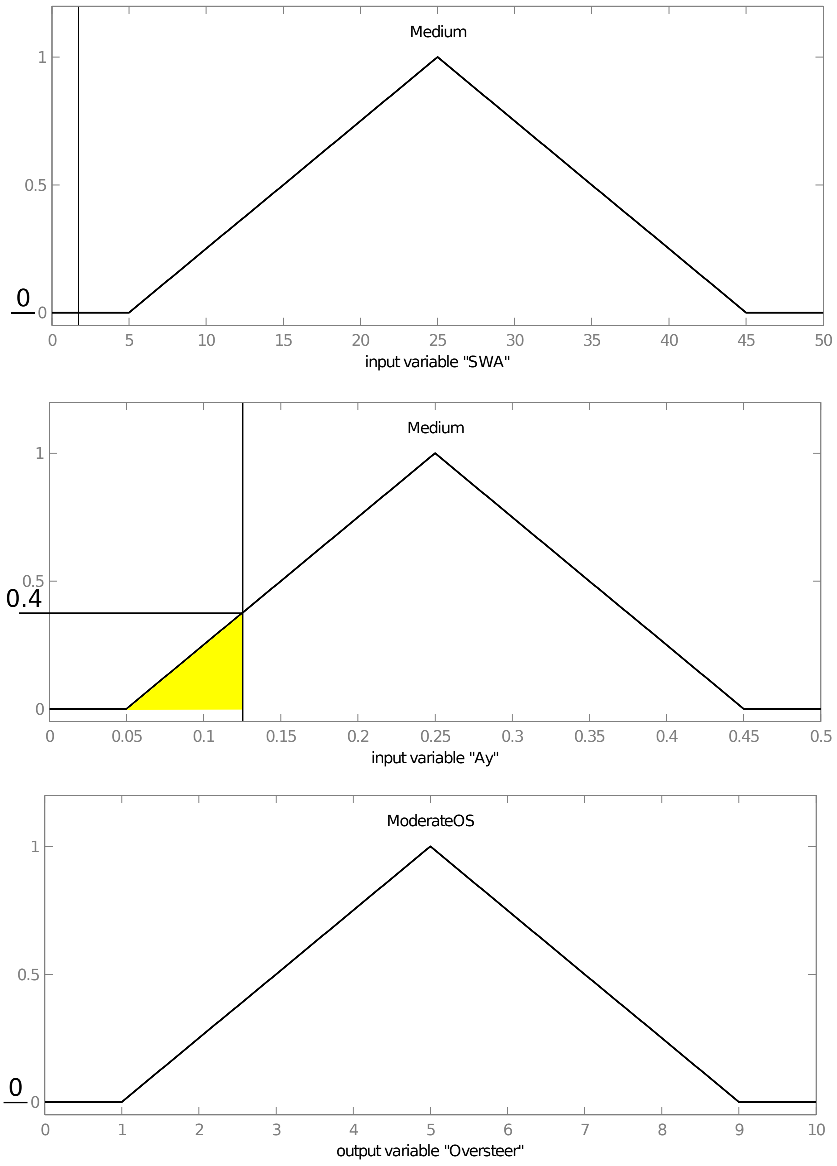


Figure 4.11: Rule 2: Medium Difference in Steering Wheel Angle and Medium Difference in Lateral Acceleration Evaluated for Medium Oversteer.

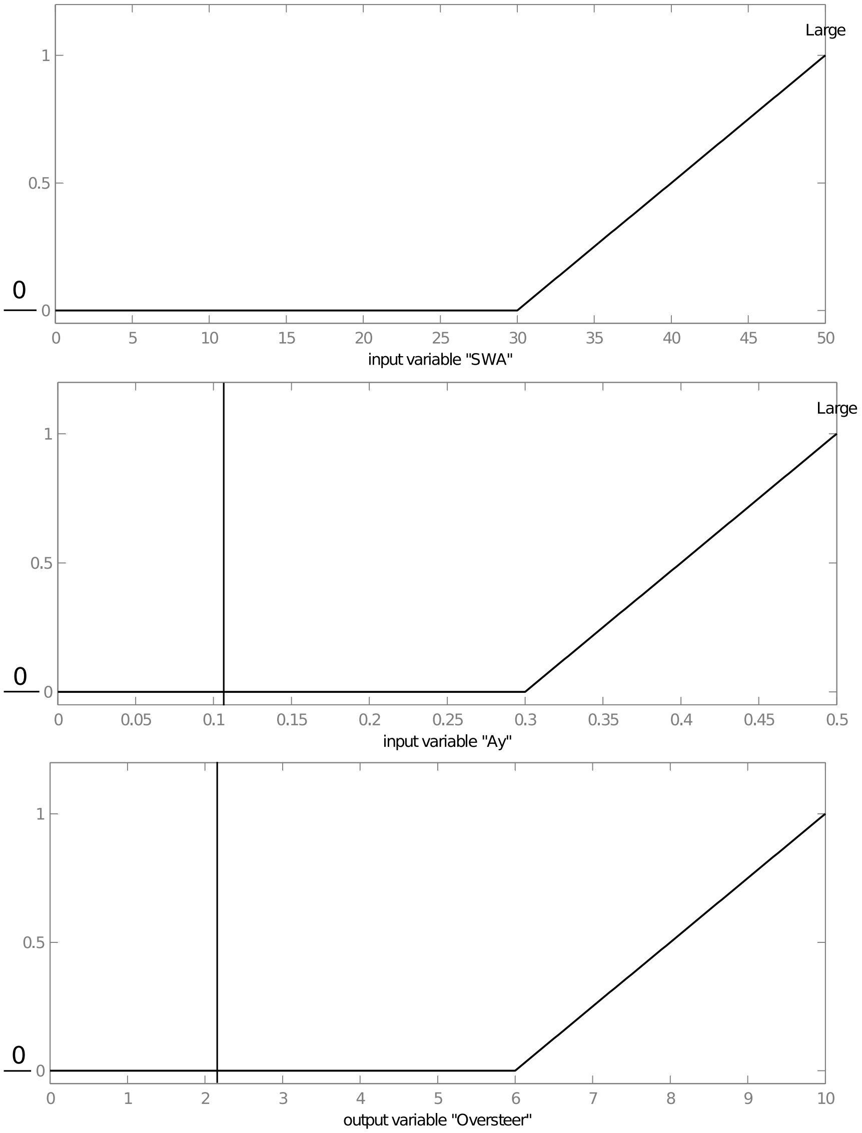


Figure 4.12: Rule 3: High Difference in Steering Wheel Angle and High Difference in Lateral Acceleration Evaluated for Heavy Oversteer.

function. Finally, Rule 6 evaluates the yaw rate (8.3 deg/s) as 0 in the large membership function and results in a 0 in the heavy oversteer membership function.

- Rule 4: If AVz is Small *then* Oversteer is No Oversteer
- Rule 5: If AVz is Medium *then* Oversteer is Moderate Oversteer
- Rule 6: If AVz is Large *then* Oversteer is Heavy Oversteer

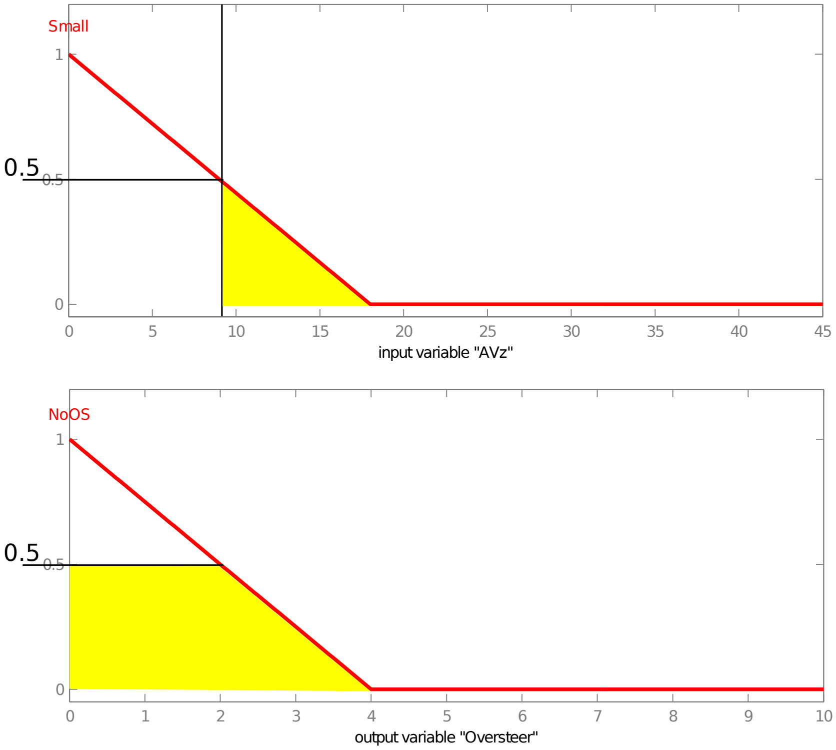


Figure 4.13: Rule 4: Low Yaw Rate Evaluated for No Oversteer.

4.5.5 Compile Result

Now that each rule has been evaluated, a single output can be calculated for time step A. The output of oversteer produced a shape for each rule in the oversteer domain. The sum of these six shapes produces a final overall shape, or membership for the oversteer at this time step. In the previous sections it can be seen that only rules 1, 4 and 5 produced any output; so, for this time step, the sum of these three shapes will be used to calculate oversteer and can be seen in Figure 4.16. From here, the centroid of this summed shape or membership function is found and the x value of the centroid is defined as the current level of oversteer.

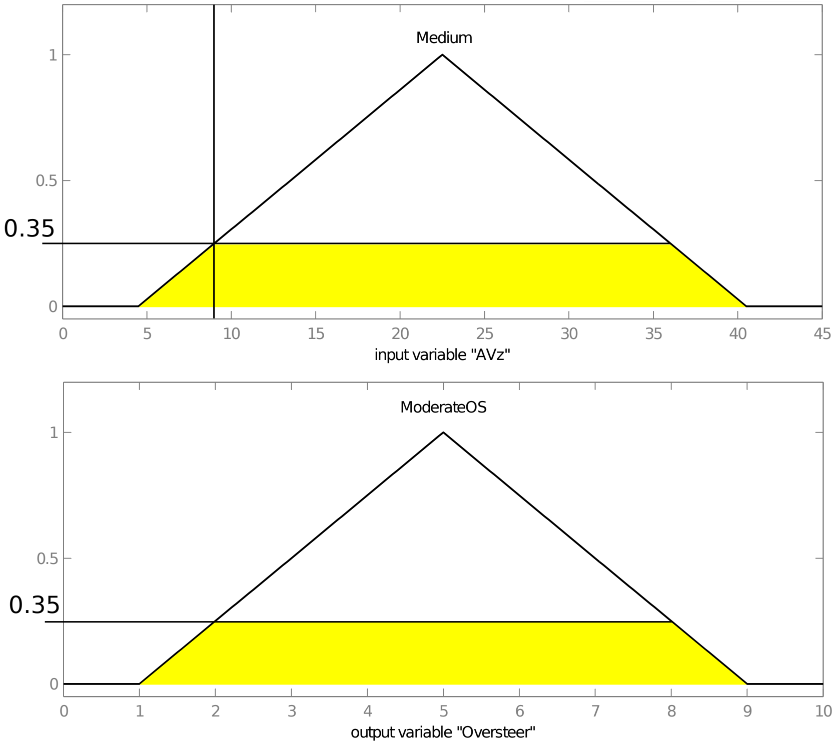


Figure 4.14: Rule 5: Medium Yaw Rate Evaluated for Moderate Oversteer.

4.6 Summary of Fuzzy Logic Oversteer Indicator

At each time step, the fuzzy logic structure evaluates the current level of oversteer as outlined above. For completeness a summary of the evaluation of level of oversteer is shown for both time steps A and B in Figure 4.17. In this figure, each of the inputs is shown in the first three columns for each of the six rules (shown as the rows). Each cell in the grid represent a portion of the rule. Each rule can contain one membership function from each of the inputs or is left blank when that input is not used. For example, Rule 1 shows the low difference in steering wheel angle input and the low difference in lateral acceleration input but, the yaw rate cell is left blank since it is unused in that rule. Rule 4 deals only with low yaw rate so, the first two cells are blank since this rule does not deal with steering or lateral acceleration. The third cell shows the low yaw rate membership function since that is used in the rule. The rules are evaluated as described in the previous section. The final column is the oversteer calculated for that rule from the inputs. The last row and last column show the final summed output of oversteer along with the resulting centroid that gives the level of oversteer for that time step.

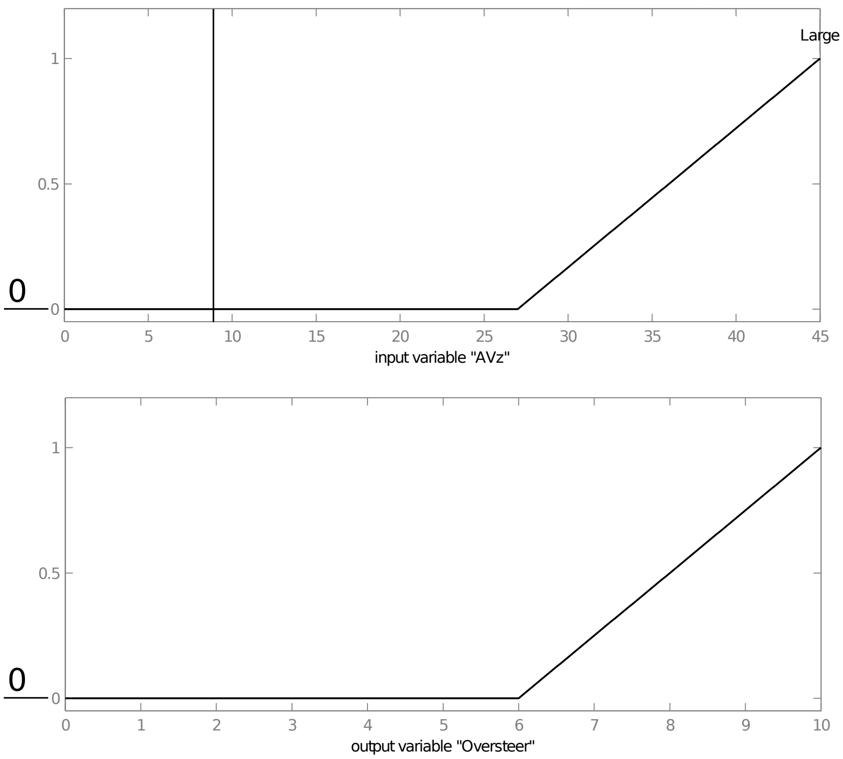
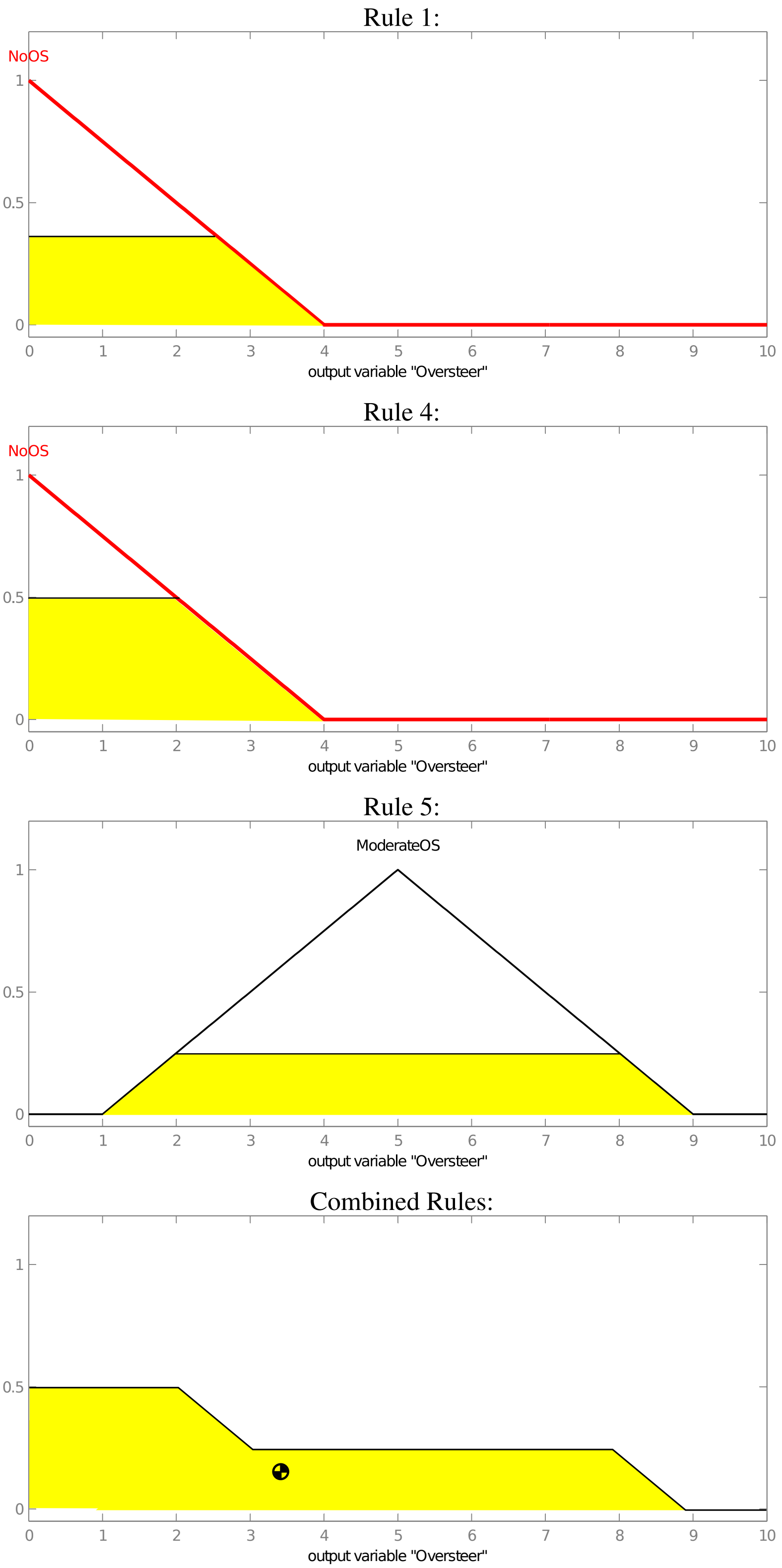


Figure 4.15: Rule 6: High Yaw Rate Evaluated for Heavy Oversteer.

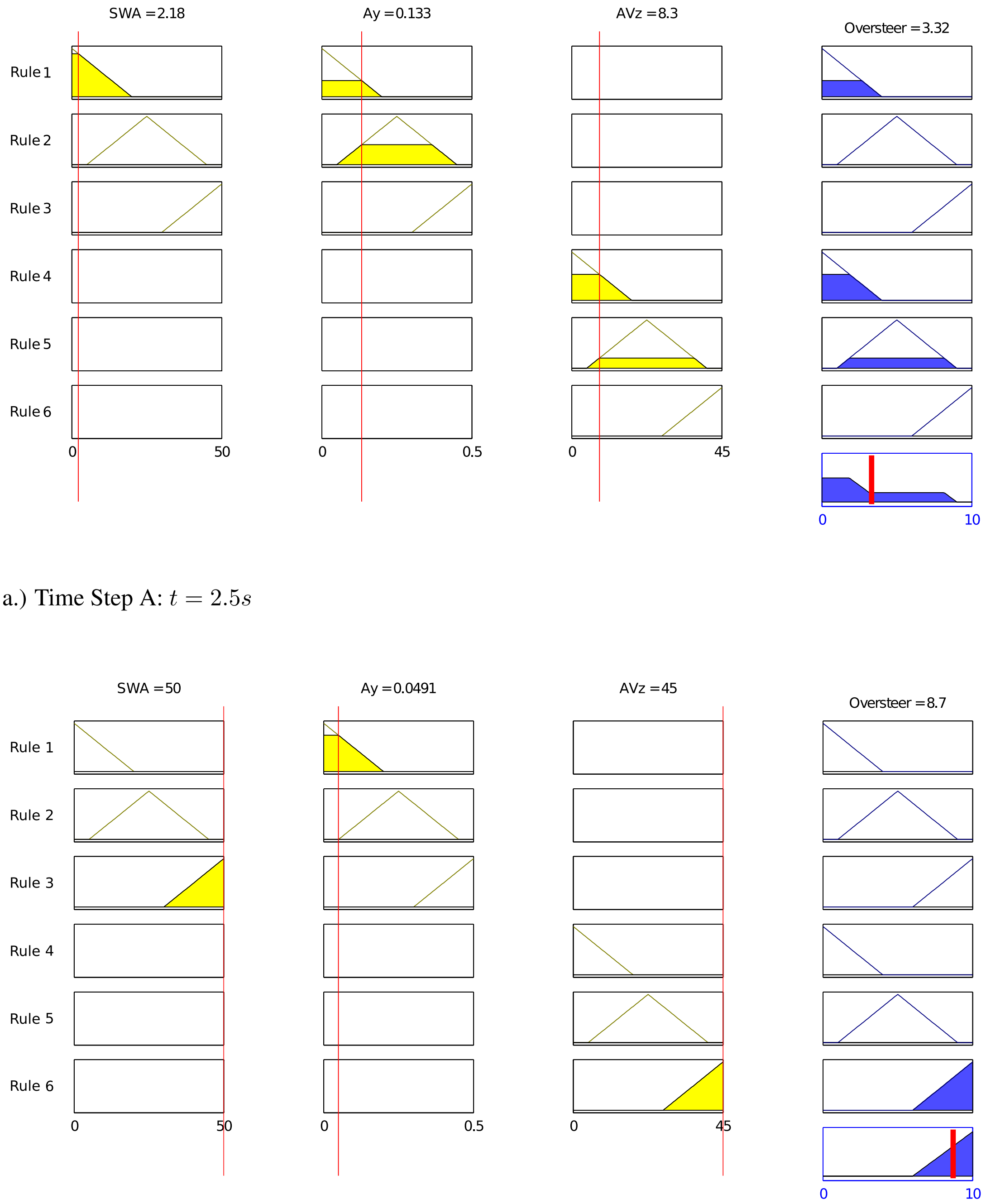

Rule 1:

Rule 4:

Rule 5:

Combined Rules:

Figure 4.16: Evaluation of Oversteer Fuzzy Logic Indicator.



a.) Time Step A: $t = 2.5s$

b.) Time Step B: $t = 6s$.

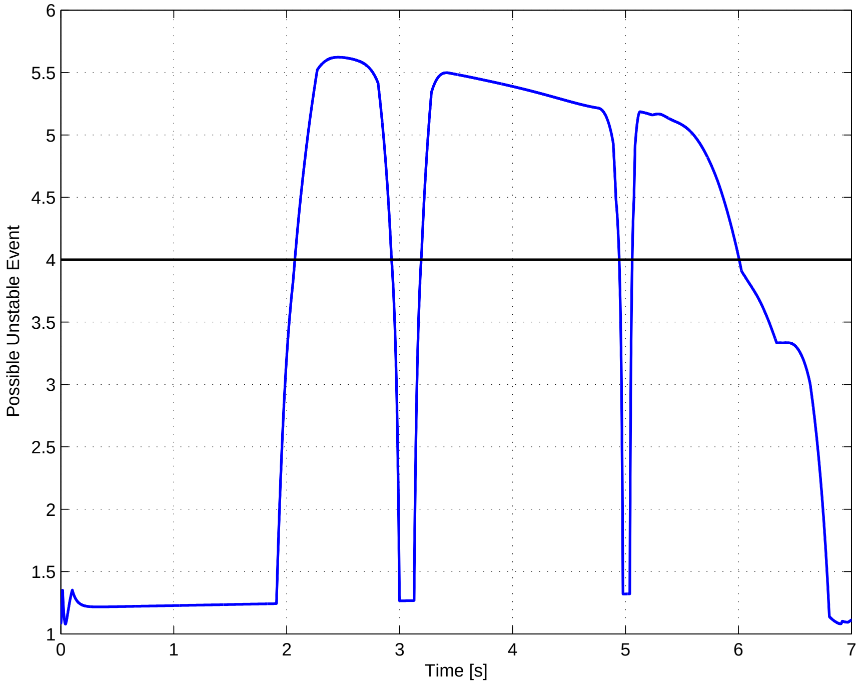
Figure 4.17: Summary of Oversteer Indicating Fuzzy Logic Structure.

4.7 Possible Unstable Event Fuzzy Logic Threshold

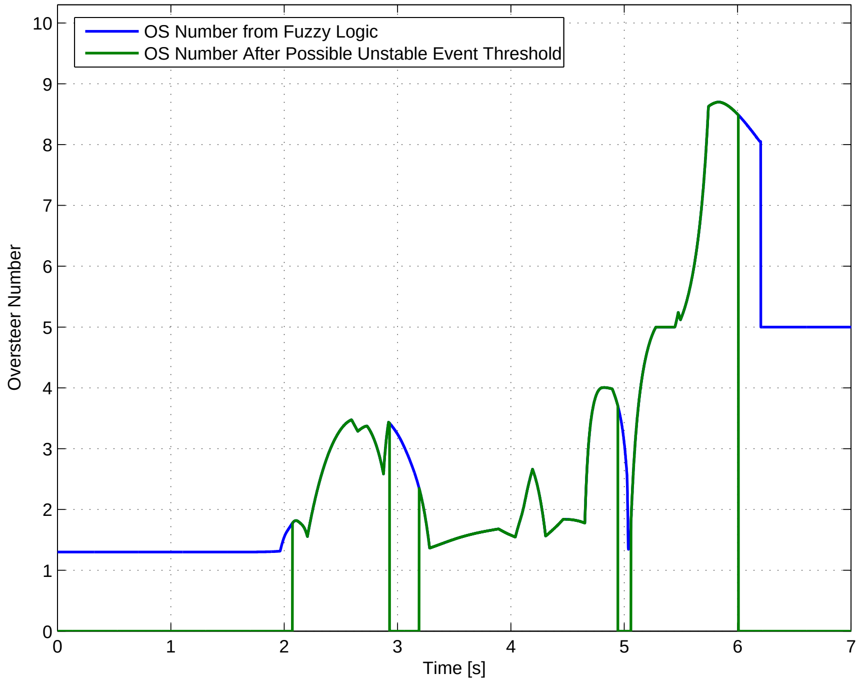
As discussed in the previous chapter an additional fuzzy logic structure is employed to determine the possibility of an unstable event. This structure cues on longitudinal velocity and lateral acceleration to determine the level of an unstable event. There are nine rules that govern this routine and are discussed in the previous chapter. The reason for this fuzzy logic structure is to eliminate any control action at low speed and non-severe maneuvers. The rules are evaluated in the same manner as in the oversteer-indicating fuzzy logic. In this research, when the possibility of an unstable event reaches a set threshold of 4 (evaluated on a 0-10 scale), the oversteer level calculated by the previous fuzzy logic structure is passed on to the ESC routine. When the possible unstable event indicator is less than 4, a 0 is passed to the rest of the ESC routine. The result of the possible unstable event fuzzy logic structure for this example can be seen in Figure 4.18a. It can be seen that in the beginning when the car is just coasting and there is no steering, the possible unstable event indicator is reading a relatively low number, 1.2, which is well below the set threshold. When the driver starts to steer the vehicle and lateral acceleration starts to rise at slightly before 2 s (seen in Figure 4.6), the possible unstable event spikes as it should and now the oversteer number calculated by the oversteer-indicating fuzzy logic structure is passed on. It should also be noted that the reason the possible unstable event spikes down at approximately 3 and 5 s is because the driver is changing direction on the course and the absolute value of lateral acceleration goes back to zero before rising again. In the next section, this problem will be accounted for. Also, it should be noted that this is the example with ESC off; therefore, after 5 s when the vehicle is spinning (seen in Figure 4.6), lateral acceleration drops off and longitudinal speed decreases dramatically resulting in a decrease in the possible unstable event number. Figure 4.18b shows how the possible unstable event indicator affects the oversteer number passed on. As discussed previously, the oversteer number is correctly changed to a 0 in the first 2 s of the DLC when there is no chance of the vehicle becoming unstable. This fuzzy logic structure also returns a 0 at the spikes at 3 and 5 s and again after 6 s. This problem will be accounted for in the next section; but, this structure is essential for preventing control action at low speeds.

4.8 Oversteer Hold Number

The final step before the level of oversteer commands the control action is that it must go through a block in Simulink which will hold the highest level of oversteer indicated for a certain amount of time. The reason for this block is that when the control action is applied, the vehicle dynamic traces that are



a.) Possible Unstable Event with Threshold.



b.) Calculated Oversteer Number Before and After Unstable Event Threshold.

Figure 4.18: Possible Unstable Event Fuzzy Logic Effect on Oversteer Number.

indicating oversteer look considerably better before the vehicle regains stability (becasue the control action is working correctly); however, this prematurely releases the brakes. By holding the highest level of oversteer for a set amount of time, the controller can apply the correct amount of control long enough to stabilize the vehicle. Also, as discussed in the precious section, when the driver changes direction, the possible unstable event indicator suddenly spikes down which returns a zero for the current oversteer level. By holding the current level of oversteer, this problem is also alleviated. Finally, when the vehicle goes into a severe spin, the possible unstable event indicator decreases because of the decreased lateral acceleration and longitudinal velocity. Again, holding the highest level of oversteer is essential in these cases to continue the control action. This block is discussed in detail in the previous chapter and its result can be seen in Figure 4.19 which illustrates how it affects the oversteer number. It is shown how holding the current level of oversteer eliminates the problems at 3 and 5 s and after 6 s but still allows for a 0 at the beginning of the run where there is no chance of an unstable event. Now, the final oversteer number can be passed and the level of control can be determined from it.

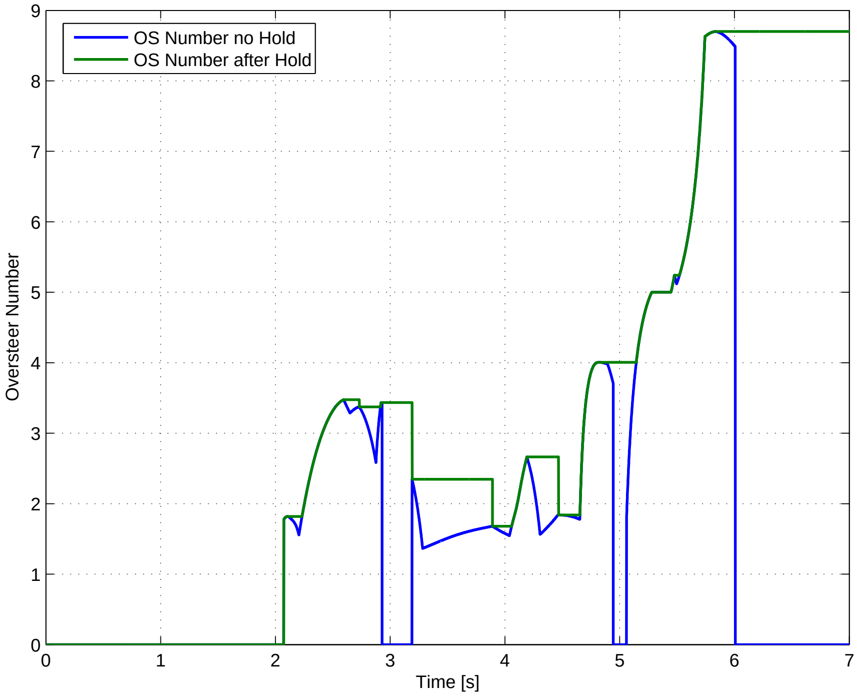


Figure 4.19: Oversteer Hold Number Evaluation.

4.9 Control Action Applied

This section will show how the controller stabilizes the vehicle by applying the control action. The trajectories for the same vehicle at 105 kph are plotted in Figure 4.20 with ESC both on and off. These show how the control action stabilizes the maneuver. The vehicle dynamic traces can be seen in Figure 4.21. The control action is derived directly from the oversteer number and is broken into ranges as seen in Figure 4.22. This is discussed in the previous chapter where no correction is no braking, moderate correction applies a braking moment proportional to the yaw acceleration, and heavy correction applies an extremely large braking moment to induce as much of a yaw moment as possible in the opposite direction of the spin. The control action is applied to either the front left or right tire as it is assumed that in a spin the rear tires are saturated and cannot aid in inducing a yaw moment. Whether the left or right tire is braked depends on which way the vehicle is spinning and is found by integrating the yaw rate to find vehicle yaw. The direction of the yaw determines which wheel to brake. A final plot of oversteer number, vehicle sideslip, and the braking moments can be seen in Figure 4.23.

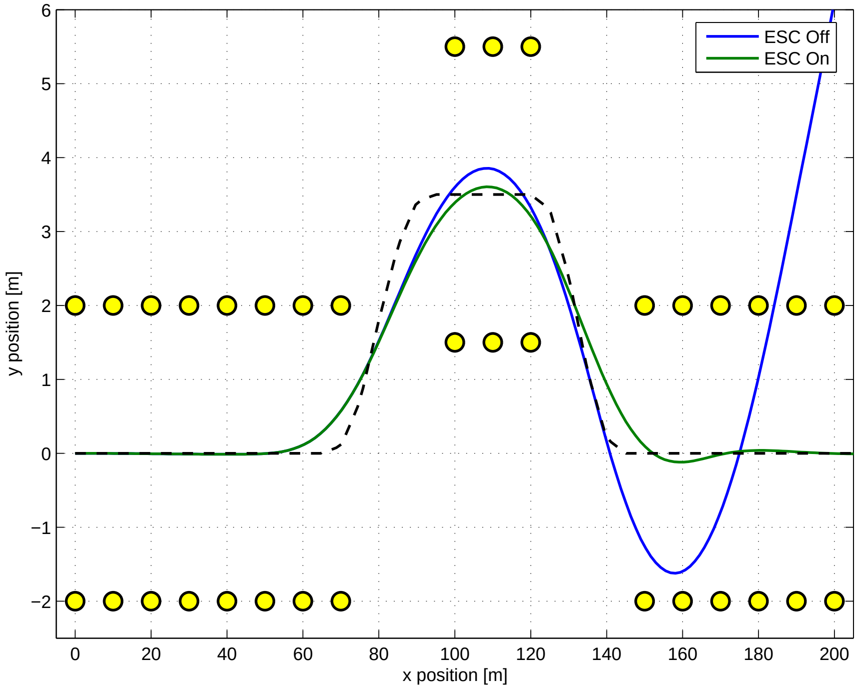


Figure 4.20: Trajectory of DLC Degraded Mini ESC On and Off (105 kph).

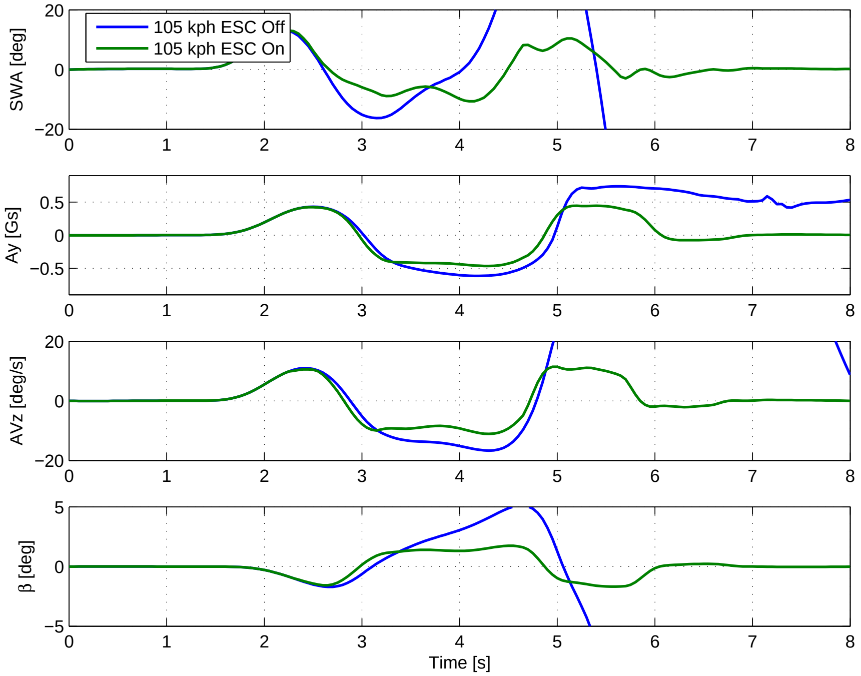


Figure 4.21: ESC On and Off Vehicle Dynamic Traces (105 kph).

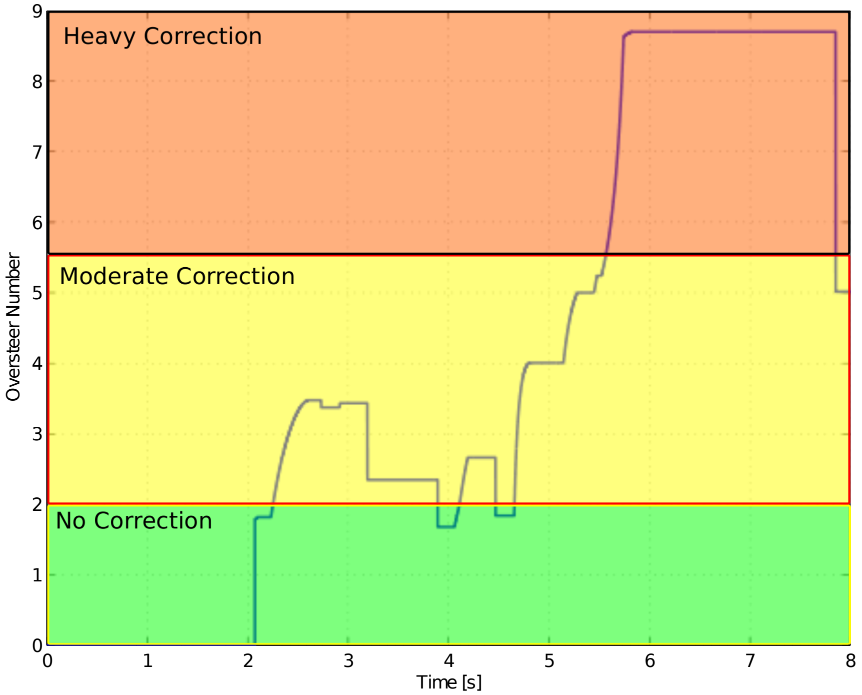


Figure 4.22: Oversteer Number for 105 kph DLC with ESC Off.

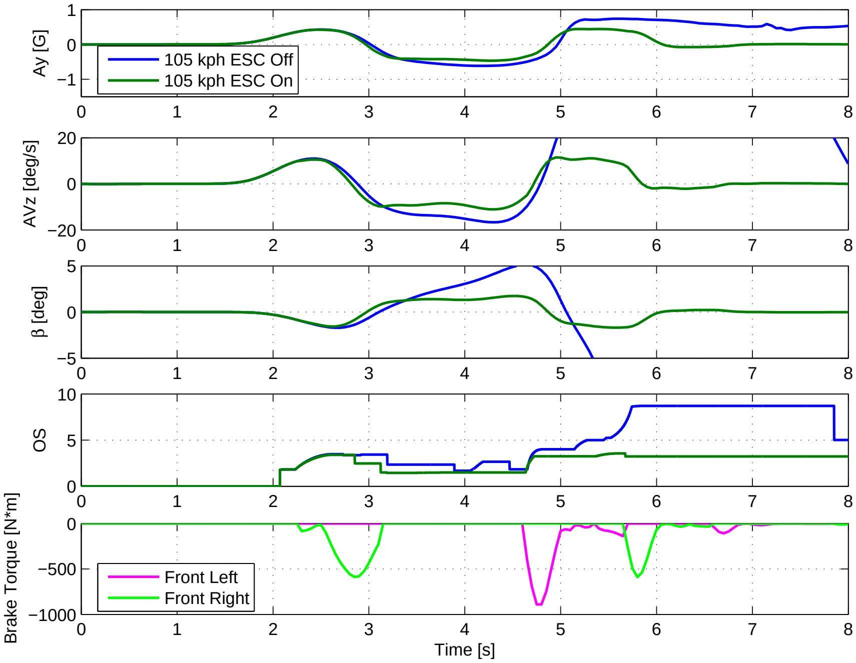


Figure 4.23: ESC On and Off Lateral Acceleration (Ay), Yaw Rate (AVz), Sideslip Angle (β), Oversteer Number, and Corrective Braking Torque.

Chapter 5

CASE STUDIES

5.1 Introduction

In this chapter, the fuzzy logic ESC strategy will be implemented in ten different case studies. These cases will encompass multiple vehicle models, configurations of the vehicle (i.e., loading and tire characteristics), driving maneuvers, and road conditions. Table 5.1 provides an overview of each case study presented in this chapter.

Case	Vehicle	Driver	Configuration	Maneuver	Tire to Road Adhesion
1	BMW Mini	CarSim	Nominal	DLC	Normal ($\mu = 0.85$)
2	BMW Mini	CarSim	Nominal	DLC	Low Mu ($\mu = 0.2$)
3	BMW Mini	CarSim	Degraded Rear Tires	DLC	Normal
4	BMW Mini	CarSim	GVW	DLC	Normal
5	BMW Mini	CarSim	Degraded Rear Tires	Fishhook	Normal
6	BMW Mini	Genta	Degraded Rear Tires	DLC	Normal
7	Sports Car	CarSim	Nominal	Understeer	Normal
8	Sedan	CarSim	Nominal	DLC	Normal
9	SUV	CarSim	Degraded Rear Tires	DLC	Normal
10	BMW Mini	CarSim	Nominal	DLC	Split Mu ($\mu = 0.2, 0.5$)

Table 5.1: Summary of Case Studies Presented

5.2 Topics Covered

5.2.1 Vehicle Models

For this research, several vehicle models were used to develop and demonstrate the performance of the stability control algorithm. The BMW Mini model developed in a joint project with BMW and Michelin was the main vehicle model used in simulations because it was validated with test data [11]. In addition to this model, several of the CarSim internal vehicle models were used (Table 5.1). These models were chosen based on their widely varying properties.

5.2.2 Driver Models

5.2.2.1 CarSim Driver Model

The primary driver model used is the internal CarSim driver model (illustrated in Figure 5.1) which is a very complex model utilizing optimal control theory [12] to calculate steer angle throughout the maneuver. For the full explanation of this controller, [12] should be consulted. The controller calculates the optimal steering action backwards from the preview horizon to the current time. In the studies presented, the default preview time of one second is used. In general, the controller uses eight feedback variables: the x and y coordinate of the front axle (X_v, Y_v), the x and y component of vehicle velocity (V_x, V_y), yaw rate ($\dot{\psi}$), yaw angle (ψ), and front and rear steer control (for factors such as suspension kinematics, u_{fo}, u_{ro}).

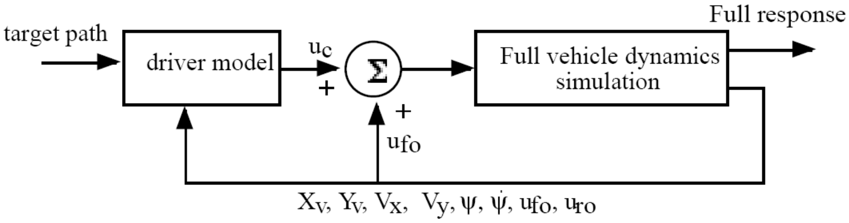

target path
driver model
u_c
+
Σ
+
u_{fo}
Full vehicle dynamics simulation
Full response
X_V, Y_V, V_X, V_y, ψ, $\dot{\psi}$, u_{fo}, u_{ro}

Figure 5.1: CarSim Driver Model as Seen in [12].

5.2.2.2 Genta Driver Model

In addition to the CarSim driver model, the driver model proposed by Genta was used for Case 6 [13]. This model is a simple path-following model which includes a driver lag time, preview distance, and driver gain. The driver's goal is to follow the path seen in Figure 5.2. He has an error of Δy from point #1 at the current time. He is looking ahead to point #2 at the look ahead distance (l) and wants to be on the path at point #2. If he remains on his current heading (ψ), he will have a lane position error of d at the time he arrives at point #2. If he is on course at point #2, he will have a heading angle, ψ_2, which is tangent to the desired path at that point. The heading angle corresponding to a line between points # 1 and # 2 is ψ_o. It is assumed that the driver will want to correct his heading by the amount $\psi_o - \psi$ at the current time instead of by the full amount $\psi_2 - \psi$. In other words, the driver perceives a lane position error ahead (d) and must correct in order to remain on course at point # 2. The relationship between the error at point # 2 (d) and the heading angles can be seen in Equation 5.1. The driver steers in response to the perceived "error" as seen in Equation 5.2. The constants used for this driver model included a driver gain of $K = 0.2$, a driver lag of $\tau = 0.2s$ and a look ahead distance of $l = 25m$. This driver lag τ accounts for the human lag of calculating and entering the steering input. For this model, the road wheel angles are substituted directly into CarSim, thus bypassing the steering dynamics.

$$\begin{gathered} \tfrac{(\Delta y - d)}{l} = tan\left(\psi_o - \psi\right) \approx \psi_o - \psi \\ d = l\left(\psi - \psi_o\right) + \Delta y \end{gathered} \tag{5.1}$$

$$\tau \frac{d\delta_{RWA}}{dt} + \delta_{RWA} = -K\frac{d}{l} = K\left[\left(\psi_o - \psi\right) - \frac{\Delta y}{l}\right] \tag{5.2}$$

5.2.3 Configurations

Three main configurations of the vehicle were tested in these studies: the nominal state, degraded rear tires, and gross vehicle weight (GVW). The nominal configuration is defined as curb weight plus driver and original equipment tires. The car with degraded rear tires simulates a worst case scenario with regard to propensity for a spin-out. For all degraded rear tire cases, the lateral handling capacity of the tire is reduced by 30 % as seen in Figure 4.1. Gross vehicle weight is defined as the vehicle with the maximum number of passengers and cargo.

5.2.4 Maneuvers

The research used three different maneuvers to test the ESC algorithm. The double lane change maneuver (DLC) seen in Figure 4.2 was the primary test used in this research because it is a very common,

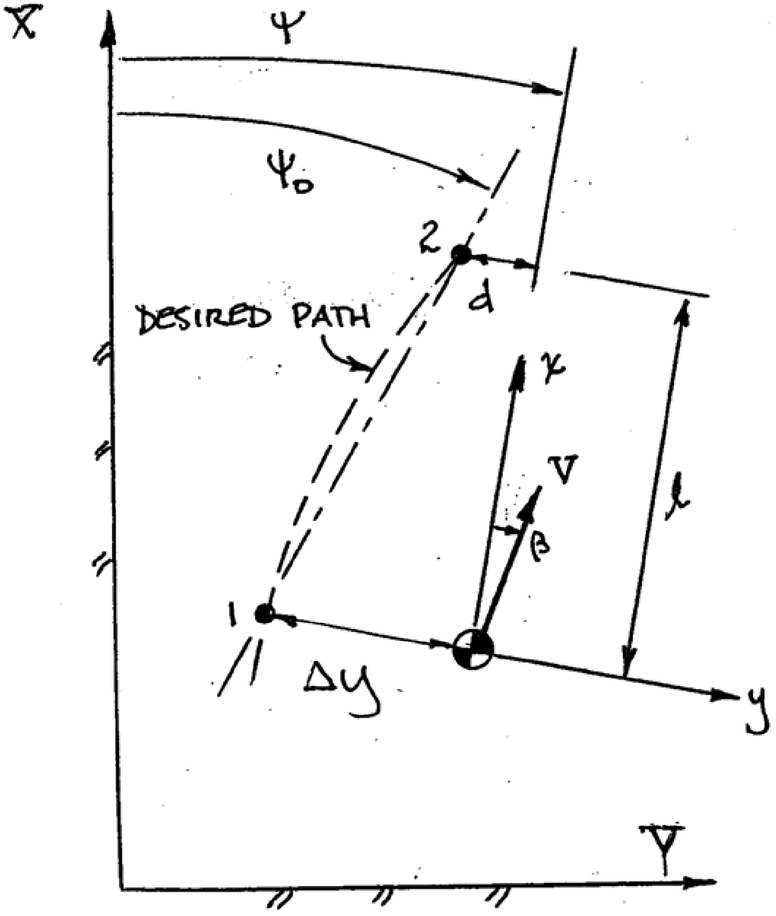


Figure 5.2: Genta Driver Model as Seen in [13].

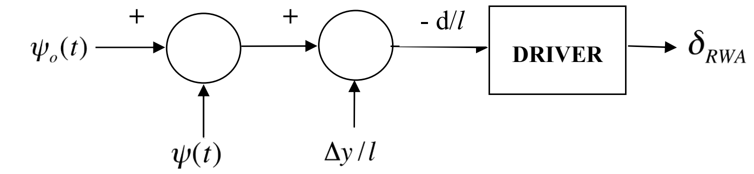


Figure 5.3: Genta Driver Model Block Diagram as Seen in [13].

real-world driving scenario. The understeer test defined in CarSim was also used. This is where a vehicle will increase speed around a 500 ft radius path until it experiences oversteer or understeer. Finally, a standard fishhook maneuver (available in CarSim) was used where the driver enters the steering input seen in Figure 5.4. This is a very harsh maneuver and is generally used with a robotic steering controller to test vehicle rollover but will also test for a spin-out in a heavily oversteering vehicle such as the cases with degraded rear tires.

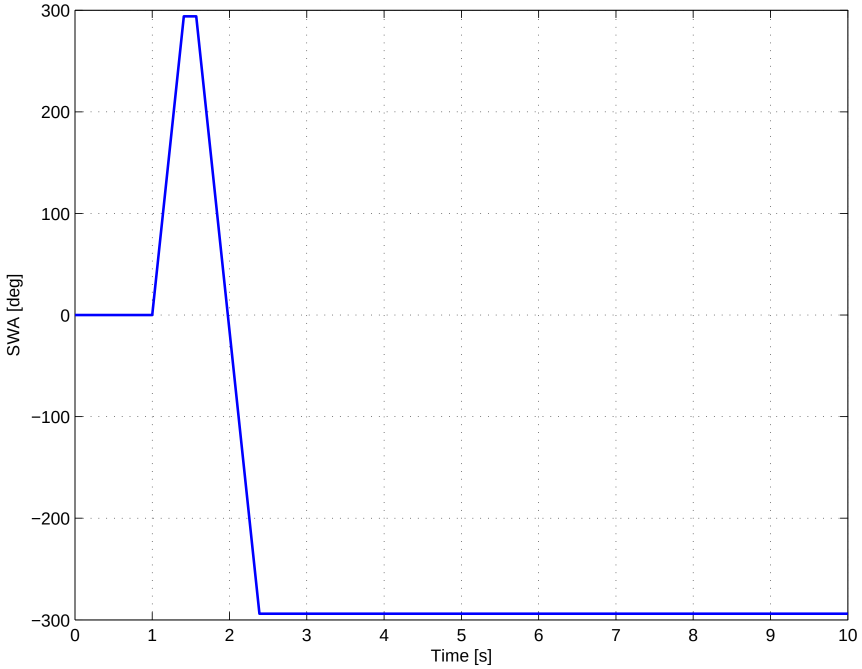


Figure 5.4: Steering Input for Fishhook Maneuver.

5.2.5 Conditions

Three conditions were tested in this research. The normal case is defined with the coefficient of friction between the tire and the road at the default value of $\mu = 0.85$. For the low mu case, $\mu = 0.2$ to simulate very slick, almost icy conditions. Lastly, a split mu test was performed where the right half of the DLC course is at a relatively high coefficient of friction ($\mu = 0.5$), and the left half of the course has a low coefficient of friction ($\mu = 0.2$). This is a very common stability control test and is used to simulate hitting a patch of ice on one side of the vehicle.

5.3 Results

5.3.1 Case 1: Nominal BMW Mini, Double Lane Change

This study consisted of the BMW Mini in its nominal configuration completing the double lane change maneuver at 185 kph. It should be noted that this is a very high speed and would prove very difficult at best and probably impossible for a real driver to traverse the DLC at this speed; however, the CarSim driver model is able to barely steer the vehicle through the DLC at this speed and if there is any increase in speed, the driver model would not be able to maintain control. Both the CarSim ESC and fuzzy logic ESC were examined to compare the fuzzy logic approach to a ”traditional” ESC algorithm that incorporates a two degree-of-freedom estimator.

The CarSim ESC strategy can be seen in Figure 5.5. It utilizes the two degree-of-freedom bicycle model to estimate the yaw rate of the vehicle based on longitudinal speed and steering wheel angle. It then compares the estimated yaw rate of the vehicle to its actual yaw rate. This error in yaw rate is then multiplied by a proportional gain (the default value is 37). When this new value exceeds a threshold (implemented in the form of a DeadZone block in Simulink which will pass a zero until the limits are reached), the control braking pressure is then passed. It uses a switch with threshold set at zero to determine which side of the vehicle to brake. If the yaw rate is negative (counterclockwise when viewed from above), it will be multiplied by the bottom set of numbers ([0 -1 0 -0.6]) to apply braking pressure at 100 % of the control amount to the right front wheel and at 60 % of the control amount to the right rear wheel. It finally passes through a saturation block to limit the braking pressure to 12 MPa. If the yaw rate is positive or clockwise, the brake pressure is applied to the left front and left rear wheels.

In this case study, both the CarSim ESC and the fuzzy logic ESC stabilized the vehicle. This can be seen in Figure 5.6. Both ESC algorithms reduced the vehicle sideslip angle. The CarSim ESC lowered the maximum sideslip angle more than the fuzzy logic ESC; however, the fuzzy logic ESC damps out the sideslip angle more quickly. The fuzzy logic ESC dramatically decreased steering wheel angle (SWA) and, by extension, driver workload. Moreover, when the driver is traversing the second turn in the lane change (2.1 s), the CarSim ESC required more steering than without ESC or the fuzzy ESC. Lateral acceleration and yaw rate are similar to sideslip in the fact that the CarSim ESC produced smaller maximum values but the fuzzy logic ESC resulted in faster decays. It should be noted that the fuzzy logic ESC was tuned to work for all ten case studies while the CarSim ESC is specifically tuned for only this vehicle in this configuration. The CarSim ESC relies on accurate values of cornering stiffness for the two degree-of-freedom vehicle model

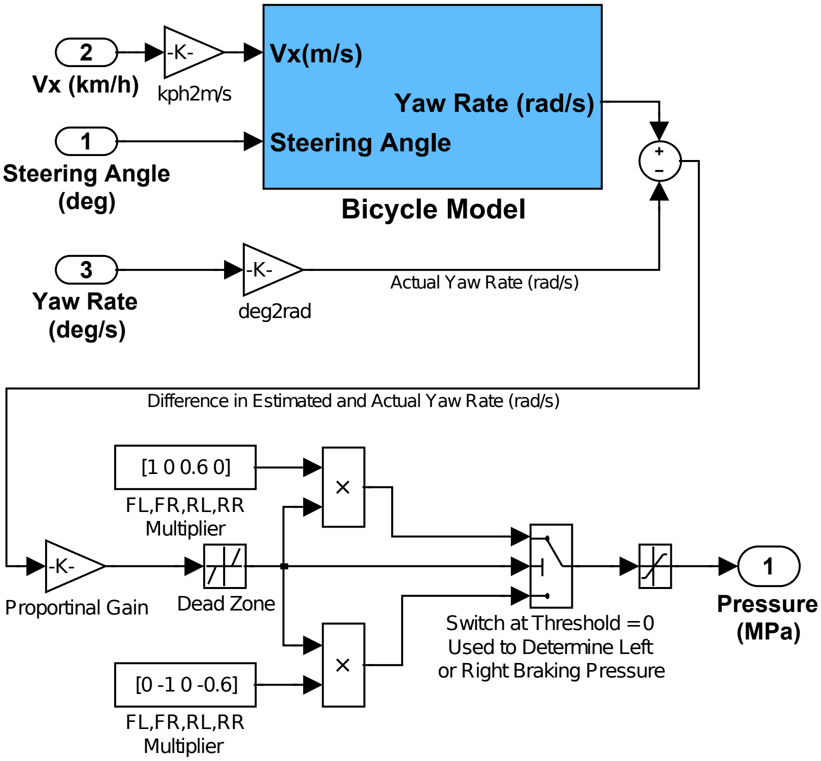


Figure 5.5: Internal CarSim ESC Strategy.

which are fed from CarSim to Simulink at the beginning of the simulation. In actual practice, it is very difficult if not impossible to obtain these cornering stiffness values in real time.

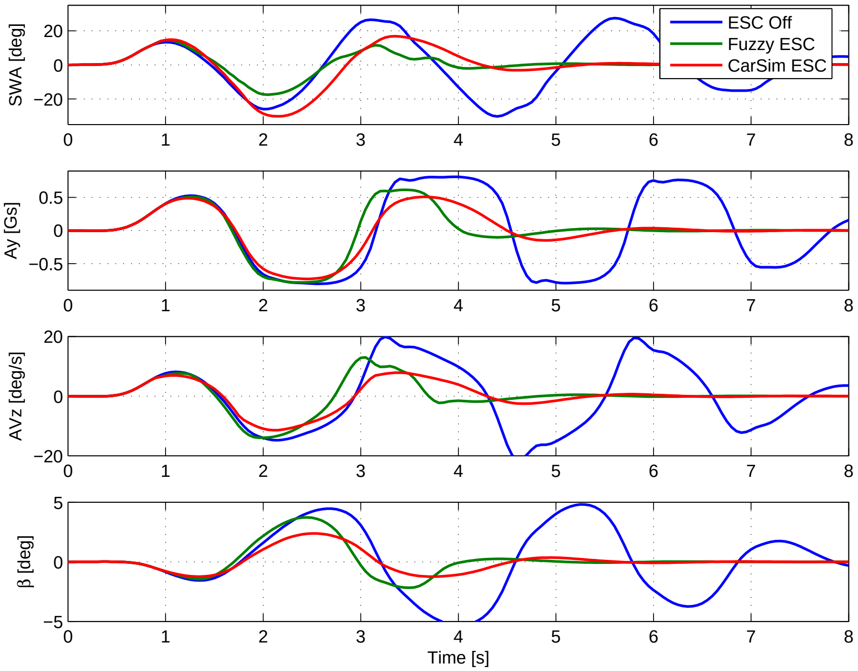


Figure 5.6: Case 01 Vehicle Dynamic Traces ($V = 185$ kph).

Examining the vehicle trajectories in Figure 5.7, it can be seen that the fuzzy logic ESC helped the vehicle track the lane change much better than without ESC or even the CarSim ESC. With no ESC or the CarSim ESC, the vehicle barely passed the lane change as it almost hit the last cone at $x = 150$m.

The vehicle yaw moment gives an indication of the amount of brake force needed to be applied to slow or stop a spin. Summing the moments about the center of gravity (Figure 5.8) from the tire longitudinal and lateral forces, Equation 5.3 can be written. This is the total yaw moment of the vehicle. The contribution of the yaw moment from the ESC algorithm can be found from the brake torque. First, the per wheel brake forces are found by dividing braking moments by the effective rolling radius of the tire. Next, the moments of these four braking forces are summed about the center of gravity which yields the yaw moment contribution from braking. It should be noted that the fuzzy ESC will only have front braking forces where the CarSim ESC will have braking forces on all four wheels.

Plotted in Figure 5.9 is the yaw moment from braking (derived from the braking torque), the total vehicle yaw moment (derived from the all of the lateral and longitudinal forces of the tires), the remainder (the difference in the total vehicle yaw moment and the contribution from braking), and the configuration

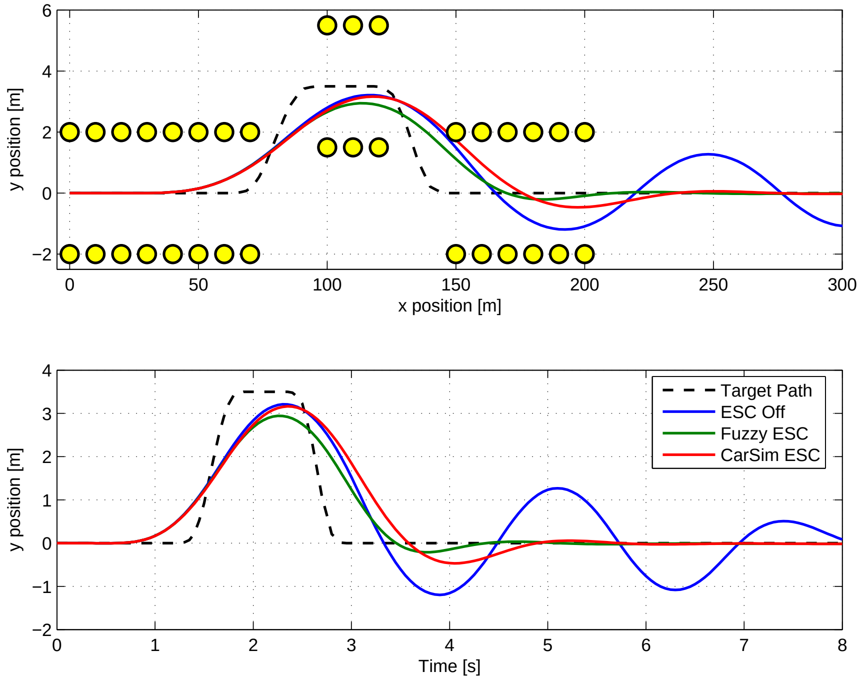


Figure 5.7: Case 01 Vehicle Trajectories ($V = 185$ kph).

with no stability control. The remainder is largely made up of the lateral forces of the tires and lowering the remainder is considered good because less lateral force generation is required from the tires. A certain amount of yaw moment is necessary for the vehicle to complete the DLC; however, it should not be excessive and should return to zero after the exit of the maneuver. From Figure 5.9, it is clear that both ESC algorithms lower the value of the yaw moment and return it to zero after the DLC (DLC course ends after 3 s from Figure 5.7).

Looking at the fuzzy ESC from 0 - 2.5 s, shows that the differential braking is helping the vehicle traverse the DLC. It produces a very similar yaw moment trace to the no ESC configuration without the extra steering effort required without ESC (steering can be seen in Figure 5.6). After 3 s, the fuzzy logic ESC is damping out the remainder of the yaw moment and returning it to zero after 4 s. The CarSim ESC does lower the maximum yaw moment compared to the no ESC configuration; however, it does not damp out until after 5 s.

$$\begin{aligned}\sum M_{CG} &= I_z\ddot{\psi} \\ &= -Fx_{FL}\left(\frac{T_f}{2}\right)+Fx_{FR}\left(\frac{T_f}{2}\right)-Fx_{RL}\left(\frac{T_r}{2}\right)+Fx_{RR}\left(\frac{Tr}{2}\right) \\ &\quad +Fy_{FL}\,(a)+Fy_{FR}\,(a)-Fy_{RL}\,(b)-Fy_{RR}\,(b) \qquad (5.3)\end{aligned}$$

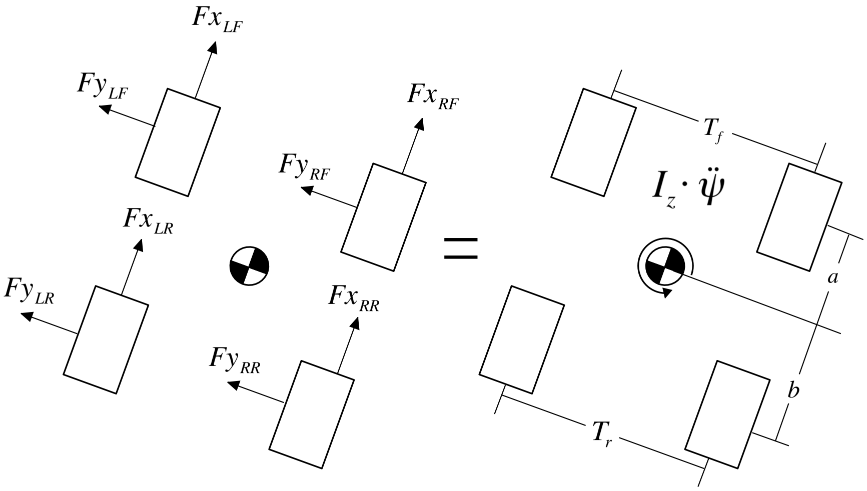


Figure 5.8: Free Body Used to Calculate Yaw Moment.

As shown in Figure 5.10, the tire forces illustrate the physics of double lane change event. The longitudinal tire forces show the corrective braking applied by each ESC strategy. The CarSim ESC applies a braking force to both the front and rear wheels with smaller values at the rear while the fuzzy ESC uses only the front tires (it is assumed that the rear tires are already saturated). The tire lateral force traces show that after completion of the lane change, both ESC algorithms return the force to zero (after 4 s). The fuzzy ESC damps out vehicle oscillation after the maneuver more quickly than the CarSim ESC. This can be seen in the lateral force traces after 4 s.

For completeness, the braking torque commanded by each controller can be seen in Figure 5.11. The fuzzy ESC requires more braking torque than the CarSim because it is only braking the front wheels as opposed to all four as in the CarSim ESC.

Power is also computed to show the consumption from each ESC strategy. This is calculated by summing moments of the braking forces (found by dividing the braking torque by the effective rolling radius of the tire) about the center of gravity. This give the yaw moment on the vehicle due to ESC braking. Next,

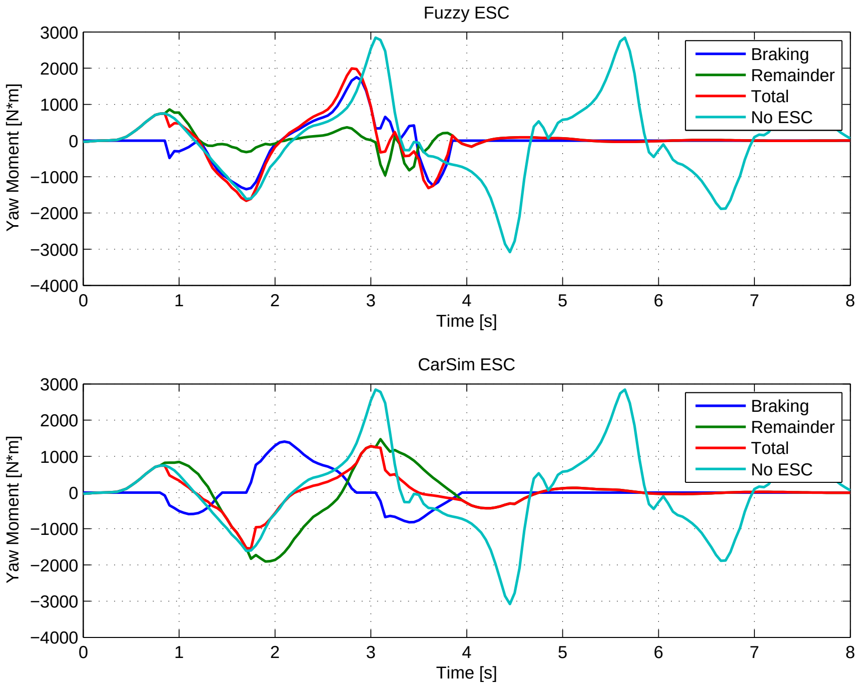


Figure 5.9: Case 01 Yaw Moment ($V = 185$ kph).

Equation 5.4 can be applied to compute power used by the controller. The power used by both ESC algorithms (Figure 5.12) is fairly close with the CarSim ESC using slightly more than the fuzzy logic approach. Integrating the curves (Equation 5.5), the total energy used during the DLC can be found. The CarSim ESC used 250.7 J (184.9 $ft \cdot lb$) while the fuzzy logic ESC used 225.8 J (166.5 $ft \cdot lb$). This give a slight advantage to the fuzzy logic algorithm.

$$Power = Moment \times \dot{\psi} \tag{5.4}$$

$$Energy = \int_{t=0}^{t=t_f} Power\,(t)\,dt \tag{5.5}$$

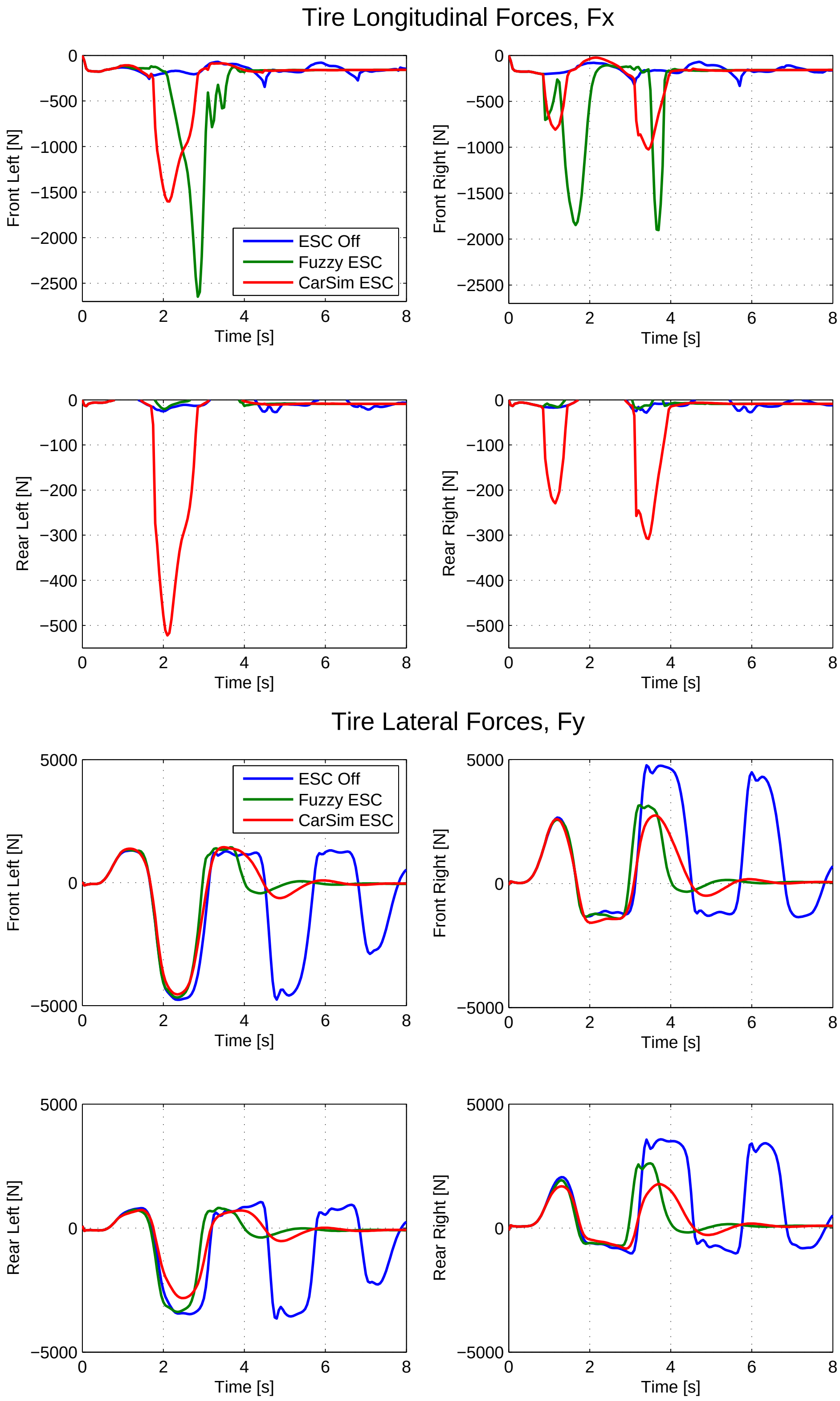


Figure 5.10: Case 01 Tire Forces ($V = 185$ kph).

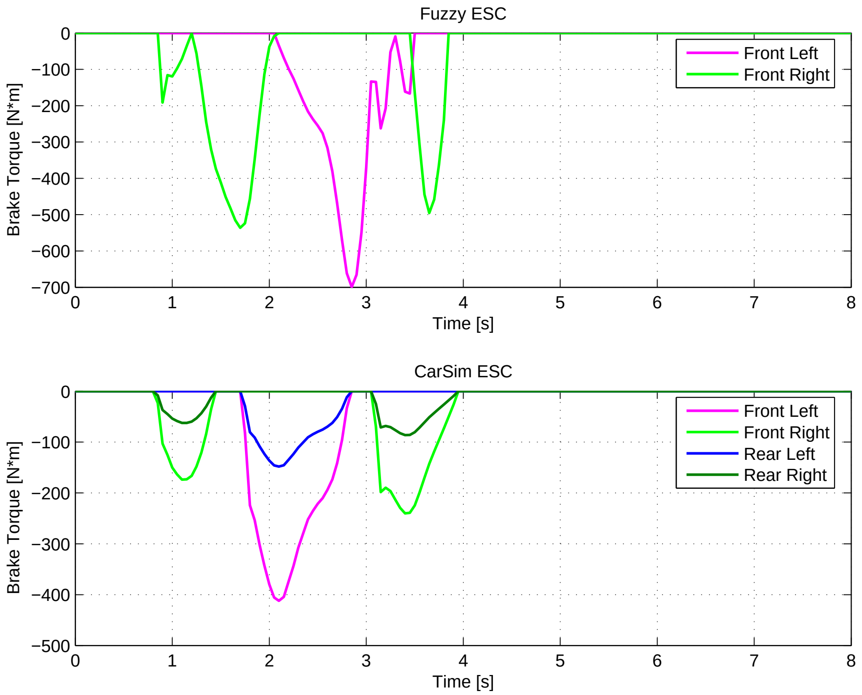


Figure 5.11: Case 01 Braking Torque from Each ESC Algorithm ($V = 185$ kph).

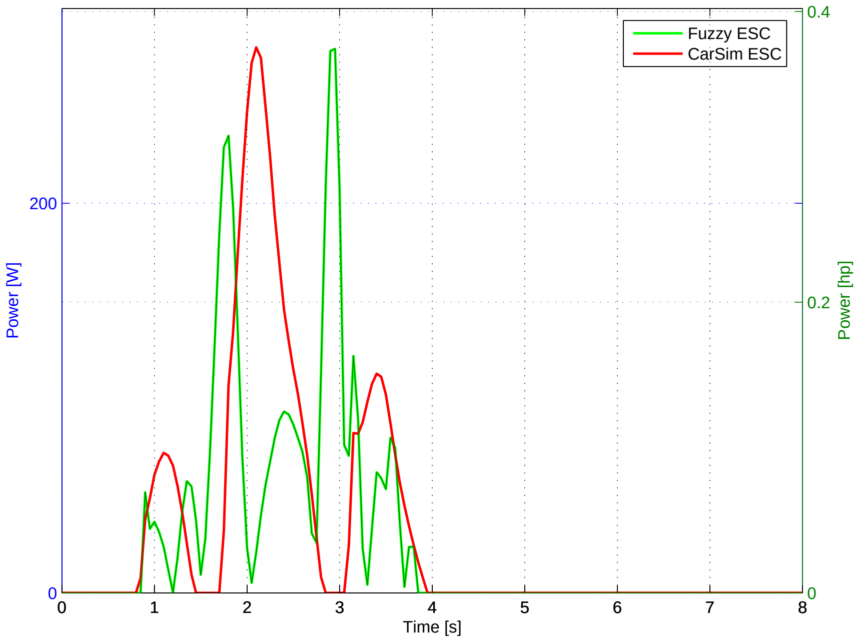


Figure 5.12: Case 01 ESC Power Used ($V = 185$ kph).

5.3.2 Case 2: Nominal BMW Mini, Double Lane Change, Low Friction Surface

This case study consisted of the nominal BMW Mini completing a lane change maneuver on a low friction ($\mu = 0.2$) surface. This will test the ESC algorithm against changing surface conditions such as a vehicle is traveling on an icy road. The vehicle trajectories with and without ESC can be seen in Figure 5.13. Both with and without ESC, the vehicle failed the maneuver because it ran through the last set of cones; however, the vehicle without ESC spun-out at the end of the test while the fuzzy ESC stabilized it. As seen in Figure 5.14, ESC dramatically reduced driver work load (i.e., steering). The vehicle used all of its lateral handling capacity (0.2 g lateral acceleration) in this run with the ESC returning the lateral acceleration back to zero. The sideslip trace shows that the case without ESC spun-out after the exit to the lane change (8 s). Figure 5.15 shows the corrective braking applied to each of the front wheels for the case with stability control. The tire forces can be seen in Figure 5.16. The longitudinal forces illustrate the corrective braking force applied to the front wheels while the lateral forces illustrate how the ESC algorithm returns them to zero after the lane change. The yaw moment plot in Figure 5.17 shows a greatly reduced magnitude while damping it to zero after the DLC.

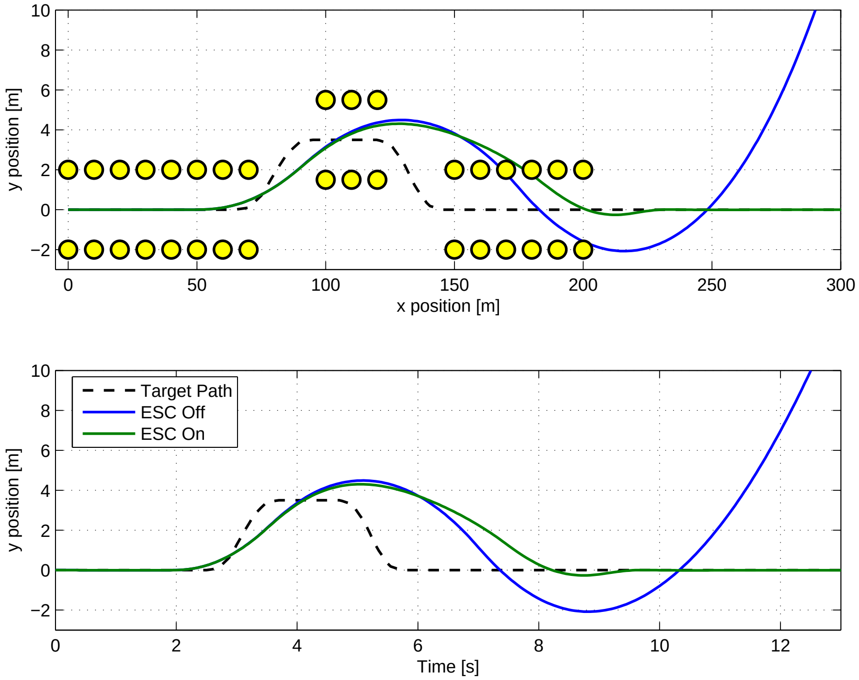


Figure 5.13: Case 02 Vehicle Trajectories ($V = 95$ kph).

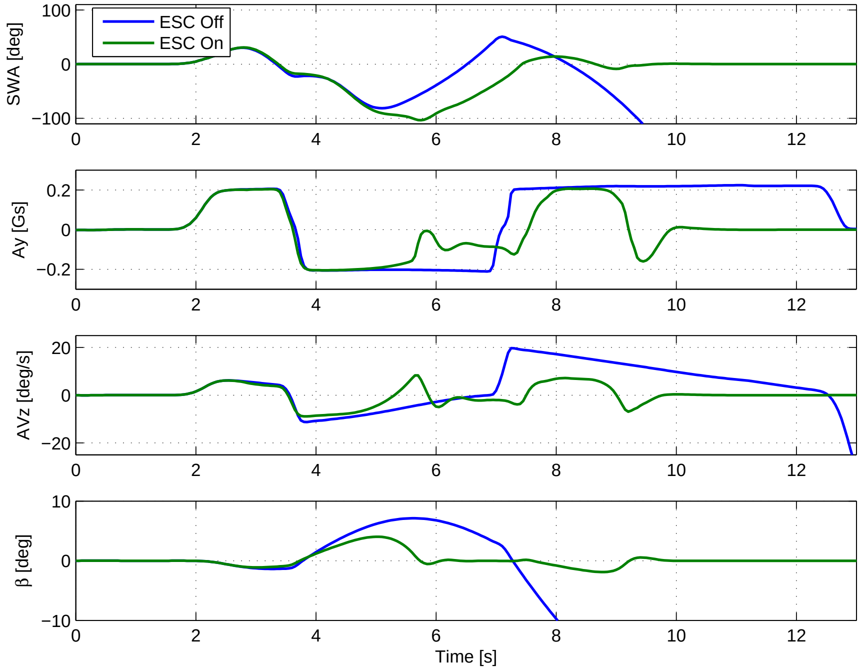


Figure 5.14: Case 02 Vehicle Dynamic Traces ($V = 95$ kph).

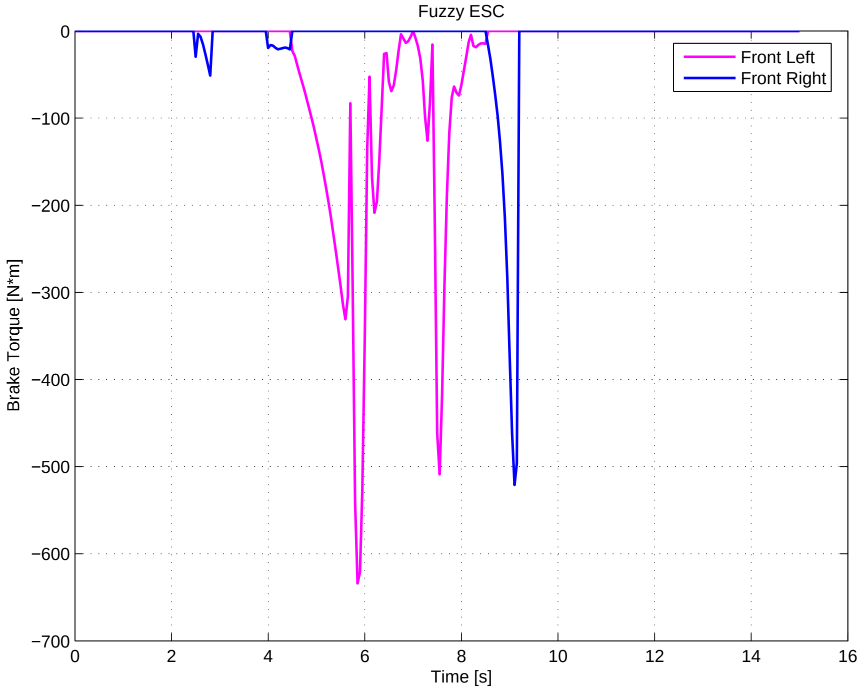


Figure 5.15: Case 02 Braking Torque from Each ESC Algorithm ($V = 95$ kph).

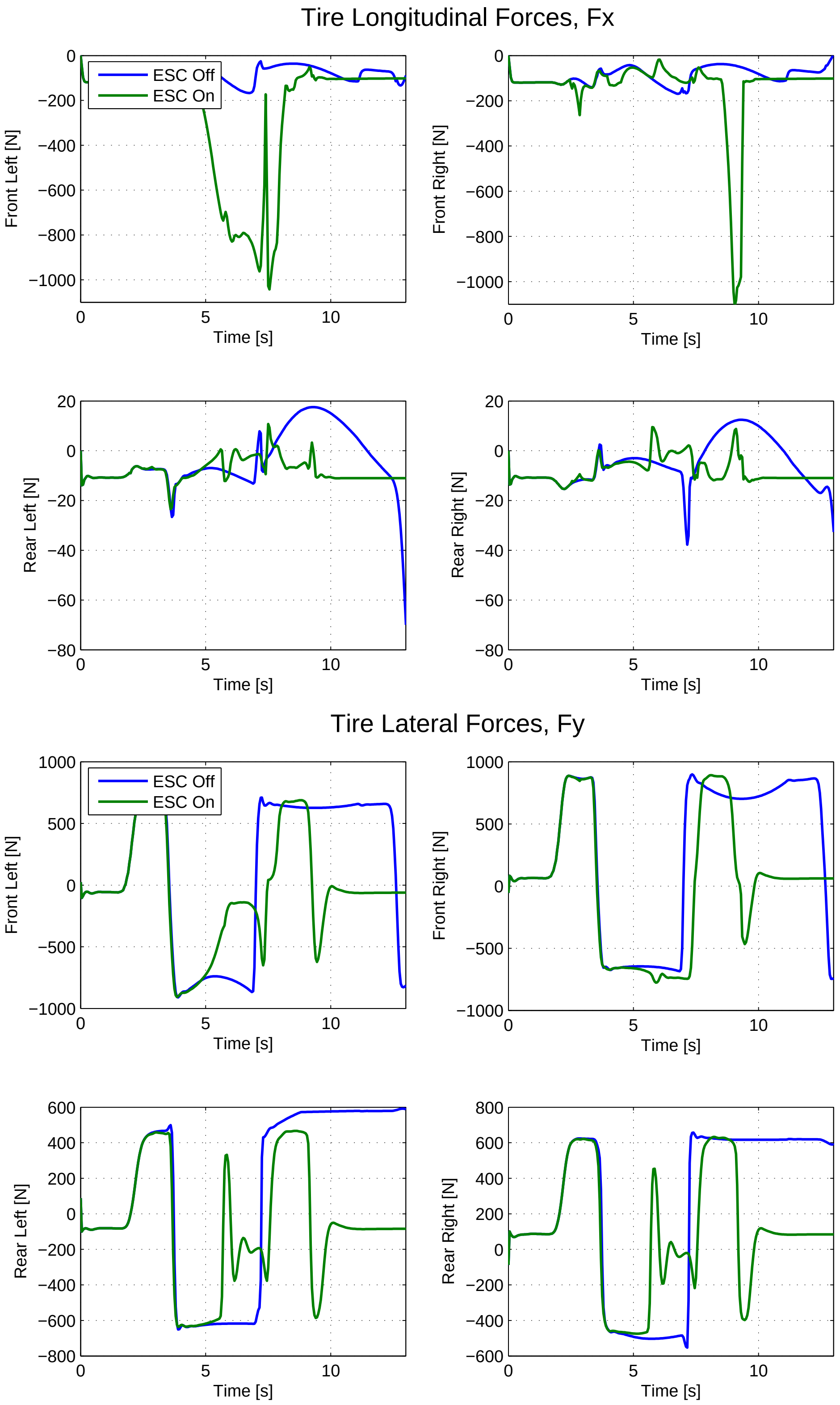


Figure 5.16: Case 02 Tire Forces ($V = 95$ kph).

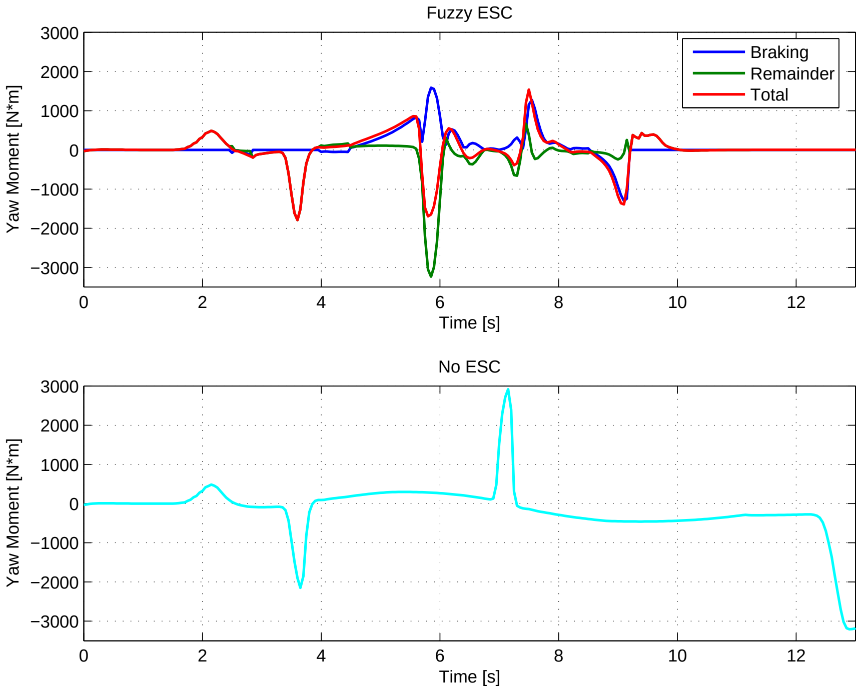


Figure 5.17: Case 02 Yaw Moment ($V = 95$ kph).

5.3.3 Case 3: BMW Mini with Degraded Rear Tires, Double Lane Change

In this study, the BMW Mini negotiated a DLC with degraded rear tires. Recall that the degraded rear tire case corresponds to a 30 % degraded lateral capacity curve as seen in Figure 4.1. The tire to road adhesion is set at $\mu = 0.85$. This represents a worst case scenario for a driver because the back tires do not have as much lateral grip as the fronts. In this case, the vehicle with ESC was able to successfully negotiate the DLC at 105 kph while the case without ESC spun-out at the end of the test much like in Case 02. Figure 5.18 shows the vehicle trajectories while Figure 5.19 shows the vehicle dynamic traces. After 5 s, the sideslip angle for the car without ESC diverged past 10 deg and the driver commanded a large steering wheel angle (SWA) in an attempt to control the car. ESC stabilized the vehicle and reduced SWA, yaw rate, and lateral acceleration after the exit of the DLC. Figure 5.20 shows the corrective braking torque applied by the ESC algorithm. The tire force traces are shown in Figure 5.21. The corrective braking forces from the ESC algorithm can be seen in the longitudinal force traces and the lateral forces were brought back to zero at the end because the car is heading straight again. As seen in Figure 5.22, the ESC system acts to reduce the yaw moment acting on the car to a level necessary to negotiate the DLC.

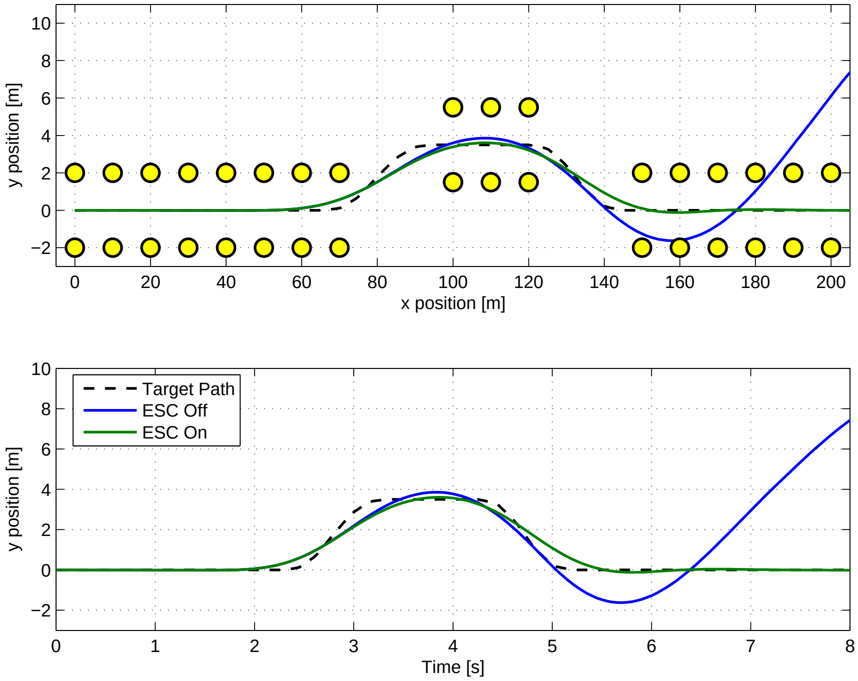


Figure 5.18: Case 03 Vehicle Trajectories ($V = 105$ kph).

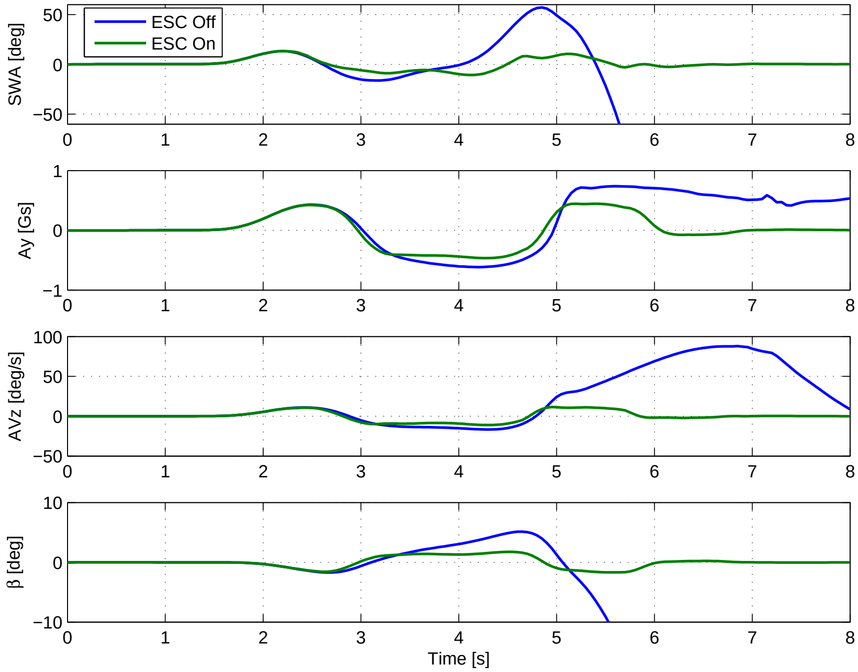


Figure 5.19: Case 03 Vehicle Dynamic Traces ($V = 105$ kph).

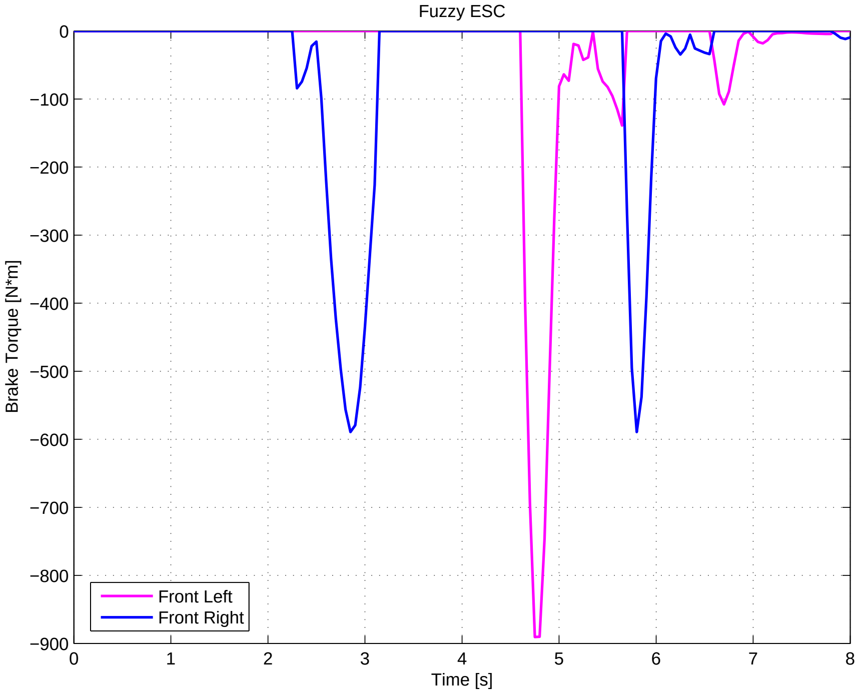


Figure 5.20: Case 03 Braking Torque from Each ESC Algorithm ($V = 105$ kph).

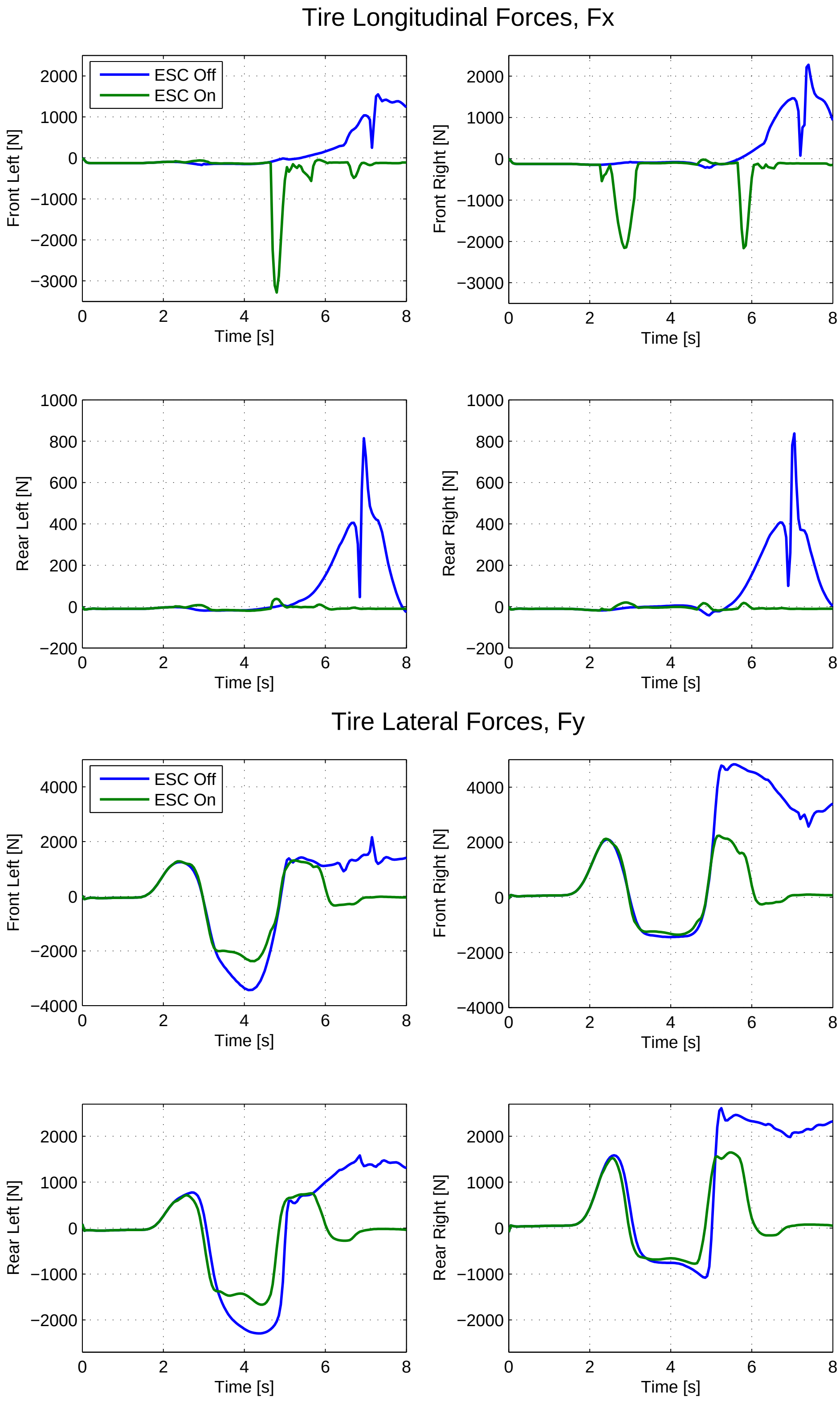


Figure 5.21: Case 03 Tire Forces ($V = 105$ kph).

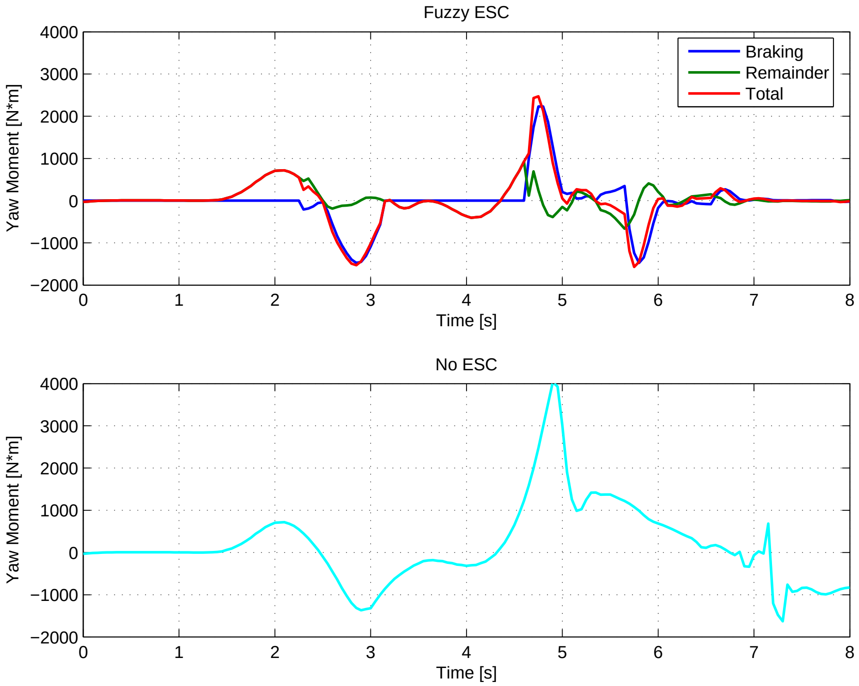


Figure 5.22: Case 03 Yaw Moment ($V = 105$ kph).

5.3.4 Case 4: Gross Vehicle Weight BMW Mini, Double Lane Change

The BMW Mini at gross vehicle weight was also evaluated in a lane change maneuver. This tests the ESC algorithm with changing vehicle loading conditions. Since the normal loads on the tires are greater, the tires have increased handling capacity and the vehicle can complete the DLC at 185 kph. Just as in Case 01, this is an extremely high rate of speed and unrealistic for real drivers; however, it does illustrate the stabilizing effects of the ESC algorithm. The cars with ESC on and with ESC off successfully negotiated the lane change. However, the car without ESC was extremely close to hitting the last cone at $x = 150$ m (Figure 5.23). Examining the vehicle dynamics in Figure 5.24, the car without ESC experiences a large amount of vehicle sideslip (5 deg) after 3 s (after the exit of the DLC) while the car with ESC has much smaller and more highly damped sideslip. Again, ESC reduced the driver's workload throughout the maneuver and dramatically decreased both lateral acceleration and yaw rate after the exit of the course. The corrective braking can be seen in Figure 5.25. The tire forces can be seen in Figure 5.26 where the lateral forces are returned to zero at the end of the run. The yaw moment (Figure 5.27) shows a decreased magnitude and is damped to zero after the exit of the DLC (3.5 s).

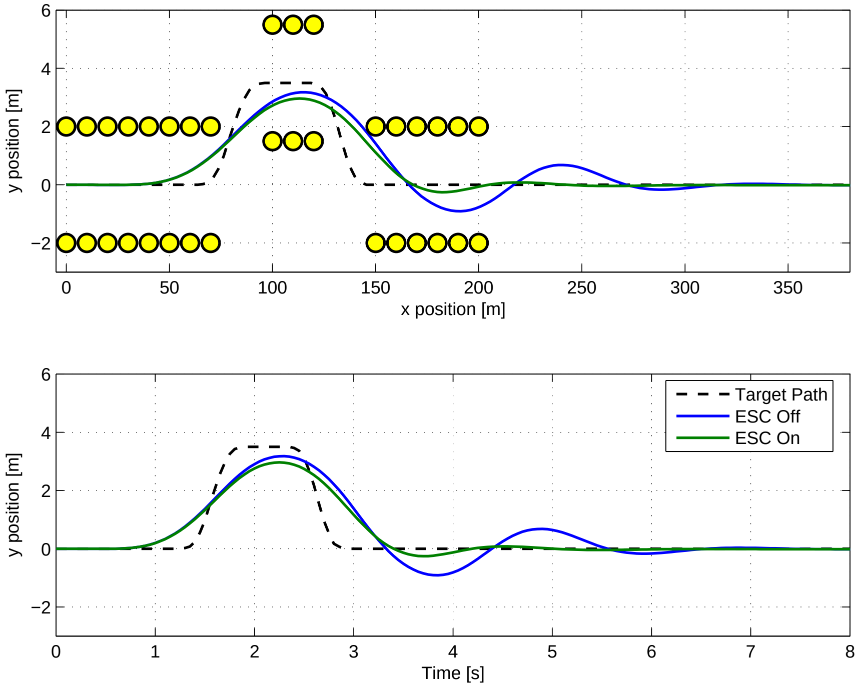


Figure 5.23: Case 04 Vehicle Trajectories ($V = 185$ kph).

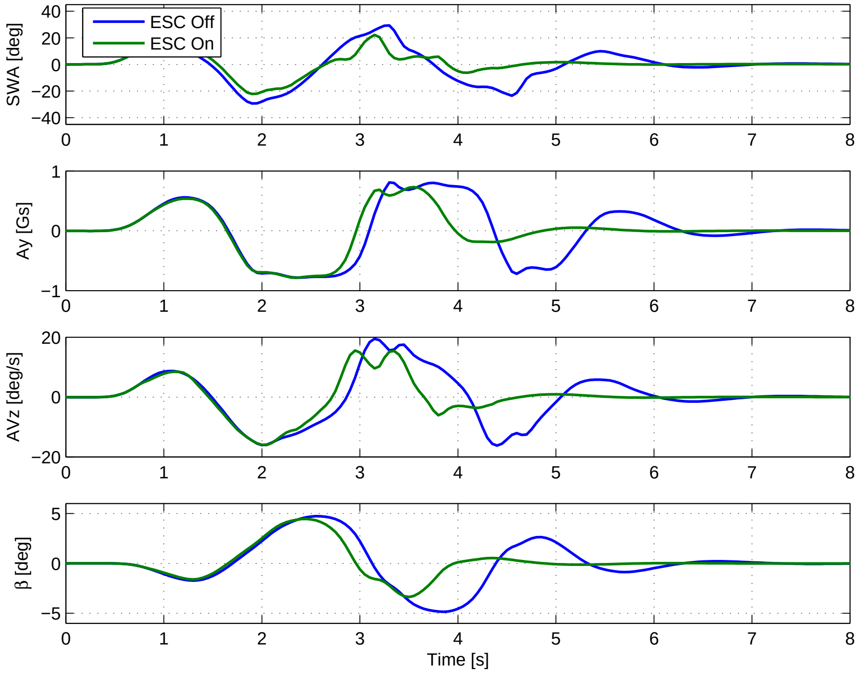


Figure 5.24: Case 04 Vehicle Dynamic Traces ($V = 185$ kph).

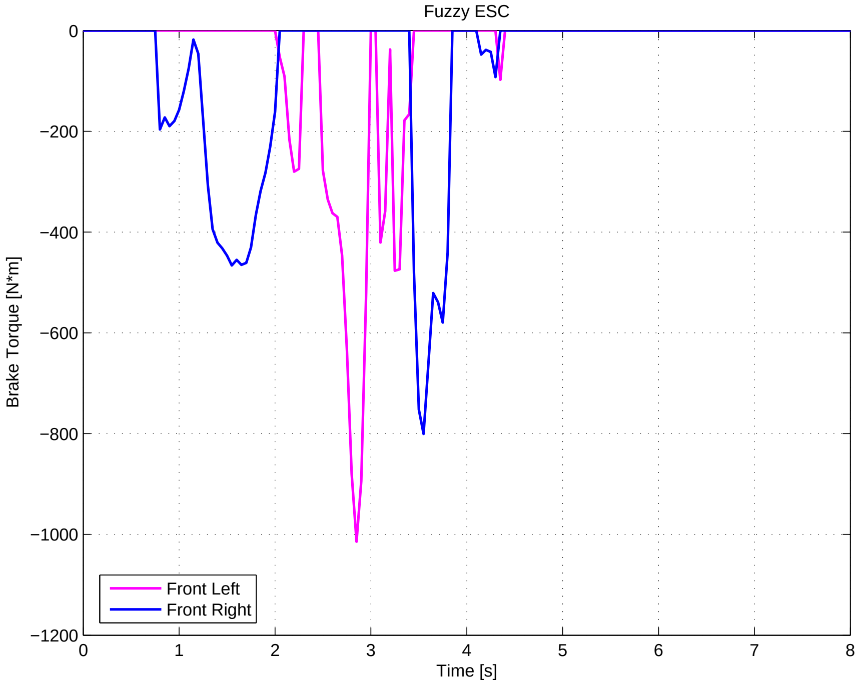


Figure 5.25: Case 04 Braking Torque from Each ESC Algorithm ($V = 185$ kph).

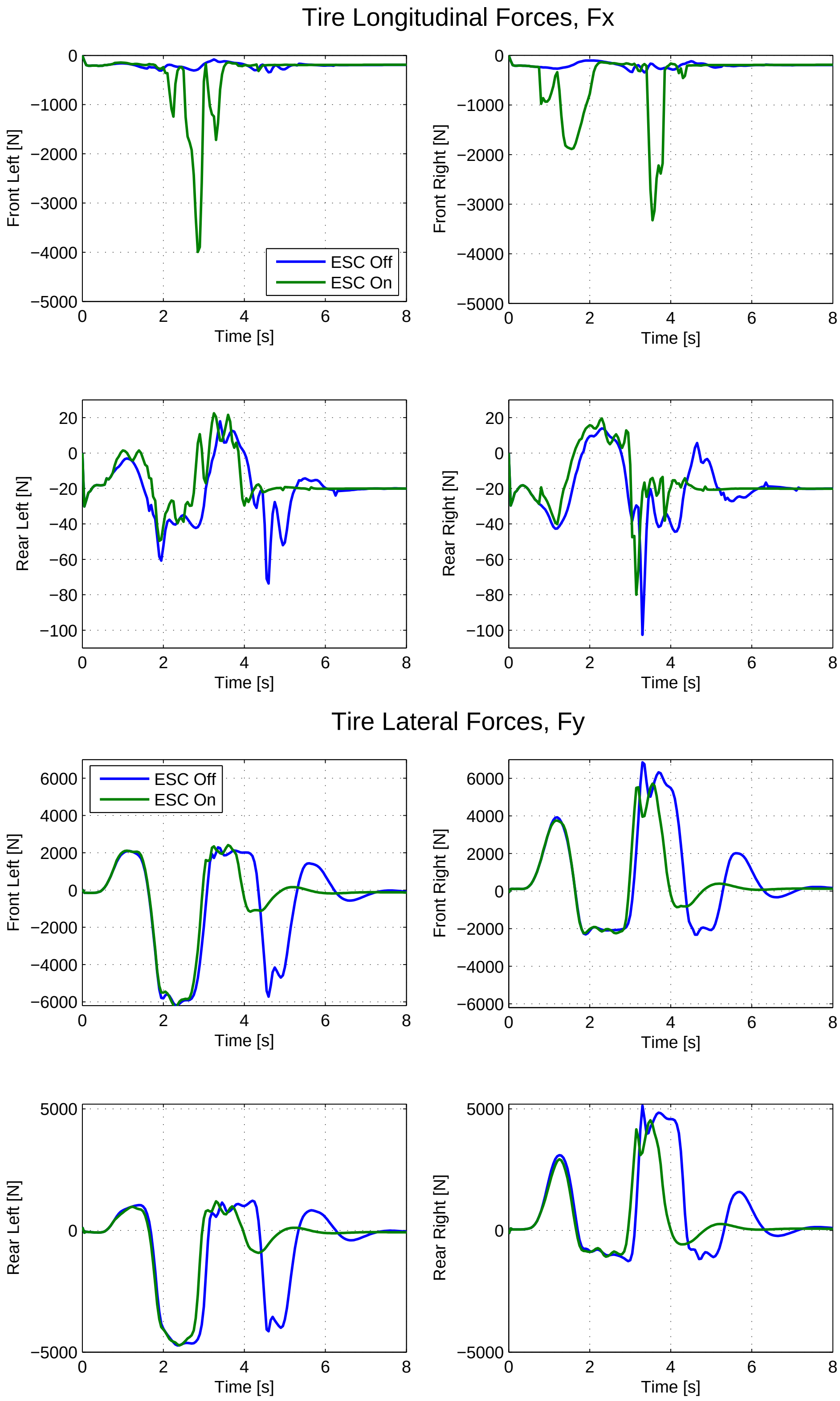


Figure 5.26: Case 04 Tire Forces ($V = 185$ kph).

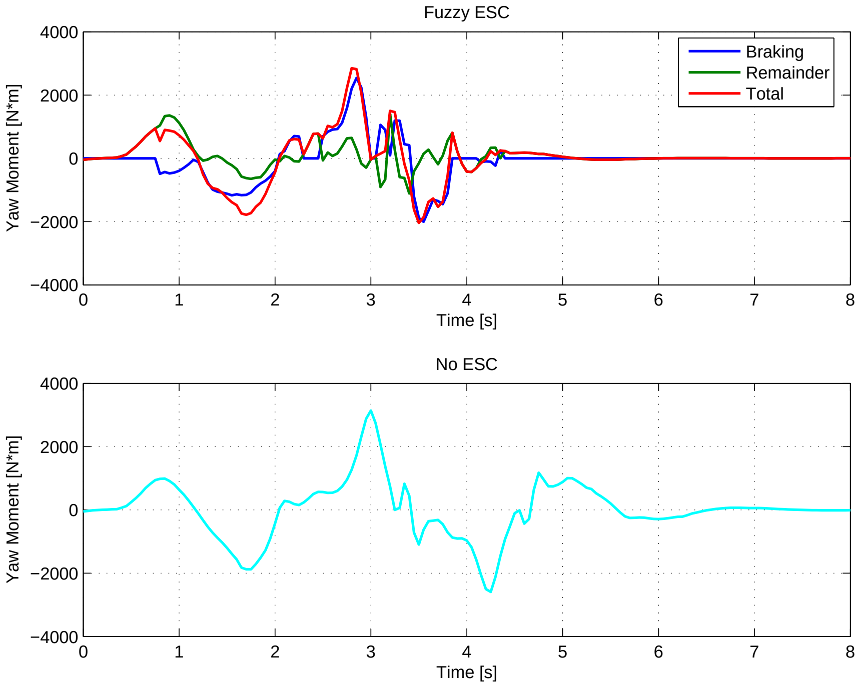


Figure 5.27: Case 04 Yaw Moment ($V = 185$ kph).

5.3.5 Case 5: BMW Mini with Degraded Rear Tires, Fishhook Maneuver

The fishhook maneuver is typically used in rollover testing; however, it can easily induce a spin with the harsh steering input. For this case the BMW Mini with degraded rear tires were used so that it would experience oversteer during this test. The trajectory plot (Figure 5.28) shows that ESC kept the vehicle headed where it should for this test without spinning while without ESC, the vehicle spun-out at about 4.5 s where it diverges off of the path. This can also be seen in the vehicle dynamics sideslip trace (Figure 5.29). The open loop steering input can also be seen in Figure 5.29. The tire forces are shown in Figure 5.31. These do not return back to zero as in the DLC because the vehicle is still cornering at the end of this test.

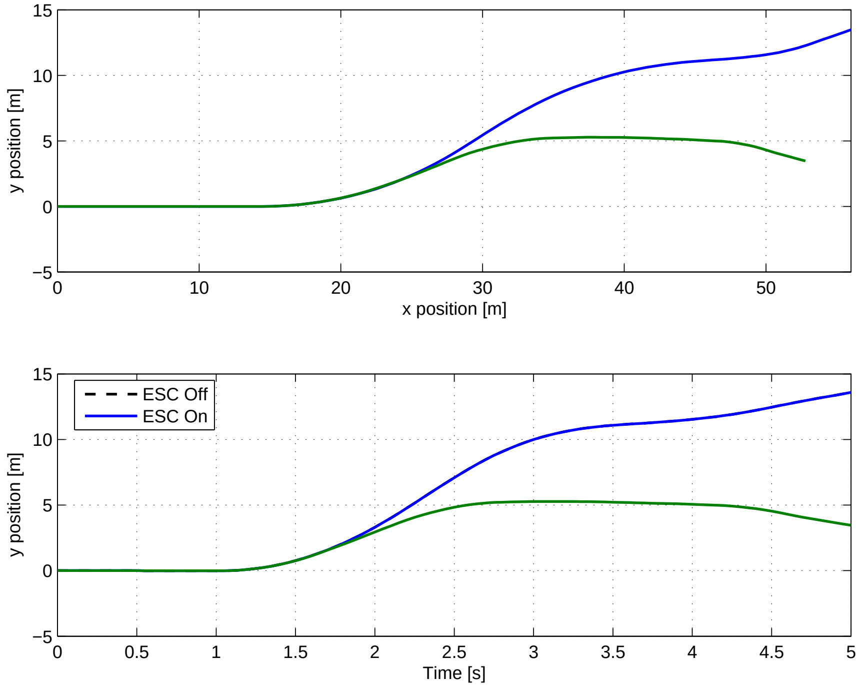


Figure 5.28: Case 05 Vehicle Trajectories ($V = 50$ kph).

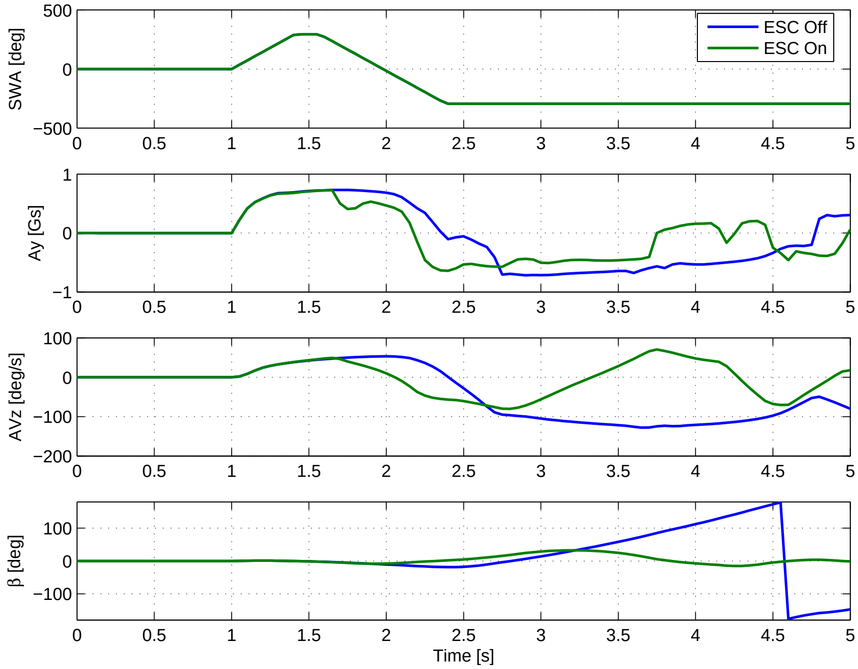


Figure 5.29: Case 05 Vehicle Dynamic Traces ($V = 50$ kph).

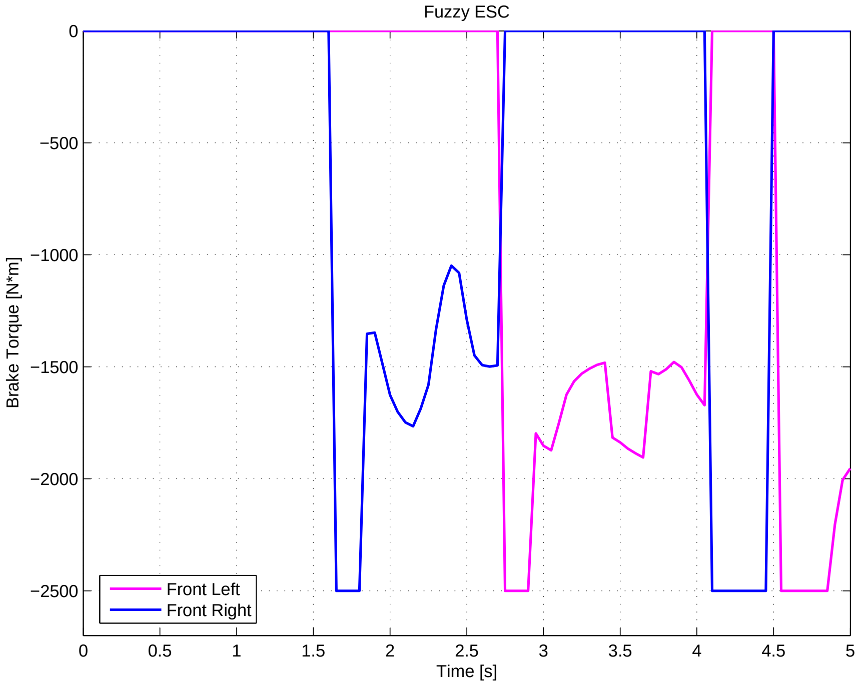


Figure 5.30: Case 05 Braking Torque from Each ESC Algorithm ($V = 50$ kph).

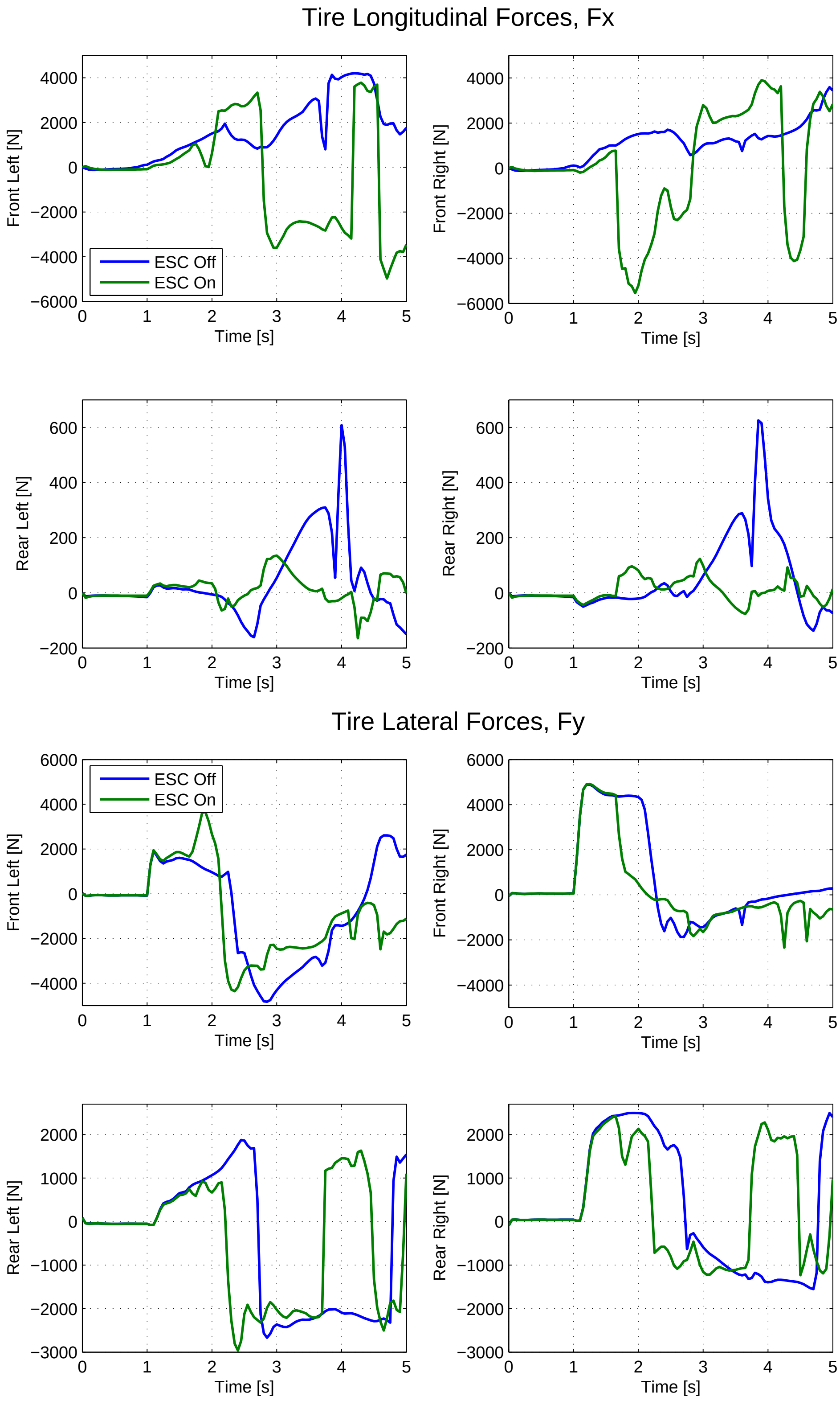


Figure 5.31: Case 05 Tire Forces ($V = 50$ kph).

5.3.6 Case 6: BMW Mini with Degraded Rear Tires, Double Lane Change, Genta Driver Model

This case study consisted of the BMW Mini with degraded rear tires and the Genta driver model which is described in detail in Section 5.2.2.2. This driver model is assumed to be more realistic than the CarSim driver model because the speeds through the lane change are closer to what an average person could do. Also, this case study tests the robustness of the fuzzy ESC with inputs from a very different driver model. For this case study, the standard speed for the lane change (80 kph) was chosen to examine the vehicle with and without ESC. Examining the trajectory plot in Figure 5.32, both configurations (with and without ESC) fail the DLC maneuver. However, it can be seen (Figure 5.33) that after 3 s, the car without ESC spun-out while the fuzzy logic ESC commanded braking (Figure 5.34) to prevent the vehicle from spinning. The spin can be seen in vehicle sideslip trace where it goes from 180 deg to -180 deg. Neither configuration (with or without ESC) produced ”good” signals in the fact that the values of sideslip, lateral acceleration, and yaw rate are quite large; however, the ESC algorithm kept the vehicle from going unstable.

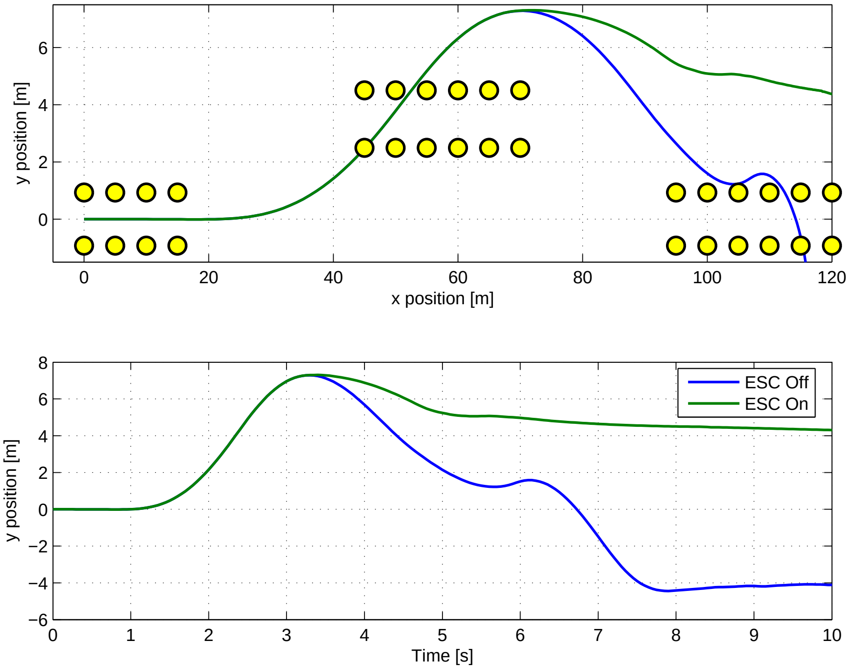


Figure 5.32: Case 06 Vehicle Trajectories ($V = 80$ kph).

The tire forces, seen in Figure 5.35, illustrate how the ESC algorithm applied braking to prevent the vehicle from spinning-out. The commanded braking can be seen in the longitudinal tire forces. It can

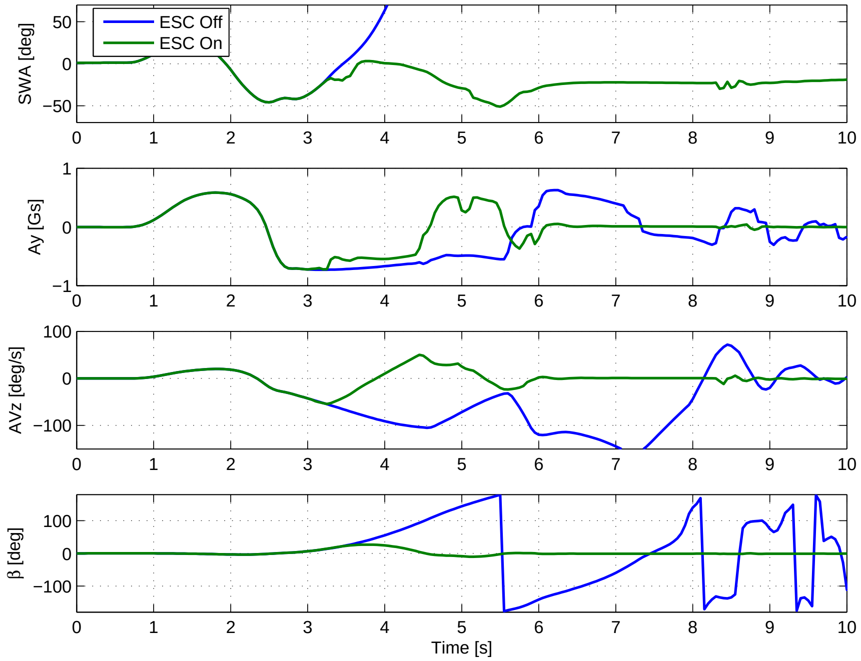


Figure 5.33: Case 06 Vehicle Dynamic Traces ($V = 80$ kph).

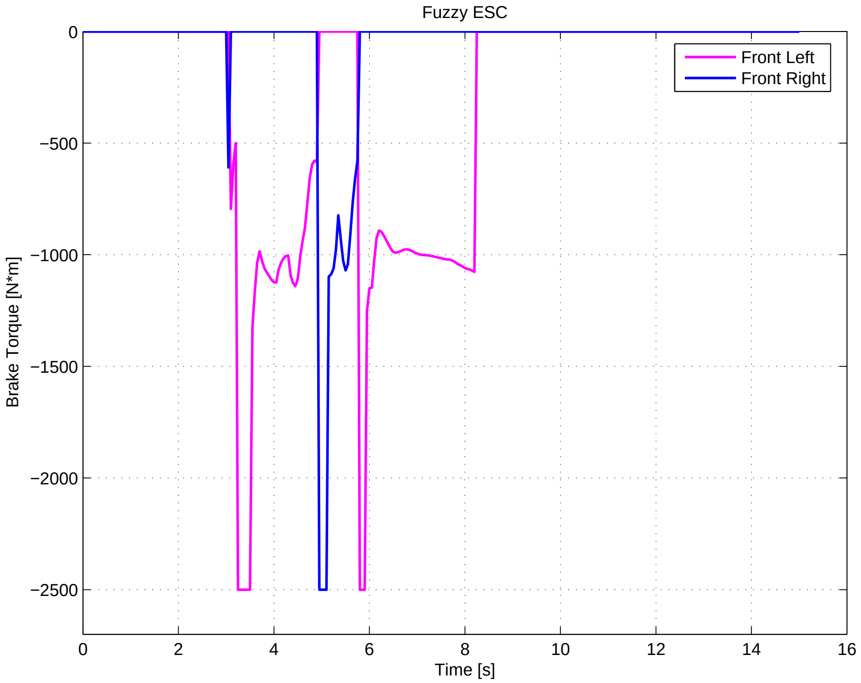


Figure 5.34: Case 06 Braking Torque from Each ESC Algorithm ($V = 80$ kph).

be seen that at 3 s, the front left wheel commanded braking for a fairly large amount of time to prevent the vehicle from spinning-out in the second turn of the DLC. In the beginning of the maneuver, a similar amount of lateral force for both with and without ESC was used; however, after 6 s, the ESC has stopped the vehicle from spinning and the vehicle is heading straight. The configuration without ESC has large spikes in its lateral force traces after 6 s which show the vehicle is still oscillating. The yaw moment plot for this case (Figure 5.36) illustrates how the ESC braking drastically lowers the magnitude of this signal and damps the vehicle oscillation at the end of the run.

The last item of interest for this case is the maximum speed at which the vehicle can traverse the lane change without going unstable with and without ESC. Neither car with or without ESC can successfully negotiate the lane change at these speeds, i.e, neither could sustain the steering input that was initiated in an attempt to negotiate the lane change. However, in the attempt, the car without ESC could traverse the lane change at a maximum speed of 71.3 kph without going unstable while the car with ESC could traverse it at 81.5 kph before going unstable. The trajectory plots can be see in Figure 5.37 while the dynamic traces can be seen in Figure 5.38. Also in Figure 5.37, it should be noted that the car is not returning to the centerline of the course because of the dramatically decreased longitudinal speed that the ESC braking has caused.

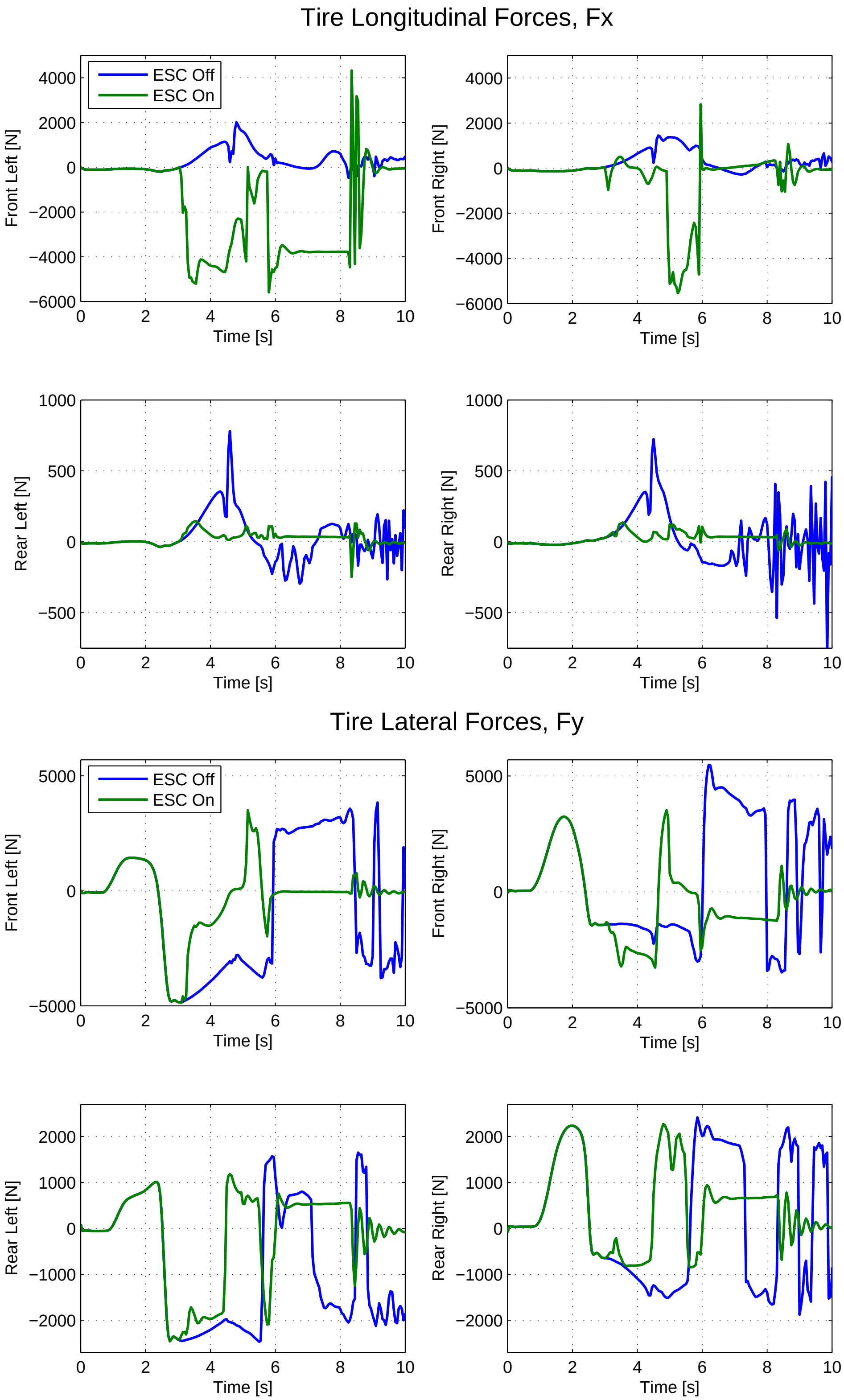


Figure 5.35: Case 06 Tire Forces ($V = 80$ kph).

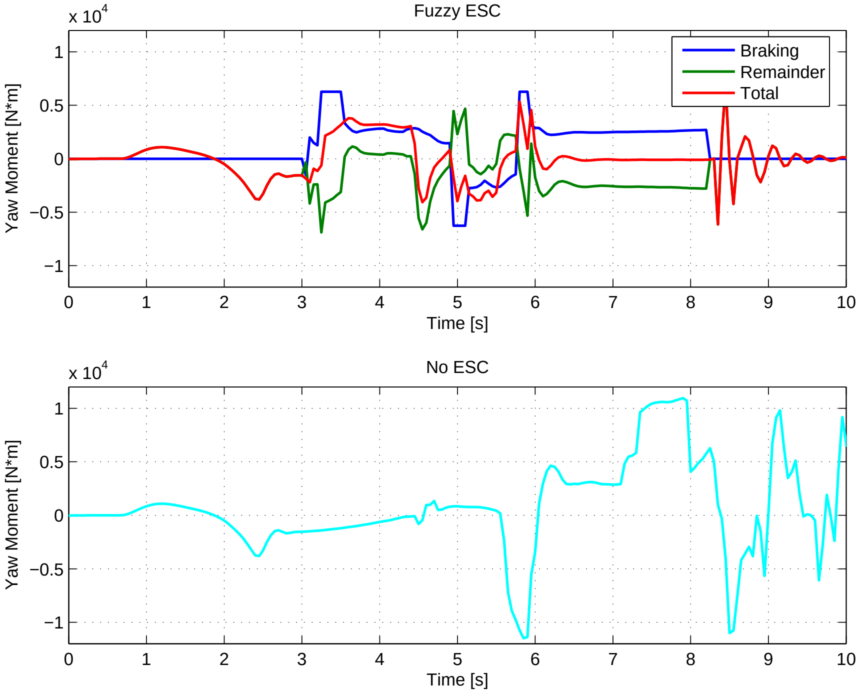


Figure 5.36: Case 06 Yaw Moment ($V = 80$ kph).

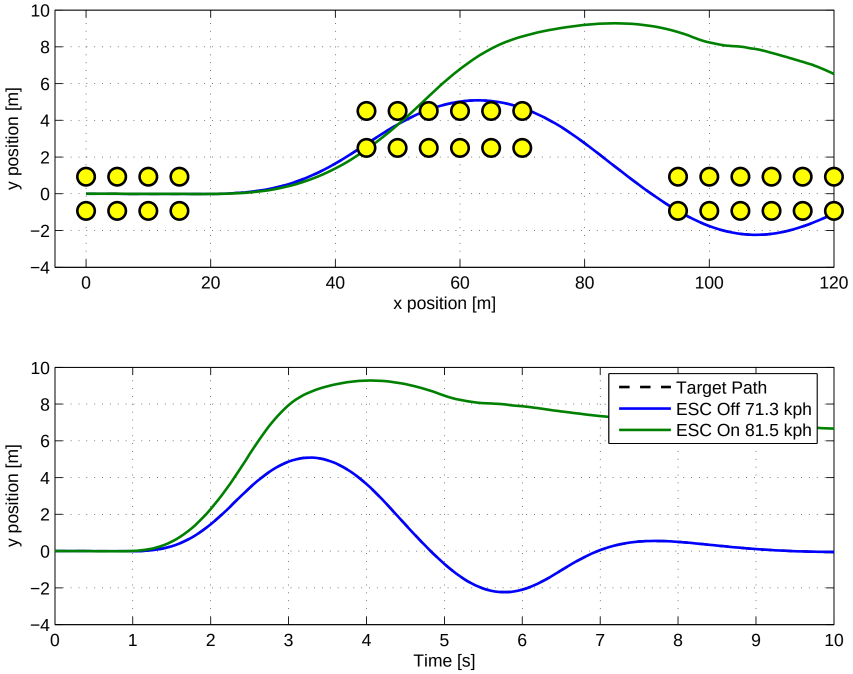


Figure 5.37: Case 06 Vehicle Trajectories Max Speed ($V = 71.3, 81.5$ kph ESC off/on).

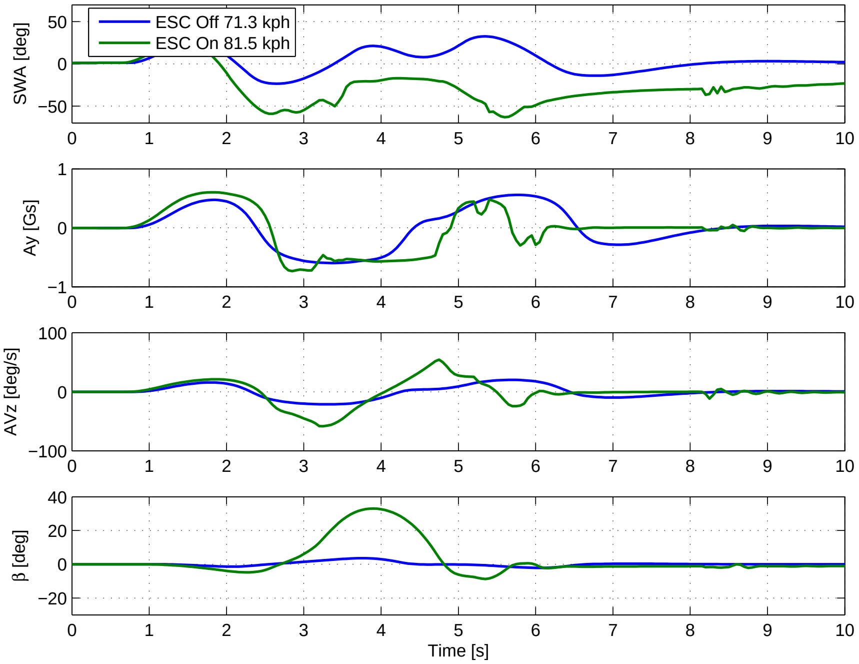


Figure 5.38: Case 06 Vehicle Dynamic Traces Max Speed ($V = 71.3, 81.5$ kph ESC off/on).

5.3.7 Case 7: CarSim Sports Car, Understeer Test

This test consisted of the CarSim Sports Car on an understeer test. This is where the vehicle increases speed around a 500 ft circular path until it experiences terminal oversteer or understeer. The test starts at 120 kph and the speed is ramped at 1.2 kph until failure. This test was chosen to test the ESC algorithm on a different maneuver with a car that is known to experience terminal oversteer. From the trajectory plot in Figure 5.39, it can be seen that the vehicle without ESC quickly loses control while the case with ESC makes it much further along the path. Without stability control, the vehicle spun-out at 122.4 kph while with ESC, it understeers off the course at 129.5 kph. Corrective braking can be seen in Figure 5.41. The steering trace (Figure 5.40) shows that the car without ESC is out of control while the driver of the car with ESC keeps the steering at a near steady state value of 24 deg while traversing the course. The lateral acceleration for the car with ESC shows the plateau effect that is expected from this test while yaw rate and side slip show very similar plateaus. The tire forces (Figure 5.42) show similar plateau effects in the lateral components.

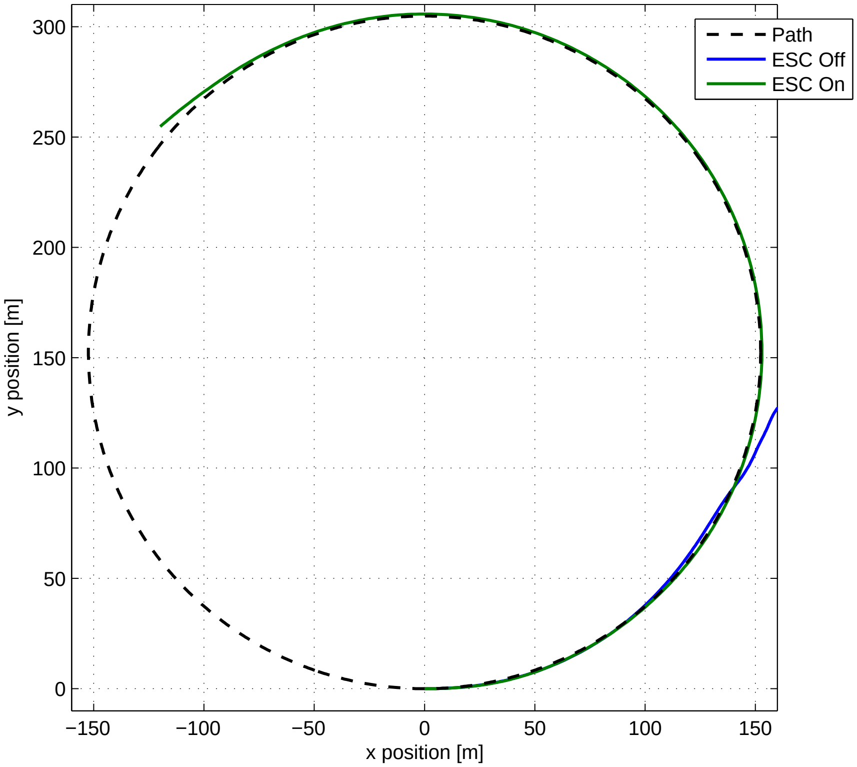


Figure 5.39: Case 07 Vehicle Trajectories.

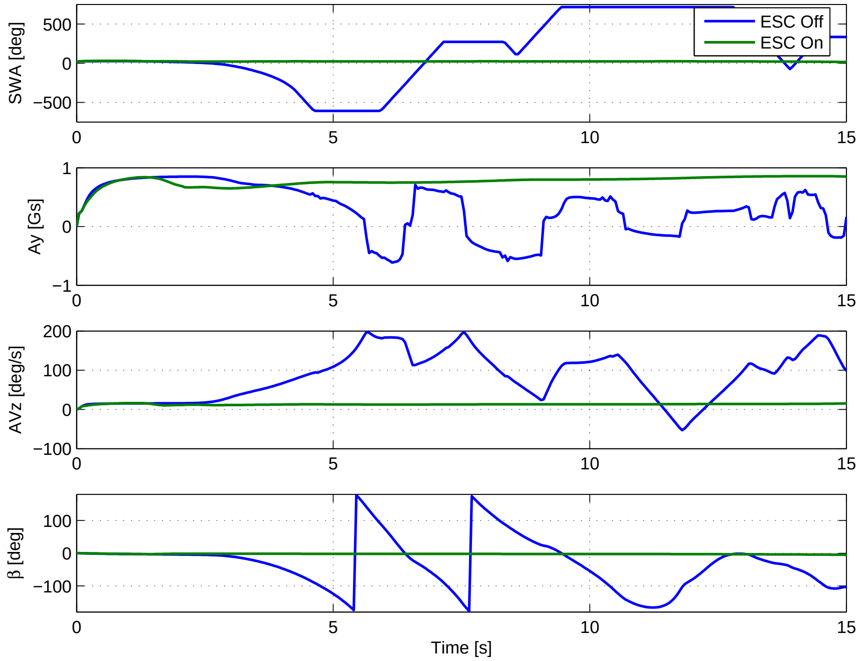


Figure 5.40: Case 07 Vehicle Dynamic Traces.

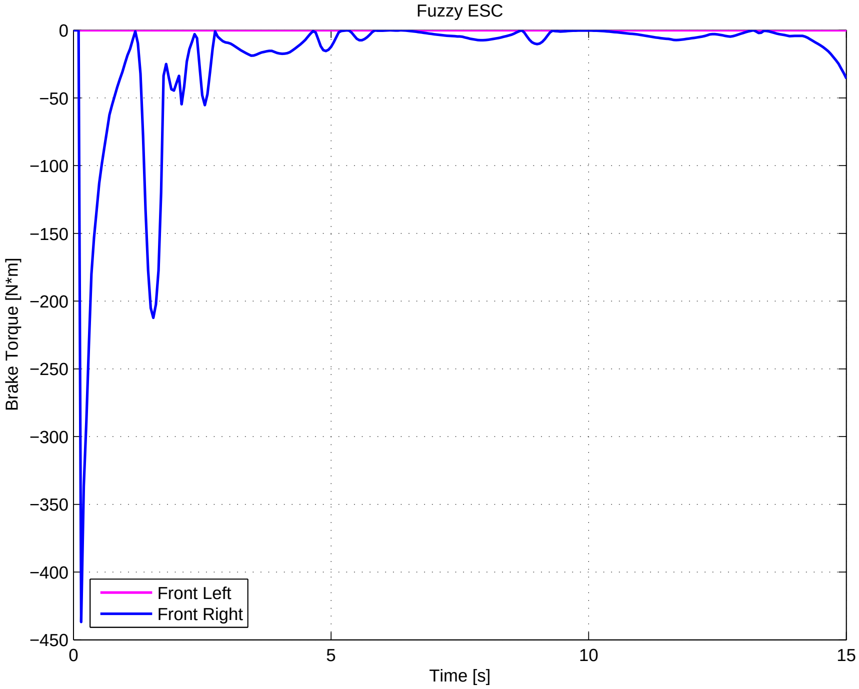


Figure 5.41: Case 07 Braking Torque from Each ESC Algorithm.

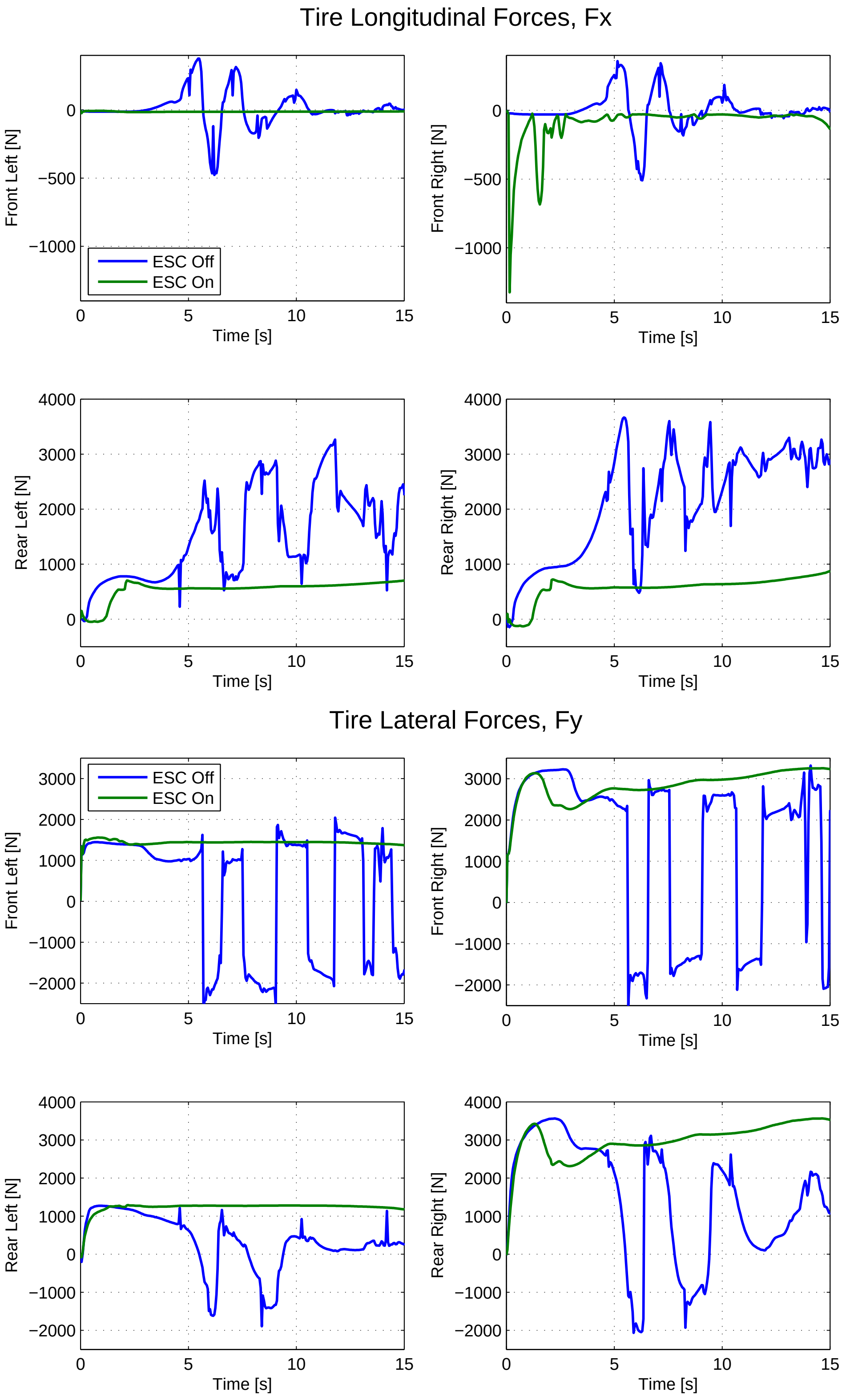


Figure 5.42: Case 07 Tire Forces.

5.3.8 Case 8: CarSim Sedan, Double Lane Change

In this study the CarSim Sedan completed a double lane change to illustrate the controller's robustness with respect to changing vehicle parameters. Both the trajectory (Figure 5.43) and vehicle dynamics (Figure 5.44) show results similar to Cases 02 and 03. The vehicle without ESC spun-out after the DLC while it was stabilized with ESC and successfully negotiated the lane change. The corrective braking can be seen in Figure 5.45. For the ESC-equipped car, the tire lateral forces (Figure 5.46) are damped to zero at the end of the lane change (after 4 s) and the longitudinal forces illustrate the corrective braking applied. The yaw moment (Figure 5.47) is similar in both cases until after the DLC where it is damped to zero for the car with ESC.

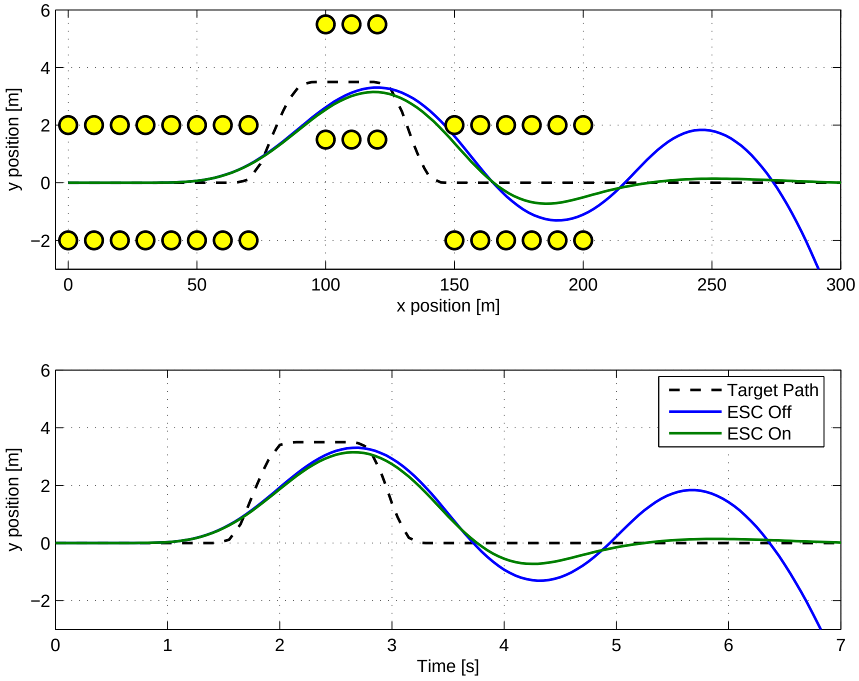


Figure 5.43: Case 08 Vehicle Trajectories ($V = 165$ kph).

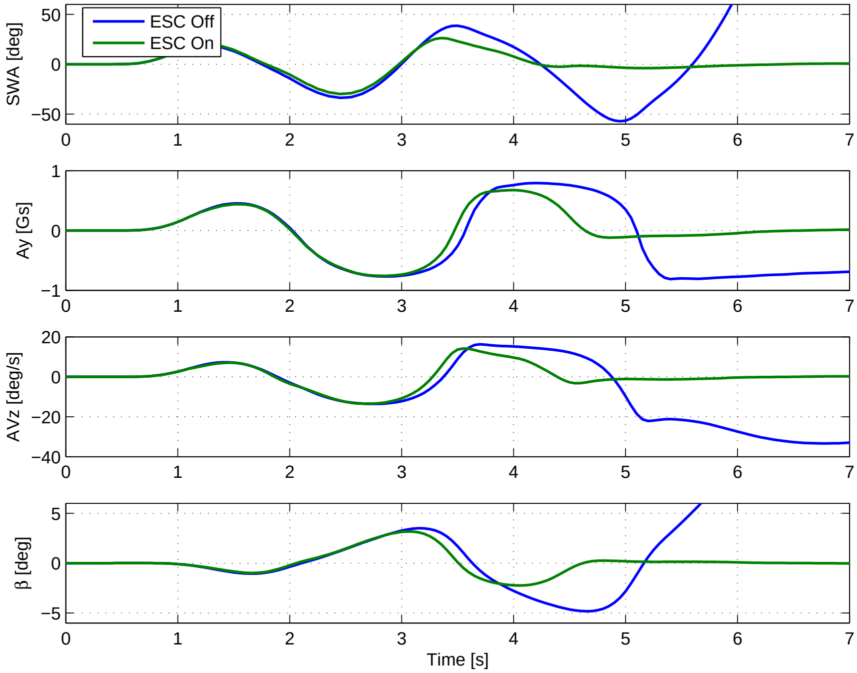


Figure 5.44: Case 08 Vehicle Dynamic Traces ($V = 165$ kph).

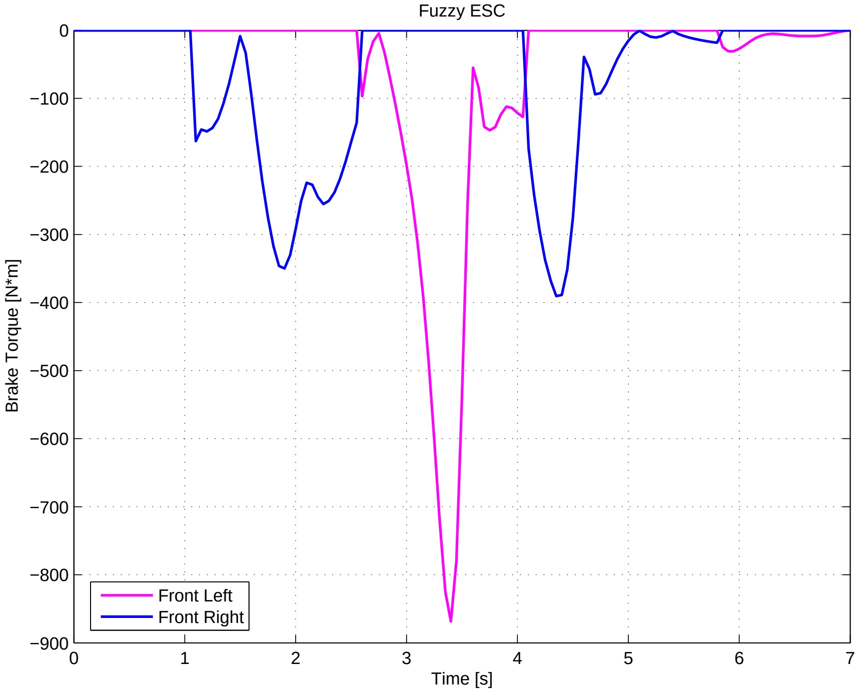


Figure 5.45: Case 08 Braking Torque from Each ESC Algorithm ($V = 165$ kph).

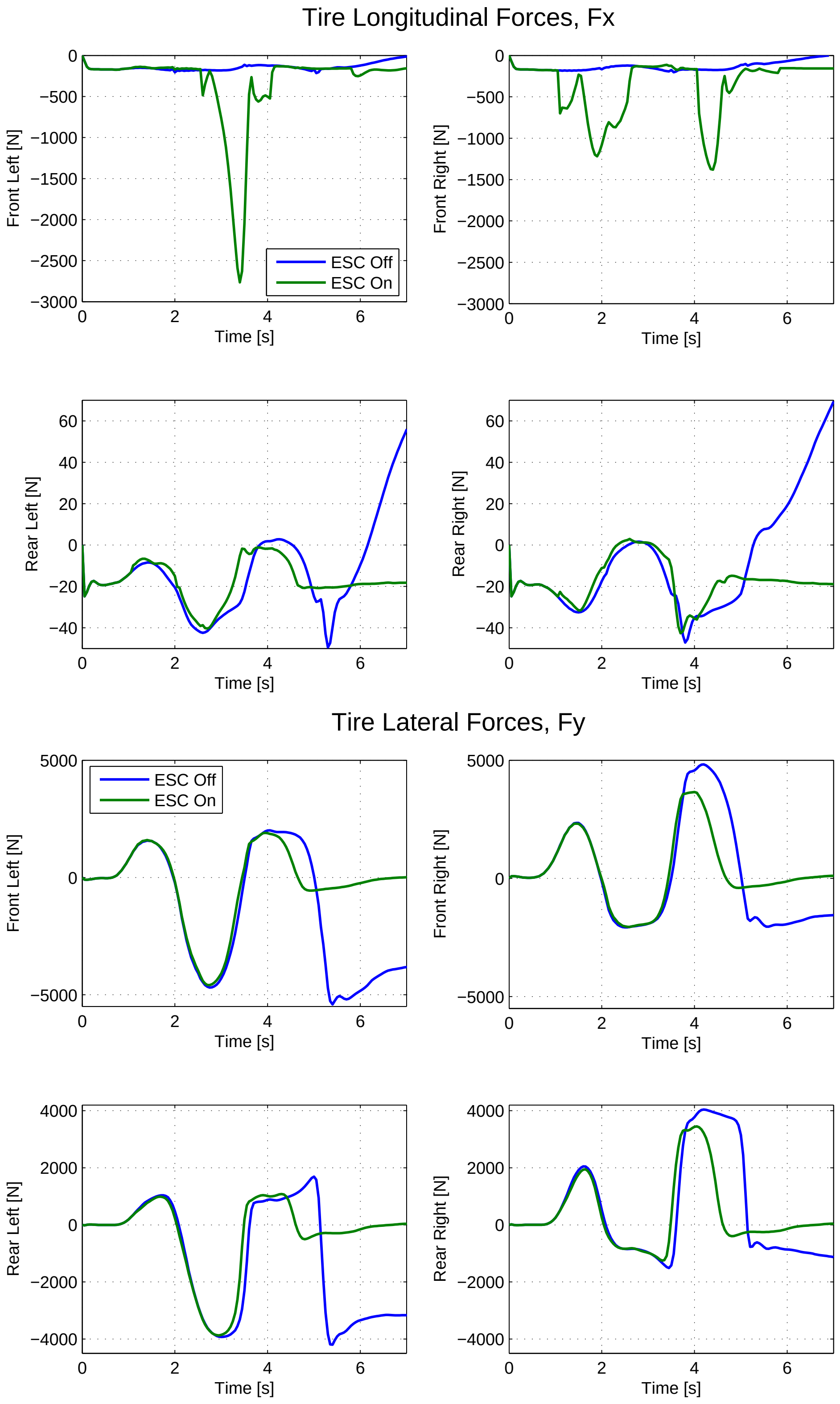


Figure 5.46: Case 08 Tire Forces ($V = 165$ kph).

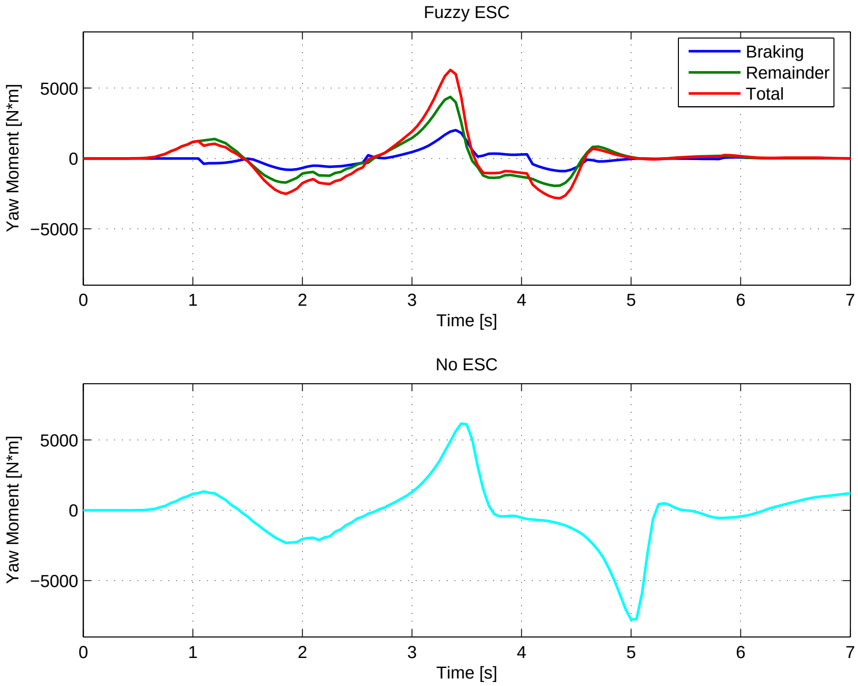


Figure 5.47: Case 08 Yaw Moment ($V = 165$ kph).

5.3.9 Case 9: CarSim SUV with Degraded Rear Tires, Double Lane Change

This case study involved the CarSim SUV in a double lane change maneuver with degraded rear tires. This case tested the ESC installed in a very different vehicle as compared with the other vehicles tested. It has a much higher center of gravity and overall, much larger dimensions. The results are similar to those of Case 02, 03, and 08. The vehicle without ESC spun-out while the car with ESC maintained stability and completed a successful lane change. The trajectory (Figure 5.48) and vehicle dynamics (Figure 5.49) show how the fuzzy ESC lowered driver workload and damped out lateral acceleration, yaw rate, and vehicle sideslip angle after the exit of the DLC (4 s). The corrective brake torque can be seen in Figure 5.50 while the tire forces illustrate the event in Figure 5.51. These show how ESC stabilizes the vehicle and keeps it heading straight at the end of the run (zero lateral forces). The yaw moment illustrates that the braking helped the vehicle traverse the maneuver (2.0 -3.5 s) and from 3.5 - 4.0 s, it damped out the yaw moment.

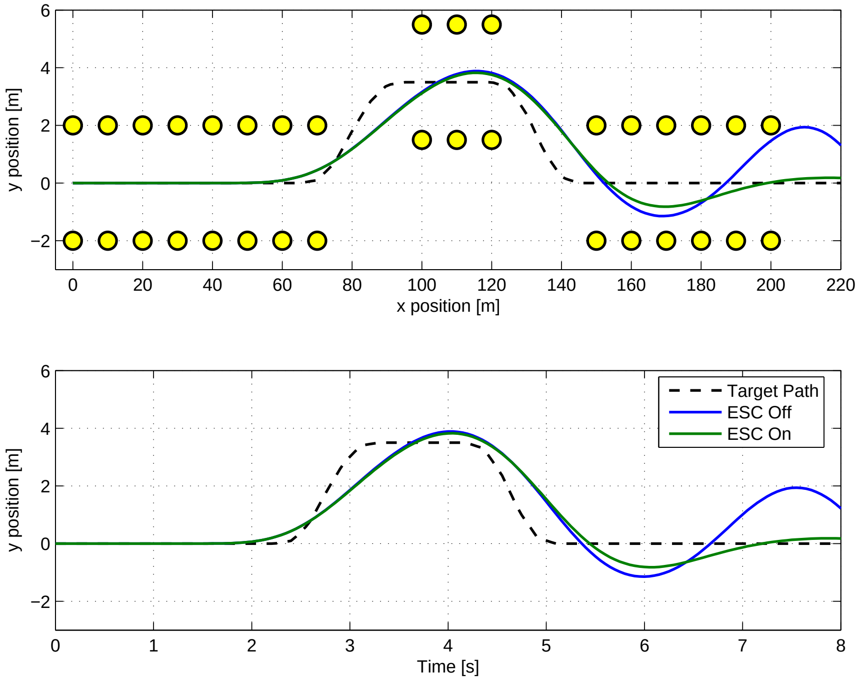


Figure 5.48: Case 09 Vehicle Trajectories ($V = 106$ kph).

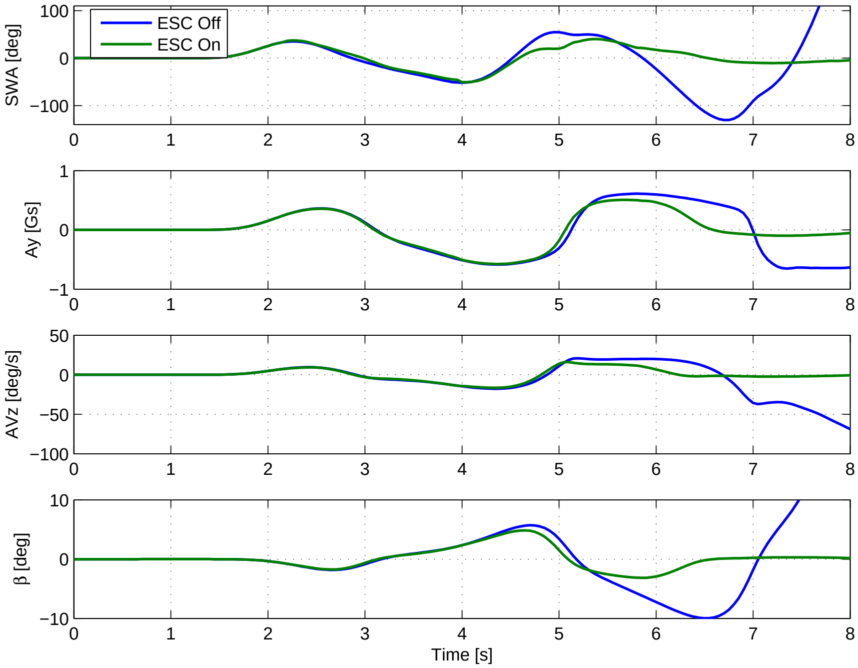


Figure 5.49: Case 09 Vehicle Dynamic Traces ($V = 106$ kph).

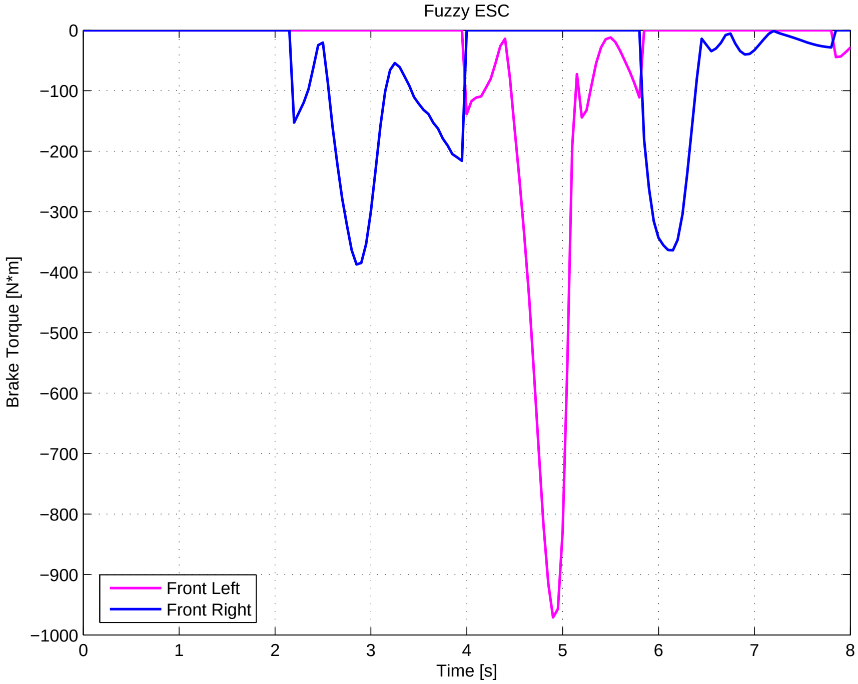


Figure 5.50: Case 09 Braking Torque from Each ESC Algorithm ($V = 106$ kph).

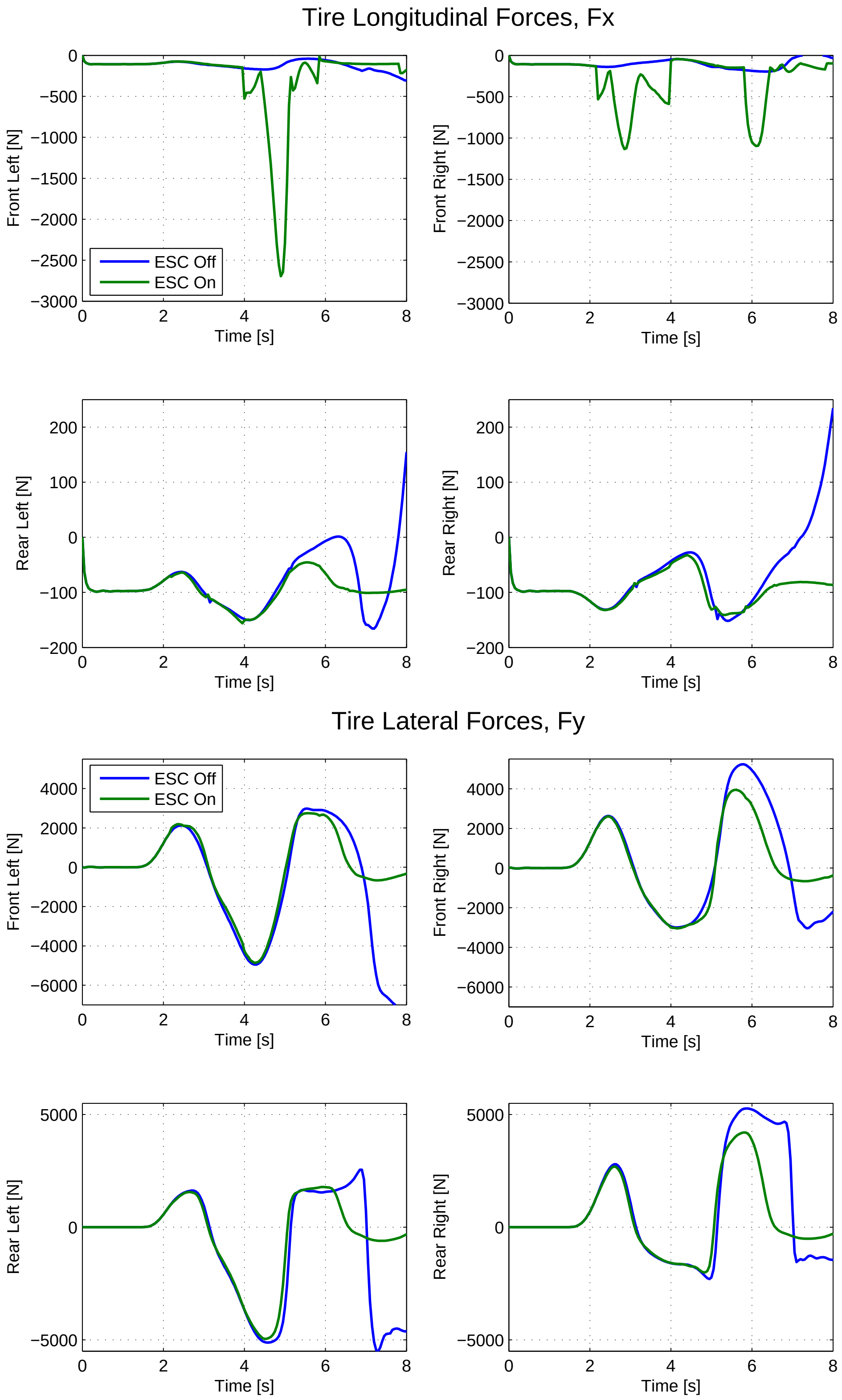


Figure 5.51: Case 09 Tire Forces ($V = 106$ kph).

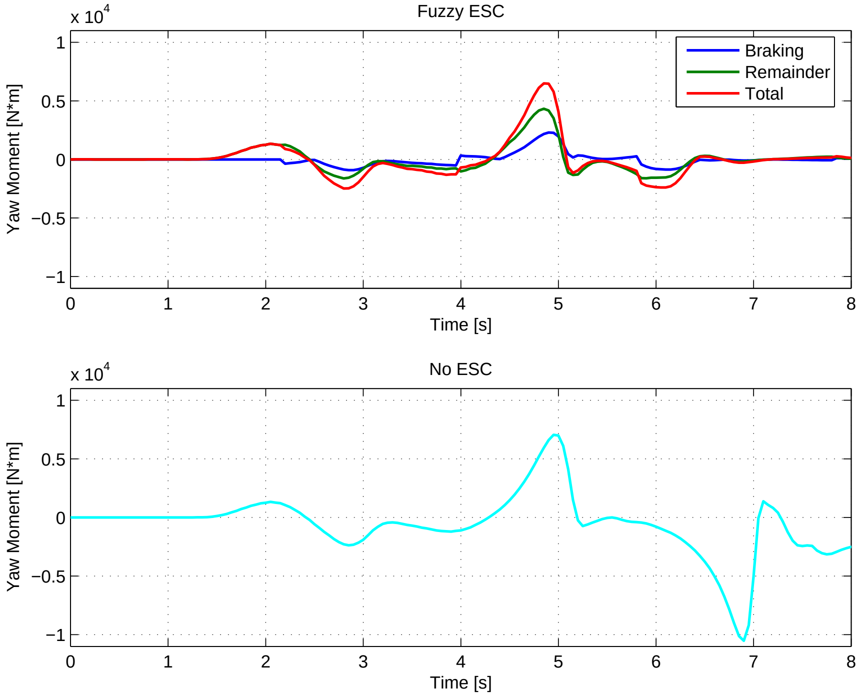


Figure 5.52: Case 09 Yaw Moment ($V = 106$ kph).

5.3.10 Case 10: Nominal BMW Mini, Double Lane Change, Split Mu Conditions

This case study illustrated split mu conditions which is a very common stability test. It simulates one side of the vehicle hitting a very slick road ($\mu = 0.2$) while the other has a higher coefficient of friction ($\mu = 0.5$). It is very similar to hitting an ice patch. As in Cases 02, 03, 08, and 09, the ESC stabilized the vehicle (although it did violate the lane change boundary at about $x = 150$ m). The same vehicle without ESC spun-out after exiting the DLC. Figures 5.53 and 5.54 illustrate the trajectory and the vehicle dynamics respectively. The control action can be seen in Figure 5.57. The tire forces throughout the run can be seen in Figure 5.56. Here it can be seen that the lateral forces are returned to zero at the end of the run with ESC and diverge (because of the spin) for the configuration with ESC off. The yaw moment (Figure 5.58) shows how the braking damped out the yaw moment after 5 s (exit of the DLC).

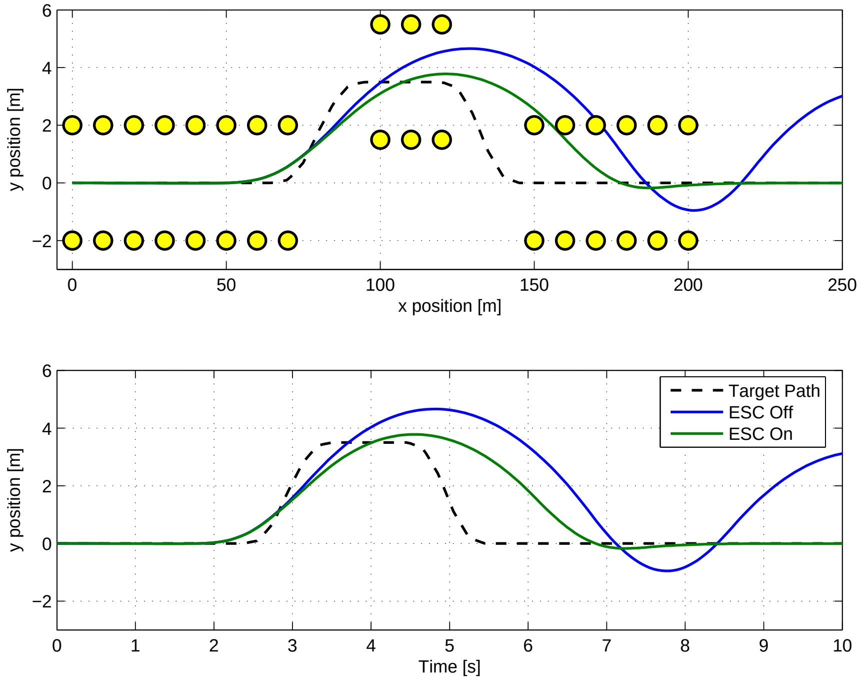


Figure 5.53: Case 10 Vehicle Trajectories ($V = 100$ kph).

5.4 Summary

In this chapter, ten case studies (Table 5.1) were presented that tested the same ESC algorithm for various cars, loading conditions, maneuvers, driver models, and road conditions. The signals used were

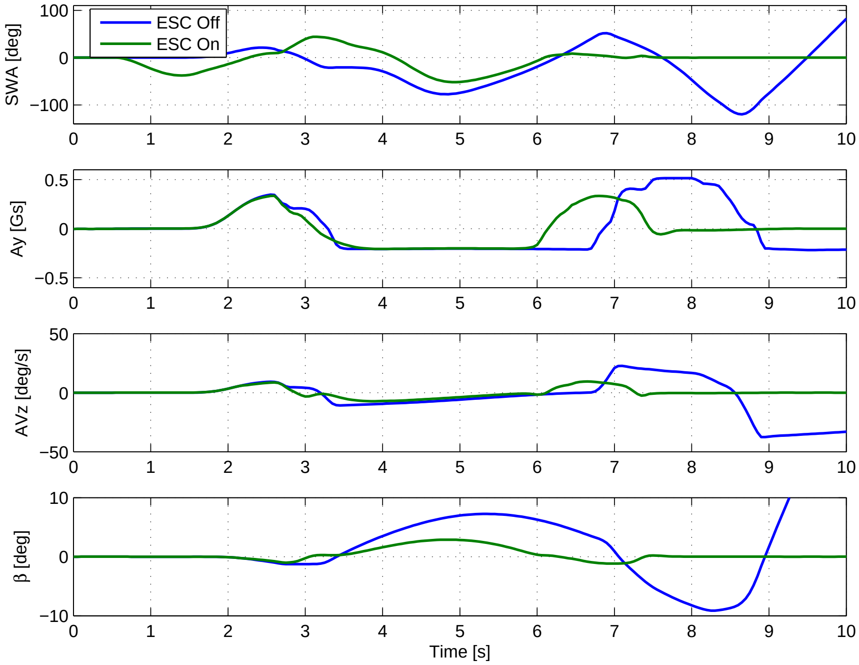


Figure 5.54: Case 10 Vehicle Dynamic Traces ($V = 100$ kph).

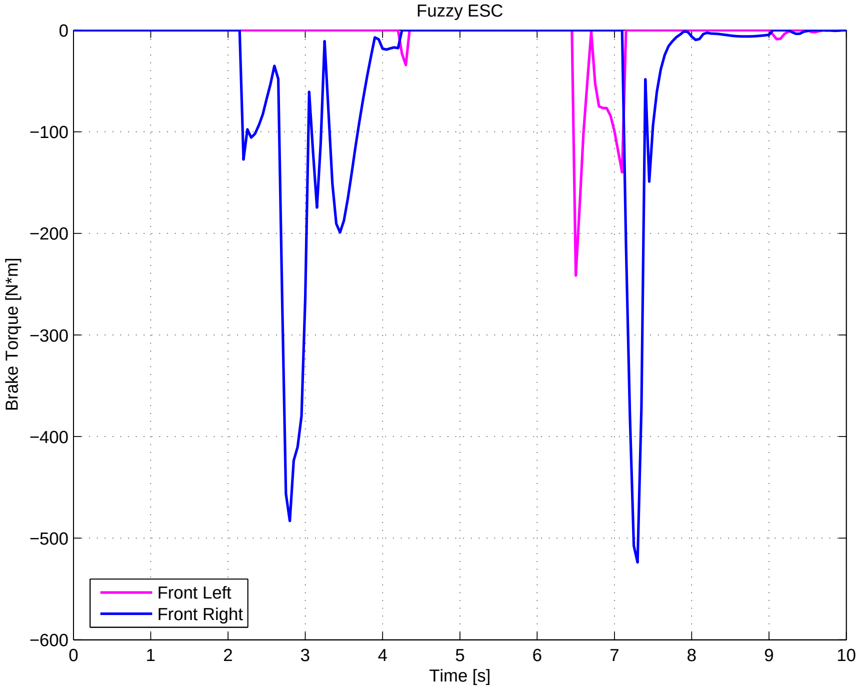


Figure 5.55: Case 10 Braking Torque from Each ESC Algorithm ($V = 100$ kph).

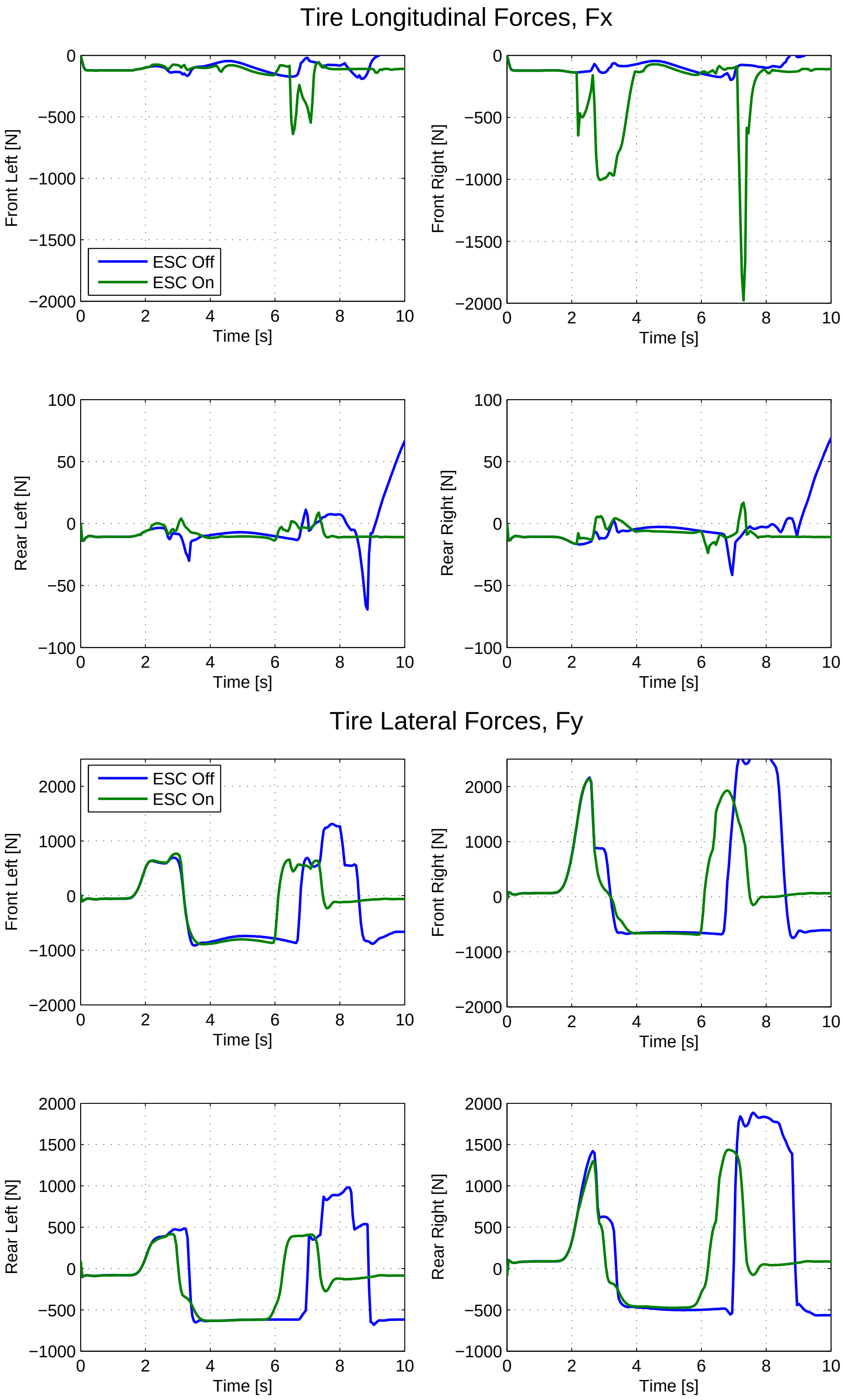


Figure 5.56: Case 10 Tire Forces ($V = 100$ kph).

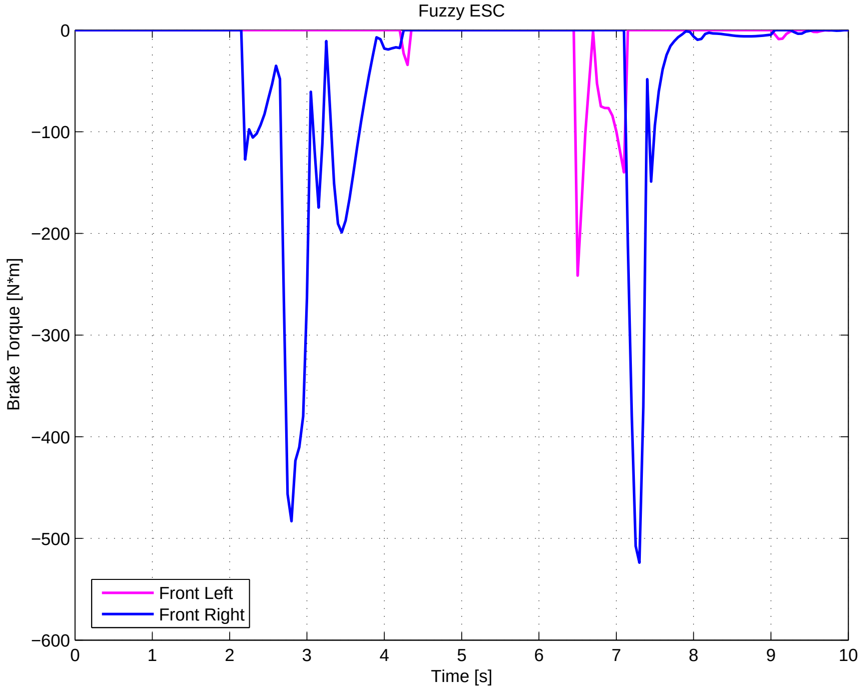


Figure 5.57: Case 10 Braking Torque from Each ESC Algorithm ($V = 100$ kph).

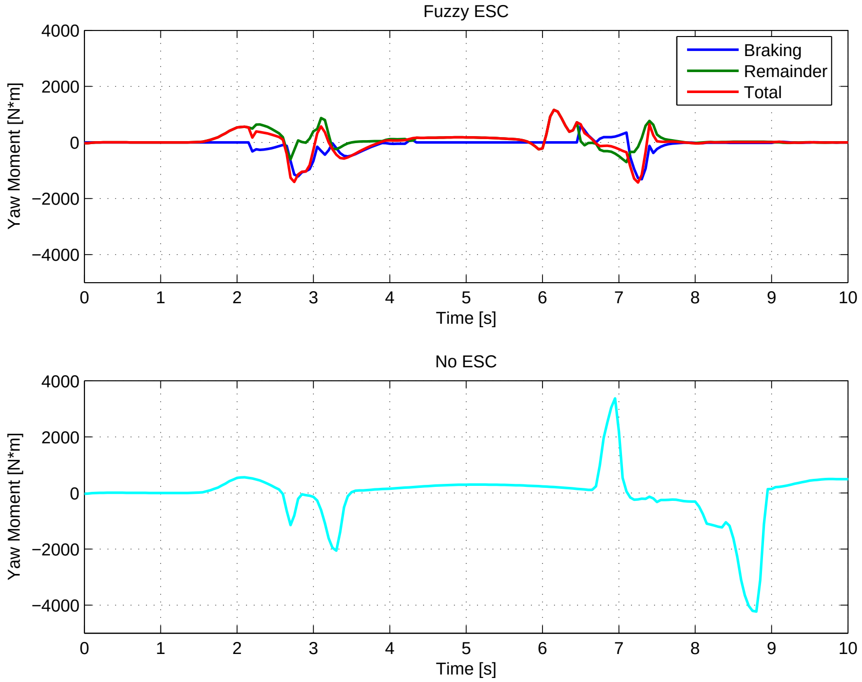


Figure 5.58: Case 10 Yaw Moment ($V = 100$ kph).

yaw rate, vehicle speed, lateral acceleration, and steering wheel angle. The results from the ten studies demonstrate the robustness of the fuzzy ESC controller in stabilizing the vehicle in each case. In Case 01, the controller was compared to a ”traditional” ESC algorithm that relies on a two degree-of-freedom estimator. Comparable results were found between the two controllers. In several of the cases (Cases 02, 03, 08, 09, and 10) the fuzzy ESC was able to prevent a spin-out. In summary, this single fuzzy ESC algorithm was able to work without tuning on all ten cases to stabilize the vehicle.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis, a fuzzy logic stability control algorithm is presented. This algorithm relies only on measurable vehicle parameters (i.e., steering wheel angle, lateral acceleration, yaw rate, and longitudinal velocity) to actuate differential braking and stabilize the vehicle. The advantage of using only measurable vehicle parameters is that the system does not rely on vehicle models to estimate vehicle states based on driver inputs. These estimators rely on accurate knowledge of the parameters used in a particular vehicle, the conditions between the tire and the road, and require additional means that add complexity to correct for inaccuracies in these estimations. Cornering stiffness ($C\alpha$) is the main parameter used in the two degree-of-freedom model commonly used in stability control and changes with age, loading, and driving conditions (coefficient of friction between the tire and road).

This thesis discussed the basic properties and relationships among the signals used to determine stability with respect to oversteer. Also, a brief overview of fuzzy logic was presented. The work of Fey [5] was extensively used in describing signals and their relationships that indicate oversteer. Also, Vaduri's work [6] in indicating understeer or oversteer with fuzzy logic was expanded to develop the fuzzy logic stability control algorithm used in this work. An example of a stable double lane along with an increased speed DLC which causes the vehicle to become unstable was presented. The signals associated with these maneuver and how the fuzzy logic ESC reacts to the instabilities and initiates differential braking to stabilize the vehicle was presented.

Finally, ten case studies which encompass several vehicles, configurations, drivers, maneuvers, and tire-to-road conditions were tested to thoroughly prove the validity of this ESC strategy. It should be noted that the same exact ESC algorithm was used without tuning for all ten cases to illustrate the robustness of the system. In conclusion, it appears that stability control of a vehicle can be achieved without the use of traditional estimators and by using only easily measurable signals.

6.2 Future Work

Future work on this topic should look at understeer; specifically, how to indicate both oversteer and understeer simultaneously and initiate control actions that will not conflict with one another. These two are very different indicators and when both are examined together, they tend to conflict. For example, when oversteer actuates a control action, the next time step looks like an understeering vehicle and it will command understeer braking.

Appendices

Appendix A

VEHICLE PARAMETERS

In this section, the parameters for each of the vehicles used in simulation are summarized. Table A.1 summarizes the BMW Mini in multiple configurations including the nominal curb plus driver (C+D) loading, degraded rear tires, and gross vehicle weight (GVW). Table A.2 summarizes the CarSim vehicle models. Tire force charactericstics are described in Appendix C.

Parameter	Units	Nominal C+D BMW Mini	Degraded Rear BMW Mini	GVW BMW Mini
Inertial Properties:				
Total Vehicle Mass	kg	1323.45	1323.45	1852.93
Sprung Mass	kg	1071.45	1071.45	1600.83
Front Weight per Wheel	N	3893.59	3893.59	5451.03
Rear Weight per Wheel	N	2595.73	2595.73	3634.02
CG Height	m	0.517	0.517	0.517
Front/Rear Distribution	%	60/40	60/40	60/40
Yaw Moment of Inertia	$kg \cdot m^2$	1750	1750	1750
Tire Properties:				
Front Effective Rolling Radius	mm	290	290	290
Rear Effective Rolling Radius	mm	290	290	290
Front Cornering Stiffness	N/deg	1311.0	1311.0	1645.3
Rear Cornering Stiffness	N/deg	943.53	660.5	1250.0
Vehicle Dimensions:				
Wheelbase	m	2.468	2.468	2.468
Front Track Width	m	1.453	1.453	1.453
Rear Track Width	m	1.475	1.475	1.475

Table A.1: BMW Mini Parameters in Multiple Configurations.

Parameter	Units	CarSim Sports Car	CarSim Sedan	CarSim SUV
Inertial Properties:				
Total Vehicle Mass	kg	1140	1530	2532
Sprung Mass	kg	1020	1370	2257
Front Weight per Wheel	N	2794.9	4502.6	6810.3
Rear Weight per Wheel	N	2794.9	2999.5	5605.0
CG Height	m	0.375	0.54	0.781
Front/Rear Distribution	%	50/50	60/40	55/45
Yaw Moment of Inertia	$kg \cdot m^2$	996	4192	3524.9
Tire Properties:				
Front Effective Rolling Radius	mm	338	335	401.4
Rear Effective Rolling Radius	mm	314	335	401.1
Front Cornering Stiffness	N/deg	1356.5	1987.0	1897.6
Rear Cornering Stiffness	N/deg	1356.5	1454.6	1104.7
Vehicle Dimensions:				
Wheelbase	m	2.33	2.78	2.95
Front Track Width	m	1.481	1.55	1.90
Rear Track Width	m	1.486	1.55	1.95

Table A.2: BMW Mini Parameters in Multiple Configurations.

Appendix B

MATLAB AND SIMULINK DOCUMENTATION

Here the Simulink block diagrams of the fuzzy ESC algorithm and the supporting MATLAB code used in running these block diagrams are presented. The home screen (Figure B.1) is the overall system that is used to run the simulation and it is divided into five sections: the input/output block, the scopes and displays, the Genta driver model, the driver throttle (used in Case 07 to ramp velocity), and the stability control system.

As mentioned in Chapter 3, Simulink and CarSim run at a fixed time-step through an S-Function block in Simulink which supports the input and output of variables. This input/output main block is the center of the ESC strategy can be seen in Figure B.2. In this block, the variables from CarSim are passed to different subsystems which will be addressed in the proceeding sections. Also, the braking torques four each wheel along with throttle and road wheel angle are exported CarSim (if turned on with the manual switches seen in Figure B.2).

The scopes and displays are for the user's convenience and are used to display vehicle parameters and export them to the MATLAB workspace to document later. The variables listed in order from the top to bottom are: time, lateral acceleration (Ay), steering wheel angle (Steer_SW), throttle, longitudinal velocity (Vx), target velocity (VxTarget), yaw angle (yaw), yaw rate (AVz), and transmission gear selected (gear). Below these seven signals are the displays and scopes for each of the for wheel's braking torques.

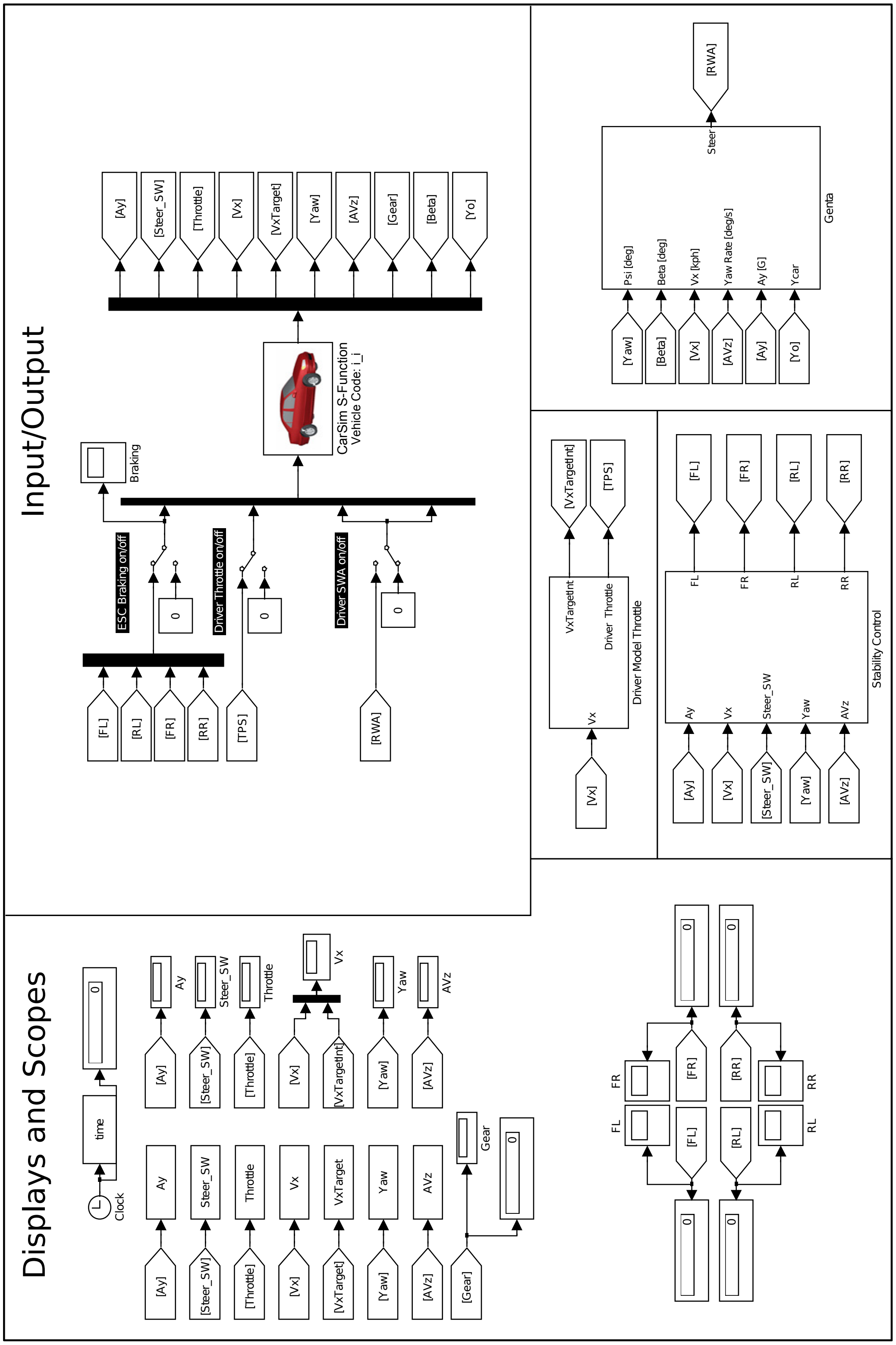


Figure B.1: Home Screen.

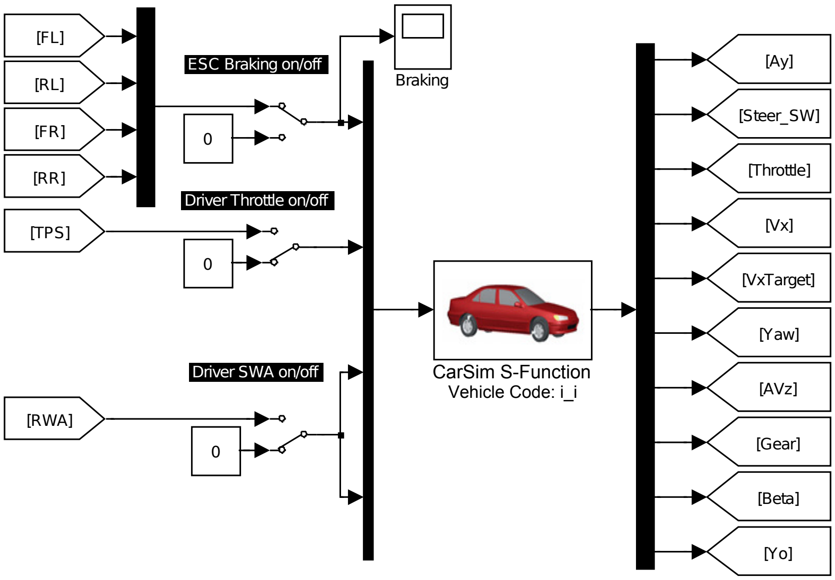


Figure B.2: Input/Output Main Block.

B.1 Stability Control Subsystem

Figure B.3 shows an overview of the stability control algorithm with its inputs and outputs as seen on the home screen. The details of the block can be seen in Figure B.4. In the stability control subsystem, there are three main subsystems: oversteer indicator, oversteer hold subsystem, and braking control. The oversteer indicator block can be seen in Figure B.5 and is discussed in detail in Chapter 3.

B.1.1 Oversteer Hold

The oversteer hold subsystem can be seen in Figure B.6 and is located in the stability control subsystem. It is discussed in detail in Section 3.5.2. Associated with this block is the code for how to deal with the held oversteer number. Based on the inputs of the current indicated oversteer number (OS), the rate of change of the oversteer number (dOS), the number held by the sample and hold block if the signal starts to fall (OShold), and the amount of time the signal is held, it will adjust the oversteer number passed to the braking control block.

B.1.1.1 Embedded MATLAB Code: OS Hold

```
function newOS = fcn(OS,dOS,OShold,HoldTime)
```

```
newOS = OS;
if dOS < 0
    if HoldTime < 5
        newOS = OShold;
    else
        newOS = OS;
    end
elseif dOS >= 0 && OS > OShold
    newOS = OS;
elseif dOS >= 0 && OS < OShold && HoldTime < 5
    newOS = OShold;
else
    newOS = OS;
end
```

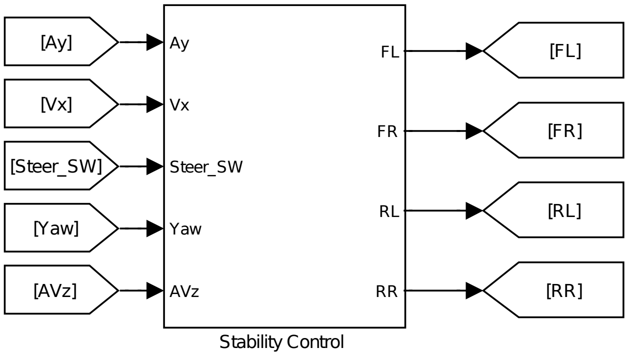


Figure B.3: Stability Control Block Subsystem.

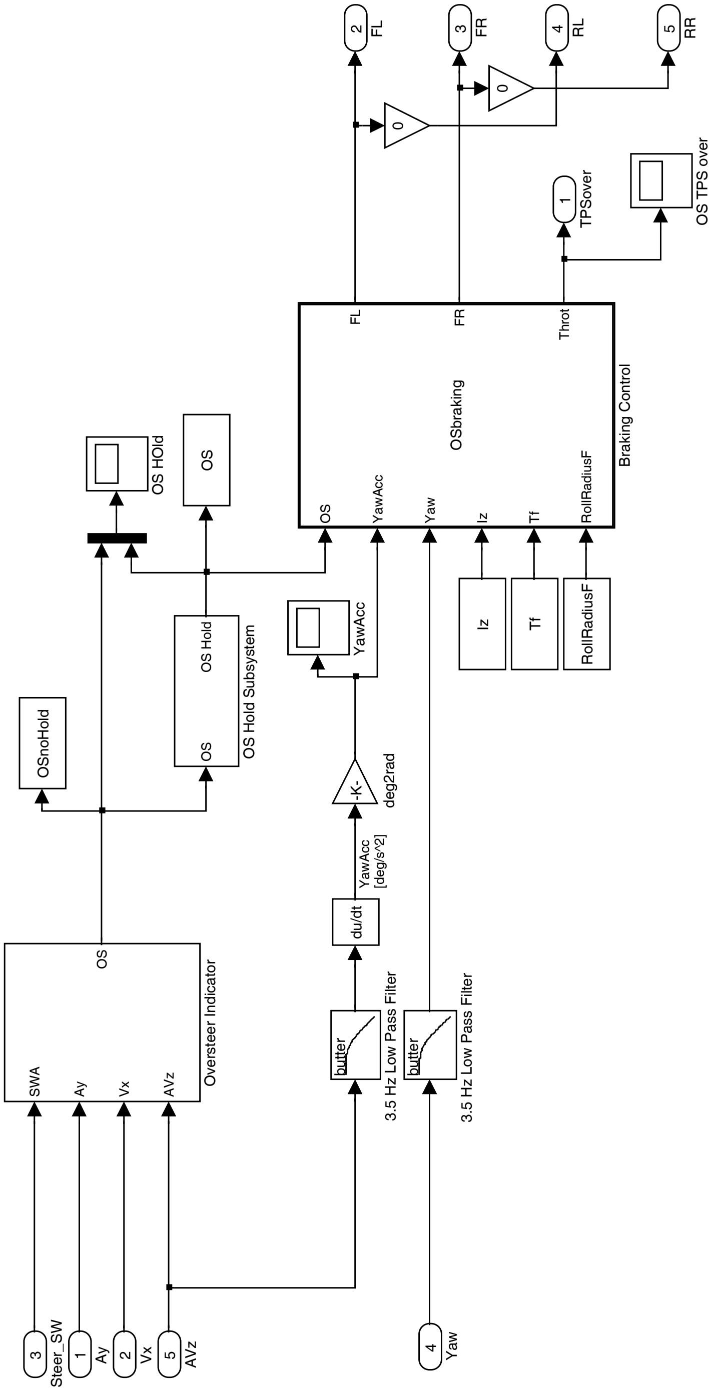


Figure B.4: Stability Control Subsystem Block.

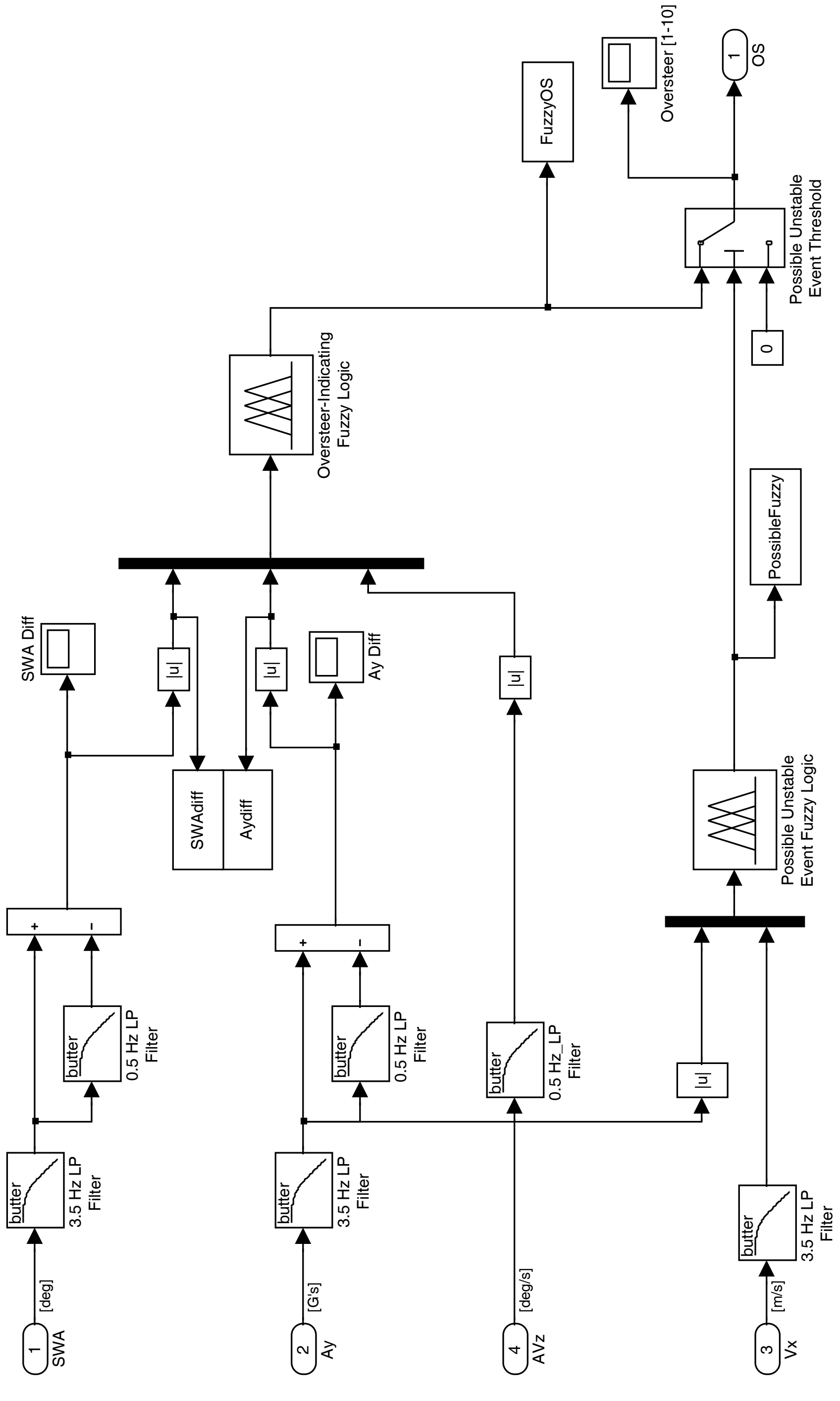


Figure B.5: Oversteer Indicator.

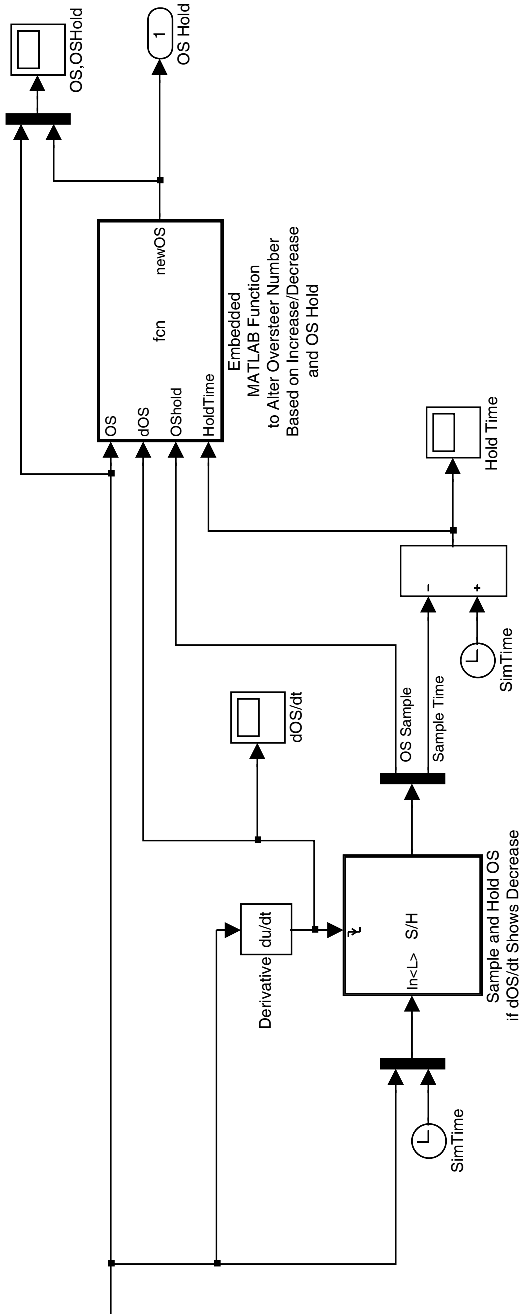


Figure B.6: Oversteer Hold Block.

B.1.2 Braking Control Block

This block is in the stability control subsystem and is an embedded MATLAB function. Below is the code found in this block. Basically, it takes the current level of oversteer and breaks the control action into three levels: no control, moderate control, and heavy control (discussed in Section 3.6). From here, it will determine whether to brake the left or right side of the vehicle based on the yaw angle.

```
function [FL,FR,Throt]  = OSbraking(OS,YawAcc,...
Yaw,Iz,Tf,RollRadiusF)
FR=0;
FL=0;
brake=0;
% Determines Amount of Braking
    if OS <= 2
        brake = 0;
    elseif OS >2 && OS <=5.5
        brake = abs(Iz*YawAcc*2/Tf*RollRadiusF)
    elseif OS > 5.5
         brake = 2500;
    end
% Determines the Side of Vehicle To Brake
    if Yaw > 0
        FR = brake;
    elseif Yaw < 0
        FL = brake;
    else
        FL = brake;
        FR = brake;
    end
```

B.2 Genta Driver Model

The Genta driver model as seen on the home screen is shown in Figure B.7. The details of this model can be seen in Figure B.8. This driver model is discussed in detail in Section 5.2.2.2.

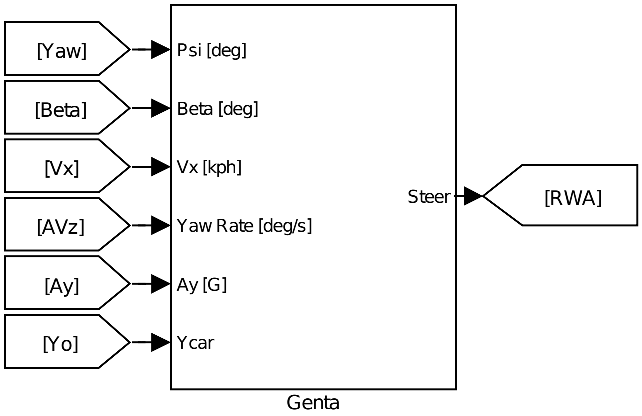


Figure B.7: Genta Main Diagram.

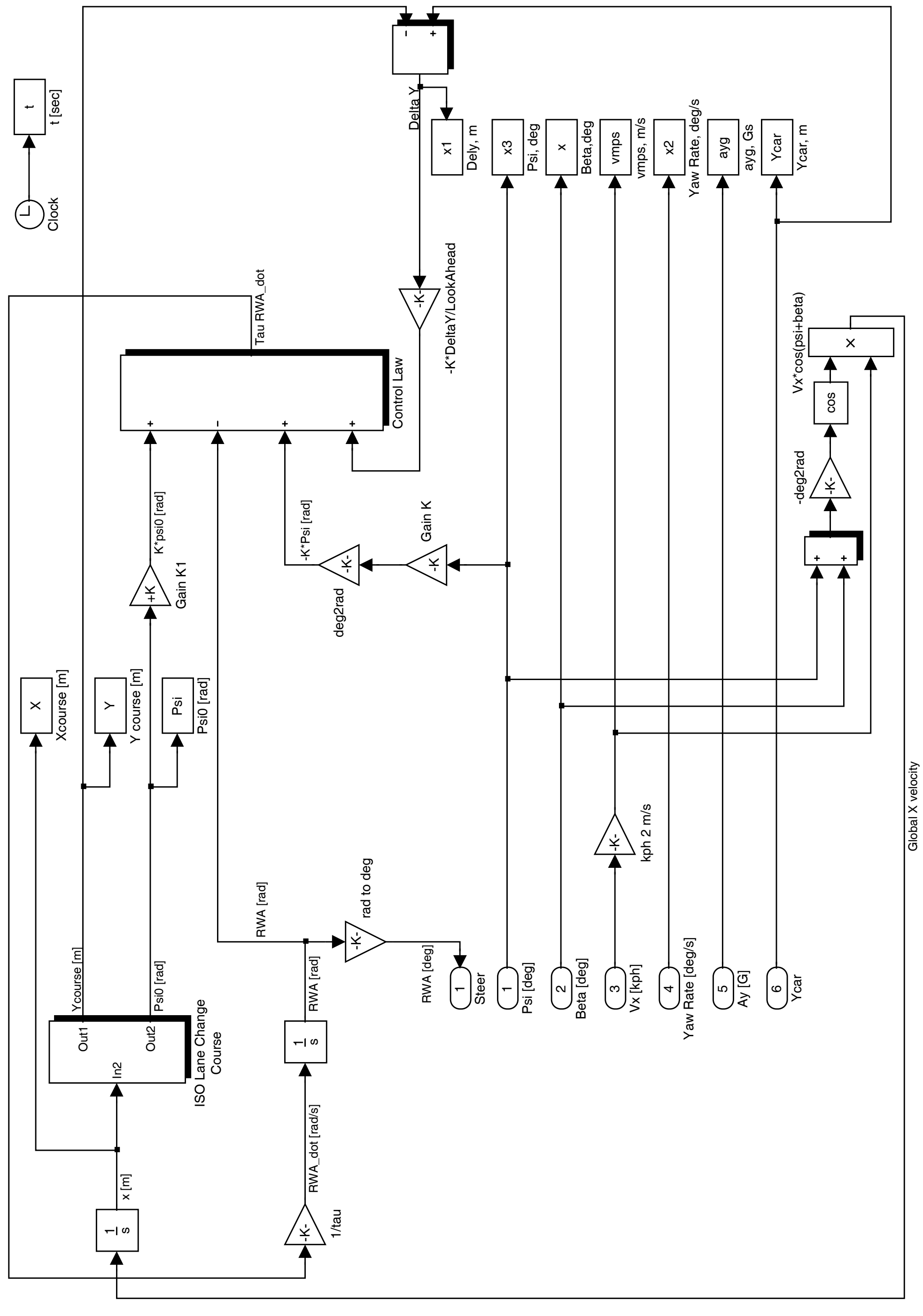


Figure B.8: Genta Block Diagram.

B.3 Driver Throttle

This block is used only in Case 07 to ramp the velocity for the understeer. It uses a simple proportional gain (0.08) that acts on the difference in the current velocity and the target velocity. The difference in velocities is multiplied by the gain to determine throttle position. Then, that signal is passed through a saturation block to force the throttle position to be within 0 and 100 %. An overview as seen on the home screen can be seen in Figure B.9 and the details illustrating the proportional control can be seen in Figure B.10.

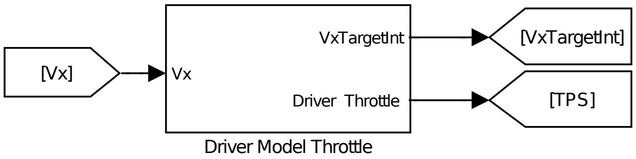


Figure B.9: Driver Throttle Overview.

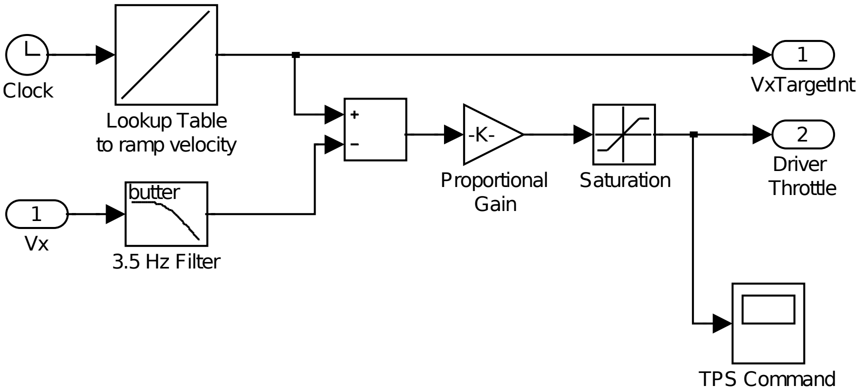


Figure B.10: Driver Throttle.

Appendix C

TIRE DATA

In this section, the tire data for each of the cars will be presented. The BMW Mini used the OE Goodyear tires described by Figures C.1 and C.2. Figure C.3 shows the lateral force versus slip angle of the tire for the cases with degraded rear tires. The CarSim sports car and sedan both used generic, small car longitudinal and lateral force curves as seen in Figures C.4 - C.5. The CarSim SUV used generic large vehicle handling curves seen in Figures C.6 and C.7. For the case with the degraded rear tires, the lateral force curve that was used can be seen in Figure C.8.

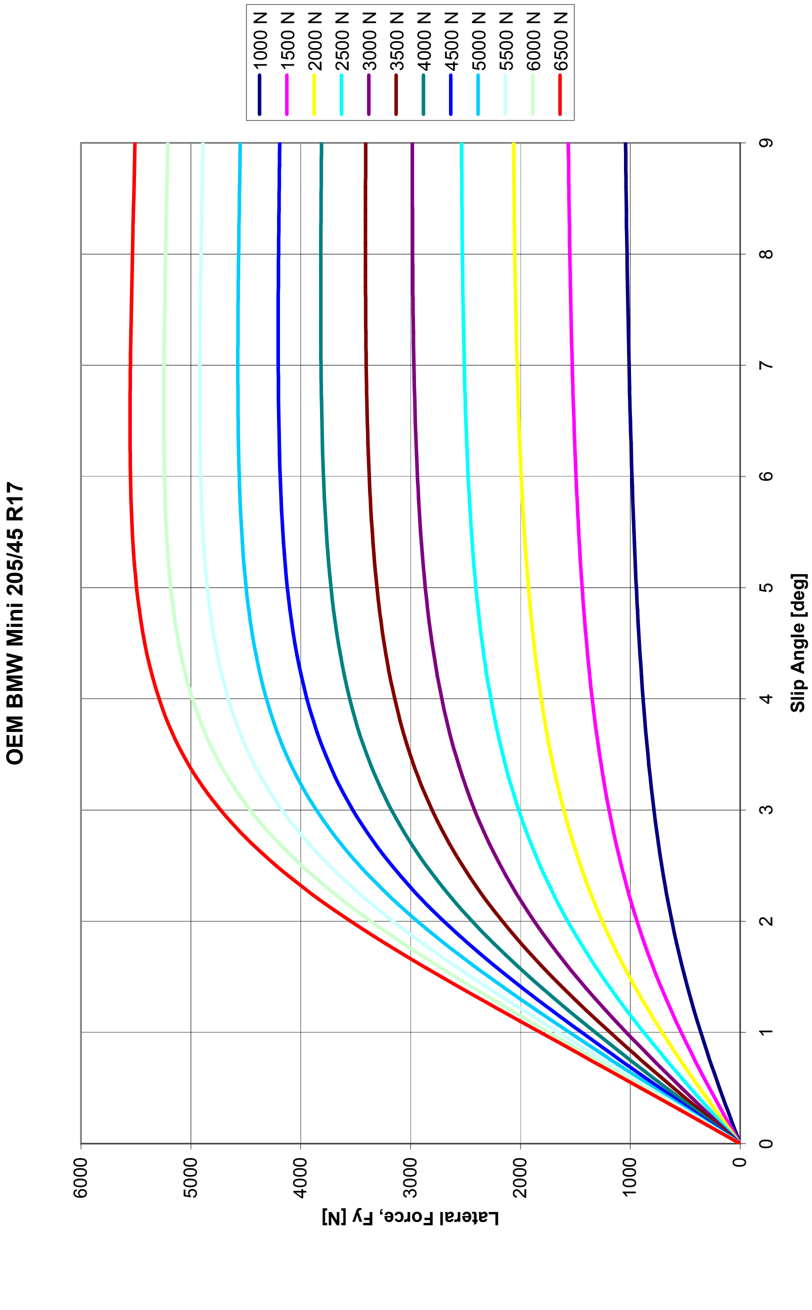


Figure C.1: Lateral Force vs. Slip Angle (BMW Mini).

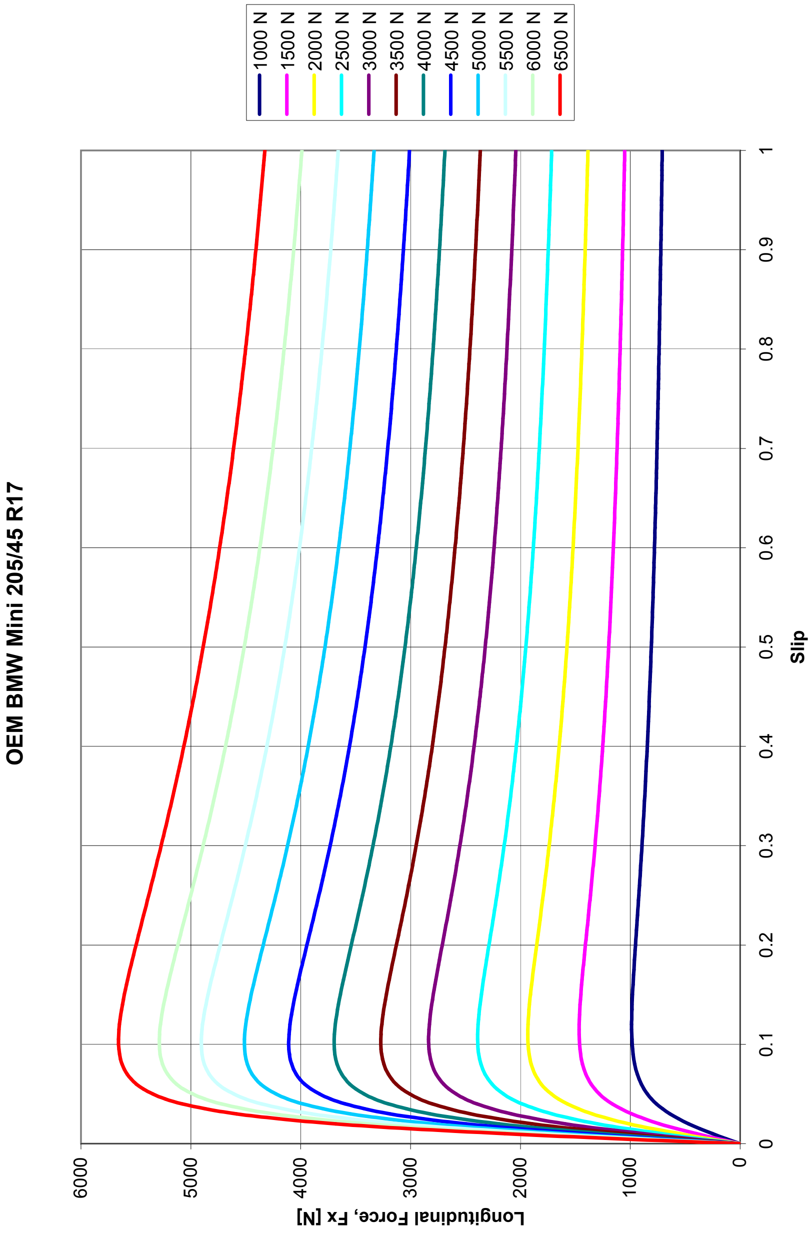


Figure C.2: Longitudinal Force vs. Slip (BMW Mini).

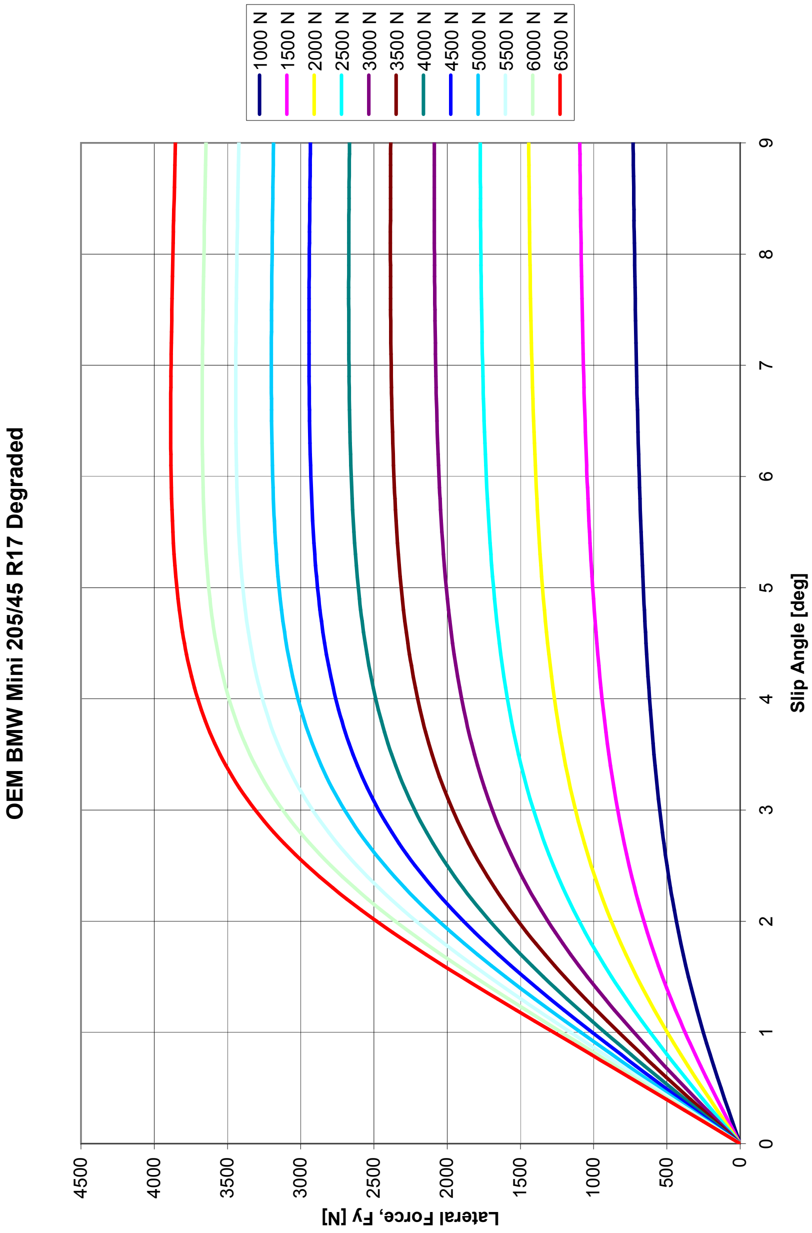


Figure C.3: Lateral Force vs. Slip Angle (BMW Mini - Degraded Tires).

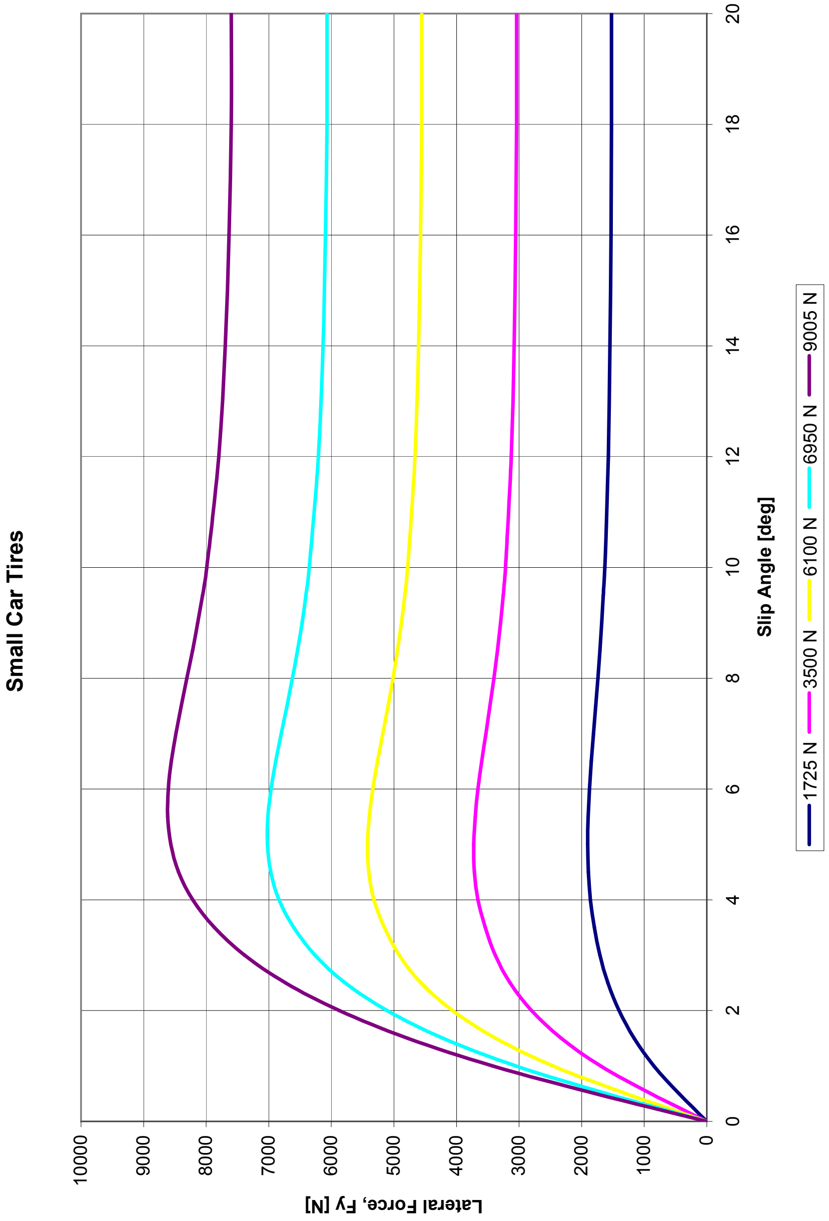


Figure C.4: Lateral Force vs. Slip Angle (CarSim - Sports Car and Sedan).

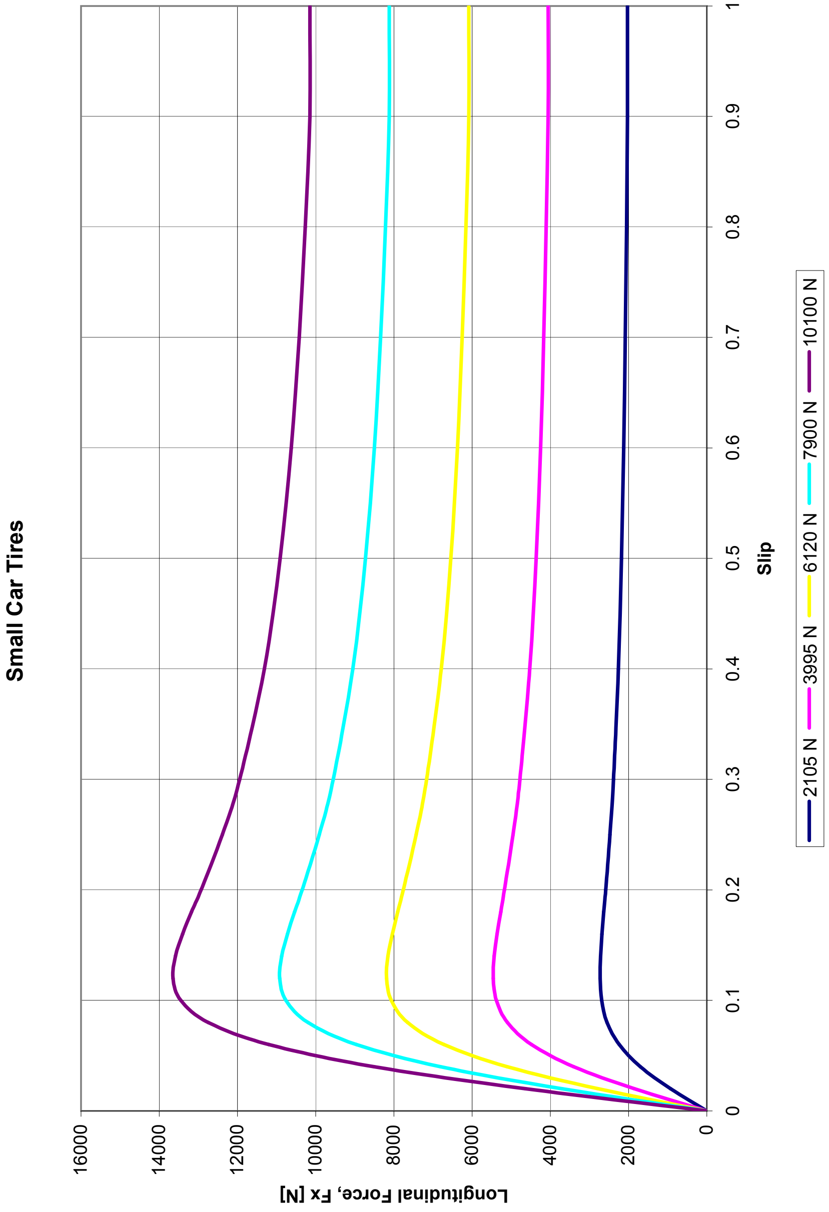


Figure C.5: Longitudinal Force vs. Slip (CarSim - Sports Car and Sedan).

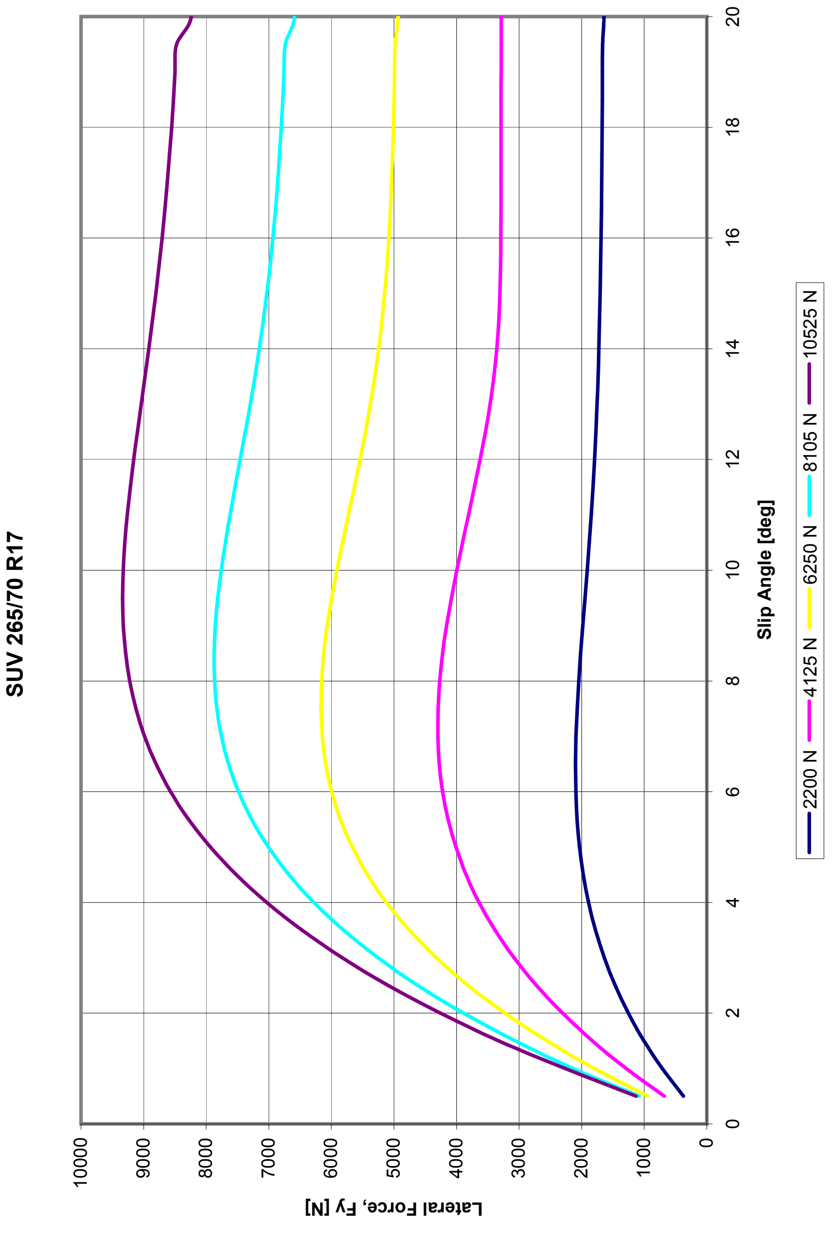


Figure C.6: Lateral Force vs. Slip Angle (CarSim SUV).

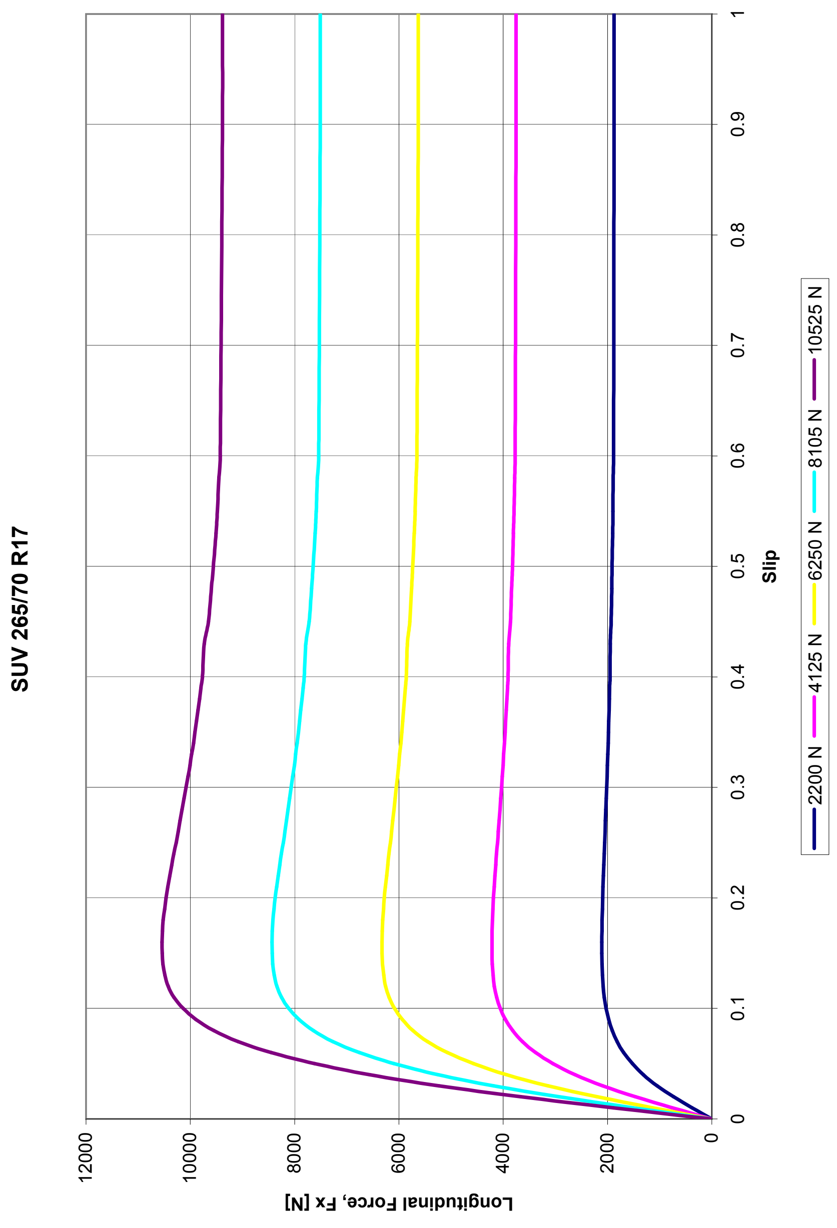


Figure C.7: Longitudinal Force vs. Slip (CarSim SUV).

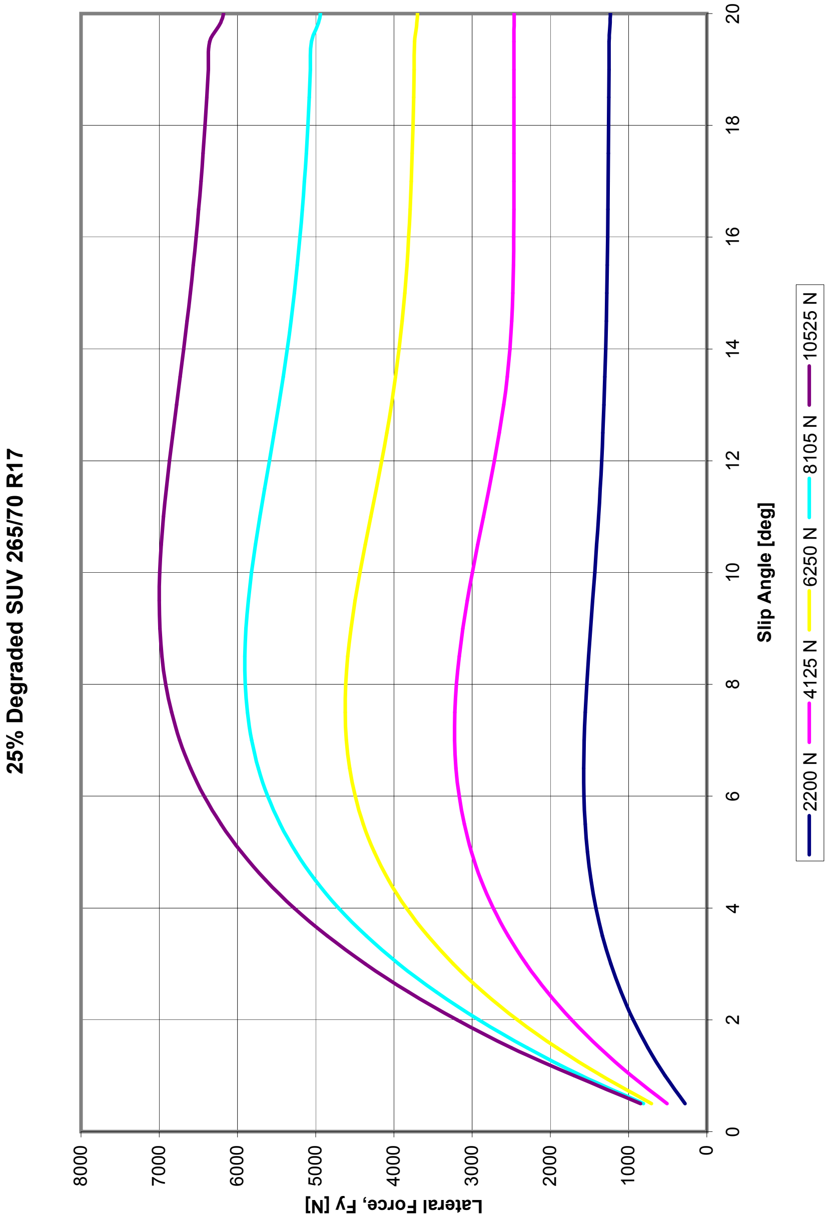


Figure C.8: Lateral Force vs. Slip Angle (CarSim SUV - Degraded Tires).

Appendix D

DATA COMPILE

In this appendix, the code used to compile the data from the simulations is presented. This is a sample code of the code to compile Case 01 and is similar to the code used to compile the rest of the case studies. To start, the ERD File Converter in CarSim was used to convert the .bin files to .mat files in order to bring the variables into MATLAB. Once each run was converted to a .mat, a simple load command could be used to load the variables and generate the appropriate figures.

```
clc
clear all
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CASE 01 - Nominal BMW MINI DLC: No ESC, Fuzzy ESC, CarSim ESC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Vehicle Parameters
    A = 0.9872;     % Distance from front axle to CG [m]
    B = 1.4808;     % Distance from rear axle to CG [m]
    Tf = 1.453;     % Front Track [m]
    Tr = 1.475;     % Rear Track [m]
    reff_f = 0.29;  % Rolling Radius Front [m]
    reff_r = 0.293; % Rolling Radius Rear [m]
%%% 185 kph DLC no ESC Run 389
```

```
a =1;
    load Run389.mat
            Ay_(:,a) = Ay;
            SWA_(:,a) = Steer_SW;
            Beta_(:,a) = Beta;
            Time_(:,a) = Time;
            AVz_(:,a) = AVz;
            Xo_(:,a) = Xo;
            Yo_(:,a) = Yo;
            X_Target_(:,a) = X_Target;
            Y_Target_(:,a) = Y_Target;
            TimeCarSim_(:,a) = Time;
            FLBrake_(:,a) = My_Bk_L1;
            FRBrake_(:,a) = My_Bk_R1;
            RLBrake_(:,a) = My_Bk_L2;
            RRBrake_(:,a) = My_Bk_R2;
            Fx_L1_(:,a) = Fx_L1;
            Fx_L2_(:,a) = Fx_L2;
            Fx_R1_(:,a) = Fx_R1;
            Fx_R2_(:,a) = Fx_R2;
            Fy_L1_(:,a) = Fy_L1;
            Fy_L2_(:,a) = Fy_L2;
            Fy_R1_(:,a) = Fy_R1;
            Fy_R2_(:,a) = Fy_R2;
            Fz_L1_(:,a) = Fz_L1;
            Fz_L2_(:,a) = Fz_L2;
            Fz_R1_(:,a) = Fz_R1;
            Fz_R2_(:,a) = Fz_R2;

%%% 185 kph DLC ESC ON Run 423
a =2;
```

```
load Run423.mat
        Ay_(:,a) = Ay;
        SWA_(:,a) = Steer_SW;
        Beta_(:,a) = Beta;
        Time_(:,a) = Time;
        AVz_(:,a) = AVz;
        Xo_(:,a) = Xo;
        Yo_(:,a) = Yo;
        X_Target_(:,a) = X_Target;
        Y_Target_(:,a) = Y_Target;
        TimeCarSim_(:,a) = Time;
        FLBrake_(:,a) = My_Bk_L1;
        FRBrake_(:,a) = My_Bk_R1;
        RLBrake_(:,a) = My_Bk_L2;
        RRBrake_(:,a) = My_Bk_R2;
        Fx_L1_(:,a) = Fx_L1;
        Fx_L2_(:,a) = Fx_L2;
        Fx_R1_(:,a) = Fx_R1;
        Fx_R2_(:,a) = Fx_R2;
        Fy_L1_(:,a) = Fy_L1;
        Fy_L2_(:,a) = Fy_L2;
        Fy_R1_(:,a) = Fy_R1;
        Fy_R2_(:,a) = Fy_R2;
        Fz_L1_(:,a) = Fz_L1;
        Fz_L2_(:,a) = Fz_L2;
        Fz_R1_(:,a) = Fz_R1;
        Fz_R2_(:,a) = Fz_R2;
    load FuzzyOutput423.mat
        FuzzyOS_(:,a) = FuzzyOS;
        OS_(:,a) = OS;
        OSnoHold_(:,a) = OSnoHold;
```

```
            PossibleFuzzy_(:,a) = PossibleFuzzy;
            SWAdiff_(:,a) = SWAdiff;
            Aydiff_(:,a) = Aydiff;
            TimeSimulink_(:,a) = time;
%%% 185 kph DLC ESC ON TRAD ESC Run
a =3;
    load Run399.mat
            Ay_(:,a) = Ay;
            SWA_(:,a) = Steer_SW;
            Beta_(:,a) = Beta;
            Time_(:,a) = Time;
            AVz_(:,a) = AVz;
            Xo_(:,a) = Xo;
            Yo_(:,a) = Yo;
            X_Target_(:,a) = X_Target;
            Y_Target_(:,a) = Y_Target;
            TimeCarSim_(:,a) = Time;
            FLBrake_(:,a) = My_Bk_L1;
            FRBrake_(:,a) = My_Bk_R1;
            RLBrake_(:,a) = My_Bk_L2;
            RRBrake_(:,a) = My_Bk_R2;
            Fx_L1_(:,a) = Fx_L1;
            Fx_L2_(:,a) = Fx_L2;
            Fx_R1_(:,a) = Fx_R1;
            Fx_R2_(:,a) = Fx_R2;
            Fy_L1_(:,a) = Fy_L1;
            Fy_L2_(:,a) = Fy_L2;
            Fy_R1_(:,a) = Fy_R1;
            Fy_R2_(:,a) = Fy_R2;
            Fz_L1_(:,a) = Fz_L1;
            Fz_L2_(:,a) = Fz_L2;
            Fz_R1_(:,a) = Fz_R1;
            Fz_R2_(:,a) = Fz_R2;

%% Trajectory
    figure
    rows = 2;
    cols = 1;
    subplot(rows,cols,1),plot(X_Target_(:,1),Y_Target_(:,1),...
         ’--k’, Xo_(:,1),Yo_(:,1),...
            Xo_(:,2),Yo_(:,2),...
            Xo_(:,3),Yo_(:,3),...
            ’linewidth’,1.5’)
        hold on
        load Cones.csv
    plot(Cones(:,1),Cones(:,2),’o’,’markersize’,10,...
        ’markerfacecolor’,’y’,’linewidth’,1.5,...
   ’MarkerEdgeColor’,’k’)
        xlim([-5 300])
        ylim([-2.5 6])
        grid
        xlabel(’x position [m]’)
        ylabel(’y position [m]’)
    subplot(rows,cols,2),...
    plot(TimeCarSim_(:,1),Y_Target_(:,1),...
         ’--k’,TimeCarSim_(:,1),Yo_(:,1),...
       TimeCarSim_(:,2),Yo_(:,2),...
               TimeCarSim_(:,3),Yo_(:,3),...
                ’linewidth’,1.5’)
        legend(’Target Path’,’ESC Off’, ’ESC On’,...
            ’Traditional ESC’,’location’,’northeast’)
        grid
        xlabel(’Time [s]’)
        ylabel(’y position [m]’)
    filename = ’ESCOnOffTrajectoryCASE01’;
    print(filename,’-depsc2’)

%% Vehicle Dynamic Traces
    close all
    figure
    rows = 4;
    cols = 1;
    subplot(rows,cols,1),plot(TimeCarSim_(:,1),SWA_(:,1),...
           TimeCarSim_(:,2),SWA_(:,2),...
  TimeCarSim_(:,3),SWA_(:,3),...
           ’linewidth’,1.5);
        grid
        ylabel(’SWA [deg]’)
        ylim([-35 35])
        legend(’ESC Off’,’ESC On’,’Traditional ESC’,...
             ’location’,’northeast’)
    subplot(rows,cols,2),plot(TimeCarSim_(:,1),Ay_(:,1),...
           TimeCarSim_(:,2),Ay_(:,2),...
  TimeCarSim_(:,3),Ay_(:,3),...
          ’linewidth’,1.5);
        grid
        ylabel(’Ay [Gs]’)
        ylim([-.9 .9])
    subplot(rows,cols,3),plot(TimeCarSim_(:,1),AVz_(:,1),...
            TimeCarSim_(:,2),AVz_(:,2),...
   TimeCarSim_(:,3),AVz_(:,3),...
            ’linewidth’,1.5);
        grid
        ylabel(’AVz [deg/s]’)
        ylim([-20 20])
    subplot(rows,cols,4),plot(TimeCarSim_(:,1),Beta_(:,1),...
          TimeCarSim_(:,2),Beta_(:,2),...
  TimeCarSim_(:,3),Beta_(:,3),...
           ’linewidth’,1.5);
        grid
        ylabel(’\beta [deg]’)
        ylim([-5 5])
        xlabel(’Time [s]’)
    filename = ’ESCOnOffVehicleDyanmicsCASE01’;
    print(filename,’-depsc2’)
%% Fuzzy Inputs and OS out
    figure
    rows = 4;
    cols = 1;
    subplot(rows,cols,1),..
    plot(TimeSimulink_(:,1),SWAdiff_(:,1),...
         ’linewidth’,1.5);
        grid
        ylabel(’SWA [deg]’)
        ylim([0 150])
    subplot(rows,cols,2),...
    plot(TimeSimulink_(:,1),Aydiff_(:,1),...
         ’linewidth’,1.5);
        grid
        ylabel(’Ay [Gs]’)
        ylim([0 .8])
    subplot(rows,cols,3),...
    plot(TimeCarSim_(:,1),abs(AVz_(:,1)),..
         ’linewidth’,1.5);
        grid
        ylabel('AVz [deg/s]')
        ylim([0 100])
    subplot(rows,cols,4),...
    plot(TimeSimulink_(:,1),FuzzyOS_(:,1),...
         'linewidth',1.5);
        grid
        ylabel('OS')
        xlabel('Time [s]')
    filename = 'FuzzyInput105kphESCoff';
    print(filename,'-depsc2')
%% Possible Unstalbe Event
    figure
    plot(TimeSimulink_(:,1),PossibleFuzzy_(:,1),...
            [0 8],[4 4],'k','linewidth',1.5)
    grid
    ylabel('Possible Unstable Event')
    xlabel('Time [s]')
    filename = 'PossibleFuzzy105kphESCoff';
    print(filename,'-depsc2')
%% OS affected by possible
    figure
    plot(TimeSimulink_(:,1),FuzzyOS_(:,1),...
            TimeSimulink_(:,1),OSnoHold_(:,1),...
            'linewidth',1.5)
    legend('OS Number from Fuzzy Logic',...
        'OS Number After Possible Unstable Event Threshold',...
        'location','northwest')
    grid
    ylabel('Oversteer Number')
    xlabel('Time [s]')
    ylim([0 10.3])
    filename = ’PossibleOSAffect’;
    print(filename,’-depsc2’)
%% OS Hold
    figure
    plot(TimeSimulink_(:,1),OSnoHold_(:,1),...
            TimeSimulink_(:,1),OS_(:,1),..
            ’linewidth’,1.5)
    legend(’OS Number no Hold’,’OS Number after Hold’,...
        ’location’,’northwest’)
    grid
    ylabel(’Oversteer Number’)
    xlabel(’Time [s]’)
    filename = ’OSHold105kphESCoff’;
    print(filename,’-depsc2’)
%% Commanded Braking
    figure
    rows = 5;
    cols =1;
    subplot(rows,cols,1),plot(TimeCarSim_(:,1),Ay_(:,1),...
         TimeCarSim_(:,2),Ay_(:,2),’linewidth’,1.5);
        grid
        ylabel(’Ay [G]’)
        ylim([-1.5 1])
        legend(’105 kph ESC Off’,’105 kph ESC On’,’location’,...
            ’southwest’)
    subplot(rows,cols,2),plot(TimeCarSim_(:,1),AVz_(:,1),...
         TimeCarSim_(:,2),AVz_(:,2),’linewidth’,1.5);
        grid
        ylabel(’AVz [deg/s]’)
        ylim([-20 20])
    subplot(rows,cols,3),plot(TimeCarSim_(:,1),Beta_(:,1),...
         TimeCarSim_(:,2),Beta_(:,2),’linewidth’,1.5);
        grid
        ylabel(’\beta [deg]’)
        ylim([-5 5])
    subplot(rows,cols,4),plot(TimeSimulink_(:,1),OS_(:,1),...
         TimeSimulink_(:,2),OS_(:,2),’linewidth’,1.5);
        grid
        ylabel(’OS’)
        ylim([0 10])
    subplot(rows,cols,5),...
            plot(TimeCarSim_(:,2),FLBrake_(:,2),...
          ’m’,TimeCarSim_(:,2),FRBrake_(:,2),...
     ’g’,’linewidth’,1.5)
        grid
        ylabel(’Brake Torque [N*m]’)
        xlabel(’Time [s]’)
    legend(’Front Left’,’Front Right’,’location’,’southwest’)
    filename = ’ESCOnOffBraking105kphDLC’;
    print(filename,’-depsc2’)
%% Commanded Braking Trad/Fuzzy
    figure
    rows = 2;
    cols =1;
    subplot(rows,cols,1),...
    plot(TimeCarSim_(:,2),FLBrake_(:,2),’m’,...
            TimeCarSim_(:,2),FRBrake_(:,2),’g’,’linewidth’,1.5)
        grid
        ylabel(’Brake Torque [N*m]’)
        xlabel(’Time [s]’)
        title(’Fuzzy ESC’)
     legend(’Front Left’,’Front Right’,’location’,’northeast’)
     subplot(rows,cols,2),...
     plot(TimeCarSim_(:,3),FLBrake_(:,3),’m’,...
             TimeCarSim_(:,3),FRBrake_(:,3),’g’,...
             TimeCarSim_(:,3),RLBrake_(:,3),
             TimeCarSim_(:,3),RRBrake_(:,3),...
             ’linewidth’,1.5)
        grid
        ylabel(’Brake Torque [N*m]’)
        xlabel(’Time [s]’)
        title(’CarSim ESC’)
        legend(’Front Left’,’Front Right’,...
                      ’Rear Left’,’Rear Right’,...
                      ’location’,’northeast’)
    filename = ’BrakeTorqueCASE01’;
    print(filename,’-depsc2’)
%% Tire Forces
    figure
    rows = 2;
    cols = 2;
    subplot(rows,cols,1),plot(TimeCarSim_(:,1),Fx_L1_(:,1),...
                              TimeCarSim_(:,2),Fx_L1_(:,2),...
                              TimeCarSim_(:,3),Fx_L1_(:,3),...
                              ’linewidth’,1.5)
        ylabel(’Front Left [N]’)
        xlabel(’Time [s]’)
        grid
        ylim([-2700 0])
    subplot(rows,cols,2),plot(TimeCarSim_(:,1),Fx_R1_(:,1),...
                              TimeCarSim_(:,2),Fx_R1_(:,2),...
                              TimeCarSim_(:,3),Fx_R1_(:,3),...
                              ’linewidth’,1.5)
        grid
        ylabel(’Front Right [N]’)
        xlabel(’Time [s]’)
        ylim([-2700 0])
    subplot(rows,cols,3),plot(TimeCarSim_(:,1),Fx_L2_(:,1),...
                              TimeCarSim_(:,2),Fx_L2_(:,2),...
                              TimeCarSim_(:,3),Fx_L2_(:,3),...
                              ’linewidth’,1.5)
        grid
        ylabel(’Rear Left [N]’)
        xlabel(’Time [s]’)
        ylim([-550 0])
     subplot(rows,cols,4),plot(TimeCarSim_(:,1),Fx_R2_(:,1),...
                               TimeCarSim_(:,2),Fx_R2_(:,2),...
                               TimeCarSim_(:,3),Fx_R2_(:,3),...
                               ’linewidth’,1.5)
        grid
        ylabel(’Rear Right [N]’)
        xlabel(’Time [s]’)
        ylim([-550 0])
        [ax,h3]=suplabel(’Tire Longitudnal Forces, Fx’,’t’);
        set(h3,’FontSize’,15)
        legend(’ESC Off’, ’ESC On’,’CarSim ESC’,’location’,...
            ’southeast’)
    filename = ’TireFxCASE01’;
    print(filename,’-depsc2’)
%% Tire Forces Fy
    figure
    rows = 2;
    cols = 2;
```

```
subplot(rows,cols,1),plot(TimeCarSim_(:,1),Fy_L1_(:,1),...
                          TimeCarSim_(:,2),Fy_L1_(:,2),...
                          TimeCarSim_(:,3),Fy_L1_(:,3),...
                          ’linewidth’,1.5)
    ylabel(’Front Left [N]’)
    xlabel(’Time [s]’)
    grid
    ylim([-5000 5000])
subplot(rows,cols,2),plot(TimeCarSim_(:,1),Fy_R1_(:,1),...
                          TimeCarSim_(:,2),Fy_R1_(:,2),...
                          TimeCarSim_(:,3),Fy_R1_(:,3),...
                          ’linewidth’,1.5)
    grid
    ylabel(’Front Right [N]’)
    xlabel(’Time [s]’)
    ylim([-5000 5000])
subplot(rows,cols,3),plot(TimeCarSim_(:,1),Fy_L2_(:,1),...
                          TimeCarSim_(:,2),Fy_L2_(:,2),...
                          TimeCarSim_(:,3),Fy_L2_(:,3),...
                          ’linewidth’,1.5)
    grid
    ylabel(’Rear Left [N]’)
    xlabel(’Time [s]’)
    ylim([-5000 5000])
 subplot(rows,cols,4),plot(TimeCarSim_(:,1),Fy_R2_(:,1),...
                           TimeCarSim_(:,2),Fy_R2_(:,2),...
                           TimeCarSim_(:,3),Fy_R2_(:,3),...
                           ’linewidth’,1.5)
    grid
    ylabel(’Rear Right [N]’)
    xlabel(’Time [s]’)
```

```
        ylim([-5000 5000])
        [ax,h3]=suplabel('Tire Lateral Forces, Fy','t');
        set(h3,'FontSize',15)
        legend('ESC Off', 'ESC On','CarSim ESC','location',...
            'northeast')
    filename = 'TireFyCASE01';
    print(filename,'-depsc2')
%% Braking Power
close all
clc
% Brake Force
    Fb_L1_ = FLBrake_./reff_f;
    Fb_R1_ = FRBrake_./reff_f;
    Fb_L2_ = RLBrake_./reff_r;
    Fb_R2_ = RRBrake_./reff_r;
% Brake Yaw Moment
    BrakeYawMom =  -Fb_L1_.*(Tf/2)...
             +Fb_R1_.*(Tf/2)...
                                 - Fb_L2_.*(Tr/2)...
                                 +Fb_R2_.*(Tr/2);
    AVzRAD_S = AVz_.*(pi/180);
    ESCPower = abs(AVzRAD_S.*BrakeYawMom);
    ESCPowerHP = (1/745.7).*ESCPower; % Converts W to hp
    figure
    [haxes,hline1,hline2]=   plotyy(TimeCarSim_(:,2),...
                                            ESCPower(:,2),...
        TimeCarSim_(:,2),ESCPowerHP(:,2),'plot','plot','r',...
        'linewidth',1.5,'linewidth',1.5);
    set(hline1,'color','g','linewidth',1.5)
    set(hline1,'color','g','linewidth',1.5)
    axes(haxes(1))
```

```
        ylabel(’Power [W]’)
        ylim([0 300])
        grid
        hold on
            plot(TimeCarSim_(:,3),ESCPower(:,3),...
             ’r’,’linewidth’,1.5)
            legend(’Fuzzy ESC’,’CarSim ESC’,...
             ’location’,’northeast’)
        hold off
    axes(haxes(2))
        ylabel(’Power [hp]’)
        ylim([0 0.402])
        xlabel(’Time [s]’)
        grid
        filename = ’ESCPowerCASE01’;
        print(filename,’-depsc2’)
        ESCnrg = [trapz(TimeCarSim_(:,1),ESCPower(:,1)),...
            trapz(TimeCarSim_(:,2),ESCPower(:,2)),...
            trapz(TimeCarSim_(:,3),ESCPower(:,3))]
%% Yaw moment
close all
clc
Fb_L1_ = FLBrake_./reff_f;
Fb_R1_ = FRBrake_./reff_f;
Fb_L2_ = RLBrake_./reff_r;
Fb_R2_ = RRBrake_./reff_r;
% Brake yaw moment
BrakeYawMom = ....
-Fb_L1_.*(Tf/2) + Fb_R1_.*(Tf/2) ...
- Fb_L2_.*(Tr/2) + Fb_R2_.*(Tr/2);
%%% Sum Forces Long and Lat about CG
```

```
TotYawMom = ...
 -Fx_L1_.*(Tf/2) + Fx_R1_.*(Tf/2)...
- Fx_L2_.*(Tr/2) + Fx_R2_.*(Tr/2)...
+Fy_L1_.*A        + Fy_R1_.*A...
- Fy_L2_.*B         - Fy_R2_.*B;
remainder = TotYawMom - BrakeYawMom;
figure
rows = 2;
cols = 1;
subplot(rows,cols,1),...
     plot(TimeCarSim_(:,2),BrakeYawMom(:,2),...
                 TimeCarSim_(:,2),remainder(:,2),...
                 TimeCarSim_(:,2),TotYawMom(:,2),...
                 TimeCarSim_(:,1),TotYawMom(:,1),...
                 ’linewidth’,1.5)
    legend(’Braking’,’Remainder’,’Total’,’No ESC’)
    xlabel(’Time [s]’)
    ylabel(’Yaw Moment [N*m]’)
    grid
    title(’Fuzzy ESC’)
subplot(rows,cols,2),...
    plot(TimeCarSim_(:,3),BrakeYawMom(:,3),...
                TimeCarSim_(:,3),remainder(:,3),...
                TimeCarSim_(:,3),TotYawMom(:,3),...
                TimeCarSim_(:,1),TotYawMom(:,1),...
                 ’linewidth’,1.5)
    legend(’Braking’,’Remainder’,’Total’,’No ESC’)
    xlabel(’Time [s]’)
    ylabel(’Yaw Moment [N*m]’)
    grid
    title(’CarSim ESC’)
```

```
        filename = ’YawMomentCASE01’;
print(filename,’-depsc2’)
```

Bibliography

[1] Department of Transportation National Highway Traffic Saftey Administration. *Federal Motor Vehicle Saftey Standards; Electronic Stability Control Systems; Controls and Displays*, 2007.

[2] John Limroth. *Real-Time Vehicle Parameter Estimation and Adaptive Stability Control*. PhD thesis, Clemson University, December 2009.

[3] E.H. Law. Me 453/653 class notes. Clemson University, Fall 2007.

[4] T. Rhyne. ”flat track test pacejka data for bmw mini oem tires”, test ranking no. j746030. Michelin America Research Center, April 2007.

[5] Buddy Fey. *Data Power Using Racecar Data Acquisition*. Towery Publishing, 1993.

[6] Sunder Vaduri. *Development of Computer Tools for Analysis of Track Test Data and for Prediction of Dynamic Handling Response for Winston Cup Cars*. PhD thesis, Clemson University, 1999.

[7] Ronald Yager and Lofti Zadeh. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Kluwer Academic Publishers, 1992.

[8] Lotfi Zadeh. *Fuzzy Logic Toolbox Forward*. The MathWorks, Inc., Natick, MA, 1995.

[9] Anon. ”carsim 8 math models”, June 2009.

[10] E. H. Law and Sunder Vaduri. Development of an expert system for the analysis of track test data. *SAE Paper 2000-01-1628*, 2000.

[11] E.H. Law. *”Transient Handling Analysis of a 2007 BMW Mini Equipped with TWEELs: Comparison of Tests and Simulation and Prameter Studies*. Clemson University, Department of Mechanical Engineering Report TR-08-119-ME-MMS, March 2009.

[12] The vehiclesim steer controller. Technical report, Mechanical Simulation, 2008.

[13] E.H. Law. Me 893 class notes. chapter 15. control of the automobile by the human driver., Spring 2009.