### Clemson University TigerPrints

#### All Theses

Theses

1-1-2009

# RELATIONSHIPS BETWEEN LANE CHANGE PERFORMANCE AND OPEN-LOOP HANDLING METRICS

Robert Powell Clemson University, rpowell@g.clemson.edu

Follow this and additional works at: http://tigerprints.clemson.edu/all\_theses Part of the <u>Engineering Mechanics Commons</u>

Please take our one minute survey!

#### **Recommended** Citation

Powell, Robert, "RELATIONSHIPS BETWEEN LANE CHANGE PERFORMANCE AND OPEN-LOOP HANDLING METRICS" (2009). *All Theses.* Paper 743.

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact <a href="mailto:awesole@clemson.edu">awesole@clemson.edu</a>.

### RELATIONSHIPS BETWEEN LANE CHANGE PERFORMANCE AND OPEN-LOOP HANDLING METRICS

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 Robert A. Powell December 2009

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

# Abstract

This work deals with the question of relating open-loop handling metrics to driverin-the-loop performance (closed-loop). The goal is to allow manufacturers to reduce cost and time associated with vehicle handling development. A vehicle model was built in the CarSim environment using kinematics and compliance, geometrical, and flat track tire data. This model was then compared and validated to testing done at Michelin's Laurens Proving Grounds using open-loop handling metrics. The open-loop tests conducted for model validation were an understeer test and swept sine or random steer test. Four commonly used handling metrics (steady state yaw rate gain, yaw rate damping ratio, yaw rate bandwidth, and lateral acceleration phase lag at 1 Hz) were extracted from the frequency response functions of the swept sine test. These are the open-loop handling parameters used to draw relationships to closed-loop performance.

Next, a driver model was coupled to the vehicle model in order to simulate a closedloop maneuver. Quadratic cost functions are then introduced as a means to measure performance through the closed-loop ISO Double Lane Change maneuver. These quadratic metrics measure path-following ability, and the mental and physical workload of the driver. Driver model parameters were determined by weighting the quadratic cost functions to select the optimum driver with the lowest total cost. In this work, highest priority was given to path-following ability in order to successfully complete the lane change without violating course boundaries. Mental workload and physical workload were given lower priority because of the short length of the maneuver.

As a means to change the vehicle model, three hypothetical tires are introduced. To

have an even greater number of configurations, the vehicle model's static weight distribution was altered. This yielded twelve different realistic vehicle configurations from which to draw conclusions. With the twelve different vehicle models considered, qualitative relationships were found between open-loop handling measures and driver-in-the-loop quadratic performance metrics. Two relationships were found to exist—when steady state yaw rate gain and yaw rate bandwidth are increased, path-following ability is enhanced. When yaw rate bandwidth is increased and lateral acceleration phase lag at 1 Hz is decreased, mental and physical workload are reduced on the driver.

# Acknowledgments

I would first like to thank my advisors, Dr. E. Harry Law and Dr. Beshahwired Ayalew, for their wisdom and guidance in helping me achieve this goal. I would also like to thank Dr. John Ziegert for being a part of my committee and providing academic assistance in my graduate course work. In addition, I would like to thank Lynn Vorys for her help in the editing process of this document.

A special thanks also goes to Michelin America's Research Center and BMW for providing the funding for this project. In addition, I would like to thank Dr. Tim Rhyne and Dr. Steve Cron of Michelin and Dr. Andreas Obieglo of BMW for their continued support throughout this research.

# Dedication

I would like to dedicate this thesis to my parents, David and Leslie, who have always believed in me and helped me achieve my goals. In addition, I would like to thank the rest of my family for their unwavering support throughout my educational career.

To my friends and office mates, thank you for making my experience at Clemson both enjoyable and educational. A special thanks goes to Anup Khekare with whom I have had the pleasure of working on this research.

# **Table of Contents**

$\mathbf{A}$	bstra	act	ii
A	cknov	wledgments	iv
D	edica	$\operatorname{tion}$	v
Li	st of	Figures	ix
Li	st of	Tables	xi
1	Intr	$\operatorname{roduction}$	1
	1.1	Introduction	1
	1.2	Research Motivation and Problem Statement	2
	1.3	Literature Review	4
		1.3.1 Open-Loop Handling Metrics	4
		1.3.2 Relation Between Subjective and Objective Testing	4
		1.3.3 Relationship Between Open and Closed-Loop Testing	5
		1.3.4 Driver Models	6
<b>2</b>	Veh	nicle Models and Open-Loop Tests	8
	2.1	Introduction	8
	2.2	Vehicle Model	9
	2.3	Open-Loop Handling Tests	10
		2.3.1 Swept Sine Test	11

		2.3.2 Understeer Test	12
	2.4	Validation of CarSim Model	13
3	Clos	sed-Loop Modeling and Metrics	16
	3.1	Introduction	16
	3.2	Driver Model A	16
	3.3	ISO Double Lane Change Course and Performance Metrics	18
		3.3.1 ISO Double Lane Change Course	18
		3.3.2 Quadratic Cost Functions	20
	3.4	Driver Parameter Selection	22
	3.5	Selection of Tire Characteristics	29
4	$\operatorname{Res}$	ults and Discussion	32
	4.1	Introduction	32
	4.2	Open-Loop and Closed-Loop Performance with Changing Tires	32
	4.3	Open-Loop and Closed-Loop Performance Metrics with Changing Weight	
		Distribution	40
	4.4	Qualitative Analysis Relating Open and Closed-Loop Handling Metrics $\ .$ .	43
5	Con	clusions and Future Work	48
	5.1	Conclusions	48
	5.2	Future Work	49
$\mathbf{A}$	Veh	icle Model Parameters	51
	A.1	Vehicle Properties	52
	A.2	Suspension Parameters	53
	A.3	Suspension Kinematic Parameters	54
	A.4	Compliance Parameters	58
	A.5	Tire Parameters	62
	A.6	Kingpin Geometry	63

В	Dri	ver Models	64
	B.1	Driver Model A	64
	B.2	Driver Model B	67
	B.3	Driver Model C	70
$\mathbf{C}$	Met	bodology for Creating the TWEELs <sup>TM</sup> $\dots \dots \dots \dots \dots \dots$	74
D	Ope	en and Closed-Loop Raw Data	78
$\mathbf{E}$	$\operatorname{Car}$	Sim File Formats	81
$\mathbf{F}$	MA	TLAB Programs	84
	F.1	Model_Validation.m	84
	F.2	TWEEL_Lateral_Force.m	93
	F.3	TWEEL_Aligning_Moment.m	105
	F.4	Spider Plot Programs	112
		F.4.1 color_index.m	112
		F.4.2 isint.m	113
		F.4.3 spider.m	114
	F.5	Frequency_Response_Functions.m	122
	F.6	Lane_Change.m	155
Re	efere	nces	167

# List of Figures

1.1	Image of Generic TWEEL <sup>TM</sup>	3
2.1	Transfer Function for RWA/SWA	10
2.2	Frequency Response Functions (LPG Test vs. CarSim Simulation) $\ldots$ .	14
2.3	Spider Chart Comparing CarSim Simulation vs. LPG Test	15
2.4	Understeer Test Comparing CarSim Simulation vs. LPG Test	15
3.1	Genta driver and vehicle closed loop system	18
3.2	ISO Lane Change Course	20
3.3	$Y_{\text{error}}$ vs. Time for each J's Lowest Cost	24
3.4	SWA vs. Time for each J's Lowest Cost	26
3.5	SWA Rate vs. Time for each J's Lowest Cost	27
3.6	$J_{\text{total}}$ for OE Tire at 78 kph	28
3.7	Cornering Stiffness versus Vertical Load for TWEELs^TM $\hfill \ldots \hfill \hfill \ldots \hfill \hfill \ldots \hfill \$	30
3.8	Lateral Force vs. Slip Angle for OE Tire	31
4.1	Frequency Response Functions Comparing OE Tire to TWEELs^TM $\ . \ . \ .$	33
4.2	Spider Plot of Open-Loop Handling Metrics with Different Tires	34
4.3	Vehicle Trajectory with OE Tire at Maximum Speed of 78 kph $\ .\ .\ .$ .	36
4.4	Vehicle Response Parameters vs. Time at 78 kph	37
4.5	Lateral Acceleration vs. Time and Distance at 78 kph	38
4.6	Lateral Position and Heading Angle vs. Time at 78 kph	39
4.7	Frequency Response Functions Comparing Weight Distributions	41

Spider Plot of Open-Loop Handling Metrics for Changing Weight Distribution	42
Relating Select Open and Closed-Loop Metrics in 2-D Plots	44
3-D Plot Comparing SS Yaw Rate Gain and Yaw Rate Bandwidth to $J_1$	45
3-D Plot Comparing Lateral Accel. Phase Lag and Yaw Rate Bandwidth to $\mathcal{J}_2$	46
3-D Plot Comparing Lateral Accel. Phase Lag and Yaw Rate Bandwidth to ${\cal J}_3$	47
Original Genta Driver Model Block Diagram	64
Driver Model A Simulink/CarSim Block Diagram	66
Driver Model B Simulink/CarSim Block Diagram	69
Driver Model C Simulink/CarSim Block Diagram	72
Horiuchi Driver/Vehicle Closed Loop System	73
BMW 3 Series Aligning Moment Used for Mini Cooper in CarSim $\ \ . \ . \ .$	77
Sample ERD file	82
	Spider Plot of Open-Loop Handling Metrics for Changing Weight Distribution         Relating Select Open and Closed-Loop Metrics in 2-D Plots         3-D Plot Comparing SS Yaw Rate Gain and Yaw Rate Bandwidth to $J_1$ 3-D Plot Comparing Lateral Accel. Phase Lag and Yaw Rate Bandwidth to $J_2$ 3-D Plot Comparing Lateral Accel. Phase Lag and Yaw Rate Bandwidth to $J_3$ Original Genta Driver Model Block Diagram         Driver Model A Simulink/CarSim Block Diagram         Driver Model C Simulink/CarSim Block Diagram         Driver Model C Simulink/CarSim Block Diagram         Block Diagram         Boriver Model C Simulink/CarSim Block Diagram         Block Diagram         Series Aligning Moment Used for Mini Cooper in CarSim         Sample ERD file

# List of Tables

3.1	$J_1$ at 78 kph Normalized to Lowest Value $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	24
3.2	$J_2$ at 78 kph Normalized to Lowest Value $\hfill \ldots \hfill \hfill \ldots \hfill \hfill \ldots \hfill \ldots \hfill \hfill \ldots \hfill \hfill \ldots \hfill \hfill \ldots \hfill \ldots \hfill \ldots \hfill \ldots \hfill \hfill \ldots \hfill \hfill \ldots \hfill \hfill \ldots \hfill \hfill \hfill \hfill \ldots \hfill \hfi$	25
3.3	$J_3$ at 78 kph Normalized to Lowest Value $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	25
3.4	$J_{\text{total}}$ for OE Tire at 78 kph	28
3.5	Driver Parameters for Driver Model A	28
3.6	Cornering Stiffness for each TWEEL <sup>TM</sup>	29
4.1	Spider Chart Data for TWEELs <sup>TM</sup> $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	35
4.2	Quadratic Cost Functions at Each Configuration's Maximum Speed $\ldots$	35
4.3	Spider Chart Data for Changing Weight Distribution	40
4.4	Quadratic Cost Functions at Each Configuration's Maximum Speed $\ldots$	42
4.5	Legend for Figures in Current Section	43
A.1	Vehicle Properties	52
A.2	Suspension Parameters	53
A.3	Left Front Suspension Kinematic Parameters	54
A.4	Right Front Suspension Kinematic Parameters	55
A.5	Left Rear Suspension Kinematic Parameters	56
A.6	Right Rear Suspension Kinematic Parameters	57
A.7	Left Front Compliance Parameters	58
A.8	Right Front Compliance Parameters	59
A.9	Left Rear Compliance Parameters	60
A.10	Right Rear Compliance Parameters	61

A.11	Tire Parameters	62
A.12	Kingpin Geometry Parameters	63
D.1	4 Parameter Data for 12 Configurations	79
D.2	Raw Values of Quadratic Cost Functions at 77 kph	79
D.3	Normalized Values of Quadratic Cost Functions at 77 kph	80

# Chapter 1

# Introduction

### 1.1 Introduction

Open-loop testing is an established practice which engineers use to objectively measure handling qualities of a vehicle. By definition, the driver should not provide any corrective action in open-loop testing thus making the tests repeatable and not dependent upon the driver. Although standards exist for how to conduct the test, there are no standards defining acceptable values for the extracted handling metrics [7]. These tests are used to establish relationships between changes being made to the vehicle and its handling properties. Mimuro et al. [2] suggest arranging commonly used handling metrics in a spider chart to depict trends in behavior. The four parameters Mimuro et al. suggested are steady state yaw rate gain, yaw rate natural frequency, yaw rate damping ratio, and lateral acceleration phase lag at 1 Hz.

While general ideas exist on which parameters improve open-loop performance, the link between open-loop handling metrics and driver-in-the-loop (i.e., closed-loop) performance is still relatively unknown. When the driver is in the loop it is more difficult to relate driver evaluation to specific and quantifiable handling metrics and thus to vehicle design. Crolla et al. point out that car manufacturers employ highly skilled drivers to subjectively measure vehicle handling, but this process is both inefficient and costly [7]. The remaining part of this chapter discusses the motivation behind researching this topic and reviews the previous work done in this field. Chapter 2 presents the vehicle model used and the open-loop tests conducted to validate the vehicle model. The driver model that was adopted and the closed-loop maneuver is explained along with the metrics used to measure performance in the maneuver in Chapter 3. In the same chapter, the different tires fitted to the vehicle are explained. Chapter 4 presents the results for both open-loop and closed-loop testing and the conclusions that were drawn regarding the relationships between the two. Finally, results will be summarized and possible extensions of this work suggested. Additional support information is included in the appendices.

### **1.2** Research Motivation and Problem Statement

Prior to marketing any new car, manufacturers must spend large amounts of time and money developing vehicles. This requires building prototype vehicles and carrying out many tests. Highly-skilled professional test drivers are used to evaluate vehicle handling and ride comfort. If the professional driver does not sign-off on the vehicle then the designer must address the driver's concerns before the vehicle can proceed in its development cycle. This method of vehicle development is both costly and inefficient.

This thesis addresses the method by which vehicle handling is evaluated. In this thesis, it is desired to relate open-loop handling metrics, which are easily modeled and simulated, to driver-in-the-loop performance. Relating these two types of testing methods will allow an engineer to produce a design which will need less iterations to receive a sign-off by the test driver thereby reducing costs and development time. In order to infer relationships, many different vehicle configurations are considered.

In this work, different vehicle configurations are achieved by altering the tires fitted to a single model. In this thesis, the new tires fitted to the vehicle are hypothetical and meant to replicate the handling characteristics of the TWEEL<sup>TM</sup> (a non-pneumatic tire) developed by Michelin. A generic image of a TWEEL<sup>TM</sup> is shown in Figure 1.1.



Figure 1.1: Image of Generic TWEEL<sup>TM</sup>

#### **1.3** Literature Review

As stated by Crolla et al., [7] there are defined standards for conducting open-loop and subjective testing. Subjective testing is practiced quite extensively in the automotive industry and manufacturers design towards established general ranges of performance. The major problem is that there is no link between closed-loop and open-loop testing. If there were, manufacturers could realize significant time and money savings.

#### 1.3.1 Open-Loop Handling Metrics

One of the more commonly cited papers for open-loop testing is by Mimuro et al. [2] in which they suggest using a pulse steer at 100 kilometers per hour (kph) to develop transfer functions for yaw rate and lateral acceleration versus steering wheel angle. From these transfer functions, four parameters are extracted which Mimuro et al. contend correlate well with subjective driver ratings. The four parameters are lateral acceleration phase lag at 1 Hz, steady state yaw rate gain, yaw rate natural frequency, and yaw rate damping ratio. To present these four parameters in a logical and easy-to-read way, Mimuro et al. arrange them in a "rhombus" or spider plot. Presenting it in this way allows the reader to easily compare the handling characteristics of multiple vehicles simultaneously.

#### 1.3.2 Relation Between Subjective and Objective Testing

In order to reduce cost and development time, research has been done to relate subjective and objective handling metrics. In a paper by Crolla et al. [7], many different open-loop handling metrics are measured and correlated to subjective driver ratings. There were sixteen vehicle configurations with forty-nine objective handling metrics measured. These open-loop tests included a steady state understeer test, step steer input (J-turn), and impulse (frequency response) test. For the steady state understeer test, some of the parameters measured were lateral acceleration versus steering wheel angle, side slip angle versus lateral acceleration, and roll angle. In the J-turn test the parameters measured included peak lateral acceleration response time, peak road wheel steer angle and response time, and peak yaw rate and response time. The authors then asked eight experienced test drivers forty-nine questions about each of the sixteen vehicle configurations. The answers were then averaged among the eight test drivers and a linear regression was applied. According to the authors, while this provides some good insight into the links between subjective and objective ratings, there is still no clear link between the two. This paper is clearly a step in the right direction for finding a link between objective and subjective testing.

In a later paper by King et al.[8], a neural network is used to correlate subjective and objective handling ratings. This paper builds off of the earlier paper by Crolla et al. [7] by using the same test data for both objective testing and driver assessments. The main difference in the two is the method of correlation. In this paper by King, a neural network is used to anticipate which parameters drivers will like better. The neural networks provided non-linear links between subjective and objective handling metrics. The problem with this method that the authors point out is that, in order to function properly, the neural network uses some of the gathered data for learning. This results in fewer data points to predict subjective ratings given the objective inputs. However, it is encouraging that the ranges provided by the authors for many open-loop parameters correlated well with subjective ratings .

#### **1.3.3** Relationship Between Open and Closed-Loop Testing

Bergman [11] uses several different tests to try to draw a link between open-loop and closed-loop tests. For the study, two different Ford production sedans were used in three different configurations: curb plus driver (C+D), gross vehicle weight (GVW), and GVW with reduced rear tire inflation pressure. The three open-loop maneuvers employed were step steer (J-turn), modified step steer, and reverse step steer. The step steer requires a driver to travel in a straight line and then input a predefined steering wheel angle as quickly as possible. The modified step steer consists of the vehicle traveling along a slightly curved path then the driver inputs the step steer in the same direction the vehicle is traveling.

The reverse step steer consisted of the vehicle following a slightly curved path in a given direction and then suddenly giving a step steer input in the opposite direction of the vehicle path. Three different lateral accelerations of 0.15g, 0.30g and 0.45g were achieved for each test. The closed-loop maneuver was a single lane change at the same levels of lateral acceleration as the open-loop tests. Bergman finds a good linear correlation between open-loop lateral acceleration response time and closed-loop lateral acceleration response time using the reverse step steer. He further states that the results are strongly influenced by the particular response parameter selected and the test procedure considered.

In a survey, Good reviewed the parameters that correlate well among open-loop, closed-loop, and subjective testing. The papers examined various subjects including accident avoidance, driver-vehicle performance, driver strategy for negotiating a maneuver, open-loop testing, closed-loop testing, and subjective testing.

In another study that Good reviewed [12], he noted that the authors, McRuer and Klien [13], cite the effects of steering ratio on closed-loop performance with an expert driver and three "representative" drivers. The interesting observation from this study is that the expert driver changed his gain as the steering ratio of the vehicle changed while the "representative" drivers did not show the same behavior. Good also cites a paper by Lincke et al. [14] in which eight unskilled drivers negotiated a double lane change at 100 kph. Lincke et al. found that the drivers' subjective ratings correlated well with the natural frequency of the vehicle with damping ratio being much less important.

#### 1.3.4 Driver Models

Another area of review in this thesis is driver models. There are many different driver models put forward and each has its own benefits and drawbacks. In this review, three will be examined. Two are proposed by Genta [15] (one a simple version of the other) and the third is proposed by Horiuchi and Yuhara [16].

In describing a driver model, Genta [15] describes some necessary aspects required in order to perform a constant speed task. These parameters are lead, lag, time delay, and gain. He then gives a first-order transfer function with each of these parameters (which will be explained in further detail in Chapter 3).

Proposing an even simpler driver model, Genta states that the lead term can be neglected and the time delay and lag can be combined as a single term. This yields a much simpler first-order transfer function that is easily implemented in MATLAB/Simulink. He also adds to the model a term where the driver is previewing the error between the current path and the desired path ahead.

In the paper by Horiuchi and Yuhara [16], they propose a driver model that has three essential characteristics: time delay and time lag which are due to mental processing time and neuromuscular system lag, respectively; lead time or predictive action of the driver; and driver gain. The model proposed has a driver that uses lateral position command as an input to the system and then uses the lateral position error and the yaw angle error as feedback into the driver model. The driver outputs a steering wheel angle to negotiate the course. This paper further goes on to give metrics for determining driver workload and task performance. These will be discussed in a later chapter.

### Chapter 2

# Vehicle Models and Open-Loop Tests

### 2.1 Introduction

Data for the vehicle and tires are necessary to create a valid handling model. A full non-linear CarSim model is presented in this thesis which was used to capture higher order effects such as the non-linearity of the tire characteristics and suspension kinematics. Kinematics and compliance (K&C) measurements were performed on the vehicle (2007 BMW Mini Cooper S) at the Michelin Americas Research Center (MARC) to characterize all of the geometric and suspension characteristics. Michelin also performed flat track testing on the original equipment (OE) tires and provided the results in the form of coefficients which could be used in Pacejka's Magic Formula [3]. Open-loop handling tests, including a swept sine and an understeer test of the vehicle, were completed at Michelin's Laurens Proving Grounds (LPG). In simulation, the same tests were performed on the CarSim model and compared to the results obtained at LPG in order to validate the CarSim vehicle model.

### 2.2 Vehicle Model

In order to have a high-fidelity and comprehensive model, commercial software was used instead of developing an in-house model. CarSim, by Mechanical Simulation, is widely used by many companies including Michelin and requires tire and suspension data to be entered as look-up tables. The CarSim model is proprietary software that uses a system of approximately 55 DOF to model the vehicle [19] although some of these degrees of freedom are for things such as the engine and transmission which have no bearing on this analyses. For the purposes of this thesis, only parameters affecting lateral dynamics were used for the analyses. All of the suspension and tire data used for the CarSim model was measured at MARC and provided via the K&C report [4] and coefficients used in Pacejka's Magic Formula [3, 5] for the tire lateral force and aligning moment curves as functions of vertical load, slip angle, and camber. Vehicle weights and CG locations for the test configurations were measured at LPG [20]. A detailed list of the required parameters and their source for the CarSim model is given in Appendix A.

CarSim also has a built-in steering system model requiring parameters such as steering inertia, hysteresis, and damping. However these parameters were not measured. As these steering parameters were not measured, testing at LPG was done not only by measuring steering wheel angle but also rack displacement. As it is more common to show handling parameters with respect to steering wheel angle (SWA), a method was devised to show CarSim output, initially calculated in road wheel angle, with respect to SWA. As track test data was recorded for both rack displacement and steering wheel angle, a transfer function was created between the two over the frequency range of interest as shown in Figure 2.1. To get the yaw rate and lateral acceleration transfer functions in terms of SWA, the magnitudes are multiplied and the phases are added. An interesting note about Figure 2.1 is that the inverse steering ratio (RWA to SWA) is not constant over the frequency range. For an understeer test, which is considered a steady state test, a steering ratio of 17.4:1 SWA to RWA was used since this is the value at the steady state frequency for Figure



Figure 2.1: Transfer Function for RWA/SWA

2.1. Transfer functions were computed in CarSim by using the Spectrum Analyzer tool to perform a fast Fourier transform (FFT) on time history data from LPG and the simulation.

### 2.3 Open-Loop Handling Tests

Open-loop testing is widely used to objectively gather information about vehicle handling with the driver removed from the loop. Although a driver is still used to perform the test, he does so without regard to feedback from the vehicle. The two open-loop tests used at LPG and in simulation are a swept sine and an understeer test.

#### 2.3.1 Swept Sine Test

A swept sine wave as a steering input gives frequency response information about the vehicle. The goal is to have as much frequency content as possible; however, in physical testing, on the high end of the frequency spectrum, a human driver is only capable of about 3 to 3.5 Hz of frequency input into the steering wheel. On the low end of the frequency spectrum, the test is usually limited by the length of the testing track because it takes a long time and, therefore, a long distance at the track to get the required low frequency content. Conversely, in simulation the maximum achievable frequency can be as high as needed but it is usually restricted to approximately 7-10 Hz. As far as the low frequency is concerned there are no limits to the length of the test track in simulation and it can be run as long as needed.

The measured parameters in this test are lateral acceleration, yaw rate, steering wheel angle and/or road wheel angle, and time. A fast Fourier transform (FFT) is computed with the time history data to transform it into frequency domain. Two transfer functions are computed from the time history data– yaw rate and lateral acceleration (Figure 2.2). From these two transfer functions, the four parameters described by Mimuro et al. [2] can be extracted. When Mimuro et al, calculate the four parameters, they fit a two degree of freedom (2 DOF) system to test data and extract the four parameters. In this work, instead of fitting a 2 DOF system to the data, the four parameters are extracted directly from the transfer functions. This necessitates a slightly different definition of yaw damping ratio and use of yaw rate bandwidth as opposed to yaw rate natural frequency. Yaw rate damping ratio in this thesis is derived from the definition of magnitude ratio which is presented in Equation 2.1.

$$|G| = \frac{1}{\sqrt{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right] + \left(2\zeta\frac{\omega}{\omega_n}\right)^2}} |_{\omega=\omega_n} = \frac{1}{2\zeta}$$
(2.1)

where G is the peak yaw rate gain divided by the steady state yaw rate gain,  $\omega_n$  is the

natural frequency, and  $\zeta$  is the damping ratio.

Mimuro et al. suggest plotting these four parameters in a spider chart which shows handling tendencies. An example of the spider chart can be seen in Figure 2.3 which compares the LPG test results to the CarSim model.

In general, when the area of the rhombus for one vehicle is larger than another (assuming consistent axes), the vehicle with the larger area is considered to have greater handling potential. It is important to note for this spider chart that when a rhombus shifts to the upper right-hand corner in relation to other rhombi, the vehicle tends towards more understeer. Whereas when a rhombus shifts toward the lower left, the vehicle tends towards less understeer. [2].

For the test at LPG on the Mini, Michelin provided both an objective handling report [20] (in which they used in-house software to process the yaw rate and lateral acceleration transfer functions) and understeer test results. Michelin also provided the raw data which included five runs of each test to Clemson for processing. All runs were averaged (by Michelin and in this thesis) in order to smooth out any irregularities in one particular test. With the raw testing data, transfer functions could then be computed in CarSim's Spectrum Analyzer tool. This allowed for consistent processing of the data and results could be stated with confidence.

#### 2.3.2 Understeer Test

Another important test for determining handling characteristics is an understeer test. In simulation, this is done by having a vehicle start from a standstill and accelerate at a constant rate while traversing a 500 foot radius circle with a track coefficient of friction of 1.0. The actual tests conducted at LPG are slightly different than what is done in simulation. This is due to the fact that, in order to get reliable results and smooth out any irregularities with one test, multiple tests have to be run and averaged (in this case five runs). If a vehicle were to be run multiple times around a circular track at the speed required for this test it would cause excessive wear on the outside tires and therefore drastically alter the handling properties of the vehicle. In simulation tire wear is not a concern therefore the test is simple to complete and process the results.

The most common data to observe from this test is steering wheel angle (SWA) versus lateral acceleration. Calculating the initial slope of the graph (approximately from 0.1g to 0.3g) will provide the understeer gradient. The understeer gradient gives an indication of how much lateral acceleration will be achieved for a given steering wheel angle. Figure 2.4 shows a comparison of LPG test data to that of the CarSim simulation for steering wheel angle versus lateral acceleration.

### 2.4 Validation of CarSim Model

Once the model was developed in CarSim, simulations were run similar to the tests conducted at LPG– specifically a swept sine test and an understeer test as described in the previous two sections. For the former test, transfer functions for yaw rate and lateral acceleration were calculated and compared to LPG test results. For comparing CarSim simulations to testing performed at LPG, all results are presented with respect to RWA in order to remove the steering system completely from validation of the model. The analyses in this document only examine the curb plus driver configuration as tested at LPG. The parameters used in the CarSim model are listed in Appendix A. Figure 2.2 shows the yaw rate and lateral acceleration transfer functions with respect to RWA. It can be seen from the figure that the transfer functions match well in the lower frequency range (less than 1.5 Hz). Low frequency results are expected to match well because kinematics and compliance data are taken in a quasi-static state. When the actual vehicle is excited at high frequencies it is less likely the simulation and LPG test results will match. Figure 2.3 is a spider chart extracting the four parameters explained by Mimuro et al. [2] comparing the CarSim simulation and LPG test results. From Figure 2.3 it is clear that two of the parameters (steady state yaw rate gain and yaw rate damping ratio) are a perfect match to the LPG test data whereas the other two (yaw rate bandwidth and lateral acceleration phase lag at 1 Hz) remain close.

Figure 2.4 shows a comparison of the CarSim simulation to LPG results for an understeer test. This figure shows a very good match up to about 0.6g. On this figure, the Ackerman angle has been subtracted in order to make the data pass through zero. The Ackerman angle is defined as L/R where L is the wheelbase of the vehicle and R is the radius of the turn. Using the results of the two tests described in this and the previous section and comparing them to the to LPG tests, the CarSim model was considered validated for the Mini Cooper. A more thorough analysis which included the case of gross vehicle weight was done in an internal report to Michelin, BMW, and Clemson [6].



Figure 2.2: Frequency Response Functions (LPG Test vs. CarSim Simulation)



Figure 2.3: Spider Chart Comparing CarSim Simulation vs. LPG Test



Figure 2.4: Understeer Test Comparing CarSim Simulation vs. LPG Test

# Chapter 3

# **Closed-Loop Modeling and Metrics**

### 3.1 Introduction

This section presents the driver model that is used in this thesis. Following the driver model is an explanation of the International Standards Organization (ISO) double lane change course [1] and the desired trajectory for the vehicle. The next section describes quadratic cost functions as metrics for measuring performance through the lane change. Using the driver model described, parameters for the driver model were selected using the quadratic cost functions. Finally, as a means to change the vehicle's handling metrics, hypothetical tires are created and their characteristics are described.

### 3.2 Driver Model A

In order to negotiate an ISO double lane change, implementing a driver model is necessary. The three driver models examined are constant speed and path-following models which only control steering wheel angle. The driver model chosen for the subsequent analyses will be presented in this section while the remaining two driver models [15, 16] are presented in Appendix B. All of these driver models are relatively simple compared to others described in the literature. However, the goal of this thesis is not to conduct an intensive study of driver models but rather find to a driver model that adequately completes the lane change and removes driver variability from the analyses.

In describing a suitable simple driver controller, Genta [15] says the driver model needs to be able to accept a small number of inputs, interpret them and then give an appropriate output. In addition, Genta states the driver is assumed to work off a reaction time delay and a neuromuscular lag. The reaction time delay takes into account the driver seeing his environment and formulating a response. The neuromuscular lag is the time it takes for the command from the brain to reach the relevant muscles and to perform the required action.

Genta then proposes Equation 3.1 for the driver model. This driver model is not unique to Genta and can be found in several other sources including one by Sutton [21]. Genta says that the time delay and lag fall into two categories—one for common drivers and one for professional drivers. In this thesis only common drivers will be considered.

$$\frac{\delta(s)}{U(s)} = K_d \frac{(1+T_L s) e^{-\tau s}}{1+T_D s}$$
(3.1)

In Equation 3.1 and Figure 3.1,  $T_L$  is the lead time given to the driver,  $T_D$  is the neuromuscular lag,  $K_d$  is the driver gain, L is the preview distance,  $\tau$  is the reaction time delay,  $Y_{error}$  is the difference between the desired and the actual path,  $\psi_{error}$  is the difference between the desired and the actual path,  $\psi_{error}$  is the difference between the desired heading angle and the actual heading angle,  $\delta(s)$  is the steering angle provided by the driver, and U(s) is the input into the driver model.

A general block diagram of the driver/vehicle combination using the transfer function in Equation 3.1 is shown in Figure 3.1. In Appendix B.1, the Simulink block diagram shows the implementation of the driver model incorporated with the CarSim vehicle model.

The benefits of this model are its relative simplicity and ease in implementing. The drawback is that at high speed the driver is very slow to react. Since this model is relatively simple yet provides all the necessary parameters needed for a suitable driver, it is the driver model chosen for the analyses.



Figure 3.1: Genta driver and vehicle closed loop system

# 3.3 ISO Double Lane Change Course and Performance Metrics

In this section, the ISO double lane change course boundaries and the desired trajectory are presented. In addition, quadratic cost functions are used as a means for objectively determining performance through the lane change.

#### 3.3.1 ISO Double Lane Change Course

In this thesis the ISO double lane change as shown in Figure 3.2 will be used as the closed-loop test. There are five sections to the course: the first has the car stay in the original lane for 15 meters; the second has the car move laterally 3.5 meters in a 30 meter distance; the third has the vehicle stay in the lane for 25 meters; the fourth has the vehicle return to the original lane in 25 meters; and the fifth is to stay in the original lane for 30 meters. For sections one, three, and five the widths are defined as 1.1T + 0.25, 1.2T + 0.25, and 1.3T + 0.25, respectively, where T is the track width of the vehicle, in meters. For sections two and four, the vehicle does not have to remain within any lateral boundaries. For simplicity, it is assumed that the trajectory the driver attempts to maintain throughout the course is in the middle of the various segments. A skilled driver would probably take a different path that would be much closer to the lane boundaries in sections one, three, and five. The desired lateral position and the heading angle defined by Genta [15] are used in this thesis. The lateral position is given by Equation 3.2 and the heading angle is given by Equation 3.3. X and Y are in units of meters.

$$\begin{cases} Y = 0 & for \ X < 15 \\ Y = \frac{3.5}{2} \left\{ 1 - \cos \left[ \frac{\pi}{30} \left( X - 15 \right) \right] \right\} & for \ 15 \le X < 45 \\ Y = 3.5 & for \ 45 \le X < 70 \\ Y = \frac{3.5}{2} \left\{ 1 + \cos \left[ \frac{\pi}{25} \left( X - 70 \right) \right] \right\} & for \ 70 \le X < 95 \\ Y = 0 & for \ 95 \le X < 125 \end{cases}$$
(3.2)

$$\begin{cases}
 \psi_0 = 0 & for \ X < 15 \\
 \psi_0 = \arctan\left\{\frac{3.5\pi}{60}\sin\left[\frac{\pi}{30}\left(X - 15\right)\right]\right\} & for \ 15 \le X < 45 \\
 \psi_0 = 0 & for \ 45 \le X < 70 & (3.3) \\
 \psi_0 = -\arctan\left\{\frac{3.5\pi}{50}\sin\left[\frac{\pi}{25}\left(X - 70\right)\right]\right\} & for \ 70 \le X < 95 \\
 \psi_0 = 0 & for \ 95 \le X < 125
 \end{cases}$$



Figure 3.2: ISO Lane Change Course

#### 3.3.2 Quadratic Cost Functions

Performance through the double lane change needs to be measured in a way that considers the vehicle's ability to follow the desired path together with the workload put upon the driver [16]. The workload on the driver includes both mental and physical components. These performance parameters can be expressed as quadratic integral cost functions computed through the test maneuver. The metric for lateral deviation from the desired path is defined by Equation 3.4.

$$J_1 = \int_{0}^{t_f} (y_c - y)^2 dt = \int_{0}^{t_f} y_e^2 dt$$
(3.4)

where  $J_1$  is the total quadratic cost,  $y_c$  is the y-coordinate of desired C.G. path, y is the actual C.G. path,  $y_e$  is the instantaneous lateral error, and  $t_f$  is the final time of the observation. Each of the lateral positions ( $y_c$ , y, and  $y_e$ ) are with respect to a global (fixed) coordinate system. The metric for physical workload can be considered using the cost function in Equation 3.5 as suggested by Horiuchi and Yuhara [16].

$$J_2 = \int\limits_0^{t_f} \delta^2 dt \tag{3.5}$$

where is  $\delta$  is the steering wheel angle. For the mental workload, Horiuchi and Yuhara suggest using the sum of the leads that their driver model (presented in Appendix B) requires. Equation 3.6 shows how they calculate mental workload.

$$J_3 = T_{Ly} + T_{L\psi} \tag{3.6}$$

However, since this analysis does not use their driver model another method for measuring mental workload had to be adopted. This was found in a thesis by Oscarsson [17] which presents a metric for mental workload as shown in Equation 3.7.

$$J_3 = \int_0^{t_f} \dot{\delta}^2 dt \tag{3.7}$$

Note that the above three quadratic measures are integrated over the entire maneuver and, as such, provide insight into how the vehicle performs for the whole duration. The objective is to minimize these cost functions through the maneuver under consideration. Minimizing  $J_1$  means maintaining good path-following ability. Minimizing  $J_2$  and  $J_3$ implies that the driver workload would be minimal. Each of these metrics in the body of this thesis will be presented in normalized form to the OE vehicle. The nominal values are shown in Appendix D in the interest of completeness. The normalized values allow for an easy comparison to a baseline vehicle, in this case the OE vehicle.

### 3.4 Driver Parameter Selection

In order to find a relationship between open-loop and closed-loop handling, it is necessary to remove driver variability from the closed-loop system. This necessitates finding a driver model that is the same for all vehicle configurations tested. It is desired for this driver model to be the best possible through the double lane change as measured by weighted quadratic cost functions that were presented in Section 3.3.2. The methodology and selected driver model parameters will be presented in this section and the driver parameters will be used throughout the remaining analyses.

Driver Model A contains the following driver parameters: gain  $(K_d)$ , lead  $(T_L)$ , lag  $(T_D)$ , time delay  $(\tau)$ , and preview distance (L). The time delay and lag can be grouped into two categories-those for professional and those for common drivers. For all drivers, these values are independent of task and considered constant [15, 16]. The analysis in this thesis will only consider common drivers. Horiuchi and Yuhara suggest a value of 0.1 seconds for both the lag and time delay terms. The next parameter to consider is the preview distance which is greatly affected by speed and driving task. Horiuchi and Yuhara [16] suggest a preview time of 1.5 seconds which is easily converted to distance based on the speed of the maneuver.

All but two parameters have now been determined—gain  $(K_d)$  and lead  $(T_L)$ . The methodology chosen for selecting these two parameters was to use the quadratic cost functions defined above and determine the values of  $K_d$  and  $T_L$  which gave the lowest overall cost. First, weights are assigned to  $J_1$ ,  $J_2$ , and  $J_3$  in order to put more or less emphasis on the individual metrics. Then a total cost is computed as shown in Equation 3.8

$$J_{total} = q_1 J_1 + q_2 J_2 + q_3 J_3 \tag{3.8}$$

where  $q_1$ ,  $q_2$ , and  $q_3$  are weighing factors. In this exercise, the path-following ability is given the highest weight because it is considered very important for the vehicle to make it through the double lane change without violating the course boundaries. The driver physical and mental workloads are given lower priority because of the length of the maneuver. The values  $q_1 = 0.75$ ,  $q_2 = 0.125$ , and  $q_3 = 0.125$  were selected for the analysis that follows, where the individual cost functions  $J_1$ ,  $J_2$ , and  $J_3$  have been normalized. The weightings will only be used for selecting appropriate driver model parameters. In Chapter 4 all J parameters will be presented individually without a weighted sum.

To select driver gain  $(K_d)$  and lead  $(T_L)$ , the simulation was first run at a speed of 70 kph using a range of gains and leads. The values of gain and lead that gave the lowest total weighted cost (lowest  $J_{\text{total}}$ ) were then used to determine maximum speed through the lane change without violating the course boundaries. If the maximum speed did not correspond to the speed at which the driver parameters were selected then the speed was increased and the process repeated. The speed at which this occurs is 78 kph with a driver gain of  $K_d = 0.75$  and a lead of  $T_L = 0.2$ .

Table 3.1 shows the computed cost  $J_1$  for a range of  $K_d$  and  $T_L$  settings through an ISO double lane change maneuver at 78 kph. The cost has been normalized with respect to the minimum raw cost, which was found to be at gain  $K_d = 0.7$  and at lead  $T_L = 0.4$ . These results were for the vehicle with the OE tire. Figures 3.3, 3.4, and 3.5 show the time history data for lateral deviation error, steering wheel angle, and steering wheel angle rate, respectively, at each J's lowest raw cost. The J values normalized to the lowest values are given as functions of  $K_d$  and  $T_L$  in Tables 3.1, 3.2, and 3.3.

Tables 3.2 and 3.3 show similar information to Table 3.1 except they are for physical workload  $(J_2)$  and mental workload  $(J_3)$ . Table 3.2 shows that the lowest raw cost for
	K <sub>d</sub>						
$T_{\rm L}$	0.55	0.60	0.65	0.7	0.75	0.8	0.85
0.1	3.10	2.82	2.63	2.45	2.26	2.12	2.01
0.2	2.05	1.80	1.60	1.45	1.33	1.26	1.26
0.3	1.71	1.54	1.43	1.33	1.24	1.15	1.15
0.4	1.51	1.31	1.13	1.00	1.22	2.26	1045.47
0.5	1.26	1.15	1.08	4.80	220.44	476.27	20.38

Table 3.1:  $J_1 \ {\rm at} \ 78 \ {\rm kph}$  Normalized to Lowest Value



Figure 3.3:  $Y_{\rm error}$  vs. Time for each J's Lowest Cost

	$K_{ m d}$						
$T_{\rm L}$	0.55	0.60	0.65	0.7	0.75	0.8	0.85
0.1	1.28	1.46	1.64	1.82	2.01	2.18	2.45
0.2	1.00	1.12	1.24	1.40	1.59	1.84	2.16
0.3	1.02	1.16	1.31	1.49	1.71	2.13	2.94
0.4	1.18	1.46	2.15	3.83	5.90	7.80	94.09
0.5	2.42	4.32	6.00	21.63	93.11	103.27	87.00

Table 3.2:  $J_2$  at 78 kph Normalized to Lowest Value

Table 3.3:  $J_3$  at 78 kph Normalized to Lowest Value

	$K_{ m d}$						
$T_{\rm L}$	0.55	0.60	0.65	0.7	0.75	0.8	0.85
0.1	1.00	1.19	1.39	1.61	1.88	2.15	2.62
0.2	1.14	1.38	1.66	2.07	2.57	3.29	4.25
0.3	1.80	2.24	2.77	3.53	4.62	6.84	11.15
0.4	3.17	4.80	9.18	19.42	27.10	30.14	25.72
0.5	12.04	23.01	28.47	33.45	27.48	26.54	29.92

physical workload is at  $T_L = 0.2$  and  $K_d = 0.55$  which is expected. For any given lead  $(T_L)$ , it can be seen that as driver gain  $(K_d)$  increases, the physical workload of the driver (disregarding the unstable cases) also increases. One can conclude that driver gain relates to physical workload on the driver.

Table 3.3 shows the lowest raw cost for mental workload is found for  $T_L = 0.1$  and  $K_d = 0.55$ . Increasing both gain and lead raises a driver's mental workload which results in the conclusion that these two parameters are related to  $J_3$ .

Figure 3.6 shows a surface plot of  $J_{\text{total}}$  with the minimum total cost indicated by the data point which corresponds to  $T_L = 0.2$  and  $K_d = 0.75$ . Table 3.4 contains the same information as Figure 3.6 but in tabular form for easy viewing of values rather than trends.



Figure 3.4: SWA vs. Time for each J's Lowest Cost



Figure 3.5: SWA Rate vs. Time for each J's Lowest Cost

	$K_{ m d}$						
$T_{\rm L}$	0.55	0.60	0.65	0.7	0.75	0.8	0.85
0.1	2.61	2.45	2.35	2.27	2.18	2.13	2.14
0.2	1.80	1.66	1.56	1.52	1.51	1.59	1.74
0.3	1.63	1.58	1.58	1.62	1.72	1.99	2.62
0.4	1.68	1.76	2.27	3.66	5.04	6.44	799.08
0.5	2.76	4.28	5.12	10.49	180.41	373.43	29.90

Table 3.4:  $J_{\text{total}}$  for OE Tire at 78 kph



Figure 3.6:  $J_{\rm total}$  for OE Tire at 78 kph

The driver parameters selected in this analysis will be used throughout the rest of the document. All relevant driver parameters are in Table 3.5.

Table 3.5: Driver Parameters for Driver Model A $\tau$  $T_{\rm D}$  $T_{\rm L}$  $K_{\rm d}$ L (sec)

	1	TD	TL	n <sub>d</sub>	L (Sec)
Parameter	0.1	0.1	0.2	0.75	1.5

#### **3.5** Selection of Tire Characteristics

Michelin has provided flat-track testing results for the OE Mini Cooper tire; however, at the time this thesis was written, TWEELs<sup>TM</sup> for the Mini had not yet been fabricated and, therefore, no data exists. Michelin did provide Pacejka coefficients for an experimental prototype of a TWEEL<sup>TM</sup> for a BMW 3 Series car. These data were used in the Magic Formula [3] to generate lateral force and aligning moment versus slip angle curves. Combining the Pacejka data for the BMW 3 Series with comments from Michelin [10] led to the creation of the three hypothetical TWEELs<sup>TM</sup> for the Mini. The effect modeled is that of the tires becoming more or less sensitive to vertical load.

One of the TWEELs<sup>TM</sup> is supposed to maintain the same total (front + rear) cornering stiffness as the OE tire but changes the split (front/rear) of the cornering stiffness. The other two TWEELs<sup>TM</sup> also change the split as well as raise the total cornering stiffness. The methodology by which the three TWEELs<sup>TM</sup> were created is discussed in more detail in Appendix C.

Using the method explained in Appendix C to create the TWEELs<sup>TM</sup>, Table 3.6 shows the "garage" values of front and rear, total, and the front to rear split of the cornering stiffness. These values for front and rear cornering stiffness are dependent on the prevailing weight distribution of the vehicle (OE F/R=60/40).

	$C_{lpha f}$	$C_{lpha r}$	Total $C_{\alpha}$ (f + r)	Front to Rear			
	(N/deg)	(N/deg)	(N/deg)	Split			
				$(C_{ m lpha f}/C_{ m lpha r})$			
OE Tire	1323	948	2271	1.40			
TWEEL <sup>TM</sup> A	1247	1030	2278	1.21			
TWEEL <sup>TM</sup> B	1446	1119	2565	1.29			
TWEEL <sup>TM</sup> C	1561	1327	2888	1.18			

Table 3.6: Cornering Stiffness for each TWEEL<sup>TM</sup>

Figure 3.7 shows the cornering stiffness versus load plot for the candidate TWEELs<sup>TM</sup>. In this plot the vertical lines mark the garage value loads (at curb + driver loading, C+D) for front and rear wheels which are 3900 and 2600 Newtons, respectively. For the OE tire,



Figure 3.7: Cornering Stiffness versus Vertical Load for TWEELs<sup>TM</sup>

cornering stiffness is almost linear over the given range of vertical loads whereas each of the three TWEELs<sup>TM</sup> is more non-linear above 3000 N. As vertical load is increased past 5000 N on the TWEELs<sup>TM</sup>, cornering stiffness remains about the same (TWEEL<sup>TM</sup> B) or decreases (TWEELs<sup>TM</sup> A and C).

Lateral force versus slip angle and vertical load curves are shown in Figure 3.8. In Figure 3.8 each of the curves represents a different vertical load on the tire with the lowest vertical load corresponding to the lowest lateral force curve. Similar plots for each of the TWEELs<sup>TM</sup> will be omitted due to the fact they are qualitatively the same as the OE tire except for the noted difference of cornering stiffness versus load.



Figure 3.8: Lateral Force vs. Slip Angle for OE Tire

## Chapter 4

# **Results and Discussion**

#### 4.1 Introduction

In this chapter results of the open-loop and closed-loop handling metrics will be examined for different configurations of the Mini Cooper. First, results will be presented for both open-loop and closed-loop performance by changing the tires using the three hypothetical TWEELs<sup>TM</sup>. Next, these same open-loop and closed-loop metrics will be presented for different weight distributions of the vehicle. Each of these vehicle configurations is considered a realistic change and therefore dramatic improvement or degradation in results is not expected. Finally, combining the TWEELs<sup>TM</sup> with the weight distribution cases yields twelve configurations which are used to find relationships between open-loop and closed-loop handling.

# 4.2 Open-Loop and Closed-Loop Performance with Changing Tires

Shown in Figure 4.1 are the transfer functions for yaw rate and lateral acceleration response of the vehicle. In the upper left-hand plot there are two things to note. First, steady state yaw rate gain is about the same for all tires except TWEEL<sup>TM</sup> B. Second, yaw



Figure 4.1: Frequency Response Functions Comparing OE Tire to TWEELs<sup>TM</sup>

rate bandwidth is much larger for TWEEL<sup>TM</sup> C than the others. In addition, in the lower right-hand plot, lateral acceleration phase lag at 1 Hz is slightly less than 40 degrees for TWEEL<sup>TM</sup> C and is separated from the other tires at 1 Hz which are clustered around 50 to 55 degrees.

Figure 4.2 shows the spider charts generated by simulating the vehicle models through swept-sine steer (open-loop) maneuvers. The transfer functions for yaw rate and lateral acceleration were created and the four parameters were extracted as described in Chapter 2. Figure 4.2 clearly shows the trends in the open-loop characteristics as the vehicle's tire sets are changed. When the spider plot shifts to the upper right, the vehicle



Figure 4.2: Spider Plot of Open-Loop Handling Metrics with Different Tires

exhibits more understeering behavior and, conversely, when it shifts to the lower left, the vehicle exhibits less understeering behavior. It can be noted also that the increased total cornering stiffness of TWEELs<sup>TM</sup> B and C (See Table 3.6 above) gives larger areas in the spider plot showing the greater handling potential of the vehicle [2], as should be expected. The larger front to rear split in cornering stiffness for TWEEL<sup>TM</sup> A compared to the OE tire accounts for the larger yaw rate bandwidth and smaller lateral acceleration phase lag at 1 Hz for TWEEL<sup>TM</sup> A. It can also be seen from Figure 4.2 that steady state yaw rate gain and yaw rate damping ratio vary only a small amount for the various configurations compared to the other two parameters on the chart. Table 4.1 contains all of the data presented in Figure 4.2 except in tabular form and also includes the understeer gradient for each configuration.

To evaluate the closed-loop handling, the double lane change performance at the maximum speed (i.e., the maximum speed at which the lane change is traversed without violating lane boundaries) for the car and driver combination is presented. Figure 4.3 shows the course taken by the vehicle with the OE Tire at 78 kph. The desired path has the vehicle

	SS Yaw Rate Gain (1/s)	Yaw Rate Bandwidth (Hz)	Yaw Rate Damping Ratio	Lateral Acceleration Phase Lag @ 1 Hz (deg)	Kus (deg/g)
OE Tire	0.42	2.35	0.48	-55.84	17.8
TWEEL <sup>TM</sup> A	0.41	2.52	0.48	-51.88	17.6
TWEEL <sup>TM</sup> B	0.45	2.61	0.49	-48.57	20.8
TWEEL <sup>TM</sup> C	0.42	3.10	0.47	-39.68	23.4

Table 4.1: Spider Chart Data for TWEELs<sup>TM</sup>

Table 4.2: Quadratic Cost Functions at Each Configuration's Maximum Speed

(F/R=60/40)	Max Speed (kph)	Normalized $J_1$	Normalized $J_2$	Normalized $J_3$
OE Tire	78	1.00	1.00	1.00
TWEEL <sup>TM</sup> A	77	1.03	1.01	0.96
TWEEL <sup>TM</sup> B	78	0.92	0.97	0.98
TWEEL <sup>TM</sup> C	78	0.91	0.90	0.84

staying in the middle of the lanes throughout the course. Figures 4.4 to 4.6 show various vehicle parameters versus time and distance.

Table 4.2 lists each individual J metric (of closed-loop lane change performance) and each vehicle's maximum speed with which it could successfully negotiate the double lane change maneuver described above without violating the course boundaries. Each J metric is normalized to the equivalent metric computed for the OE tire although TWEEL<sup>TM</sup> A is normalized at a slightly lower speed than the other three. Since these are cost functions, the goal is to minimize them as much as possible and therefore a value less than 1.00 is considered an improvement over the OE Tire. TWEEL<sup>TM</sup> C gives between a 9 and 15% improvement in each J-parameter as compared to the OE tire. TWEELs<sup>TM</sup> A, B, and C each show an improvement in  $J_3$  and TWEELs<sup>TM</sup> B and C show improvements in both  $J_2$ and  $J_3$ .

Using Figure 4.2 and Table 4.2 together one observes that the normalized  $J_2$  and  $J_3$  appear to decrease as lateral acceleration phase lag at 1 Hz decreases and as yaw rate bandwidth increases. Normalized  $J_1$  shows this same trend except for the vehicle with TWEEL<sup>TM</sup> A.



Figure 4.3: Vehicle Trajectory with OE Tire at Maximum Speed of 78 kph



Figure 4.4: Vehicle Response Parameters vs. Time at 78 kph



Figure 4.5: Lateral Acceleration vs. Time and Distance at 78 kph



Figure 4.6: Lateral Position and Heading Angle vs. Time at 78 kph

Weight Dist. (F/R)	SS Yaw Rate Gain (1/s)	Yaw Rate Bandwidth (Hz)	Yaw Rate Damping Ratio	Lateral Acceleration Phase Lag @ 1 Hz (deg)	$\begin{array}{c} {\rm Kus} \\ {\rm (deg/g)} \end{array}$
$\begin{array}{c} 60/40, \\ (OE \text{ Tire}) \end{array}$	0.42	2.35	0.48	-55.84	17.8
$\begin{array}{c} 55/45, \\ (OE \text{ Tire}) \end{array}$	0.47	2.27	0.49	-61.53	13.2
$\begin{array}{c} 50/50, \\ (OE \text{ Tire}) \end{array}$	0.51	2.12	0.49	-67.72	9.4

Table 4.3: Spider Chart Data for Changing Weight Distribution

## 4.3 Open-Loop and Closed-Loop Performance Metrics with Changing Weight Distribution

It was desired to change another first-order effect on handling dynamics, namely the static weight distribution, to see if further conclusions can be drawn between open-loop and closed-loop performance metrics. The nominal weight distribution of the vehicle is 60/40 (front to rear). Two other configurations with front to rear distributions of 55/45 and 50/50 were simulated considering the OE tire set for both configurations. It is noted that only weights carried by the axles were changed, not the yaw inertias of the vehicle. Figure 4.7 shows the yaw rate and lateral acceleration transfer functions for the changing weight distributions. From Figure 4.7 there are two important points to note-the large differences between steady state yaw rate gain for each of the cases in the upper left-hand plot and, in the lower left-hand plot, the lateral acceleration phase lag at 1 Hz.

Figure 4.8 is a spider chart of the simulated vehicle through a swept sine test where the weight distribution of the vehicle was changed. As the C.G. of the vehicle moves rearward, the vehicle has greater steady state yaw rate gain and larger lateral acceleration phase lag at 1 Hz (Figure 4.7). The shift of the rhombi to the lower left with the C.G. moving to the rear also indicates that the vehicle tends toward less understeer. Table 4.3 shows the data in Figure 4.8 in tabular form along with the understeer gradient for each configuration.



Figure 4.7: Frequency Response Functions Comparing Weight Distributions



Figure 4.8: Spider Plot of Open-Loop Handling Metrics for Changing Weight Distribution

Weight Dist. $(F/R)$	$(F/R) \begin{vmatrix} Max \text{ Speed} \\ (kph) \end{vmatrix} \text{ Normalized } J_1$		Normalized $J_2$	Normalized $J_3$
60/40, (OE Tire)	78	1.00	1.00	1.00
55/45, (OE Tire)	78	0.98	0.96	1.02
50/50, (OE Tire)	77	0.86	1.60	2.03

Table 4.4: Quadratic Cost Functions at Each Configuration's Maximum Speed

Table 4.4 contains the quadratic cost functions normalized to the nominal weight distribution at each configuration's maximum speed. The configuration that stands out the most is the 50/50 weight distribution. It shows that the driver expends much more physical and mental effort (larger  $J_2$  and  $J_3$ ) to complete the lane change in this configuration while tracking the course ( $J_1$ ) somewhat better than in the nominal configuration albeit at a slightly lower speed.

Table 4.9. Degend for Figures in Ourfeit Section					
Configuration (Weight Distribution, Tire)	Marker	Configuration (Weight Distribution, Tire)	Marker		
60/40, OE Tire	Δ	55/45, TWEEL <sup>TM</sup> B	×		
60/40, TWEEL <sup>TM</sup> A	$\nabla$	55/45, TWEEL <sup>TM</sup> C	+		
60/40, TWEEL <sup>TM</sup> B	$\triangleleft$	50/50, OE Tire	0		
60/40, TWEEL <sup>TM</sup> C	$\triangleright$	50/50, TWEEL <sup>TM</sup> A			
55/45, OE Tire	尊	50/50, TWEEL <sup>TM</sup> B	•		
55/45, TWEEL <sup>TM</sup> A	\$	50/50, TWEEL <sup>TM</sup> C	*		

Table 4.5: Legend for Figures in Current Section

# 4.4 Qualitative Analysis Relating Open and Closed-Loop Handling Metrics

In this section, it is desired to present additional qualitative comparisons between the four-parameter open-loop handling measures and the closed-loop quadratic performance measures by combining the considerations in the previous two sections to cover more configurations of the vehicle. Therefore each tire is combined with each weight distribution yielding twelve configurations. The results are presented in Figures 4.9, 4.10, 4.11, and 4.12 with Table 4.5 listing the symbols used in the figures. To remove speed as a variable in this analysis, all configurations were simulated at a speed of 77 kph, the maximum speed with which all configurations could successfully complete the lane change.

First, the upper two plots in Figure 4.9 show the lateral acceleration phase lag at 1 Hz and yaw rate bandwidth compared to the physical workload as measured by  $J_2$ . These plots show that physical workload is reduced as yaw rate bandwidth increases and lateral acceleration phase lag at 1 Hz decreases. The lower plots of Figure 4.9 show yaw rate bandwidth and lateral acceleration phase lag at 1 Hz compared to the mental workload ( $J_3$ ). The plots clearly show that mental workload is reduced as yaw rate bandwidth increases and lateral acceleration phase lag at 1 Hz decreases.



Figure 4.9: Relating Select Open and Closed-Loop Metrics in 2-D Plots



Steady State Yaw Rate Gain, 1/s

Figure 4.10: 3-D Plot Comparing SS Yaw Rate Gain and Yaw Rate Bandwidth to  ${\cal J}_1$ 

Figure 4.10 is a 3-D plot showing the observed relationship between steady state yaw rate gain and yaw rate bandwidth versus the metric for lateral deviation from the desired path  $(J_1)$ . The points stretch from smaller yaw rate bandwidth and steady state yaw rate gain and extend towards greater yaw rate bandwidth and steady state yaw rate gain. As should be expected, the plot suggests that as steady state yaw rate gain and yaw rate bandwidth increase, the metric  $J_1$  is reduced, indicating improved tracking performance.

Figures 4.11 and 4.12 show lateral acceleration phase lag at 1 Hz and yaw rate bandwidth versus  $J_2$  and  $J_3$ , respectively. This is the same information as presented in Figure 4.9 but is shown in three dimensional space to get a better understanding of how mental  $(J_2)$  and physical  $(J_3)$  workload correspond with both parameters. The same conclusion is drawn- as lateral acceleration phase lag at 1 Hz decreases and yaw rate bandwidth increases



Lateral Accel. Phase Lag, deg

Figure 4.11: 3-D Plot Comparing Lateral Accel. Phase Lag and Yaw Rate Bandwidth to  $J_2$ 

physical and mental workload are reduced.

It can also be observed that the more aggressive design, TWEEL<sup>TM</sup> C (in terms of highest overall cornering stiffness), consistently gives lower  $J_1$ ,  $J_2$ , and  $J_3$  costs regardless of the weight distribution. However, in some cases, the particular vehicle characteristic that was changed does have an effect on whether specific relationships hold. For example, the 50/50 OE tire case shows larger yaw rate bandwidth and lower lateral acceleration phase lag at 1 Hz than the 50/50 TWEEL A, but it also gives the largest  $J_2$  and  $J_3$  costs.

Other pairs of performance metrics were also considered (e.g. steady state yaw rate gain and yaw rate bandwidth versus mental workload), but no clear qualitative relationships were detected and therefore those results are not presented.



Lateral Accel. Phase Lag, deg



Figure 4.12: 3-D Plot Comparing Lateral Accel. Phase Lag and Yaw Rate Bandwidth to  $J_3$ 

### Chapter 5

# **Conclusions and Future Work**

#### 5.1 Conclusions

In this thesis, it was attempted to establish relationships between traditional openloop handling metrics and quadratic cost functions measuring closed-loop performance. To this end, a vehicle model was built for the Mini Cooper and was validated with track testing data. In addition, a driver model that included lead, lag, time delay, and preview distance parameters was constructed. Quadratic cost functions were then adopted to take into account lane change tracking performance and driver mental and physical workload. Driver parameters were then selected at a given speed using the quadratic cost functions to minimize the total cost with high priority given to successfully completing the lane change. Mental and physical workloads were given less priority due to the short length of the maneuver. The selected driver parameters were then used for the remaining analyses to determine relationships between open-loop and closed-loop performance metrics.

TWEEL<sup>TM</sup> characteristics were then presented in a cornering stiffness versus load plot to show the load sensitivity of the TWEEL<sup>TM</sup>. The TWEELs<sup>TM</sup> and changing the weight distribution of the vehicle were used as a means of changing the vehicle configuration. The different vehicle configurations were used to make a qualitative analysis between the fourparameter (open-loop) handling performance metrics and double lane change quadratic metrics. The following clear relationships exist:

- As steady state yaw rate gain and yaw rate bandwidth increased, path-following ability  $(J_1)$  is generally enhanced.
- As yaw rate bandwidth increases and lateral acceleration phase lag at 1 Hz decreases, both physical  $(J_2)$  and mental  $(J_3)$  workload on the driver are reduced.

Relationships among the other performance metrics were found to be less definitive with the approach taken here and the vehicle (and its configurations) considered in the present work. It was also noted that even for the specific relationships identified, the extent of the correlation could be a function of the vehicle design characteristics (cornering stiffness or weight distribution) that were changed to extract the relationships.

#### 5.2 Future Work

Future work in this area should consider many different configurations for many different vehicles to give a large array of data upon which to make statistical correlations. The different vehicles should come from a variety of different classes including large and small sedans, sports cars, SUVs, and trucks. Given the current data, the car at a loading corresponding to gross vehicle weight could be included in the correlation to double the number of data points.

In addition, a more sophisticated method of correlation could be possible with more data points. In order to validate closed-loop performance to a baseline vehicle, testing needs to be conducted measuring the quadratic cost functions described in this thesis. A more complex driver model could be coupled to the vehicle model in order to better emulate driver behavior. APPENDICES

# Appendix A

# Vehicle Model Parameters

Some of the documents referenced in this appendix are internal documents to Michelin, BMW, and Clemson University and are not, therefore, available to any other parties. Vehicle yaw inertia was not measured as part of K&C testing. However, once the model was built with all other parameters known, yaw inertia was changed until a good match was achieved between LPG testing and simulation. In the case for curb plus driver a value of 1750  $kg - m^2$  was found to match.

# A.1 Vehicle Properties

Description	Curb + Driver	Units	Source
Total vehicle mass	1323	kg	[20] Page 1
Wheelbase	2468	mm	[4] Figure 2
Track width (front)	1453	mm	[4] Figure 2
Track width (rear)	1475	mm	[4] Figure 2
CG location longitudinally	081	mm	[99] Table 9
(distance from front axle - a)	901		[22] Table 2
CG location longitudinally	1487	mm	[99] Table 9
(distance from rear axle – b)	1407		
CG location vertically	517	mm	[22]Table 2

 Table A.1: Vehicle Properties

## A.2 Suspension Parameters

Description	Curb + Driver	Units	Source
Front spring force vs. Compression	Spring Rate Curve	N/mm	[4] Section ZF Figure Z1
Rear spring force vs. Compression	Spring Rate Curve	N/mm	[4] Section ZR Figure Z1
Difference between overall roll stiffness and that provided by springs (front)	577	N-m/deg	<ul><li>[4] Vehicle summary,</li><li>Roll test, Table 4 &amp;</li><li>Table 41</li></ul>
Difference between overall roll stiffness and that provided by springs (rear)	422	N-m/deg	<ul><li>[4] Vehicle summary,</li><li>Roll test, Table 4 &amp;</li><li>Table 41</li></ul>
Front shock travel rate vs. Damping force	Front Shock Curve	N-s/mm	[4] Section A
Rear shock travel rate vs. Damping force	Rear Shock Curve	N-s/mm	[4] Section A

Table A.2:	Suspension	Parameters
------------	------------	------------

### A.3 Suspension Kinematic Parameters

Description	Curb + Driver	Units	Source	
Left Front				
Wheel spin moment of inertia	1.637	$kg - m^2$	[23]Inertial Parameters and CG locations, values are scaled using rolling radius	
Static toe left front	0.125	deg	[4] Table 1A Loaded Geometry Summary	
Static camber left front	-0.28	deg	[4] Table 1A Loaded Geometry Summary	
Camber change due to jounce	Camber Curve Sheet	deg	[4] Section ZF Figure Z3	
Lateral displacement change due to jounce	Lateral Displacement Sheet	mm	[4] Section ZF Figure Z6	
Toe steer due to jounce	Toe Change Sheet	deg	[4] Section ZF Figure Z2	
Caster change due to jounce	Caster Curve Sheet	deg	[4] Section ZF Figure Z8	
Longitudinal movement of wheel due to jounce	Longitudinal Movement Sheet	mm	[4] Section ZF Figure Z7	

 Table A.3: Left Front Suspension Kinematic Parameters

Description	Curb + Driver	Units	Source	
Right Front				
			[23] Inertial	
			Parameters and CG	
Wheel spin moment of inertia	1.637	$kg-m^2$	locations, values are	
			scaled using rolling	
			radius	
Static too right front	0.195	deg	[4] Table 1A Loaded	
Static toe right from	0.125	ueg	Geometry Summary	
Static combon night front	0.69	der	[4] Table 1A Loaded	
Static camber right front	-0.62	deg	Geometry Summary	
Cambon shanna dua ta isumaa	Camber Curve	deg	[4] Section ZF Figure	
Camper change due to jounce	Sheet		Z3	
I stand dimber and show as	Lateral		[4] Section ZE Eimme	
Lateral displacement change	Displacement	mm	[4] Section ZF Figure	
due to jounce	Sheet		Z0	
	Toe Change	deg	[4] Section ZF Figure	
Toe steer due to jounce	Sheet		Z2	
	Caster Curve	dom	[4] Section ZF Figure	
Caster change due to jounce	Sheet	ueg	Z8	
Longitudinal movement of wheel due to jounce	Longitudinal		[4] Section ZE Eimme	
	Movement	mm	[4] Section ZF Figure	
	Sheet			

 Table A.4: Right Front Suspension Kinematic Parameters

Description	Curb + Driver	Units	Source	
Left Rear				
		$kg - m^2$	[23] Inertial	
			Parameters and CG	
Wheel spin moment of inertia	1.376		locations, values are	
			scaled using rolling	
			radius	
Static too left mean	0.175	dor	[4] Table 1A Loaded	
Static toe left rear	0.175	ueg	Geometry Summary	
	-1.7	deg	[4] Table 1A Loaded	
Static camper left rear			Geometry Summary	
Comban aban na dua ta isana a	Camber Curve	deg	[4] Section ZR Figure	
Camper change due to jounce	Sheet		Z13	
Lateral displacement change due to jounce	Lateral	mm	[4] Section ZR Figure Z16	
	Displacement			
	Sheet			
Toe steer due to jounce	Toe Change	deg	[4] Section ZR Figure	
	Sheet		Z12	
Longitudinal movement of wheel due to jounce	Longitudinal		[4] Section 7D Eigen	
	Movement	$\mathbf{m}\mathbf{m}$	[4] Section ZK Figure	
	Sheet		217	

Table A.5: Left Rear Suspension Kinematic Parameters

Description	Curb + Driver	Units	Source	
Right Rear				
		$kg - m^2$	[23] Inertial	
			Parameters and CG	
Wheel spin moment of inertia	1.376		locations, values are	
			scaled using rolling	
			radius	
Static too right roop	0.175	dor	[4] Table 1A Loaded	
Static toe right rear	0.175	ueg	Geometry Summary	
	-1.68	deg	[4] Table 1A Loaded	
Static camper right rear			Geometry Summary	
Combon shanne due to journe	Camber Curve	deg	[4] Section ZR Figure	
Camper change due to jounce	Sheet		Z13	
Lateral displacement change due to jounce	Lateral	mm	[4] Section ZR Figure Z16	
	Displacement			
	Sheet			
Teo steen due te ieunee	Toe Change	deg	[4] Section ZR Figure	
10e steer due to jounce	Sheet	ueg	Z12	
Longitudinal movement of wheel due to jounce	Longitudinal	mm	[4] Section 7D Firming	
	Movement		[4] Section ZR Figure	
	Sheet			

Table A.6: Right Rear Suspension Kinematic Parameters

## A.4 Compliance Parameters

Description	Curb + Driver	Units	Source
	Left Front		
Coefficient of change in toe per change in Fx	6.30E-05	deg/N	[4] Section XF, List XF
Coefficient of change in steer angle per change in Fy	-4.00E-05	$\rm deg/N$	[4] Section YF, List YF
Coefficient of change in steer angle per change of tire Mz	4.90E-03	deg/N-m	[4] Section YF, List YF
Coefficient of change in inclination angle per change in Fy	1.94E-04	deg/N-m	[4] Section YF, List YF
Coefficient of change in inclination angle per change in Mz	-6.80E-03	deg/N-m	[4] Section YF, List YF
Coefficient of change in X displacement per change in Fx	1.84E-03	m mm/N	[4] Section XF, List XF
Coefficient of change in Y displacement per change in Fy	2.41E-04	mm/N	[4] Section YF, List YF

 Table A.7: Left Front Compliance Parameters

Description	Curb + Driver	Units	Source	
Right Front				
Coefficient of change in toe	8 00F 05	deg/N	[4] Section XF, List	
per change in Fx	8.00E-05	ueg/1	XF	
Coefficient of change in steer	2 105 05	$\rm deg/N$	[4] Section YF, List	
angle per change in Fy	-3.10E-03		YF	
Coefficient of change in steer	4.82E.02	dog/N m	[4] Section YF, List	
angle per change of tire Mz	4.83E-03	deg/IN-m	YF	
Coefficient of change in			[4] Section VE List	
inclination angle per change	1.93E-04	deg/N-m	[4] Section YF, List	
in Fy			Ϋ́́Γ	
Coefficient of change in			[4] Continue VE Lint	
inclination angle per change	-6.87E-03	deg/N-m	[4] Section YF, List	
in Mz			ΙΓ	
Coefficient of change in X			[4] Castion VE List	
displacement per change in	1.95E-03	mm/N	[4] Section AF, List	
Fx			AF	
Coefficient of change in Y	9.25E 04	mm/N		
displacement per change in Fy	2.35E-04	11111/19	[4] Section YF List YF	

 Table A.8: Right Front Compliance Parameters
Description	Curb + Driver	Units	Source		
Left Rear					
Coefficient of change in toe	dog/N		[4] Section XR , List		
per change in Fx	0.00E+00	ucg/1	XR		
Coefficient of change in steer	2 10E 0E	$\rm deg/N$	[4] Section YR , List		
angle per change in Fy	-2.10E-05		YR		
Coefficient of change in steer	9.925.04	dog/N m	[4] Section YR , List		
angle per change of tire Mz	2.33E-04		YR		
Coefficient of change in			[4] Continue VD List		
inclination angle per change	2.74 E-04	deg/N-m	[4] Section 1 K, List		
in Fy			IK		
Coefficient of change in			[4] Continue VD List		
inclination angle per change	tion angle per change -9.03E-03		[4] Section T.K., List		
in Mz			ΪK		
Coefficient of change in X			[4] C+; VD - L;-+		
displacement per change in	$0.00E{+}00$	mm/N	[4] Section AR, List		
Fx			АК		
Coefficient of change in Y	4.94E.04	mm/N	[4] Section YR , List		
displacement per change in Fy	4.84E-04		YR		

 Table A.9: Left Rear Compliance Parameters

Description	Curb + Driver	Units	Source		
Right Rear					
Coefficient of change in toe	0.00E+00 deg/N		[4] Section XR , List		
per change in Fx	0.001+00	408/11	XR		
Coefficient of change in steer	8 00F 06	deg /N	[4] Section YR , List		
angle per change in Fy	-8.0012-00	ueg/14	YR		
Coefficient of change in steer	8.67F 04	deg/N-m	[4] Section YR , List		
angle per change of tire Mz	8.07E-04		YR		
Coefficient of change in inclination angle per change in Fy	2.77E-04	deg/N-m	[4] Section YR , List YR		
Coefficient of change in inclination angle per change in Mz	-9.40E-03	deg/N-m	[4] Section YR , List YR		
Coefficient of change in X displacement per change in Fx	0.00E+00	mm/N	[4] Section XR , List XR		
Coefficient of change in Y displacement per change in Fy	4.86E-04	mm/N	[4] Section YR , List YR		

 Table A.10: Right Rear Compliance Parameters

#### A.5 Tire Parameters

Description	Curb + Driver	Units	Source	
Coning note	020	N/mm	[24] XYZ Rigidity	
Spring rate	232		Test	
Maximum allowed force	100000	Ν	CarSim Default	
Polling registance coefficient	CarSim			
Rolling resistance coemcient	Default			
Slip angle versus Fy	(curve)		[5] Michelin	
	Pacejka data		[5] Michelli	
Clin angle yereya Mg	(curve)		[5] Michelin	
Sup angle versus wz	Pacejka data		[5] Michelli	
Camber thrust coefficient	(curve)		[5] Michelin	
versus Fz	Pacejka data		[5] Michelin	
Tipo latoral valevation lar ath	260 0	mm	[24] XYZ Rigidity	
The lateral relaxation length	308.2	111111	Test	

Table A.11: Tire Parameters

#### A.6 Kingpin Geometry

Description	Curb + Driver	Units	Source	
Left Lateral offset @ center	51 16	mm	[4] Section S. List S.	
(spindle offset)	51.10		[4] Section 5, List 5	
Right Lateral offset @ center	F1 07	mm		
(spindle offset)	51.37	mm	[4] Section S, List S	
Left kingpin inclination angle	11.37	deg	[4] Section S, List S	
Right kingpin inclination	11.20	der		
angle	11.09	ucg	[4] Section 5, List 5	
Left X coord. Of KP @ center		mm		
(spindle trail)	-0.05	111111	[4] Section S, List S	
Right X coord. Of KP @	<b>7</b> 79	mm		
center (spindle trail)	-7.73	111111	[4] Section S, List S	
Left caster angle	2.81	deg	[4] Section S, List S	
Right caster angle	3.43	deg	[4] Section S, List S	

 Table A.12: Kingpin Geometry Parameters

### Appendix B

### **Driver Models**

#### B.1 Driver Model A

This model was briefly explained in Section 3.2 with a general block diagram and the key parameters of the model. This section takes a more in-depth look at the actual MATLAB/Simulink block diagram and how it operates. The general block diagram is once again shown here in Figure B.1. Driver model A uses a look-up table for both commanded lateral position (Y) and heading angle ( $\psi$ ). Lateral position error is then divided by the preview distance (L) and that quotient and yaw angle error are passed through the driver transfer function which outputs a steer angle so the vehicle dynamics can be calculated.

Figure B.2 contains the actual Simulink block diagram used for analysis. Starting at the "ISO Lane Change Course" both the lateral position error and heading angle error



Figure B.1: Original Genta Driver Model Block Diagram

are being used by the driver model to determine the next time step's output. Then the lateral position is divided by the preview distance, which in the MATLAB code running the Simulink model, is given by a time and converted to a distance based on speed of the maneuver. Then the terms are added together and passed through the driver model which includes the gain  $(K_d)$ , time delay  $(\tau)$ , lead  $(T_L)$  and lag  $(T_D)$  transfer function, then through a rate and a saturation limiter.

The rate limiter block is intended to simulate the rate at which a driver can input steering wheel angle into the vehicle. This value was obtained from Day and Metz [25] and is set to a value of 720 degrees/second for hand wheel angle. However, since this model is using road wheel angle as an input into CarSim, the steering rate (720 deg/sec) is divided by the steering ratio (13.8:1) provided in the K&C report [4]. The saturation block is intended to replicate the physical limits of the steering system by limiting the maximum hand wheel angle to 405 degrees [4] which is then divided by the steering ratio to get the value in terms of road wheel angle. The steer angle is then passed into CarSim as road wheel angle and CarSim then computes all of the vehicle dynamics. The outputs from CarSim are: vehicle heading angle, side slip angle, forward velocity, yaw rate, lateral acceleration, lateral position of the vehicle in the global reference frame, steering wheel torque, and slip angles at each of the tires. Some of these variables are stored in the workspace and used for post-processing in MATLAB while others are used as feedback for the next time step. Then the X-position of the vehicle is calculated based on the forward velocity  $(V_x)$ , heading angle  $(\psi)$ , and side slip angle  $(\beta)$  so the model can determine the next commanded Y-position and heading angle. Finally, each of the quadratic cost functions are computed and sent to the workspace for post-processing.



Figure B.2: Driver Model A Simulink/CarSim Block Diagram

#### B.2 Driver Model B

This section presents a simplified version of driver model A presented in Section 3.2. Genta explains that Equation 3.1 can be simplified by neglecting the lead time and combining all lags and delays into  $\tau$  [15]. Reducing Equation 3.1 by this method yields Equation B.1.

$$\frac{\delta(s)}{U(s)} = K_d e^{-\tau s} \tag{B.1}$$

Next,  $e^{-\tau s}$  can be approximated as a power series and truncated to only the linear term, ignoring higher order terms. Rewriting the equation and substituting  $\psi$  for U(s) gives Equation B.2.

$$\tau \delta(t) + \delta(t) = -K_d \left[ \psi(t) - \psi_0(t) \right] \tag{B.2}$$

In Equation B.2,  $\tau$  is the time delay of the driver,  $\delta$  is the steer angle provided by the driver,  $K_d$  is the gain,  $\psi$  is the heading angle of the vehicle, and  $\psi_0$  is the commanded heading angle.

Finally, Genta proposes adding a term to account for the predictive behavior of the driver. This term is in the form of a distance the driver is looking ahead of his position (i.e., the preview distance). Equation B.3 is the control law used in the Simulink model which is used in co-simulation with CarSim and includes the driver preview distance.

$$\tau \dot{\delta}(t) + \delta(t) = -K_d \left[ \psi(t) - \psi_0(t) + \frac{\Delta y(t)}{L} \right]$$
(B.3)

In Equation B.3,  $\Delta y(t)$  is the lateral deviation of the C.G. of the vehicle from the prescribed path and L is the preview distance.

The benefit of driver model B is that it is very simple with only a time delay, gain, and look ahead distance. The drawback to this driver model is that the driver does not have any lead which makes the model very slow to react. Because of this drawback the model is considered unacceptable for this thesis.

. Equation B.3 is the control law used in the MATLAB/Simulink model. This differential equation uses the difference between desired heading angle and actual heading angle, the difference between desired lateral position of the C.G. and actual lateral position of the C.G., and the current steering wheel angle as inputs. The output of the control law is steering wheel angle rate which is integrated and passed into CarSim which calculates the vehicle dynamics. Some of the output variables are then used in the next time step for the driver model .



Figure B.3: Driver Model B Simulink/CarSim Block Diagram

#### B.3 Driver Model C

Another driver model evaluated in this thesis is one presented by Horiuchi and Yuhara [16]. As with the model presented in Equation 3.1, this model contains gain, lead, lag, and time delay terms. The authors propose a closed loop system as shown in Figure B.4. The outer loop "is needed to make the vehicle accurately follow the desired path, while the inner loop that feeds back yaw angles is necessary to give appropriate damping to the closed loop system [16]."

Compared to driver model A, this model has separate gains  $(K_y \text{ and } K_{\psi})$  and lead terms  $(T_{Ly} \text{ and } T_{L\psi})$  for both the lateral position and the heading angle. It also contains a time delay  $(e^{-\tau s})$  and a first order lag  $(T_1)$  that the authors say are due to central nervous processing time and dynamics of the muscular system. An interesting point that the authors make about the delays is that they are independent of the task the driver is performing, therefore their values are considered constant. According to Horiuchi and Yuhara, the driver will adjust his or her own gains and leads to achieve the desired closed loop system.

The transfer function for the lateral position is given in Equation B.4 below.

$$\frac{\psi(s)}{y_{error}(s)} = K_y \left( T_{Ly} s + 1 \right) \tag{B.4}$$

The transfer function for the yaw angle is given in Equation B.5 below.

$$\frac{\delta(s)}{\psi_{error}(s)} = K_{\psi} \left( T_{L\psi} + 1 \right) \frac{e^{-\tau s}}{T_1 s + 1} \tag{B.5}$$

For driver model C, the benefits are the driver model maintains the course very well as compared to the two previous driver models A and B. The drawback for model C is that it has more parameters than the previous two. This makes it more difficult to optimize the driver parameters for the double lane change maneuver. Because of this drawback, this model was not chosen for the analysis in this thesis.

Figure B.5 shows the MATLAB/Simulink model. The model works by inputting

the commanded lateral position and finding the error between that and the actual lateral position which is then divided by the preview distance (L) and passed through the lateral position transfer function. That value is then added to the heading angle error and passed through the heading angle transfer function. As with the driver model A, this model also uses a rate limiter to mimic the rate at which a real driver can input steering wheel angle. The value of this block is 720 degrees/second [25] divided by the steering ratio (13.8:1) [4]. A saturation block is also included to mimic the physical limits of the steering system which is considered to be 405 degrees of hand wheel angle [4] divided by the steering ratio. This steer angle is then sent to CarSim which computes all of the vehicle dynamics then outputs those variables back into Simulink. Some of those parameters are written to the workspace for post-processing in MATLAB while others are used for the next time step within the model. The next X-position for the model is calculated using the forward velocity  $(V_x)$ , yaw angle  $(\psi)$ , and side slip angle  $(\beta)$  which is then sent into the ISO Lane Change course to find the next commanded Y-position and heading angle.



Figure B.5: Driver Model C Simulink/CarSim Block Diagram



Figure B.4: Horiuchi Driver/Vehicle Closed Loop System

### Appendix C

# Methodology for Creating the TWEELs<sup>TM</sup>

One of the methods used to change the vehicle was by fitting it with TWEELs<sup>™</sup> but since they do not currently exist for the Mini Cooper, hypothetical ones had to be created based on comments from Michelin [10] and data from the BMW 3 Series TWEEL<sup>™</sup>. The only characteristic of the TWEEL<sup>™</sup> being investigated in this work is the load sensitivity which affects the cornering stiffness versus load plot. Michelin characterizes tire force and moment curves using coefficients which can then be used in Pacejka's Magic Formula [3]. Pacejka's equations for lateral force, as Michelin uses them, are presented in Equation C.1

$$Y = D \cdot \sin \left( C \cdot \arctan \left( B \cdot phi \right) \right)$$

$$phi = (1 - E) \cdot delta + \frac{E}{B} \arctan \left( B \cdot delta \right)$$

$$D = (a_1 \cdot Z + a_2) \cdot Z$$

$$BCD = \left( a_3 \cdot \sin \left( 2 \cdot \arctan \left( \frac{Z}{a_4} \right) \right) \right) (1 - a_5 \cdot |gamma|)$$

$$B = \frac{BCD}{(C \cdot D)}$$

$$C = a_0$$

$$E = a_6 \cdot Z + a_7$$
(C.1)

where Y is the lateral force in Newtons, delta is the slip angle of the tire in degrees, Z is the vertical load in kilo-Newtons, gamma is the camber angle in degrees, and  $a_i$  are the coefficients provided by Michelin. Note that this formulation eliminates offsets due to conicity and plysteer at zero slip angle. The part that is of interest for this thesis is BCD which is the cornering stiffness. Assuming camber is zero, it is evident that the cornering stiffness depends on the coefficients  $a_3$ ,  $a_4$ , and the vertical load. From the 3 Series TWEEL<sup>TM</sup> the coefficients are  $a_3 = 1392.9544$  and  $a_4 = 8.9047$ . Using these coefficients yields a curve similar to what is desired for the Mini but slightly lower overall cornering stiffness than desired. The goal was to create three TWEELs<sup>TM</sup> with each having one of the following three properties: the same overall cornering stiffness as the OE tire (front + rear) while changing the split (front/rear), increased overall cornering stiffness from the OE tire while changing the split, and raising the overall goal for each of these TWEELs<sup>TM</sup> was to make them a realistic change in cornering stiffness that Michelin could achieve and would provide desirable handling characteristics for the vehicle.

Using the BMW 3 Series TWEEL<sup>TM</sup> as a baseline,  $a_3$  and  $a_4$  were modified by certain percentages iteratively to find the desired cornering stiffness versus load curve. Equation C.2 shows how  $a_3$  and  $a_4$  were modified to achieve the different hypothetical TWEELs<sup>TM</sup>. It was found in this range of  $a_3$  and  $a_4$  that  $a_3$  controls the overall cornering stiffness and that  $a_4$  controls the slope of the curve. Using this method yields Figure 3.7.

$TWEEL^{\rm TM}A$	$a_3 = 1392.9544 \times 0.94$	$a_4 = 8.9047 \times 0.60$	
$TWEEL^{\rm tm}B$	$a_3 = 1392.9544 \times 1.20$	$a_4 = 8.9047 \times 0.76$	(C.2)
$TWEEL^{\mathrm{TM}}C$	$a_3 = 1392.9544 \times 1.15$	$a_4 = 8.9047 \times 0.55$	

The aligning moment used for the Mini was the BMW 3 Series TWEEL<sup>TM</sup> aligning moment because Michelin has not provided much detail of the Mini TWEEL's<sup>TM</sup> aligning moment characteristics. It is anticipated that the Mini TWEEL<sup>TM</sup> will be similar to the 3 Series TWEEL<sup>TM</sup> because the same property that influences the load sensitivity will govern the aligning moment. The shear band, as shown in Figure ?? is very stiff compared to a standard pneumatic tire and therefore deforms less. This causes a shorter moment arm and therefore produces less aligning moment than a pneumatic tire.

The Pacejka Magic Formula equations which Michelin uses for the aligning moment are shown in Equation C.3

$$N = D \cdot \sin \left( C \cdot \arctan \left( B \cdot phi \right) \right)$$

$$phi = (1 - E) \cdot delta + E/B \arctan \left( B \cdot delta \right)$$

$$D = (c_1 \cdot Z + c_2) \cdot Z$$

$$BCD = (c_3 \cdot Z + c_4) \cdot Z \cdot e^{-c_5 \cdot Z} \cdot (1 - c_6 \cdot |gamma|)$$

$$B = \frac{BCD}{C \cdot D}$$

$$C = c_0$$

$$E = (c_7 \cdot Z^2 + c_8 \cdot Z + c_9) \cdot (1 - c_{10} \cdot |gamma|)$$
(C.3)

where N is the aligning moment in Newton-meters, delta is the slip angle of the tire in degrees, Z is the vertical load on the tire in kilo-Newtons, gamma is the camber angle in degrees, and  $c_i$  are the coefficients provided by Michelin. Using Equation C.3 the aligning moment curves used in CarSim are generated and shown in Figure C.1. In Figure C.1, it should be noted that each curve corresponds to increasing vertical load; therefore as more vertical load is applied more aligning moment is created by the tire.



Figure C.1: BMW 3 Series Aligning Moment Used for Mini Cooper in CarSim

### Appendix D

## **Open and Closed-Loop Raw Data**

In this appendix, the data used in Section 4.4 for drawing relationships between open-loop and closed-loop handling will be presented. Table D.1 shows all of the open-loop handling metrics for each of the twelve cases. Table D.2 contains the raw values for each of the quadratic cost functions. The double lane change was conducted at 77 kph for all cases in order to remove speed as a variable. Table D.3 contains the entire list of normalized quadratic metrics. Each case is normalized to the OE vehicle (tire and weight distribution). This is the data used to create the relationships presented in Section 4.4.

	SS Yaw	Yaw Rate	Yaw Rate	Lateral
Weight Dist., Tire	Rate Gain	Bandwidth	Damping	Acceleration Phase
	(1/s)	(Hz)	Ratio	Lag $@$ 1 Hz (deg)
60/40, OE Tire	0.42	2.35	0.48	-55.84
60/40, TWEEL <sup>TM</sup> A	0.42	2.52	0.48	-51.88
60/40, TWEEL <sup>TM</sup> B	0.45	2.61	0.49	-48.57
60/40, TWEEL <sup>TM</sup> C	0.42	3.10	0.47	-39.68
55/45, OE Tire	0.47	2.27	0.49	-61.53
55/45, TWEEL <sup>TM</sup> A	0.49	2.35	0.49	-60.32
55/45, TWEEL <sup>TM</sup> B	0.50	2.52	0.49	-54.85
55/45, TWEEL <sup>TM</sup> C	0.48	2.89	0.48	-47.06
50/50, OE Tire	0.51	2.12	0.49	-67.72
50/50, TWEEL <sup>TM</sup> A	0.56	2.04	0.49	-69.98
50/50, TWEEL <sup>TM</sup> B	0.56	2.27	0.50	-62.27
50/50, TWEEL <sup>TM</sup> C	0.53	2.61	0.49	-54.61

Table D.1: 4 Parameter Data for 12 Configurations

Table D.2: Raw Values of Quadratic Cost Functions at 77 kph  $\,$ 

Weight Dist., Tire	$J_1, m^2 \cdot sec$	$J_2, rad^2 \cdot sec$	$J_3, rad^2/sec$
60/40, OE Tire	1.17	2.98	84.95
60/40, TWEEL <sup>TM</sup> A	1.26	3.07	85.33
60/40, TWEEL <sup>TM</sup> B	1.07	2.80	79.68
60/40, TWEEL <sup>TM</sup> C	1.06	2.66	70.73
55/45, OE Tire	1.14	2.78	83.26
55/45, TWEEL <sup>TM</sup> A	1.21	2.92	87.97
55/45, TWEEL <sup>TM</sup> B	1.05	2.61	78.18
55/45, TWEEL <sup>TM</sup> C	1.01	2.46	69.16
50/50, OE Tire	1.05	4.87	180.07
50/50, TWEEL <sup>TM</sup> A	1.12	4.05	147.70
50/50, TWEEL <sup>TM</sup> B	0.98	3.84	152.97
50/50, TWEEL <sup>TM</sup> C	0.98	2.49	80.44

Table D.3: Normalized Values of Quadratic Cost Functions at 77 kph

Weight Dist., Tire	Normalized $J_1$	Normalized $J_2$	Normalized $J_3$
60/40, OE Tire	1.00	1.00	1.00
60/40, TWEEL <sup>TM</sup> A	1.08	1.03	1.00
60/40, TWEEL <sup>TM</sup> B	0.92	0.94	0.94
60/40, TWEEL <sup>TM</sup> C	0.90	0.89	0.83
55/45, OE Tire	0.97	0.93	0.98
55/45, TWEEL <sup>TM</sup> A	1.03	0.98	1.04
55/45, TWEEL <sup>TM</sup> B	0.89	0.88	0.92
55/45, TWEEL <sup>TM</sup> C	0.86	0.83	0.81
50/50, OE Tire	0.90	1.64	2.12
50/50, TWEEL <sup>TM</sup> A	0.96	1.36	1.74
50/50, TWEEL <sup>TM</sup> B	0.84	1.29	1.80
50/50, TWEEL <sup>TM</sup> C	0.84	0.84	0.95

#### Appendix E

### **CarSim File Formats**

When using CarSim for data analysis, whether it be the Spectrum Analyzer tool or the time history solver, it was necessary to post-process the data in an easy-to-read and presentable manner. To this end, the data from CarSim was imported into MATLAB for post-processing. CarSim uses two file formats for data storage–an ERD header file, developed by the University of Michigan at Ann Arbor, and a .bin file which is a binary file that contains all of the data. The data must then be imported into MATLAB so it can be used in a meaningful way. The steps for doing this follow with sample MATLAB code. This method was developed by Anup Khekare, a fellow graduate student working on the TWEEL<sup>TM</sup> project.

- Read the .bin file into MATLAB  $fid = fopen('FFT133\_Run1396.bin', 'r');$
- Decode the binary file *D=fread(fid, 'float32');*

At this point the data is contained in one long column with nothing distinguishing the different channels in the data. Therefore it is necessary to break the data into multiple rows of data containing things such as frequency, yaw rate gain, etc. In order to do this two things have to be known-the number of samples per channel and the number of channels in the ERD file. These two things can be found in the ERD file as shown in Figure E.1.

Number of Channels Number of Samples per Channel FFT133\_Run1361 ERDSILEV2.00 10, 110, 110, 40, 1, .250000 TITLE TWEEL #5 <Load Sensitive TWEEL--Swept Sine> Ο. , SHORTNAMFreq. Steer\_SWAVz Ay YawRateGYawRatePYawRateCAy GAy PAy с LONGNAMEFrequency PSD of Steering wheel angle PSD of Yaw rate (body-fixed), vePSD of Lateral acceleration, vehGain: AVz / Steer\_SW Phase: AVz / Steer\_SW Coherence: AVz / Steer\_SW Gain: Ay / Steer\_SW Phase: Ay / Steer\_SW Coherence: Ay / Steer\_SW UNITSNAMHz deg2/Hz deg/2/Hzg's2/Hz \*\*\* deg - g's/deg deg -GENNAME Frequency PSD of Angle PSD of Lateral acceleration PSD of Angular rate Transfer Function Gain Coherence Function Transfer Function Phase Transfer Function Gain Transfer Function Phase Coherence Function HISTORY CarSim 7.01 (Torsionally Rigid): Vehicle-suspension arrangement: i\_i HISTORY Input parameter file was C:\CarSim\_Data\Runs\Run1361\_all.par HISTORY Run made at 16:23 on Mar. 31, 2009 HISTORY PSD functions computed at 9:28 on May 6, 2009 RIGIBODY-(body-fixed) Venicia Vehicle yaw (body-fixed) Vehicle total CG 550000 , 574 Vehicle yaw Steering wheel total CG Vehicle yaw (body-fixed) Vehicle yaw (body-fixed) Vehicle yaw (body-fixed) Vehicle total CG 4183 , 2.28318 Vehicle total CG Vehicle total CG , 2.28318 .000000 .574183 2.28318 . .867586E-01, .000000 .000000 .000000 .000000 deg/s g's . .000000 RMSUNITS END

Figure E.1: Sample ERD file

The final part consists of converting this long column of data into the different channels. The MATLAB code is shown below. Once this is done the data can then be post-processed as necessary.

for m=1:110FFT133\_Run1396(:,m)=D(((10\*(m-1))+1):(10\*m)); end

#### Appendix F

## **MATLAB** Programs

#### F.1 Model\_Validation.m

% binary data in this format.

FFT\_115\_CURB\_21\_01=zeros(17,90);

for m=1:90 % 90 corresponds to the number of samples per channel

FFT\_115\_CURB\_21\_01(:,m)=D(((17\*(m-1))+1):(17\*m));

% 17 corresponds to the number of channels in the ERD file end

%% Read in the data from the CarSim FFT 115 CURB\_21\_02

fid = fopen('FFT115\_CURB\_21\_2\_20.bin','r');

% bin file associated with the Michelin's raw RWA data treated % in CARSIM

D=fread(fid,'float32');

% float32 has been used to decode, since by default CARSIM

% stores binary data in this format.

FFT\_115\_CURB\_21\_02=zeros(17,90);

for m=1:90 % 90 corresponds to the number of samples per channel

FFT\_115\_CURB\_21\_02(:,m)=D(((17\*(m-1))+1):(17\*m));

% 17 corresponds to the number of channels in the ERD file end

%% Read in the data from the CarSim FFT 115 CURB\_21\_03

fid = fopen('FFT115\_CURB\_21\_3\_20.bin','r');

% bin file associated with the Michelin's raw RWA data treated % in CARSIM

D=fread(fid,'float32'); % float32 has been used to decode,

% since by default CARSIM stores binary data in this format.

FFT\_115\_CURB\_21\_03=zeros(17,90);

for m=1:90 % 90 corresponds to the number of samples per channel

FFT\_115\_CURB\_21\_03(:,m)=D(((17\*(m-1))+1):(17\*m));

% 17 corresponds to the number of channels in the ERD file end

%% Read in the data from the CarSim FFT 115 CURB\_21\_04 fid = fopen('FFT115\_CURB\_21\_4\_20.bin','r'); % bin file associated with the Michelin's raw RWA data % treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format. FFT\_115\_CURB\_21\_04=zeros(17,90); for m=1:90 % 90 corresponds to the number of samples per % channel

FFT\_115\_CURB\_21\_04(:,m)=D((((17\*(m-1))+1):(17\*m));

% 17 corresponds to the number of channels in the ERD file end

%% Read in the data from the CarSim FFT 115 CURB\_21\_05 fid = fopen('FFT115\_CURB\_21\_5\_20.bin','r'); % bin file associated with the Michelin's raw RWA data % treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default % CARSIM stores binary data in this format. FFT\_115\_CURB\_21\_05=zeros(17,90); for m=1:90 % 90 corresponds to the number of samples % per channel

FFT\_115\_CURB\_21\_05(:,m)=D((((17\*(m-1))+1):(17\*m));

% 17 corresponds to the number of channels in the ERD file end

%% Average all of the LPG test data
FFT\_115\_CURB\_21\_Frequency\_Average= (FFT\_115\_CURB\_21\_05(1,:)+
FFT\_115\_CURB\_21\_04(1,:)+FFT\_115\_CURB\_21\_03(1,:)+...
FFT\_115\_CURB\_21\_02(1,:)+ FFT\_115\_CURB\_21\_01(1,:))/5;

```
FFT_115_CURB_21_YRGain_Average= (FFT_115_CURB_21_05(9,:)+
FFT_115_CURB_21_04(9,:)+FFT_115_CURB_21_03(9,:)+...
FFT_115_CURB_21_02(9,:)+ FFT_115_CURB_21_01(9,:))/5;
```

```
FFT_115_CURB_21_YRPhase_Average= (FFT_115_CURB_21_05(10,:)+
FFT_115_CURB_21_04(10,:)+FFT_115_CURB_21_03(10,:)+...
FFT_115_CURB_21_02(10,:)+ FFT_115_CURB_21_01(10,:))/5;
```

FFT\_115\_CURB\_21\_AyGain\_Average= (FFT\_115\_CURB\_21\_05(12,:)+
FFT\_115\_CURB\_21\_04(12,:)+FFT\_115\_CURB\_21\_03(12,:)+...
FFT\_115\_CURB\_21\_02(12,:)+ FFT\_115\_CURB\_21\_01(12,:))/5;

FFT\_115\_CURB\_21\_AyPhase\_Average= (FFT\_115\_CURB\_21\_05(13,:)+
FFT\_115\_CURB\_21\_04(13,:)+FFT\_115\_CURB\_21\_03(13,:)+...
FFT\_115\_CURB\_21\_02(13,:)+ FFT\_115\_CURB\_21\_01(13,:))/5;

% Read in the data from the CarSim FFT 114 Run1214

fid = fopen('FFT114\_Run1214.bin','r');

% bin file associated with the Michelin's raw RWA

% data treated in CARSIM

D=fread(fid,'float32');

% float32 has been used to decode, since by default % CARSIM stores binary data in this format. FFT114\_Run1214=zeros(10,110);

for m=1:110 % 104 corresponds to the number of samples % per channel

FFT114\_Run1214(:,m)=D(((10\*(m-1))+1):(10\*m));

% 6 corresponds to the number of channels in the ERD file end

%% Read in the data from the CarSim FFT 114 Run1349 fid = fopen('FFT115\_Run1349.bin','r'); % bin file associated with the Michelin's raw RWA % data treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default % CARSIM stores binary data in this format. FFT115\_Run1349=zeros(17,110); for m=1:110 % 110 corresponds to the number of samples % per channel

FFT115\_Run1349(:,m)=D(((17\*(m-1))+1):(17\*m));

% 17 corresponds to the number of channels in the ERD file end

```
%% Used for report to Michelin and BMW
%
% figure
% subplot(2,2,1)
% plot(FFT114_Run1214(1,:),FFT114_Run1214(5,:),'-b',...
% FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_YRGain_Average,...
% '-g',FFT115_Run1349(1,:),FFT115_Run1349(9,:),'-r','linewidth',1.5)
% ylabel('Yaw Rate Gain (1/s)','fontweight','bold','fontsize',12)
% legend('CarSim @ 2.1 bar Current TF','LPG Test @ 2.1 bar',...
%'CarSim @ 2.1 bar Previous TF', 'location', 'southwest')
% grid on
% xlim([0.25 4])
% title('Yaw Rate Gain/Road Wheel Angle (C+D)', 'fontweight',...
% 'bold', 'fontsize',12)
%
% subplot(2,2,3)
% plot(FFT114_Run1214(1,:),FFT114_Run1214(6,:),'-b',
% FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_YRPhase_Average,...
%'-g',FFT115_Run1349(1,:), FFT115_Run1349(10,:),'-r','linewidth',1.5)
% xlabel('Frequency (Hz)', 'fontweight', 'bold', 'fontsize',12)
% ylabel('Yaw Rate Phase (deg)','fontweight','bold','fontsize',12)
% legend('CarSim @ 2.1 bar Current TF','LPG Test @ 2.1 bar',...
%'CarSim @ 2.1 bar Previous TF', 'location', 'southwest')
% grid
% xlim([0.25 4])
% title('Yaw Rate Phase/Road Wheel Angle (C+D)','fontweight',...
%'bold','fontsize',12)
%
```

```
% subplot(2,2,2)
% plot(FFT114_Run1214(1,:),FFT114_Run1214(8,:),'-b',...
% FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_AyGain_Average,...
% '-g',FFT115_Run1349(1,:),FFT115_Run1349(12,:),'-r','linewidth',1.5)
% ylabel('Lateral Accel. Gain (g/deg)','fontweight','bold',...
% 'fontsize',12)
% legend('CarSim @ 2.1 bar Current TF','LPG Test @ 2.1 bar',...
% 'CarSim @ 2.1 bar Previous TF', 'location', 'northeast')
% grid
% xlim([0.25 4])
% title('Lateral Accel. Gain/Road Wheel Angle (C+D)',...
% 'fontweight', 'bold', 'fontsize', 12)
%
% subplot(2,2,4)
% plot(FFT114_Run1214(1,:),FFT114_Run1214(9,:),'-b',...
% FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_AyPhase_Average,...
% '-g',FFT115_Run1349(1,:),FFT115_Run1349(13,:),'-r','linewidth',1.5)
% xlabel('Frequency (Hz)', 'fontweight', 'bold', 'fontsize',12)
% ylabel('Lateral Accel. Phase (deg)','fontweight','bold',...
% 'fontsize',12)
% legend('CarSim @ 2.1 bar Current TF','LPG Test @ 2.1 bar',...
% 'CarSim @ 2.1 bar Previous TF', 'location', 'northwest')
% grid
% xlim([0.25 4])
% title('Lateral Accel. Phase/Road Wheel Angle (C+D)',...
% 'fontweight', 'bold', 'fontsize',12)
% print -djpeg -r300 FRF_21barC_D
% % % %
```

```
% data=[-55.9 2.49 7.23 0.49;-60.3 2.00 7.36 0.49;-60.7 1.6 9.6 0.5]';
% legend1={'CarSim @ 2.1 bar Current TF', 'LPG Test @ 2.1 bar',...
% 'CarSim @ 2.1 bar Previous TF'};
% axes_label = {'Lateral Accel Phase @ 1 Hz' '(deg)';...
% 'Yaw Rate Bandwidth' '(Hz)'; 'SS Yaw Rate Gain' '(deg/s)/deg';...
% 'Yaw Rate Damping Ratio' "};
% [f, o] = spider(data,'LPG Test Data Compared to CarSim ...
% Simulation (C+D)',
% [-80 -40 ;0 3.5;0 10;0 0.6],axes_label,legend1);
% print -djpeg -r300 spider_21barC_D
%
% Used for Thesis validation
figure
subplot(2,2,1)
plot(FFT114_Run1214(1,:),FFT114_Run1214(5,:),'-.b',...
FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_YRGain_Average,...
'-k','linewidth',1.5)
ylabel('Yaw Rate/RWA, Gain (1/s)','fontweight','bold',...
'fontsize',12)
legend('Simulation', 'Track Test', 'location', 'southwest')
grid on
xlim([0.25 4])
% title('Yaw Rate Gain/Road Wheel Angle', 'fontweight', 'bold',...
'fontsize',12)
```

```
subplot(2,2,3)
plot(FFT114_Run1214(1,:),FFT114_Run1214(6,:),'-.b',...
FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_YRPhase_Average...
,'-k','linewidth',1.5) %
xlabel('Frequency (Hz)','fontweight','bold','fontsize',12)
ylabel('Yaw Rate/RWA, Phase (deg)','fontweight','bold',...
'fontsize',12)
legend('Simulation','Track Test','location','southwest')
grid
xlim([0.25 4]) %
title('Yaw Rate Phase/Road Wheel Angle','fontweight','bold',...
'fontsize',12)
```

```
subplot(2,2,2)
plot(FFT114_Run1214(1,:),FFT114_Run1214(8,:),'-.b',...
FFT_115_CURB_21_Frequency_Average,FFT_115_CURB_21_AyGain_Average...
,'-k','linewidth',1.5)%
ylabel('Lateral Accel./RWA, Gain (g/deg)','fontweight','bold',...
'fontsize',12)
legend('Simulation','Track Test','location','northeast')%
grid
xlim([0.25 4]) %
title('Lateral Accel. Gain/Road Wheel Angle','fontweight',...
'bold','fontsize',12)
```

```
subplot(2,2,4)
plot(FFT114_Run1214(1,:),FFT114_Run1214(9,:),'-.b',...
FFT_115_CURB_21_Frequency_Average, FFT_115_CURB_21_AyPhase_Average,...
```

```
'-k','linewidth',1.5)%
xlabel('Frequency (Hz)', 'fontweight', 'bold', 'fontsize',12)
ylabel('Lateral Accel./RWA, Phase (deg)','fontweight','bold',...
'fontsize',12)
legend('Simulation', 'Track Test', 'location', 'northwest')%
grid xlim([0.25 4]) %
title('Lateral Accel. Phase/Road Wheel Angle','fontweight',...
'bold','fontsize',12)
print -djpeg -r100 FRF_Thesis_Sim_LPG
data=[-55.9 2.49 7.23 0.49;-60.3 2.00 7.36 0.49]'; %
legend1={'Simulation','Track Test'};%
axes_label = {'Lateral Accel Phase @ 1 Hz' '(deg)';...
'Yaw Rate Bandwidth' '(Hz)';'SS Yaw Rate Gain' '(deg/s)/deg';...
'Yaw Rate Damping Ratio' "};
[f, o] = spider(data,",[-80 -40 ;0 3.5;0 10;0 0.6],axes_label,legend1);
```

#### F.2 TWEEL\_Lateral\_Force.m

#### 

%Lateral Tire force estimation %Lateral Force Symbol = Y, Units = N %Slip Angle Symbol = delta, Units = deg %Vertical Load Symbol = Z, Units = kN %Camber Angle Symbol = gamma, Units = deg %Pajecka coefficients for OE Mini Goodyear Tire

a0=1.2252;

a1=-40.0094;

a2=1114.4301;

a3=1992.8228;

a4=10.2723;

a5=0.00531;

a6=-0.54271;

a7=1.0502;

```
a8=-0.072587;
```

```
a9=-0.015939;
```

a10=-0.19808;

```
a111=15.4667;
```

```
a112=-3.5313;
```

```
a12=37.9298;
```

```
a13=-43.3789;
```

```
%%
```

%Pacjeka Magic Formula for Mini OE tire Z=0.5:0.5:7; % range of vertical loads ZN=Z\*1000; %vertical loads in N

```
delta=0:0.1:9.1; %slip angle
```

```
gamma_OE=0; % camber angle
```

```
Y_OE=zeros(length(delta),length(Z)); %preallocate
for i=1:length(delta)
for k=1:length(Z)
    D_OE=((a1*Z(k))+a2)*Z(k);
    BCD_OE=(a3*sin(2*atan(Z(k)/a4)))*(1-a5*abs(gamma_OE));
    C_OE=a0;
    B_OE=BCD_OE/(C_OE*D_OE);
    E_OE=a6*Z(k)+a7;
    phi_OE(i)=(1-E_OE)*(delta(i))+E_OE/B_OE*atan(B_OE*(delta(i)));
    Y_OE(i,k)=D_OE*sin(C_OE*atan(B_OE*phi_OE(i)));
```

 $\operatorname{end}$ 

```
end
```

```
figure
```

```
plot(delta,Y_OE,'linewidth',1.5)
ylabel('Lateral Force, N','fontweight','bold')
xlabel('Slip Angle, deg','fontweight','bold')
%title('OE Tire Lateral Force versus Slip Angle','fontweight','bold')
legend('0.5 kN','1.0 kN','1.5 kN','2.0 kN','2.5 kN', '3.0 kN',...
'3.5 kN', '4.0 kN','4.5 kN','5.0 kN','5.5 kN','6.0 kN','6.5 kN',...
'7.0 kN','Location','best')
grid on
```
%% Pacjeka Magic Formula for TWEEL B

gamma\_1=0; %camber change versus load

a3\_run1=1392.9544; % Pacejka coefficients a4\_run1=8.9047; % for the 3 Series TWEEL

x3\_1=1.2; %percent changing original Pacejka
x4\_1=0.76; %coefficients for 3 Series TWEEL

a3\_1=a3\_run1\*x3\_1; % Modifiied Pacejka coefficients
a4\_1=a4\_run1\*x4\_1; % for a new load insensitive TWEEL

```
Y_1=zeros(length(delta),length(Z)); %preallocate
for i=1:length(delta) %
for k=1:length(Z)
    D_1=((a1*Z(k))+a2)*Z(k);
    BCD_1=(a3_1*sin(2*atan(Z(k)/a4_1)))*(1-a5*abs(gamma_1));
    C_1=a0;
    B_1=BCD_1/(C_1*D_1);
    E_1=a6*Z(k)+a7;
    phi_1=(1-E_1)*(delta(i))+E_1/B_1*atan(B_1*(delta(i)));
    Y_1(i,k)=D_1*sin(C_1*atan(B_1*phi_1));
```

 $\operatorname{end}$ 

```
figure
```

```
plot(delta,Y_1,'linewidth',1.5)
ylabel('Lateral Force, N','fontweight','bold')
xlabel('Slip Angle, deg','fontweight','bold')
title('Load Sensitive TWEEL B Lateral Force versus Slip Angle',...
'fontweight','bold')
legend('0.5 kN','1.0 kN','1.5 kN','2.0 kN','2.5 kN', '3.0 kN',...
'3.5 kN','4.0 kN','4.5 kN','5.0 kN','5.5 kN','6.0 kN','6.5 kN',...
'7.0 kN','Location','best')
grid on
```

%% Pacjeka Magic Formula for TWEEL C

gamma\_1=0; %camber change versus load

```
x3_2=1.15; % percent changing original Pacejka
x4_2=0.55; % coefficients for 3 Series TWEEL
a3_2=a3_run1*x3_2; % Modifiled Pacejka coefficients
a4_2=a4_run1*x4_2; % for a new load insensitive TWEEL
```

```
Y_2=zeros(length(delta),length(Z)); %preallocate
```

for i=1:length(delta)

```
for k=1:length(Z)
```

```
D_2=((a1*Z(k))+a2)*Z(k);
BCD_2=(a3_2*sin(2*atan(Z(k)/a4_2)))*(1-a5*abs(gamma_1));
C_2=a0;
B_2=BCD_2/(C_2*D_2);
E_2=a6*Z(k)+a7;
```

```
phi_2=(1-E_2)*(delta(i))+E_2/B_2*atan(B_2*(delta(i)));
Y_2(i,k)=D_2*sin(C_2*atan(B_2*phi_2));
```

end

figure

```
plot(delta,Y_2,'linewidth',1.5)
ylabel('Lateral Force, N','fontweight','bold')
xlabel('Slip Angle, deg','fontweight','bold')
title('Load Sensitive TWEEL C Lateral Force versus Slip Angle',...
'fontweight','bold')
legend('0.5 kN','1.0 kN','1.5 kN','2.0 kN','2.5 kN', '3.0 kN',...
'3.5 kN', '4.0 kN','4.5 kN','5.0 kN','5.5 kN','6.0 kN','6.5 kN',...
'7.0 kN','Location','best')
grid on
```

%% Pacjeka Magic Formula for TWEEL A

gamma\_1=0; %camber change versus load

x3\_3=0.94; % percent changing original Pacejka x4\_3=0.6; % coefficients for 3 Series TWEEL

a3\_3=a3\_run1\*x3\_3; % Modifiied Pacejka coefficients a4\_3=a4\_run1\*x4\_3; % for a new load insensitive TWEEL

Y\_3=zeros(length(delta),length(Z)); %preallocate
for i=1:length(delta)

```
for k=1:length(Z)
```

```
D_3=((a1*Z(k))+a2)*Z(k);
BCD_3=(a3_3*sin(2*atan(Z(k)/a4_3)))*(1-a5*abs(gamma_1));
C_3=a0;
B_3=BCD_3/(C_3*D_3);
E_3=a6*Z(k)+a7;
phi_3=(1-E_3)*(delta(i))+E_3/B_3*atan(B_3*(delta(i)));
Y_3(i,k)=D_3*sin(C_3*atan(B_3*phi_3));
```

end

## figure

```
plot(delta,Y_3,'linewidth',1.5)
ylabel('Lateral Force, N','fontweight','bold')
xlabel('Slip Angle, deg','fontweight','bold')
title('Load Sensitive TWEEL A Lateral Force versus Slip Angle',...
'fontweight','bold')
legend('0.5 kN','1.0 kN','1.5 kN','2.0 kN','2.5 kN', '3.0 kN',...
'3.5 kN', '4.0 kN','4.5 kN','5.0 kN','5.5 kN','6.0 kN','6.5 kN',...
'7.0 kN','Location','best')
grid on
```

## %%

```
%calculate mu Fy/Fz
% calculate
for j=1:k
```

```
latss1(j)=Y_1(92,j);
```

```
mu_1(j)=latss1(j)/ZN(j);
latss2(j)=Y_2(92,j);
mu_2(j)=latss2(j)/ZN(j);
latss3(j)=Y_3(92,j);
mu_3(j)=latss3(j)/ZN(j);
```

 $\operatorname{end}$ 

```
muavg_1=mean(mu_1);
muavg_2=mean(mu_2);
muavg_3=mean(mu_3);
```

```
fprintf('Mu average for TWEEL #6, Fy/Fz %.3f \n',muavg_3)
disp(' ')
fprintf('Mu average for TWEEL #7, Fy/Fz %.3f \n',muavg_1)
disp(' ')
fprintf('Mu average for TWEEL #8, Fy/Fz %.3f \n',muavg_2)
disp(' ')
```

% plot Calpha vs. load for each of the tires

for i=1:length(Z)

```
BCD_0E(i)=(a3*sin(2*atan(Z(i)/a4)));
```

```
end
```

```
for i=1:length(Z)
```

BCD\_1(i)=(a3\_1\*sin(2\*atan(Z(i)/a4\_1)));

```
for i=1:length(Z)
```

```
BCD_2(i)=(a3_2*sin(2*atan(Z(i)/a4_2)));
```

```
for i=1:length(Z)
```

```
BCD_3(i)=(a3_3*sin(2*atan(Z(i)/a4_3)));
```

end

```
p_OE=polyfit(Z,BCD_OE,2); % find the polynomial
p_1=polyfit(Z,BCD_1,2); % fits to the Calpha
p_2=polyfit(Z,BCD_2,2); % vs. load curves
p_3=polyfit(Z,BCD_3,2);
```

```
fit_OE=polyval(p_OE,Z); % uses the fit to find the
fit_1=polyval(p_1,Z); % Calpha vs. load and
fit_2=polyval(p_2,Z); % compare to the actual curve
fit_3=polyval(p_3,Z);
```

```
figure
subplot(2,2,1)
plot(Z,BCD_OE,Z,fit_OE)
grid on
legend('Original Curve','Fit','location','best')
```

subplot(2,2,2)
plot(Z,BCD\_1,Z,fit\_1)

```
grid on
legend('Original Curve', 'Fit', 'location', 'best')
subplot(2,2,3)
plot(Z,BCD_2,Z,fit_2)
grid on
legend('Original Curve','Fit','location','best')
subplot(2,2,4)
plot(Z,BCD_3,Z,fit_3)
grid on
legend('Original Curve','Fit','location','best')
% disp('C2 for OE')
% disp(p_OE(1))
% disp(' ')
% disp('C2 for TWEEL #3')
% disp(p_1(1))
% disp(' ')
% disp('C2 for TWEEL #4')
% disp(p_2(1))
% disp(' ')
% percent_1=(1-p_1(1)/p_OE(1))*100;
% percent_2=(1-p_2(1)/p_OE(1))*100;
```

```
%mark each load condition on the plot
x1=ones(1,100)*2600; %vertical line to show Rear C+D
```

```
y1=linspace(800,1400);
```

x2=ones(1,100)\*3300; %vertical line to show Rear GVW
y2=linspace(1000,1600);
x3=ones(1,100)\*3900; %vertical line to show Front C+D

y3=linspace(1000,1700);

x4=ones(1,100)\*4300; %vertical line to show Front GVW
y4=linspace(1100,1700);

```
% figure
```

```
% Figure
% plot(ZN,BCD_OE,'k-',ZN,BCD_3,'b-.',ZN,BCD_1,'g--',ZN,BCD_2,...
% 'r:',x1,y1,'-k',x3,y3,'-k','linewidth',2)
% xlabel('Vertical Load Fz (N)','fontweight','bold','fontsize',13)
% ylabel('Cornering Stiffness (N/deg)','fontweight',...
% 'bold','fontsize',13)
% title('Cornering Stiffness vs Vertical Load',...
% 'fontweight','bold','fontsize',13)
% grid on
% legend('OE Mini Tire','TWEEL A','TWEEL B','TWEEL C',...
% 'location','southeast')
% text(1000,1700,'Front C+D','fontweight','bold','fontsize',13)
% text(3800,800,'Rear GVW','fontweight','bold','fontsize',13)
% text(5400,1100,'Front GVW','fontweight','bold','fontsize',13)
% text(3300,500,'Rear C+D','fontweight','bold','fontsize',13)
```

% print -djpeg -r200 Calpha\_vs\_load\_TWEEL\_Thesis

## 

### figure

```
plot(ZN,BCD_OE,'k-',ZN,BCD_3,'b-.',ZN,BCD_1,'g--',ZN,BCD_2,...
```

```
'r:',x1,y1,'-k',x3,y3,'-k','linewidth',2)
```

xlabel('Vertical Load Fz (N)','fontweight','bold','fontsize',13)
ylabel('Cornering Stiffness (N/deg)','fontweight','bold','fontsize',13)
% title('Cornering Stiffness vs Vertical Load','fontweight',...
'bold','fontsize',13)

grid on

```
CalphaF=(a3*sin(2*atan(3.9/a4)));
CalphaR=(a3*sin(2*atan(2.6/a4)));
CalphaF1=(a3_1*sin(2*atan(3.9/a4_1)));
CalphaR1=(a3_1*sin(2*atan(2.6/a4_1)));
CalphaF2=(a3_2*sin(2*atan(3.9/a4_2)));
CalphaR2=(a3_2*sin(2*atan(2.6/a4_2)));
CalphaF3=(a3_3*sin(2*atan(3.9/a4_3)));
```

```
disp(' ')
fprintf('Total Front and Rear Cornering Stiffness Nominal
OE Tire, N/deg %. 1f \n',Calp haF+CalphaR)
disp(' ')
fprintf('Front/Rear Split of Cornering Stiffness Nominal
OE Tire %.3f \n',CalphaF/CalphaR)
```

```
disp(' ')
fprintf('Total Front and Rear Cornering Stiffness
TWEEL B, N/deg %.1f \n',CalphaF1+CalphaR1)
disp(' ')
fprintf('Front/Rear Split of Cornering Stiffness
TWEEL B %.3f \n',CalphaF1/CalphaR1)
disp(' ')
fprintf('Total Front and Rear Cornering Stiffness
TWEEL C, N/deg %.1f \n',CalphaF2+CalphaR2)
disp(' ')
fprintf('Front/Rear Split of Cornering Stiffness
TWEEL C %.3f \n',CalphaF2/CalphaR2)
disp(' ')
fprintf('Total Front and Rear Cornering Stiffness
TWEEL A, N/deg %.1f \n',CalphaF3+CalphaR3)
disp(' ')
fprintf('Front/Rear Split of Cornering Stiffness
TWEEL A %.3f \n',CalphaF3/CalphaR3)
disp(' ')
```

# F.3 TWEEL\_Aligning\_Moment.m

### 

close all clc

%Aligning Torque

%Aligning Torque Symbol = N, Units = N-m %Slip Angle Symbol = delta, Units = deg %Vertical Load Symbol = Z, Units = kN %Camber Angle Symbol = gamma, Units = deg %Pajecka coefficients for 3 Series TWEEL

c0\_18=3.2132; c1\_18=2.1634;

- c2\_18=1.2014;
- c3\_18=-0.79116;
- c4\_18=-1.4124;
- c5\_18=0;
- c6\_18=0;
- c7\_18=-0.16576;
- c8\_18=0.86122;
- c9\_18=-1.4411;
- c10\_18=-0.091491;
- c11\_18=0.22728;
- c12\_18=-0.022053;
- c13\_18=0.11096;
- c14\_18=0.42018;
- c15\_18=-2.6859;
- c16\_18=0.39968;
- c17\_18=2.4225;
- %% Run 1

Z=0.75:0.5:5.75; % range of vertical loads
ZN=Z\*1000; %vertical load in N
delta=0:0.1:9.1; %slip angle
tranpdelta=delta';

%Aligning Moment calculations using Pacekja formulas for i=1:length(delta)

for k=1:length(Z)
gamma(k)=0;

```
D=((c1_18*Z(k))+c2_18)*Z(k);
BCD=(c3_18*Z(k)+c4_18)*Z(k)*exp(-c5_18*Z(k))...
*(1-c6_18*abs(gamma(k)));
C=c0_18;
B=BCD/(C*D);
E=(c7_18*Z(k)^2+c8_18*Z(k)+c9_18)*(1-c10_18*abs(gamma(k)));
phi=(1-E)*delta(i)+E/B*atan(B*delta(i));
N_1(i,k)=-D*sin(C*atan(B*phi));
```

end

figure

```
plot(delta,N_1,'linewidth',2)
ylabel('Aligning Moment, N-m','fontweight','bold')
xlabel('Slip Angle, deg','fontweight','bold')
%title('TWEEL Run 1 Aligning Moment versus Slip Angle',...
% 'fontweight','bold')
legend('0.75 kN','1.25 kN','1.75 kN','2.25 kN','2.75 kN','3.25 kN',...
'3.75 kN','4.25 kN','4.75 kN','5.25 kN','5.75 kN','Location','best')
grid on
```

%% Run 2

%Aligning Moment calculations using Pacekja formulas

- c0\_24=3.3569;
- c1\_24=2.2141;
- c2\_24=0.95712;
- c3\_24=-0.78064;

- c4\_24=-1.3766;
- c5\_24=0;
- c6\_24=0;
- c7\_24=-0.13542;
- c8\_24=0.58656;
- c9\_24=-0.74952;
- c10\_24=-0.092824;
- c11\_24=0.21832;
- c12\_24=-0.019861;
- c13\_24=0.18277;
- c14\_24=0.37042;
- c15\_24=-2.4142;
- c16\_24=0.48819;
- c17\_24=4.963;

for i=1:length(delta)

```
for k=1:length(Z)
gamma(k)=0;
D=((c1_24*Z(k))+c2_24)*Z(k);
BCD=(c3_24*Z(k)+c4_24)*Z(k)*exp(-c5_24*Z(k))...
*(1-c6_24*abs(gamma(k)));
C=c0_24;
B=BCD/(C*D);
E=(c7_24*Z(k)^2+c8_24*Z(k)+c9_24)*(1-c10_24*abs(gamma(k)));
phi=(1-E)*delta(i)+E/B*atan(B*delta(i));
N_2(i,k)=-D*sin(C*atan(B*phi));
```

## figure

plot(delta,N\_2,'linewidth',2)

ylabel('Aligning Moment, N-m','fontweight','bold')

xlabel('Slip Angle, deg','fontweight','bold')

title('TWEEL Run 2 Aligning Moment versus Slip Angle',...

'fontweight', 'bold')

legend('0.75 kN','1.25 kN','1.75 kN','2.25 kN','2.75 kN','3.25 kN',...
'3.75 kN','4.25 kN','4.75 kN','5.25 kN','5.75 kN','Location','best')
grid on

%% Run 3

%Aligning Moment calculations using Pacekja formulas

- c0\_30=3.2817;
- c1\_30=2.2078;
- c2\_30=1.2771;
- c3\_30=-0.77012;
- c4\_30=-1.3408;
- c5\_30=0;
- c6\_30=0;
- c7\_30=-0.1052;
- c8\_30=0.30701;
- c9\_30=-0.54349;
- c10\_30=-0.075644;
- c11\_30=0.21944;
- c12\_30=0.00011484;
- c13\_30=0.12673;

- c14\_30=0.37153;
- c15\_30=-2.388;
- c16\_30=0.60807;
- c17\_30=5.9436;

```
for i=1:length(delta)
```

```
for k=1:length(Z)
gamma(k)=0;
D=((c1_30*Z(k))+c2_30)*Z(k);
BCD=(c3_30*Z(k)+c4_30)*Z(k)*exp(-c5_30*Z(k))...
*(1-c6_30*abs(gamma(k)));
C=c0_30;
B=BCD/(C*D);
E=(c7_30*Z(k)^2+c8_30*Z(k)+c9_30)*(1-c10_30*abs(gamma(k)));
phi=(1-E)*delta(i)+E/B*atan(B*delta(i));
N_3(i,k)=-D*sin(C*atan(B*phi));
```

```
end
```

```
figure
```

plot(delta,N\_3,'linewidth',2)

ylabel('Aligning Moment, N-m','fontweight','bold')

xlabel('Slip Angle, deg','fontweight','bold')

title('TWEEL Run 3 Aligning Moment versus Slip Angle',...

'fontweight', 'bold')

legend('0.75 kN','1.25 kN','1.75 kN','2.25 kN','2.75 kN','3.25 kN',... '3.75 kN','4.25 kN','4.75 kN','5.25 kN','5.75 kN','Location','best') grid on

# F.4 Spider Plot Programs

Written by Michael Arant of Michelin North America and a Clemson graduate student.

## F.4.1 color\_index.m

```
function [val] = color_index(len)
% get unique colors
% function [val] = color_index(len)
%
% inputs 1
% len number of objects class integer
%
% outputs 1
% val color vector class real
%
% michael arant
if nargin < 1 | nargout < 1; help color_index; error('I / O error');</pre>
end
if len == 1
    val = [0 \ 0 \ 0];
else
% initial color opsibilities (no white)
col = [0 0 0]
0 0 1
```

```
0 1 1
0 1 0
1 1 0
1 0 1
1 0 0];
switch len
```

```
case 1, col([2 3 4 5 6 7],:) = [];
case 2, col([1 2 3 5 6],:) = [];
case 3, col([1 3 5 6],:) = [];
case 4, col([3 5 6],:) = [];
case 5, col([5 6],:) = [];
case 6, col(6,:) = [];
```

```
% number of requested colors
val = zeros(len,3); val(:,3) = linspace(0,1,len)';
```

```
val(:,1) = interp1q(linspace(0,1,size(col,1))',col(:,1),val(:,3));
val(:,2) = interp1q(linspace(0,1,size(col,1))',col(:,2),val(:,3));
val(:,3) = interp1q(linspace(0,1,size(col,1))',col(:,3),val(:,3));
```

 $\operatorname{end}$ 

## F.4.2 isint.m

```
function [res] = isint(val)
% determines if value is an integer
% function [res] = isint(val)
```

```
%
    % inputs 1
    % val
               value to be checked
                                                class real
    %
    % outputs 1
    % res
               result (1 is integer, 0 is not) class integer
    %
                         may 15, 2004
    % michael arant
    if nargin < 1; help isint; error('I / O error'); end</pre>
    % check for real number
    if isreal(val) & isnumeric(val)
    \% check for integer
         if round(val) == val
            res = 1;
         else
            res = 0;
         end
    else
         res = 0;
    end
F.4.3 spider.m
    function [f, ca, o] = spider(data,tle,rng,lbl,leg,f)
```

```
% create a spider plot for ranking the data
% function [f, ca, o] = spider(data,tle,rng,lbl,leg,f)
%
```

```
% inputs 6 - 5 optional
% data
          input data (# axes (M) x # data sets (N))
                                                            class real
% tle
          spider plot title
                                                             class char
% rng
          peak range of the data (Mx1 or Mx2)
                                                             class real
% lbl
          cell vector axes names (Mxq) in [name unit] pairs class cell
          data set legend identification (1xN)
% leg
                                                             class cell
% f
          figure handle or plot handle
                                                             class real
%
% outptus 3 - 3 optional
% f
          figure handle
                                                          class integer
% x
          axes handle
                                                             class real
% 0
         series object handles
                                                             class real
%
% michael arant - jan 30, 2008
% data check
if nargin < 1; help spider; error('Need data to plot'); end
% size segments and number of cases
[r c] = size(data);
% exit for too few axes
if r < 3
    errordlg('Must have at least three measuremnt axes')
    error('Program Termination: Must have a minimum of three axes')
end
% title
```

```
if ~exist('tle','var') || isempty(tle) || ~ischar(tle)
```

```
tle = ";
```

```
% check for maximum range
if ~exist('rng','var') || isempty(rng) || ~isreal(rng)
 % no range given or range is in improper format
 % define new range
 rng = [min([min(data,[],2) zeros(r,1)],[],2) max(data,[],2)];
 % check for negative minimum values
 if ~isempty(ismember(-1,sign(data)))
 % negative value found - adjust minimum range
 for ii = 1:r
 % negative range for axis ii - set new minimum
 if min(data(ii,:)) < 0
 rng(ii,1) = min(data(ii,:)) - ...
 0.25 * (max(data(ii,:)) - min(data(ii,:)));
 end
 end
```

```
elseif size(rng,1) ~= r
% insuffent range definition
uiwait(msgbox(char('Range size must be Mx1 -
    number of axes x 1', ...
    sprintf('%g axis ranges defined, %g axes exist',
        size(rng,1),r))))
```

```
error(sprintf('%g axis ranges defined, %g axes exist',
       size(rng,1),r))
elseif size(rng,2) == 1
    % assume range is a maximum range - define minimum
    rng = sort([min([zeros(r,1) min(data,[],2) - ...
       0.25 * (max(data,[],2) - min(data,[],2))],[],2) rng],2);
% elseif size(rng,1) ~= r || size(rng,2) ~= 2
% % invalid range
% uiwait(msgbox('Range must have same number of rows
%
      as data set'))
  error('Invalid range specification')
%
end
% check for axis labels
if ~exist('lbl','var') || isempty(lbl)
    \% no labels given - define a default lable
    lbl = cell(r,1);
       for ii = 1:r;
           lbl(ii) = cellstr(sprintf('Axis %g',ii));
       end
elseif size(lbl,1) ~= r
    if size(lbl,2) == r
       lbl = lbl';
    else
```

```
uiwait(msgbox(char('Axis labels must be Mx1 -
```

number of axes x 1', ...

end

```
elseif ischar(lbl)
```

```
% check for charater labels
lbl = cellstr(lbl);
```

 $\operatorname{end}$ 

```
if ~exist('leg','var') || isempty(leg)
  % no data legend - define default legend
  leg = cell(1,c);
    for ii = 1:c;
        leg(ii) = cellstr(sprintf('Set %g',ii));
    end
elseif numel(leg) ~= c
    uiwait(msgbox(char('Data set label must be 1XN -
        1 x number of sets', ...
        sprintf('%g data sets labeled, %g exist',numel(leg),c))))
    error(sprintf('%g data sets labeled, %g exist',numel(leg),c))
```

```
end
% check for figure or axes
if ~exist('f','var')
    % no figure or axes requested - generate new ones
    f = figure; ca = gca(f); cla(ca); hold on
elseif ismember(f,get(0,'children')')
    % existing figure - clear and set up
    clf(f); ca = gca(f); cla(ca); hold on
elseif isint(f)
    % generating a new figure
    figure(f); ca = gca(f); cla(ca); hold on
else
    \% may be an axes - may be garbage
    try
       %is this an axes?
        if ismember(get(f,'parent'),get(0,'children')')
           % existing figure axes - use
           ca = f; f = get(f, 'parent'); cla(ca); hold on
        end
    catch
       % make new figure and axes
       disp(sprintf('Invalid axes handle %g passed.
           Generating new figure',f))
        f = figure; ca = gca(f); cla(ca); hold on
```

```
% clear figure and set limits
set(ca,'visible','off');
set(f,'color','w')
set(ca,'xlim',[-1.25 1.25],'ylim',[-1.25 1.25]);
axis(ca,'equal','manual') % title
text(0,1.25,tle,'horizontalalignment','center','fontweight','bold');
```

```
% define data case colors
col = color_index(c);
```

```
% scale by range
angw = linspace(0,2*pi,r+1)';
mag = (data - rng(:,1) * ones(1,c)) ./ (diff(rng,[],2) * ones(1,c));
% wrap data (close the last axis to the first)
ang = angw(1:end-1);
angwv = angw * ones(1,c);
magw = [mag; mag(1,:)];
```

```
% make the plot
% define the axis locations
start = [zeros(1,r); cos(ang')]; stop = [zeros(1,r); sin(ang')];
% plot the axes
plot(ca,start,stop,'color','k','linestyle','-'); axis equal
% plot axes markers
inc = 0.25:.25:1; mk = .025 * ones(1,4); tx = 4 * mk; tl = 0:.25:1;
```

```
% loop each axis ang plot the line markers and labels
for ii = 1:r
  % plot tick marks
  plot(ca,[[cos(ang(ii)) * inc + sin(ang(ii)) * mk]; ...
      [cos(ang(ii)) * inc - sin(ang(ii)) * mk]], ...
      [[sin(ang(ii)) * inc - cos(ang(ii)) * mk]], ...
      [[sin(ang(ii)) * inc + cos(ang(ii)) * mk]],'color','k')
  % label the tick marks
  for jj = 1:4
```

```
temp = text([cos(ang(ii)) * inc(jj) +% 90 corresponds to the number of samples
% 6 corresponds to the number of channels in the ERD file FFT_126_CURB_21_
tx(jj)], ...
```

[sin(ang(ii)) \* inc(jj) - cos(ang(ii)) \* tx(jj)], ... num2str(chop(rng(ii,1) + inc(jj)\*diff(rng(ii,:)),2)));

```
% temp = text([cos(ang(ii)) * tl(jj) + sin(ang(ii)) * tx(jj)], ...
```

```
% [sin(ang(ii)) * tl(jj) - cos(ang(ii)) * tx(jj)], ...
```

```
% num2str(chop(rng(ii,1) + tl(jj)*diff(rng(ii,:)))));
```

```
set(temp,'fontsize',8)
% flip the text alignment for lower axes
    if ang(ii) >= pi
        set(temp,'HorizontalAlignment','right')
```

end

```
% label each axis
```

```
temp = text([cos(ang(ii)) * 1.1 + sin(ang(ii)) * 0], ...
```

[sin(ang(ii)) \* 1.1 - cos(ang(ii)) \* 0], char(lbl(ii,:)));

% flip the text alignment for right side axes

```
if ang(ii) > pi/2 && ang(ii) < 3*pi/2
```

```
set(temp,'HorizontalAlignment','right')
```

end

## end

```
% plot the data
o = polar(ca,angw*ones(1,c),magw);
% apply the legend
temp = legend(ca,o,leg,'location','northwest');
% set the legend location out of the plot window
set(temp,'position',get(temp,'position') .* [1.25 1 1 1])
% set the color for each data set
for ii = 1:c
  % set(o(ii),'color',col(ii,:),'linewidth',2.2)
  set(o(1),'color','k','linestyle','-.','linewidth',2.2)
  set(o(2),'color','b','linestyle','--','linewidth',2.2)
  set(o(3),'color','g','linestyle','--','linewidth',2.2)
  set(o(4),'color','r','linestyle',':','linewidth',2.2)
```

end

# F.5 Frequency\_Response\_Functions.m

% transfer function for going from RWA to SWA

- % run1=60/40, OE Tire
- % run2=60/40, Tire 1
- % run3=60/40, Tire 2
- % run4=60/40, Tire 3
- % run5=55/45, OE Tire
- % run6=50/50, OE Tire
- % run7=50/50, Tire 1
- % run8=50/50, Tire 2
- % run9=50/50, Tire 3
- % run10=55/45, Tire 1
- % run11=55/45, Tire 2
- % run12=55/45, Tire 3

%% Read in the data from the CarSim FFT 126 CURB\_21\_01
fid = fopen('FFT126\_CURB\_21\_1\_20\_SWA.bin','r');
% bin file associated with the Michelin's raw RWA/SWA data

% treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

for m=1:90

% 90 corresponds to the number of samples per channel % 6 corresponds to the number of channels in the ERD file FFT\_126\_CURB\_21\_01(:,m)=D(((6\*(m-1))+1):(6\*m));

end

%% Read in the data from the CarSim FFT 126 CURB\_21\_02 fid = fopen('FFT126\_CURB\_21\_2\_20\_SWA.bin','r'); % bin file associated with the Michelin's raw RWA/SWA data % treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

for m=1:90

% 90 corresponds to the number of samples per channel % 6 corresponds to the number of channels in the ERD file FFT\_126\_CURB\_21\_02(:,m)=D(((6\*(m-1))+1):(6\*m));

%% Read in the data from the CarSim FFT 126 CURB\_21\_03
fid = fopen('FFT126\_CURB\_21\_3\_20\_SWA.bin','r');
% bin file associated with the Michelin's raw RWA/SWA data
% treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default CARSIM
% stores binary data in this format.

for m=1:90

% 90 corresponds to the number of samples per channel % 6 corresponds to the number of channels in the ERD file FFT\_126\_CURB\_21\_03(:,m)=D(((6\*(m-1))+1):(6\*m));

end

%% Read in the data from the CarSim FFT 126 CURB\_21\_04 fid = fopen('FFT126\_CURB\_21\_4\_20\_SWA.bin','r'); % bin file associated with the Michelin's raw RWA/SWA data % treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

for m=1:90

% 90 corresponds to the number of samples per channel % 6 corresponds to the number of channels in the ERD file FFT\_126\_CURB\_21\_04(:,m)=D(((6\*(m-1))+1):(6\*m)); %% Read in the data from the CarSim FFT 126 CURB\_21\_05 fid = fopen('FFT126\_CURB\_21\_5\_20\_SWA.bin','r'); % bin file associated with the Michelin's raw RWA/SWA % data treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

for m=1:90

% 90 corresponds to the number of samples per channel % 6 corresponds to the number of channels in the ERD file FFT\_126\_CURB\_21\_05(:,m)=D(((6\*(m-1))+1):(6\*m));

end

%% Read in the data from the CarSim FFT 133 Run1353 Nominal % OE Tire and nominal weight distribution fid = fopen('FFT133\_Run1353.bin','r'); % bin file associated with the Michelin's raw RWA data % treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

FFT133\_Run1353=zeros(10,110);

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1353(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

```
run1=FFT133_Run1353;
```

```
%% Read in the data from the CarSim FFT 133 Run1362
% TWEEL A, nominal weight distribution
fid = fopen('FFT133_Run1396.bin','r');
% bin file associated with the Michelin's raw RWA data
% treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default CARSIM
% stores binary data in this format.
```

```
FFT133_Run1396=zeros(10,110);
```

for m=1:110

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1396(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

```
run2=FFT133_Run1396;
```

%% Read in the data from the CarSim FFT 133 Run1363 % TWEEL B, nominal weight distribution fid = fopen('FFT133\_Run1363.bin','r'); % bin file associated with the Michelin's raw RWA data % treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

```
FFT133_Run1363=zeros(10,110);
```

for m=1:110

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1363(:,m)=D(((10\*(m-1))+1):(10\*m));

#### end

```
run3=FFT133_Run1363;
```

%% Read in the data from the CarSim FFT 133 Run1364 % TWEEL C, nominal weight distribution fid = fopen('FFT133\_Run1364.bin','r'); % bin file associated with the Michelin's raw RWA % data treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by default CARSIM % stores binary data in this format.

FFT133\_Run1364=zeros(10,110);

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1364(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

```
run4=FFT133_Run1364;
```

```
%% Read in the data from the CarSim FFT 133
% Run 1400 55/45 weight dis.
fid = fopen('FFT133_Run1403.bin','r');
% bin file associated with the Michelin's raw RWA
% data treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default
% CARSIM stores binary data in this format.
FFT133_Run1403=zeros(10,110);
```

for m=1:110

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1403(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

run5=FFT133\_Run1403;

%% Read in the data from the CarSim FFT 133

% Run 1400 50/50 weight dis.

fid = fopen('FFT133\_Run1399.bin','r');

% bin file associated with the Michelin's raw RWA

% data treated in CARSIM D=fread(fid,'float32');

% float32 has been used to decode, since by default

% CARSIM stores binary data in this format.

```
FFT133_Run1399=zeros(10,110);
```

for m=1:110

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1399(:,m)=D(((10\*(m-1))+1):(10\*m));

#### end

```
run6=FFT133_Run1399;
```

%% Read in the data from the CarSim FFT 133 % Run 1404 50/50 weight dis. with tire 1 fid = fopen('FFT133\_Run1404.bin','r'); % bin file associated with the Michelin's raw % RWA data treated in CARSIM D=fread(fid,'float32'); % float32 has been used to decode, since by % default CARSIM stores binary data in this format. FFT133\_Run1404=zeros(10,110); % 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1404(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

```
run7=FFT133_Run1404;
```

```
%% Read in the data from the CarSim FFT 133
% Run 1405 50/50 weight dis. with tire 2
fid = fopen('FFT133_Run1405.bin','r');
% bin file associated with the Michelin's raw
% RWA data treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by
% default CARSIM stores binary data in this format.
```

```
FFT133_Run1405=zeros(10,110);
```

for m=1:110

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1405(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

```
run8=FFT133_Run1405;
```
```
%% Read in the data from the CarSim FFT 133
% Run1406 50/50 weight dis. with tire 3
fid = fopen('FFT133_Run1406.bin','r');
% bin file associated with the Michelin's raw
% RWA data treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default
% CARSIM stores binary data in this format.
```

```
FFT133_Run1406=zeros(10,110); for m=1:110
```

```
% 110 corresponds to the number of samples per channel
% 10 corresponds to the number of channels in the ERD file
FFT133_Run1406(:,m)=D(((10*(m-1))+1):(10*m));
```

end

```
run9=FFT133_Run1406;
```

```
%% Read in the data from the CarSim FFT 133
% Run1407 55/45 weight dis. with tire 1
fid = fopen('FFT133_Run1407.bin','r');
% bin file associated with the Michelin's raw
% RWA data treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default
% CARSIM stores binary data in this format.
```

```
FFT133_Run1407=zeros(10,110);
for m=1:110
```

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1407(:,m)=D(((10\*(m-1))+1):(10\*m));

end

run10=FFT133\_Run1407;

%% Read in the data from the CarSim FFT 133
% Run1408 55/45 weight dis. with tire 2
fid = fopen('FFT133\_Run1408.bin','r');
% bin file associated with the Michelin's raw RWA
% data treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default
% CARSIM stores binary data in this format.

```
FFT133_Run1408=zeros(10,110);
for m=1:110
```

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1408(:,m)=D(((10\*(m-1))+1):(10\*m));

```
end
```

run11=FFT133\_Run1408;

% Read in the data from the CarSim FFT 133

% Run1409 55/45 weight dis. with tire 3
fid = fopen('FFT133\_Run1409.bin','r');
% bin file associated with the Michelin's raw
% RWA data treated in CARSIM D=fread(fid,'float32');
% float32 has been used to decode, since by default
% CARSIM stores binary data in this format.

FFT133\_Run1409=zeros(10,110);

for m=1:110

% 110 corresponds to the number of samples per channel % 10 corresponds to the number of channels in the ERD file FFT133\_Run1409(:,m)=D(((10\*(m-1))+1):(10\*m));

end

```
run12=FFT133_Run1409;
```

%% Average the Frequency, Gain and Phase for the RWA/SWA
% data from CarSim for the Michelin data

CURB\_21\_RWA\_SWA\_Frequency=(FFT\_126\_CURB\_21\_05(1,:)+... FFT\_126\_CURB\_21\_04(1,:)+FFT\_126\_CURB\_21\_03(1,:)+... FFT\_126\_CURB\_21\_02(1,:)+FFT\_126\_CURB\_21\_01(1,:))/5;

CURB\_21\_RWA\_SWA\_Gain=(FFT\_126\_CURB\_21\_05(4,:)+... FFT\_126\_CURB\_21\_04(4,:)+FFT\_126\_CURB\_21\_03(4,:)+... FFT\_126\_CURB\_21\_02(4,:)+FFT\_126\_CURB\_21\_01(4,:))/5;

```
CURB_21_RWA_SWA_Phase=(FFT_126_CURB_21_05(5,:)+...
FFT_126_CURB_21_04(5,:)+FFT_126_CURB_21_03(5,:)+...
FFT_126_CURB_21_02(5,:)+FFT_126_CURB_21_01(5,:))/5;
```

```
GP=[CURB_21_RWA_SWA_Gain;CURB_21_RWA_SWA_Phase];
freq_rad=(CURB_21_RWA_SWA_Frequency)*(2*pi);
load prep.mat
prepf = idfilt(prep,[1.57,22]);
% band pass filterd frq in rad/s since input
% freq in prep is in rad/s
```

```
spad = etfe(prepf,12,10000);
[mag,phase,w] = bode(spad);
mag=squeeze(mag);
phase=squeeze(phase);
freq_Hz=w/(2*pi);
mag1 = interp1(freq_Hz,mag,run1(1,:),'linear','extrap');
phase1= interp1(freq_Hz,phase,run1(1,:),'linear','extrap');
```

## %%

```
% Adjust the max value of j to give you the
% max no. of runs you are analyzing
for j=1:12
    if j==1
        temp=run1;
    elseif j==2
        temp=run2;
```

```
temp=run7;
elseif j==8
  temp=run8;
elseif j==9
  temp=run9;
elseif j==10
  temp=run10;
elseif j==11
  temp=run11;
elseif j==12
  temp=run12;
end
```

elseif j==3

elseif j==4

elseif j==5

elseif j==6

elseif j==7

temp=run3;

temp=run4;

temp=run5;

temp=run6;

```
FFT_freq_SWA(j,:)=(temp(1,:));
FFT_Yaw_rateG_SWA(j,:)=mag1.*(temp(5,:));
FFT_Yaw_rateP_SWA(j,:)=phase1+(temp(6,:));
```

```
% FFT_Yaw_rateG_SWA1(j,:)=CURB_21_RWA_SWA_Gain.*(temp(5,:));
FFT_AyG_SWA(j,:)=mag1.*(temp(8,:));
FFT_AyP_SWA(j,:)=phase1+(temp(9,:));
% FFT_AyP_SWA1(j,:)=CURB_21_RWA_SWA_Phase+(temp(9,:));
```

```
end
```

%% plot the results figure subplot(2,2,1) plot(FFT\_freq\_SWA(1,:),FFT\_Yaw\_rateG\_SWA(1,:),'k-',... FFT\_freq\_SWA(2,:),FFT\_Yaw\_rateG\_SWA(2,:),'b-.',... FFT\_freq\_SWA(3,:),FFT\_Yaw\_rateG\_SWA(3,:),'g--',... FFT\_freq\_SWA(4,:),FFT\_Yaw\_rateG\_SWA(4,:),'r:','linewidth',1.5) ylabel('Yaw Rate/SWA, Gain (1/s)', 'fontweight', 'bold', 'fontsize', 12) legend('Nominal OE Tire', 'TWEEL A', 'TWEEL B', 'TWEEL C',... 'location','southwest') grid on xlim([0.25 4]) title('Yaw Rate Gain/Road Wheel Angle (C+D)',... 'fontweight', 'bold', 'fontsize', 12) subplot(2,2,3)plot(FFT\_freq\_SWA(1,:),FFT\_Yaw\_rateP\_SWA(1,:),'-k',...

FFT\_freq\_SWA(2,:),FFT\_Yaw\_rateP\_SWA(2,:),'-.b',... FFT\_freq\_SWA(3,:),FFT\_Yaw\_rateP\_SWA(3,:),'--g',...

```
FFT_freq_SWA(4,:),FFT_Yaw_rateP_SWA(4,:),':r','linewidth',1.5)
```

```
xlabel('Frequency (Hz)','fontweight','bold','fontsize',12)
ylabel('Yaw Rate/SWA, Phase (deg)','fontweight','bold','fontsize',12)
legend('Nominal OE Tire','TWEEL A','TWEEL B','TWEEL C',...
```

```
'location', 'northeast')
```

#### grid

xlim([0.25 4])

title('Yaw Rate Phase/Road Wheel Angle (C+D)','fontweight',...

```
'bold', 'fontsize', 12)
```

```
subplot(2,2,2)
plot(FFT_freq_SWA(1,:),FFT_AyG_SWA(1,:),'-k',FFT_freq_SWA(2,:),...
```

```
FFT_AyG_SWA(2,:),'-.b',FFT_freq_SWA(3,:),FFT_AyG_SWA(3,:),...
'--g',FFT_freq_SWA(4,:),FFT_AyG_SWA(4,:),':r','linewidth',1.5)
```

ylabel('Lateral Accel./SWA, Gain (g/deg)','fontweight',...

'bold','fontsize',12)

legend('Nominal OE Tire','TWEEL A','TWEEL B',...

'TWEEL C', 'location', 'northeast')

#### grid

```
xlim([0.25 4])
```

title('Lateral Accel. Gain/Road Wheel Angle (C+D)',...

```
'fontweight', 'bold', 'fontsize', 12)
```

subplot(2,2,4)

plot(FFT\_freq\_SWA(1,:),FFT\_AyP\_SWA(1,:),'-k',FFT\_freq\_SWA(2,:),...

FFT\_AyP\_SWA(2,:),'-.b',FFT\_freq\_SWA(3,:),FFT\_AyP\_SWA(3,:),... '--g',FFT\_freq\_SWA(4,:),FFT\_AyP\_SWA(4,:),':r','linewidth',1.5)

xlabel('Frequency (Hz)','fontweight','bold"bold','fontsize',12)
legend('Nominal OE Tire','TWEEL A','TWEEL B',...

'TWEEL C', 'location', 'northeast') ,'fontsize',12)

ylabel('Lateral Accel./SWA, Phase (deg)','fontweight',...

'bold','fontsize',12)

legend('Nominal OE Tire','TWEEL A','TWEEL B','TWEEL C',...

'location','southeast') grid xlim([0.25 4])

title('Lateral Accel. Phase/Road Wheel Angle (C+D)',...

'fontweight', 'bold', 'fontsize', 12)

print -djpeg -r300 FRF\_Thesis\_C\_D

```
%% 4 parameters for thesis
for k=1:12
SS_yaw_rate_gain(k) = interp1(FFT_freq_SWA(k,:),...
```

FFT\_Yaw\_rateG\_SWA(k,:),0.28,'linear','extrap');

damping\_coeff(k)=1/(2\*max(FFT\_Yaw\_rateG\_SWA(k,:))/...

SS\_yaw\_rate\_gain(k));

```
yaw_rate_3db_gain(k)=0.7079.*(SS_yaw_rate_gain(k));
tempo=FFT_Yaw_rateG_SWA(k,:);
k;
s=1;
clear high1
clear low1
while s<=max(size(FFT_Yaw_rateG_SWA))</pre>
    if tempo(s)<(0.7079*SS_yaw_rate_gain(k))</pre>
        low1(s)=s;
    elseif tempo(s)>(0.7079*SS_yaw_rate_gain(k))
        high1(s)=s;
    else
    end
s=s+1;
end
for w=1:length(low1)
    if low1(1)==0;
        low1(1)=[];
    end
end
high1(end);
three_db_bandwidth(k)=FFT_freq_SWA(k,low1(1));
```

Ay\_phase\_1Hz(k)=FFT\_AyP\_SWA(k,46);

```
% 46 is the element no. corresponding to 1 Hz
% in frequency vector FFT_freq_SWA
end
```

```
% data for the changing cornering stiffness
% (OE Tire, Tire 1, Tire 2, Tire3)
data=[Ay_phase_1Hz(1) three_db_bandwidth(1) SS_yaw_rate_gain(1)...
```

damping\_coeff(1); Ay\_phase\_1Hz(2) three\_db\_bandwidth(2)... SS\_yaw\_rate\_gain(2) damping\_coeff(2); Ay\_phase\_1Hz(3)... three\_db\_bandwidth(3) SS\_yaw\_rate\_gain(3) damping\_coeff(3);... Ay\_phase\_1Hz(4) three\_db\_bandwidth(4) SS\_yaw\_rate\_gain(4)... damping\_coeff(4)]';

data1=[Ay\_phase\_1Hz(1) three\_db\_bandwidth(1) SS\_yaw\_rate\_gain(1)...

damping\_coeff(1); Ay\_phase\_1Hz(5) three\_db\_bandwidth(5)... SS\_yaw\_rate\_gain(5) damping\_coeff(5); Ay\_phase\_1Hz(6)... three\_db\_bandwidth(6) SS\_yaw\_rate\_gain(6) damping\_coeff(6)]';

data12=[Ay\_phase\_1Hz(1) three\_db\_bandwidth(1) SS\_yaw\_rate\_gain(1)...

damping\_coeff(1); Ay\_phase\_1Hz(2) three\_db\_bandwidth(2)... SS\_yaw\_rate\_gain(2) damping\_coeff(2); Ay\_phase\_1Hz(3)... three\_db\_bandwidth(3) SS\_yaw\_rate\_gain(3) damping\_coeff(3);... Ay\_phase\_1Hz(4) three\_db\_bandwidth(4) SS\_yaw\_rate\_gain(4)... damping\_coeff(4); Ay\_phase\_1Hz(5) three\_db\_bandwidth(5)... SS\_yaw\_rate\_gain(5) damping\_coeff(5); Ay\_phase\_1Hz(6)... three\_db\_bandwidth(6) SS\_yaw\_rate\_gain(6) damping\_coeff(6);... Ay\_phase\_1Hz(7) three\_db\_bandwidth(7) SS\_yaw\_rate\_gain(7)... damping\_coeff(7); Ay\_phase\_1Hz(8) three\_db\_bandwidth(8)... SS\_yaw\_rate\_gain(8) damping\_coeff(8); Ay\_phase\_1Hz(9)... three\_db\_bandwidth(9) SS\_yaw\_rate\_gain(9) damping\_coeff(9);... Ay\_phase\_1Hz(10) three\_db\_bandwidth(10) SS\_yaw\_rate\_gain(10)... damping\_coeff(10); Ay\_phase\_1Hz(11) three\_db\_bandwidth(11)... SS\_yaw\_rate\_gain(11) damping\_coeff(11); Ay\_phase\_1Hz(12)... three\_db\_bandwidth(12) SS\_yaw\_rate\_gain(12) damping\_coeff(12)]';

legend1={'Nominal OE Tire', 'TWEEL A', 'TWEEL B', 'TWEEL C'};
axes\_label = {'Lateral Accel Phase @ 1 Hz' '(deg)';...

'Yaw Rate Bandwidth' '(Hz)'; 'SS Yaw Rate Gain' '(deg/s)/deg';... 'Yaw Rate Damping Ratio' "};

[f, o] = spider(data,",[-60 -30 ;0 3.5;0.1 0.5;0 0.6],axes\_label,... legend1);

print -djpeg -r300 spider\_Thesis\_TWEELS

```
%% spider plot for example
data1=[Ay_phase_1Hz(1) three_db_bandwidth(1) SS_yaw_rate_gain(1)...
damping_coeff(1)]';
axes_label = {'Lateral Accel Phase @ 1 Hz', '(deg)';...
```

'Yaw Rate Bandwidth' '(Hz)'; 'SS Yaw Rate Gain' '(deg/s)/deg';...

'Yaw Rate Damping Ratio' "};

[f, o] = spider(data1,'Sensitivity to Load for TWEELs (C+D)',...

[-70 -50 ;0 3.5;0.1 0.5;0 0.6],axes\_label);

print -djpeg -r300 spider\_Thesis\_example

%% plot frequency response functions for weight distrubutions figure

subplot(2,2,1)

plot(FFT\_freq\_SWA(1,:),FFT\_Yaw\_rateG\_SWA(1,:),FFT\_freq\_SWA(5,:),...

FFT\_Yaw\_rateG\_SWA(5,:),':',FFT\_freq\_SWA(6,:),...

FFT\_Yaw\_rateG\_SWA(6,:),'-.','linewidth',1.5)

ylabel('Yaw Rate/SWA, Gain (1/s)','fontweight','bold','fontsize',12)
grid on

```
% legend('OE Tire','55/45 Dis.','50/50 Dis.','location','northeast')
xlim([0.25 4])
```

```
subplot(2,2,3)
plot(FFT_freq_SWA(1,:),FFT_Yaw_rateP_SWA(1,:),FFT_freq_SWA(5,:),...
FFT_Yaw_rateP_SWA(5,:),':',FFT_freq_SWA(6,:),...
FFT_Yaw_rateP_SWA(6,:),'-.','linewidth',1.5)
```

xlabel('Frequency (Hz)','fontweight','bold','fontsize',12)

ylabel('Yaw Rate/SWA, Phase (deg)','fontweight','bold',...

```
'fontsize',12)
```

### grid

```
% legend('OE Tire','55/45 Dis.','50/50 Dis.','location','northeast')
```

```
xlim([0.25 4])
```

```
subplot(2,2,2)
```

plot(FFT\_freq\_SWA(1,:),FFT\_AyG\_SWA(1,:),FFT\_freq\_SWA(5,:),...

```
FFT_AyG_SWA(5,:),':',FFT_freq_SWA(6,:),FFT_AyG_SWA(6,:),...
'-.','linewidth',1.5)
```

ylabel('Lateral Accel./SWA, Gain (g/deg)','fontweight','bold',...

```
'fontsize',12)
```

#### grid

% legend('OE Tire','55/45 Dis.','50/50 Dis.','location','northeast')
xlim([0.25 4])

```
subplot(2,2,4)
plot(FFT_freq_SWA(1,:),FFT_AyP_SWA(1,:),FFT_freq_SWA(5,:),...
```

```
FFT_AyP_SWA(5,:),':',FFT_freq_SWA(6,:),FFT_AyP_SWA(6,:),...
'-.','linewidth',1.5)
```

```
xlabel('Frequency (Hz)','fontweight','bold','fontsize',12)
ylabel('Lateral Accel./SWA, Phase (deg)','fontweight',...
```

'bold','fontsize',12)

#### grid

```
legend('OE Tire','55/45 Dis.','50/50 Dis.','location','southeast')
xlim([0.25 4])
% % print -djpeg -r300 FRF_
```

%% plot correlation plots

load 'C:\Documents and Settings\rpowell\My Documents\Thesis
\IJVD Paper\Journal Article\Normalized\_Ji'

J1=[OE\_Tire\_60\_40(1) Tire\_1\_60\_40(1) Tire\_2\_60\_40(1)...

Tire\_3\_60\_40(1) OE\_Tire\_55\_45(1) OE\_Tire\_50\_50(1)... Tire\_1\_50\_50(1) Tire\_2\_50\_50(1) Tire\_3\_50\_50(1)... Tire\_1\_55\_45(1) Tire\_2\_55\_45(1) Tire\_3\_55\_45(1)];

J2=[OE\_Tire\_60\_40(2) Tire\_1\_60\_40(2) Tire\_2\_60\_40(2)...

Tire\_3\_60\_40(2) OE\_Tire\_55\_45(2) OE\_Tire\_50\_50(2)... Tire\_1\_50\_50(2) Tire\_2\_50\_50(2) Tire\_3\_50\_50(2)... Tire\_1\_55\_45(2) Tire\_2\_55\_45(2) Tire\_3\_55\_45(2)]; J3=[OE\_Tire\_60\_40(3) Tire\_1\_60\_40(3) Tire\_2\_60\_40(3)... Tire\_3\_60\_40(3) OE\_Tire\_55\_45(3) OE\_Tire\_50\_50(3)... Tire\_1\_50\_50(3) Tire\_2\_50\_50(3) Tire\_3\_50\_50(3)... Tire\_1\_55\_45(3) Tire\_2\_55\_45(3) Tire\_3\_55\_45(3)];

```
subplot(2,2,4)
```

plot(Ay\_phase\_1Hz(1), J3(1), '^k', Ay\_phase\_1Hz(2), J3(2), 'vk',...

Ay\_phase\_1Hz(3),J3(3),'<k',Ay\_phase\_1Hz(4),J3(4),'>k',... Ay\_phase\_1Hz(5),J3(5),'hb',Ay\_phase\_1Hz(10),J3(10),'pb',... Ay\_phase\_1Hz(11),J3(11),'xb',Ay\_phase\_1Hz(12),J3(12),'+b',... Ay\_phase\_1Hz(6),J3(6),'or',Ay\_phase\_1Hz(7),J3(7),'sr',... Ay\_phase\_1Hz(8),J3(8),'.r',Ay\_phase\_1Hz(9),J3(9),'\*r',... 'MarkerSize',10)

legend('60/40, OE Tire','60/40, Tire 1','60/40, Tire 2',... '60/40, Tire 3','55/45, OE Tire','55/45, Tire 1',... '55/45, Tire 2','55/45, Tire 3','50/50, OE Tire',... '50/50, Tire 1','50/50, Tire 2','50/50, Tire 3','location',... 'best')

xlabel('Lateral Acceleration Phase Lag at 1 Hz, deg',...

```
'fontweight', 'bold') ylabel('J3 Normalized', 'fontweight', 'bold') grid
```

```
subplot(2,2,3)
```

```
plot(three_db_bandwidth(1), J3(1), '^k', three_db_bandwidth(2), ...
```

```
J3(2),'vk',three_db_bandwidth(3),J3(3),'<k',...
three_db_bandwidth(4),J3(4),'>k',three_db_bandwidth(5),...
J3(5),'hb',three_db_bandwidth(6),J3(6),'or',...
three_db_bandwidth(7),J3(7),'sr',three_db_bandwidth(8),...
J3(8),'.r',three_db_bandwidth(9),J3(9),'*r',...
three_db_bandwidth(10),J3(10),'pb',three_db_bandwidth(11),...
J3(11),'xb',three_db_bandwidth(12),J3(12),'+b',...
'MarkerSize',10)
```

```
xlabel('Yaw Rate Bandwidth, Hz','fontweight','bold')
ylabel('J3 Normalized','fontweight','bold')
grid
```

```
% subplot(3,2,1)
```

```
% plot(SS_yaw_rate_gain(1),J1(1),'^k',SS_yaw_rate_gain(2),J1(2),...
%
      'vk',SS_yaw_rate_gain(3),J1(3),'<k',SS_yaw_rate_gain(4),...
%
      J1(4), '>k', SS_yaw_rate_gain(5), J1(5), 'hb', SS_yaw_rate_gain(6)...
%
      ,J1(6),'or',SS_yaw_rate_gain(7),J1(7),'sr',SS_yaw_rate_gain(8),...
%
      J1(8),'.r',SS_yaw_rate_gain(9),J1(9),'*r',SS_yaw_rate_gain(10),...
%
      J1(10), 'pb', SS_yaw_rate_gain(11), J1(11), 'xb',...
%
      SS_yaw_rate_gain(12), J1(12), '+b', 'MarkerSize', 10)
% xlabel('Steady State Yaw Rate Gain, 1/s','fontweight','bold')
% ylabel('J1 Normalized', 'fontweight', 'bold')
% grid
% subplot(2,2,1)
% plot(three_db_bandwidth(1),J2(1),'^k',three_db_bandwidth(2),...
      J2(2),'vk',three_db_bandwidth(3),J2(3),'<k',...
%
%
      three_db_bandwidth(4), J2(4), '>k', three_db_bandwidth(5), J2(5), ...
%
      'hb',three_db_bandwidth(6),J2(6),'or',three_db_bandwidth(7),...
%
      J2(7), 'sr', three_db_bandwidth(8), J2(8), '.r',...
%
      three_db_bandwidth(9), J2(9), '*r', three_db_bandwidth(10),...
%
      J2(10), 'pb', three_db_bandwidth(11), J2(11), 'xb',...
%
      three_db_bandwidth(12), J2(12), '+b', 'MarkerSize', 10)
% xlabel('Yaw Rate Bandwidth, Hz', 'fontweight', 'bold')
% ylabel('J2 Normalized','fontweight','bold')
% grid
subplot(2,2,2)
plot(Ay_phase_1Hz(1), J2(1), '^k', Ay_phase_1Hz(2), J2(2), 'vk',...
    Ay_phase_1Hz(3), J2(3), '<k', Ay_phase_1Hz(4), J2(4), '>k',...
```

```
Ay_phase_1Hz(5), J2(5), 'hb', Ay_phase_1Hz(10), J2(10), 'pb',...
```

```
Ay_phase_1Hz(11), J2(11), 'xb', Ay_phase_1Hz(12), J2(12), '+b',...
Ay_phase_1Hz(6), J2(6), 'or', Ay_phase_1Hz(7), J2(7), 'sr',...
Ay_phase_1Hz(8), J2(8), '.r', Ay_phase_1Hz(9), J2(9), '*r',...
'MarkerSize', 10)
```

legend('60/40, OE Tire', '60/40, Tire 1', '60/40, Tire 2',...

'60/40, Tire 3','55/45, OE Tire','55/45, Tire 1',... '55/45, Tire 2','55/45, Tire 3','50/50, OE Tire',... '50/50, Tire 1','50/50, Tire 2','50/50, Tire 3',... 'location','best')

xlabel('Lateral Acceleration Phase Lag at 1 Hz, deg',...

'fontweight', 'bold')

```
ylabel('J2 Normalized','fontweight','bold')
grid
```

```
% subplot(3,2,2)
% plot(three_db_bandwidth(1),J1(1),'^k',three_db_bandwidth(2),...
%
      J1(2),'vk',three_db_bandwidth(3),J1(3),'<k',...
%
      three_db_bandwidth(4), J1(4), '>k', three_db_bandwidth(5), J1(5), ...
%
      'hb',three_db_bandwidth(6),J1(6),'or',three_db_bandwidth(7),...
%
      J1(7), 'sr', three_db_bandwidth(8), J1(8), '.r',...
%
      three_db_bandwidth(9),J1(9),'*r',three_db_bandwidth(10),...
%
      J1(10), 'pb', three_db_bandwidth(11), J1(11), 'xb',...
%
      three_db_bandwidth(12), J1(12), '+b', 'MarkerSize', 10)
% xlabel('Yaw Rate Bandwidth, Hz', 'fontweight', 'bold')
% ylabel('J1 Normalized','fontweight','bold')
% grid
```

```
% subplot(2,2,1)
% plot(SS_yaw_rate_gain(1), J1(1), '^k', SS_yaw_rate_gain(2), J1(2),...
%
      'vk',SS_yaw_rate_gain(3),J1(3),'<k',SS_yaw_rate_gain(4),J1(4),...
%
      '>k',SS_yaw_rate_gain(5),J1(5),'hb',SS_yaw_rate_gain(6),J1(6),...
%
      'or', SS_yaw_rate_gain(7), J1(7), 'sr', SS_yaw_rate_gain(8), J1(8), ...
%
      '.r',SS_yaw_rate_gain(9),J1(9),'*r',SS_yaw_rate_gain(10),J1(10),...
%
      'pb',SS_yaw_rate_gain(11),J1(11),'xb',SS_yaw_rate_gain(12),...
%
      J1(12),'+b','MarkerSize',10)
% xlabel('Steady State Yaw Rate Gain, 1/s', 'fontweight', 'bold')
% ylabel('J1 Normalized to OE Tire', 'fontweight', 'bold')
% grid
subplot(2,2,1)
plot(three_db_bandwidth(1), J2(1), '^k', three_db_bandwidth(2), ...
```

J2(2),'vk',three\_db\_bandwidth(3),J2(3),'<k',... three\_db\_bandwidth(4),J2(4),'>k',three\_db\_bandwidth(5),J2(5),... 'hb',three\_db\_bandwidth(6),J2(6),'or',three\_db\_bandwidth(7),... J2(7),'sr',three\_db\_bandwidth(8),J2(8),'.r',... three\_db\_bandwidth(9),J2(9),'\*r',three\_db\_bandwidth(10),... J2(10),'pb',three\_db\_bandwidth(11),J2(11),'xb',... three\_db\_bandwidth(12),J2(12),'+b','MarkerSize',10)

```
xlabel('Yaw Rate Bandwidth, Hz','fontweight','bold')
ylabel('J2 Normalized to OE Tire','fontweight','bold')
grid
```

```
stem3(SS_yaw_rate_gain(1),three_db_bandwidth(1),J1(1),'^k')
hold on
stem3(SS_yaw_rate_gain(2),three_db_bandwidth(2),J1(2),'vk')
hold on
stem3(SS_yaw_rate_gain(3),three_db_bandwidth(3),J1(3),'<k')</pre>
hold on
stem3(SS_yaw_rate_gain(4),three_db_bandwidth(4),J1(4),'>k')
hold on
stem3(SS_yaw_rate_gain(5),three_db_bandwidth(5),J1(5),'hb')
hold on
stem3(SS_yaw_rate_gain(10),three_db_bandwidth(10),J1(10),'pb')
hold on
stem3(SS_yaw_rate_gain(11),three_db_bandwidth(11),J1(11),'xb')
hold on
stem3(SS_yaw_rate_gain(12),three_db_bandwidth(12),J1(12),'+b')
hold on
stem3(SS_yaw_rate_gain(6),three_db_bandwidth(6),J1(6),'or')
hold on
stem3(SS_yaw_rate_gain(7),three_db_bandwidth(7),J1(7),'sr')
hold on
stem3(SS_yaw_rate_gain(8),three_db_bandwidth(8),J1(8),'.r')
hold on
stem3(SS_yaw_rate_gain(9),three_db_bandwidth(9),J1(9),'*r')
hold on
grid on
xlabel('Steady State Yaw Rate Gain, 1/s','fontweight','bold')
ylabel('Yaw Rate Bandwidth, Hz', 'fontweight', 'bold')
zlabel('J1 Normalized','fontweight','bold')
```

```
figure stem3(Ay_phase_1Hz(1),three_db_bandwidth(1),J2(1),'^k')
hold on
stem3(Ay_phase_1Hz(2),three_db_bandwidth(2),J2(2),'vk')
hold on
stem3(Ay_phase_1Hz(3),three_db_bandwidth(3),J2(3),'<k')</pre>
hold on
stem3(Ay_phase_1Hz(4),three_db_bandwidth(4),J2(4),'>k')
hold on
stem3(Ay_phase_1Hz(5),three_db_bandwidth(5),J2(5),'hb')
hold on
stem3(Ay_phase_1Hz(10),three_db_bandwidth(10),J2(10),'pb')
hold on
stem3(Ay_phase_1Hz(11),three_db_bandwidth(11),J2(11),'xb')
hold on
stem3(Ay_phase_1Hz(12),three_db_bandwidth(12),J2(12),'+b')
hold on
stem3(Ay_phase_1Hz(6),three_db_bandwidth(6),J2(6),'or')
hold on
stem3(Ay_phase_1Hz(7),three_db_bandwidth(7),J2(7),'sr')
hold on
stem3(Ay_phase_1Hz(8),three_db_bandwidth(8),J2(8),'.r')
hold on
stem3(Ay_phase_1Hz(9),three_db_bandwidth(9),J2(9),'*r')
hold on
grid on
```

```
xlabel('Lateral Accel. Phase Lag, deg','fontweight','bold')
ylabel('Yaw Rate Bandwidth, Hz','fontweight','bold')
zlabel('J2 Normalized','fontweight','bold')
hold off
```

#### 

```
figure
stem3(Ay_phase_1Hz(1),three_db_bandwidth(1),J3(1),'^k')
hold on
stem3(Ay_phase_1Hz(2),three_db_bandwidth(2),J3(2),'vk')
hold on
stem3(Ay_phase_1Hz(3),three_db_bandwidth(3),J3(3),'<k')</pre>
hold on
stem3(Ay_phase_1Hz(4),three_db_bandwidth(4),J3(4),'>k')
hold on
stem3(Ay_phase_1Hz(5),three_db_bandwidth(5),J3(5),'hb')
hold on
stem3(Ay_phase_1Hz(10),three_db_bandwidth(10),J3(10),'pb')
hold on
stem3(Ay_phase_1Hz(11),three_db_bandwidth(11),J3(11),'xb')
hold on
stem3(Ay_phase_1Hz(12),three_db_bandwidth(12),J3(12),'+b')
hold on
stem3(Ay_phase_1Hz(6),three_db_bandwidth(6),J3(6),'or')
hold on
stem3(Ay_phase_1Hz(7),three_db_bandwidth(7),J3(7),'sr')
hold on
stem3(Ay_phase_1Hz(8),three_db_bandwidth(8),J3(8),'.r')
```

%% spider plot for all cases

legend1={'60/40, OE Tire', '60/40, Tire 1', '60/40, Tire 2',...

'60/40, Tire 3','55/45, OE Tire','50/50, OE Tire',... '50/50, Tire 1','50/50, Tire 2','50/50, Tire 3',... '55/45, Tire 1','55/45, Tire 2','55/45, Tire 3',};

axes\_label = {'Lateral Accel Phase @ 1 Hz' '(deg)';...

'Yaw Rate Bandwidth' '(Hz)'; 'SS Yaw Rate Gain' '(deg/s)/deg'; ... 'Yaw Rate Damping Ratio' "};

[f, o] = spider(data12,",[-70 -30 ;0 3.5;0.0 0.6;0 0.6],...

axes\_label,legend1);

%% plot the spider plot for IVJD

data=[Ay\_phase\_1Hz(1) three\_db\_bandwidth(1) SS\_yaw\_rate\_gain(1)...

```
damping_coeff(1); Ay_phase_1Hz(5) three_db_bandwidth(5)...
SS_yaw_rate_gain(5) damping_coeff(5); Ay_phase_1Hz(6) ...
three_db_bandwidth(6) SS_yaw_rate_gain(6) damping_coeff(6)]';
```

legend1={'60/40 Weight Dis. (OE Tire)',...

```
'55/45 Weight Dis. (OE Tire)',...
'50/50 Weight Dis. (OE Tire)'};
```

axes\_label = {'Lateral Accel Phase @ 1 Hz' '(deg)';...

'Yaw Rate Bandwidth' '(Hz)'; 'SS Yaw Rate Gain' '(deg/s)/deg';...
'Yaw Rate Damping Ratio' "};

[f, o] = spider(data1,",[-70 -30 ;0 2.5;0.0 0.6;0 0.6],...

axes\_label,legend1);

%% spider plot for IJVD paper

legend1={'OE Tire','Tire 1','Tire 2','Tire 3'};

axes\_label = {'Lateral Accel Phase @ 1 Hz' '(deg)';...

'Yaw Rate Bandwidth' '(Hz)'; 'SS Yaw Rate Gain' '(deg/s)/deg';...

'Yaw Rate Damping Ratio' "};

[f, o] = spider(data,",[-70 -30 ;0 3.5;0.0 0.6;0 0.6],...

axes\_label,legend1);

print -djpeg -r500 spider\_IJVD

# F.6 Lane\_Change.m

#### 

Kst=13.8;	% Hand wheel/Road wheel ratio
K= 0.75; %[0.55	0.6 0.65 0.7 0.75 0.8 0.85];
lead=0.2; % [0.1	0.2 0.3 0.4 0.5]; %
tau=0.1;	% time delay, sec
TD=0.1;	% time lag, sec
vkph1=78;	% Speed, kph MUST ALSO BE IN CARSIM !!

% q1=0.5; % weighting for J1 for selection of driver model % q2=0.25; % weighting for J2 for selection of driver model % q3=0.25; % weighting for J3 for selection of driver model q1=0.75; % weighting for J1 for selection of driver model q2=0.125; % weighting for J2 for selection of driver model q3=0.125; % weighting for J3 for selection of driver model

```
vmps1=vkph1/3.6;
lookaheadtime=1.5; % preview time for the driver model
elm=lookaheadtime*vmps1; % preview time converted into distance
LookAhead=elm;
```

fTrack=1.453; % front track width of the Mini
rTrack=1.475; % rear track width of the Mini
aTrack=(fTrack+rTrack)/2; % average of track width of the Mini f/r
hTrack=aTrack/2; % half track width of the Mini
disp(', ')

% loop for determining the best parameters for gain and lead

```
for i=1:length(K)
```

for j=1:length(lead)
 Kd=K(i);
 TL=lead(j);

```
%
```

```
xm1=0:15;
Y1=zeros(size(xm1)); % Y, m
Psi01=zeros(size(xm1)); % rad
%
```

xm2=16:45;

```
Y2=(3.5/2)*(1 - cos((pi/30)*(xm2-15))); % Y, m
Psi02=atan((3.5*pi/60)*sin((pi/30)*(xm2-15))); % rad
xm3=46:70;
Y3=3.5*ones(size(xm3)); % Y, m
Psi03=zeros(size(xm3)); % rad
xm4=71:95;
Y4=(3.5/2)*(1 + cos((pi/25)*(xm4-70))); % Y, m
Psi04=-atan((3.5*pi/50)*sin((pi/25)*(xm4-70))); % rad
xm5=96:200;
Y5=zeros(size(xm5)); % Y, m
Psi05=zeros(size(xm5)); % rad
```

```
Ycourse=[Y1 Y2 Y3 Y4 Y5]; % Y of course, m
Xcourse=[xm1 xm2 xm3 xm4 xm5]; % X, m
Psi0=[Psi01 Psi02 Psi03 Psi04 Psi05]; % rad
%
%
% Define course boundaries, ISO TR 3888
% lane widths
%
% section 1 (0-15 m): width = 6.1 ft (1.86 m)
% section 2 (15-45 m): width = undefined
% section 3 (45-70 m): width = 0.6 ft (2.01 m)
% section 4 (70-95 m): width = undefined
% section 5&6 (95-125 m): width = 7.1 ft (2.15 m)
```

LC1x=[0 15]; % meters

```
LC1u=[(1.1*aTrack+0.25)/2 (1.1*aTrack+0.25)/2];
% meters (formula for lane width in section 1 is:...
% meters
           1.1*B + 0.25 where B is width of the car)
LC1L=[-(1.1*aTrack+0.25)/2 - (1.1*aTrack+0.25)/2];
%
LC3x=[45 70];
                        % meters
LC3u=[3.5+(1.2*aTrack+0.25)/2 3.5+(1.2*aTrack+0.25)/2];
% meters (formula for lane width in section 3 is:...
% meters
            1.2*B + 0.25 where B is the width of the car)
LC3L=[3.5-(1.2*aTrack+0.25)/2 3.5-(1.2*aTrack+0.25)/2];
%
LC5x=[95 125];
                                % meters
LC5u=[(1.3*aTrack+0.25)/2 (1.3*aTrack+0.25)/2];
% meters
          (formula for lane width in section 5 is:...
% meters
            1.3*B + 0.25 where B is the width of the car)
LC5L=[-(1.3*aTrack+0.25)/2 -(1.3*aTrack+0.25)/2];
%
t0=0;
         % Start time, sec
         % End time, sec
tf=10;
decimate=10001;
sim('Genta_Lead_LaneChange')
% Redefined Del(Y) = Ycar - Ycourse
% clc
% disp('ISO Lane Change with Mini')
fprintf('v, Initial Speed,
                                           %.2f \n', vmps_nom(1)*3.6)
                                  kph
```

```
158
```

fprintf('K, Gain, SWA/Psi, deg/deg %.2f \n',Kd)
fprintf('tau, Time lag, sec %.3f \n',tau)
fprintf('elm, Prediction distance, m %.2f \n',elm)
fprintf('Prediction Time, sec %.2f \n',lookaheadtime)
fprintf('Lead Time, sec %.2f \n',TL)

% My output parameters from Simulink are:

- % x1
- % x2
- % x3
- % x4
- % ayg
- % Ycar

#### 

% Definition of States

- % x1 % beta in deg for actual Psi0
- % x2 % yaw rate in deg/s for actual Psi0
- % x3 % yaw or heading angle psi in deg
- % x4 % del-y (lat disp), m for actual Psi0 (body axes)
- % x5 % SWA, deg

```
disp(' ')
disp('MAXIMUM VALUES')
disp(' ')
disp('Max(abs)values of:')
disp('Beta,deg Yaw Rate,deg/s Y-Ycar,m ayg, Gs')
disp([max(abs(x1)) max(abs(x2)) max(abs(Y-Ycar)) max(abs(ayg))])
```

```
disp(' ')
disp(' ')
disp('Task Performance Values')
disp(' ')
disp('J1 = int(del-y), cm<sup>2</sup>*sec
                                  % Measure of lateral deviation')
disp('
                                     % from path')
disp(' ')
disp('J2 = int(SWA^2)*dt, deg^2,sec % Measure of physical workload')
disp(' ')
disp('J3 = int[d(SWA)/dt]^2*dt, (deg/s)^2*sec')
disp('
                                 % Measure of mental workload')
disp('
                                 % involves use of lead by driver')
disp(' ')
disp(' ')
fprintf('J1, m^2*sec
                                               %.5e \n',J1(end))
fprintf('J2, rad^2*sec
                                              %.5e \n',J2(end))
fprintf('J3, (rad/s)^2*sec
                                               %.5e \n',J3(end))
disp(' ')
vkph=vmps_nom(end)*3.6;
J11(j,i)=J1(end);
J22(j,i)=J2(end);
J33(j,i)=J3(end);
    end
end
Ji=[J11 J22 J33];
```

#### 

```
D=zeros(1,length(WC_L1_Y));
for i=1:length(WC_L1_Y)
    if WC_L1_Y(i)>=4.5034 %
       D(i)=1;
    elseif WC_R1_Y(i)<=-1.0766 %
       D(i)=1;
    elseif WC_R1_Y(i)>=-1.0766 %
       D(i)=0;
    elseif WC_L1_Y(i)<=4.5034 % f
       D(i)=0;
    elseif WC_R1_Y(1820)<=2.4966 % for 80 kph
       D(i)=1; %
```

elseif WC\_L1\_Y(4316)>=1.0766 % for 80 kph

```
D(i)=1;
```

end

```
end
```

```
sum=sum(D);
```

disp('Did Vehicle Successfully Complete Double Lane Change?')

```
disp('If value is = to 0 then successful if')
disp('value is > than 0 then fail')
disp(sum)
```

```
% Ypsiup=(Ycar+hTrack).*(sin(x3/57.3)+cos(x3/57.3));
% Ypsidown=(Ycar-hTrack).*(sin(x3/57.3)+cos(x3/57.3));
% normalized J1, J2, J3 to the minimum for that set
```

for i=1:length(K)

```
for j=1:length(lead)
J1norm(j,i)=J11(j,i)/min(min(J11));
J2norm(j,i)=J22(j,i)/min(min(J22));
J3norm(j,i)=J33(j,i)/min(min(J33));
```

end

end

```
Jtotal=J1norm*q1+J2norm*q2+J3norm*q3;
figure
surf(K(1:6),lead(1:4),Jtotal(1:4,1:6))
xlabel('Driver Gain','fontweight','bold')
ylabel('Driver Lead','fontweight','bold')
zlabel('Jtotal normalized','fontweight','bold')
print -djpeg -r300 Jtotal_Driver_Selection
```

```
plot(X,x4.*cos(x3)+Y,'linewidth',1.5)
```

```
plot(WC_L1_X,WC_L1_Y,'k--',WC_R1_X,WC_R1_Y,'k--','linewidth',1.5)
legend('Tire Path') %'Desired CG Course',
hold on
plot(LC1x,LC1u,LC1x,LC1L,...
LC3x,LC3u,LC3x,LC3L,...
LC5x,LC5u,LC5x,LC5L,'linewidth',3)
hold off
grid
ylabel('Y, m','fontweight','bold')
xlabel('X, m','fontweight','bold')
```

```
title('ISO TR 3888 Lane Change Course','fontweight','bold')
print -djpeg -r300 DLC_Tire_Path
```

```
subplot(221),plot(t,Y-Ycar,'linewidth',1.5)
grid
xlabel('t, sec','fontweight','bold')
ylabel('Y-Ycar, m','fontweight','bold')
```

```
subplot(222),plot(t,x1,'linewidth',1.5)
grid
xlabel('Time, sec','fontweight','bold')
ylabel('Beta, deg','fontweight','bold')
```

```
subplot(223),plot(t,SWA,'linewidth',1.5)
grid
xlabel('Time, sec','fontweight','bold')
```

```
ylabel('SWA, deg','fontweight','bold')
```

```
subplot(224),plot(t,x2,'linewidth',1.5)
grid
xlabel('Time, sec','fontweight','bold')
ylabel('r, deg/s','fontweight','bold')
title('Yaw Rate','fontweight','bold')
print -djpeg -r300 DLC_Response_Vals
```

```
subplot(211),plot(t,ayg,'linewidth',1.5)
xlabel('Time, sec','fontweight','bold')
ylabel('Lat Acc, Gs','fontweight','bold')
grid
```

```
subplot(212),plot(X,ayg,'linewidth',1.5)
xlabel('X Distance, m','fontweight','bold')
ylabel('Lat Acc, Gs','fontweight','bold')
grid
print -djpeg -r300 DLC_Lat_accel
```

```
figure
subplot(211),plot(t,vmps_nom*3.6,'linewidth',1.5)
xlabel('Time, sec','fontweight','bold')
ylabel('Speed, kph','fontweight','bold')
grid
```

```
subplot(212),plot(X,vmps_nom*3.6,'linewidth',1.5)
```

```
xlabel('X Distance, m','fontweight','bold')
ylabel('Speed, kph','fontweight','bold')
grid
print -djpeg -r300 DLC_Speed
```

#### figure

```
subplot(211),plot(t,x4.*cos(x3)+Y, t,Y,'--','linewidth',1.5)
subplot(211),plot(t,Ycar, t,Y,'--','linewidth',1.5)
xlabel('Time, sec','fontweight','bold')
ylabel('y, m','fontweight','bold')
grid
legend('Ycar','Ycourse')
```

```
subplot(212),plot(t,Psi*57.3, t,x3,'--','linewidth',1.5)
xlabel('Time, sec','fontweight','bold')
ylabel('Heading Angle, deg','fontweight','bold')
grid
legend('Psi0-com','Psi')
print -djpeg -r300 DLC_Heading_Angle
```

```
plot(X,Y,'linewidth',1.5) %X,Ycar,'--',
legend('Desired CG Path','Actual CG Path')
hold on
plot(LC1x,LC1u,LC1x,LC1L,...
LC3x,LC3u,LC3x,LC3L,...
```

```
LC5x,LC5u,LC5x,LC5L,'linewidth',3)
```

hold off

grid

ylabel('Y, m','fontweight','bold')

xlabel('X, m','fontweight','bold')

title('ISO TR 3888 Lane Change Course','fontweight','bold')

% print -djpeg -r300 DLC\_CG\_Path

# References

- Anon., "Test Track for a Severe Lane Change Maneuver," 1-ed, International Standard Organization, ISO 3888, 1975.
- [2] Mimuro, T., Ohsaki, M., Yasunaga, H., and Satoh, K., "Four Parameter Evaluation Method of Lateral Transient Response," SAE Paper 901734, Society of Automotive Engineers, Warrendale, PA, September 1990.
- [3] Pacejka, H., <u>Tire and Vehicle Dynamics</u>, 1-ed, Society of Automotive Engineers, Warrendale, PA, 2002.
- [4] McKibben, D., and Frey, N., "2007 BMW Mini Examination of Suspension & Steering Kinematics & Compliance", Michelin Report FW576, March 2007.
- [5] Rhyne, T., "Flat Track Test Pacejka Data for BMW MINI OEM Tires", Test Ranking No. J7H46030, Michelin America Research Center, April, 2007
- [6] Powell, R., Khekare, A., Law, E. H., Ayalew, B., "Transient Handling Analysis of a 2007 BMW Mini Equipped with TWEELs: Comparison of Tests & Simulation and Parameter Studies," Clemson University, Department of Mechanical Engineering Report TR-08-119-ME-MMS, March 2009.
- [7] Crolla, D., Chen, D., Whitehead, J., and Alstead, C., "Vehicle Handling Assessment Using a Combined Subjective-Objective Approach," SAE Paper 980226, Society of Automotive Engineers, Warrendale, PA, February 1998.
- [8] King, R., Crolla, D., Ash, H., and Whitehead, J., "Identification of Subjective-Objective Vehicle Handling Links Using Neural Networks for the Foresight Vehicle", SAE Paper 2002-01-1126, Society of Automotive Engineers, Warrendale, PA, March 2002.
- [9] Lincke, W., Richter, B., and Schmidt, R., "Simulation and measurement of driver vehicle handling performance." SAE Paper 730489, Society of Automotive Engineers, Warrendale, PA, May 1973.
- [10] Meetings with Dr. Rhyne, Steve Cron, BMW representatives, and Clemson representatives, June 2008-September 2008
- [11] Bergman, W., "Correlation Between Open and Closed Loop Vehicle Handling Measurements," SAE Paper 800846, Society of Automotive Engineers, Warrendale, PA, June 1980.
- [12] Good, M., "Sensitivity of Driver-Vehicle Performance to Vehicle Characteristics Revealed in Open-loop Tests," *Vehicle System Dynamics*, Volume 6, Issue 4, pp. 245-277, Netherlands, October 1977.
- [13] McRuer, D. and Klein, R., Automobile Controllability-Driver/Vehicle Response for Steering Control, Vol. I Summary Report, U.S. Dept. of Transportation, DOT HS-801-407, February 1975.
- [14] Lincke, W., Richter, B., and Schmidt, R., "Simulation and Measurement of Driver Vehicle Handling Performance," SAE Paper 730489, Society of Automotive Engineers, Warrendale, PA, May 1973.
- [15] Genta, G., <u>Motor Vehicle Dynamics: Modeling and Simulation</u>, 1-ed, World Scientific, Singapore, 1997.
- [16] Horiuchi, S. and Yuhara, N., "An Analytical Approach to the Prediction of Handling Qualities of Vehicles With Advanced Steering Control System Using Multi-Input Driver Model," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, Volume 122, Issue 3, pp. 490-497, United States, September 2000.
- [17] Oscarsson, M. "Variable Vehicle Dynamics Design-Objective Design Methods," Master's Thesis performed in Vehicular Systems, Department of Electrical, Linkopings University, July 2003.
- [18] Law, E. H., ME 453/653 Class Notes, Clemson University, Fall 2007.
- [19] Anon., "CarSim 8 Math Models," Mechanical Simulation Corporation. http://www.carsim.com/downloads/pdf/Math\_Models\_C80.pdf. June 2009.
- [20] Henderson, J., "2007 BMW Mini, TWEEL Auto Study, Project ID A7255A, Dry Objective Handling Test Report", Test No. 172899, Michelin America Research Center, April2007.
- [21] Sutton, R., <u>Modelling Human Operators in Control System Design</u>, Research Studies Press LTD. Taunton, Somerset, England. 1990.
- [22] Law, E. H., "Development of Input Parameters for a Simulation of the Vertical Ride Response of a Model Year 2007, BMW Mini REVISED," Clemson Report, Report No. TR-07-120-ME-MMS, December 28, 2007.
- [23] Law, E. H., "Development of Input Parameters for a Simulation of the Lateral Dynamic Response of a Model year 2000, C-5 Corvette," Clemson Report, Report No. TR-00-102-ME-MMS, August 2000.
- [24] Rhyne, T. XYZ Rigidity Test for BMW Mini OEM Tires, Test No. 172421, Michelin America Research Center, April 2007.

[25] Day, Terry D., Metz, L. Daniel, "The Simulation of Driver Inputs Using a Vehicle Driver Model," SAE Paper 2000-01-1313, Society of Automotive Engineers, Warrendale, PA, March 2000.