Integrated Vehicle Stability and Power Management Controls for Electric Vehicles

by

Andy Wong

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

An integrated vehicle controller is presented for electric vehicles using independently driven wheel motors. This topology takes an optimal control approach to enhancing a vehicle's performance, stability, and energy consumption metrics simultaneously in a unified software structure. The logical output of this algorithm is a set of re-distributed wheel torques, to create torque vectoring for stability-focused yaw rate tracking, and longitudinal biasing to modify motor load for energy savings. A real-time numerical approach to solving the optimization problem is also presented, and shown to offer benefits over a closed form analytic approach. In this, solution constraints are used to link considerations such as nonlinear motor limits, tire friction envelopes, and lower-level traction control loops. To test the efficacy of this control structure, two vehicle test platforms were constructed as retrofits of production gas SUVs for electric drive. For this, the component layout is given, followed by an explanation of the software code structure as performed in a Simulink/Carsim/dSpace environment. Results from these platforms are given, with experimental and simulation data for traction control, yaw performance tracking and drive cycle power consumption. Proven performance over a variety of maneuvers and surface conditions further demonstrate the controller's stability and suitability for mass production.

Acknowledgments

I'd like to thank General Motors for providing the initial concept of the control approach, their technical support, and guidance through the years with their bi-weekly meetings. I'd also like to thank the Ontario Research Fund (ORF) and Automotive Partnership Canada (APC), who provided the large amount of funding needed to make such a project possible. Finally, my gratitude goes to Professor Amir Khajepour, who built our work environment and guided so much of our research.

Dedication

This is dedicated to my parents, who have always taught me to value education above all else.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recently, increasing attention has been placed on hybrid and electric vehicles as a cost-effective way of reducing reliance on oil. Historically, full-electric vehicles have not been widespread due to technological and cost restraints, but trends are improving for the long term future. Vehicles such as the Nissan Leaf, Chevrolet Volt, and Tesla S have shown that it is feasible to release a commercially viable electric car for the masses. Over time, full electric drives look to replace internal combustion engines as technology matures. This transition allows for distributed wheel motors at each corner, since electric motors could be produced smaller and implemented with fewer parts than their gasoline equivalents. Given independent wheel actuation and faster response times over traditional clutch and differential 4WD systems, a new generation of algorithms are enabled to exert greater control.

This thesis outlines the design process of such an electric vehicle, produced with the intention of researching new control algorithms. A major goal of the project was to create production-like qualities in packaging, weight distribution, and performance. Given the project's association with the General Motors Corporation, a pair of 2010 Chevrolet Equinoxes were chosen as the base platform, one to serve in RWD and one in 4WD modes. These vehicles are kept with stock suspension and tires, and independently driven for each active wheel. It is meant to be indicative of an outwardly production-like electric SUV, as shown in Figure 1.1.

By retrofitting a vehicle, effort was saved by not attempting major mechanical design. This allowed more resources for supporting the research group's primary focus in vehicle controls. The control system described in this thesis is comprehensive, and encompasses performance, stability, and energy management considerations. At the core of this system

Figure 1.1: Electric Equinox Platform

is the Holistic Corner Controller (HCC), a flexible algorithm which dynamically redistributes tire forces between driven wheels to create torque vectoring for vehicle control. To accomplish this, the algorithm uses state feedback from the vehicle, largely from sensors, state and parameter estimation algorithms. Together, these provide information for feedback such as body acceleration and tire forces, which the HCC uses to calculate control actions.

The HCC is set up as an optimization problem, where a cost function is used to determine the best actuator output given a desired yaw moment, under the constraints of tire friction limits, number of available actuators, and motor dynamics. Weights within the cost function may be tuned according to the vehicle's design objective - emphasis may

be placed on stability, handling, or comfort. Most importantly, the main benefit of this approach is a unification of the control structure, since the HCC can optimize against multiple control objectives in one single step.

The dynamic aspects of the HCC respond most when tire limits are reached, and tire slips result in an error between the desired and actual response. A large portion of driving is not performed at these limits of handling however, and consists instead of start-stop accelerations with mild turns. This led to the development of an energy management component, which generates the baseline torque at each wheel while minimizing energy usage during periods of low performance requirements. This algorithm reduces energy usage by finding a torque distribution between the front and rear axles at any given speed, such that the weighted average of the two efficiencies is at a maximum.

Chapter 2 outlines the theoretical basis of the controller, while Chapter 3 outlines the platform's hardware design. Software implementation of the structure is described in Chapter 4. Finally, results are given in Chapter 5 for a variety of drive cycles and test maneuvers.

1.1 Literature Review

The HCC is a formulation which maintains vehicle stability by generating an additional force and yaw moment on the vehicle via lateral differentials in tire force. The maximum deliverable forces at each tire are constrained however, and varies with the dynamic state of the vehicle. In order to achieve a desired state, the tires are controlled by supplementary traction or braking torques. Effective systems already exist in the market place, by companies such as Bosch who supply various forms of Anti-Lock Braking Systems (ABS) [2] and Electronic Stability Control (ESC) [25] for large Original Equipment Manufacturers (OEMs). In traditional gas-engine vehicles, brakes and active clutches are used to distribute torque and create yaw moments, and generally act as a control layer separated from the rest of the vehicle [4]. In a term coined as "peaceful coexistence", OEMs have had little control over the inner workings of these stability programs, and higher-level controls have had to be tuned around a number of unknowns.

In literature, examples of research have been published which solve specific aspects of yaw moment control, such as: active steering[24] [34], torque vectoring [33] [30] [36], or traction control [39] [35]. However, there are few that consider all factors at the same time.

This gives motivation for the development of a unified structure, which can perform all major aspects of vehicle control at once, such that maximum performance can be extracted without margins for "peaceful coexistence". Furthermore, the projected availability of distributed-motor vehicles place high-bandwidth actuators within the OEMs' reach, allowing higher fidelity control as compared to traditional mechanical systems.

A unified control structure similar to the HCC was first proposed by Y. Hattori [21], in a closed form objective function that was solved using nonlinear optimization techniques for differential torque and braking signals. Later developments saw the addition of active steering capabilities [23], but one major setback for both solutions is the use of slip ratios inside the cost function, which needs a subsequent tire model to attain patch forces at each corner. Because of this requirement, more parameters are required to arrive at a solution. In Hattori's implementation for example, a brush tire model is utilized, which requires cornering stiffness and vertical coefficients of stiffness, among other things. These parameters change as a function of road conditions and tire model, which may limit its practical deployment. The HCC improves upon this, by directly solving for desired forces at each wheel patch instead. As a result, tire models and associated slip ratios do not need to be considered in the analytic solution. Ultimately, lower control loops using force estimation or motor torque feedback may be used to carry out desired forces. The evolution of this work was first proposed by a GM patent in [17], and elaborated upon in [15] and [1].

The main objective function used in the HCC may be solved analytically, or numerically with bounded constraints on the solution. Several options exist for solving constrained optimization problems, including interior point methods [20], active-set methods [18], augmented Lagrangian methods [9], trust-region reflective methods [28], and other gradient methods [29]. In this thesis's implementation an active set method was found to perform best, given the speed and availability of third-party solvers.

In addition to vehicle dynamics, a practical vehicle controller may also need to integrate other considerations, such as actuator fault tolerance, or energy use optimization. This is the point at which no unified controllers in literature have been found to provide all of the aforementioned features.

For example, [16] specifies a target function to optimize energy consumption for longitudinal and lateral motion. However, performance under low friction coefficient conditions, and interaction with traction control algorithms are undefined. Similarly, [31] deploys a piece-wise, condition based algorithm to decompose a non-convex motor efficiency map into sub-regions for optimization. Due to its rule-based structure, the system is unable to adapt for changing road friction limits, and stability problems may arise when ideal assumptions are not met.

In the proposed HCC solution, a smooth numerical search is proposed instead, for a range of possible front/rear distribution ratios at each instance in time. In this, motor models predict the efficiency for each ratio. Dynamic weighting functions are described, which shifts bias away from energy management and towards traction control as a function of road friction coefficients.

As a result, the HCC described in the rest of this thesis constitutes a novel approach to comprehensive vehicle controls, encapsulating a large number of control topics within a single architecture and easily applied across a range of different scenarios.

1.2 Contributions

Within this thesis, focus has been given to my specific contributions, and the work applied to each module to transform the HCC from a simulated theoretical work into a working controller for a full sized vehicle. The work laid out is a snapshot of the vehicles and their software structure, as presented to General Motors at their Milford testing facility in October of 2013. Due to the large scope of this large project however, success would not have been possible without contributions by my fellow lab mates. A summary of our team's logical division at this time is given below in Figure 1.2.

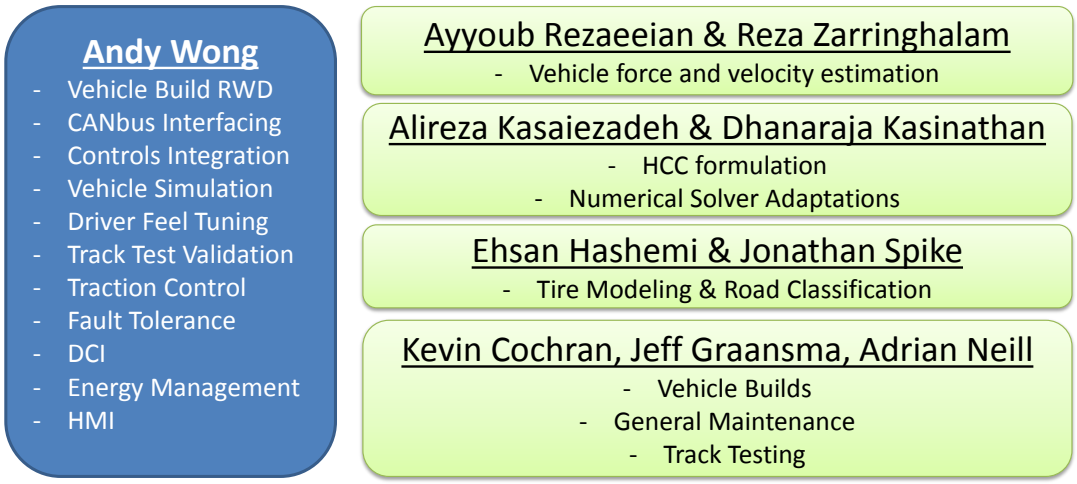


Figure 1.2: Contributions

Numerous people have worked on specific aspects of the system before myself. Prior to the Equinox, team members Alireza Kasaiezadeh, Abtin Athari, Ayyoub Rezaeeian and Jonathan Spike applied initial versions of the HCC, DCI, vehicle estimation, and tire estimation algorithms respectively on an older Opel hatchback platform using a GM supplied code base, with the aid of Kevin Cochran. The first part of my work was to build a new physical platform to alleviate the physical limitations of the Opel. The second part was to design a new software structure that brought the Waterloo-developed modules into a cohesive system for the first time. The final part involved developing solutions for things that did not function as expected.

For the first (RWD) vehicle build, technicians Kevin Cochran and Adrian Neill gave me help as we wired up retrofit components, and laid out the trunk space for interfacing to the

dSpace computer. Two base vehicle chassis which included the electric power trains were purchased from AMP motors. Later, Jeff Graansma designed a front motor mount for the second platform and worked with Kevin to replicate the layout for our 4WD platform. For physical testing, I got much help from our technicians Kevin Cochran and Jeff Graansma for booking the facilities, towing cars, and gathering data. For these tests, I designed the dSpace control-desk HMIs used to debug and configure our vehicles with in real time.

In the context of full vehicle integration, I was responsible for the high-level integrated vehicle control code, with help from Kevin Cochran for setting up signal flows, such as the interfacing of test modules initially carried over from the Opel platform. CAN-Bus network interfacing, Global-A security bypasses, and subsequent code layout changes were performed by myself. Other contributions involve development of the Simulink code's CarSim simulation branch, which utilizes a model built by GM measurements to provide virtual sensor feedback. Using this, I performed many full-vehicle simulations using GM-specified maneuvers to tune responses, and track/debug problems with lower-level function blocks such as the estimators. I regularly updated this environment, and pushed it to lab mates for the the purposes of virtual code validation before track days.

At the logical level, my contributions were motivated by requirements to showcase a working car in real life, and the additional challenges that ensue. For example, initial DCI modules exhibited instabilities, including oscillations due to filtering phase delays, poor tracking of vehicle states, and path deviations due to sensor noise and integration error. My contribution to the DCI involved the change to a stable reference model, the implementation of an atemporal dead-band filter to avoid phase offsets and oscillation, and shifting the architecture from yaw-moment control to that of a yaw-rate control in order to solve rate-derivative noise sensitivity and steady state errors. Related to the DCI was driver feel tuning which also fell under my responsibility. In this, I utilized track days to tune vehicle model parameters and pedal mappings to achieve desirable vehicle responsiveness. Along these lines, I also simulated the feel of an automatic transmission inside the car using mapped electric creep torques to achieve user expectations of a "normal" feel.

The HCC approach was initially provided by GM, and modified by Saber Fallah and Alireza Kasaiezadeh. As such, parts of the initial mathematical formulation, and literature references contained in this thesis are shared with these authors due to the common background. In its original format, tire force estimations from Ayyoub Rezaeeian and Reza Zarringhalam were combined with road friction values from Ehsan Hashemi and Jonathan Spike to draw tire limit ellipses. Stacked estimation errors led to inaccurate calculations however, and slip control was unsuccessful. My contribution in this regard was the implementation of a traction control system using a constrained numerical HCC formulation, solved by an open-source solver configured by Dhanaraja Kasinathan. This differed from

a simplified approach initially created by Alireza Kasaiezadeh, which removed the tire ellipse completely and replaced it with independent PID loops at each wheel outside of the HCC, operating off wheel acceleration feedback. Rate-derivative noise from the wheel encoders led to erratic performance in that case, and the uncoordinated wheel actions gave undesirable net yaw moments as a consequence of overriding optimized HCC outputs. In my contribution, the tire ellipse is maintained for operation within a tire's grip margin, so as to not interfere with regular control. Once limits are surpassed, a slip-based formulation affecting the bounds within the HCC objective function corrects the offending wheels and redistributes torque to compensate for the control action. This was significant in maintaining optimized, neutral yaw control even under dynamic maneuvers.

Additional contributions to the HCC involve the consideration of actuator torque limits, which is a dynamic function of speed in electric motors. The original analytic formulation had no provisions for this, and instead applied a constant penalty weight in the optimization process to discourage actuation. Under harsh maneuvers, it is possible for the HCC to request forces past deliverable limits, and saturation resulted in squarewave-like control outputs that gave non-ideal behaviour. Under the new paradigm, motor data are used to dynamically constrain desired corner forces, which resulted in smoother control requests that follow the optimization process.

Motor limits also have a direct impact on fault tolerant control. An initial solution was provided by Alireza Kasaiezadeh which shrank the dimensionality of an actuator contribution matrix within the objective function, followed by a doubling of torque request on the failed side to maintain longitudinal force. This however, resulted in unwanted net yaw moments due to errors between commanded and achievable motor forces. For example, anything over a 50% throttle request would result in actuator saturation on the compromised side and led to yawing. Drivetrain asymmetry under fault exaggerates errors, and resulted in oscillations about neutral yaw requests. My contribution in this area was to utilize the constrained solver to specify faults as a reduced tire force bound, while governing the remaining wheels with dynamic motor limits. This way, the formulation dimensionality did not need to be changed, and all control outputs were achievable. As a consequence, straight-line and dynamic performance under failed states were significantly improved.

Finally, the initial idea for the energy management feature arose from a study I performed while specifying possible motors for the platform. In this, it was determined that longitudinal torque distribution was effective in shifting peak motor efficiency points for energy savings. Alireza Kasaiezadeh carried this procedure forward by creating a parallel search algorithm to determine optimal distribution ratios at each instance of time. Integration into the HCC framework, vehicle simulations, and drive cycle simulations were performed by myself.

Chapter 2

Integrated Vehicle Stability and Energy Management Control

In the literature, individual aspects of vehicle dynamics have been widely studied, but there have been a limited number of works which offer a practical, integrated control strategy that considers all aspects of vehicle dynamics simultaneously. The integrated vehicle controller outlined in this chapter proposes to alleviate this problem, with the control structure shown in Figure 2.1.

A mathematical formulation for the control topology will be given, which optimizes vehicle dynamics performance and energy consumption simultaneously. Broadly speaking, two main components work together to provide a comprehensive, integrated vehicle control.

First, target vehicle states are output by the Driver Command Interpreter (DCI), which uses a vehicle model to generate desired behavior. This model is compared against feedback signals from the actual vehicle response. Error between the desired and feedback signals are used to generate corrective signals, sent to the Holistic Corner Controller (HCC) for actuation. The general HCC formulation accepts a target vehicle state (CG forces and yaw moments) as input, and generates an output involving tire and suspension forces, or active steering angles. At the time of this writing, the formulation has been physically tested on a platform supplying 4 wheel independent drive with manual front steering. Without active steering, the HCC is restricted from directly affecting lateral forces, and thus performs primarily in a torque vectoring mode to track desired yaw moments. An analytic general solution for the optimization process, followed by a numerical solver approach to constrain solution values is shown to provide a means of integrating traction control and actuator limits within the objective cost function.

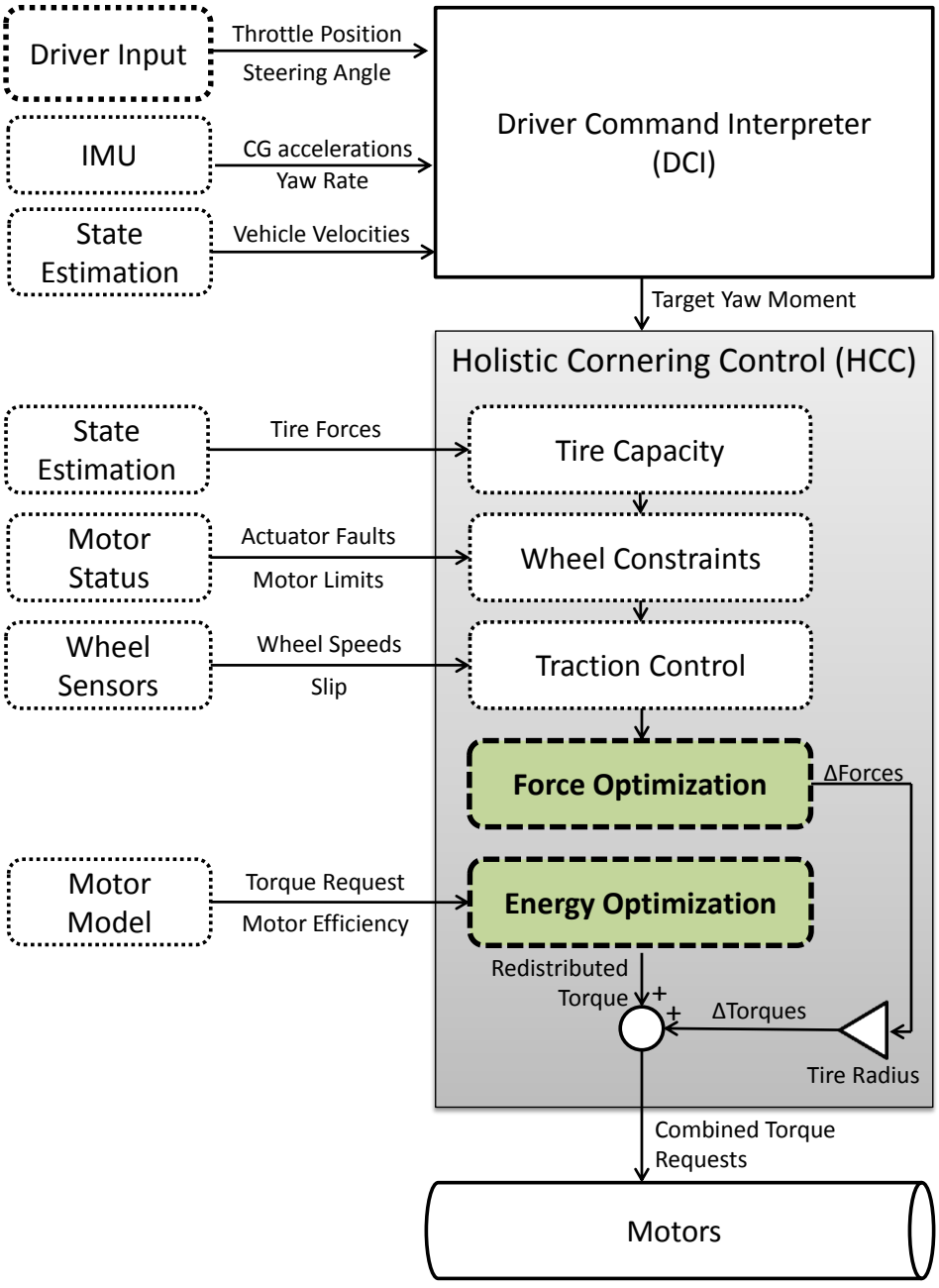


Figure 2.1: Control Structure

2.1 Background Information

2.1.1 Vehicle Parameters

Information about parameters specific to the Equinox platform, and the definition of variables used in subsequent sections are provided below in Table 2.1.

Table 2.1: Vehicle Parameters

Vehicle Width	1.842m	w
Vehicle Height	1.684m	h
Vehicle Wheelbase	2.858m	L
Front CG Distance	1.42m	a
Rear CG Distance	1.44m	b
Ground Clearance	0.198m	c
Vehicle Mass(2WD)	2047kg	m
Vehicle Mass(4WD)	2306kg	m
Tire Radius	0.365m	r_{eff}
Rolling Resistance	0.02	μ_R
Road Friction Coefficient	0-1	μ
Drag Coefficient	0.36	C_d
Air Density	1.1839 $\frac{kg}{m^3}$	ρ_{air}

2.1.2 Vehicle Dynamics

The main motivation behind a vehicle stability controller is the minimization of slip, in order to maintain the best possible tracking of any given driver input. A vehicle's ability to control the motion of its mass depends ultimately on its tires, and the amount of friction force they can apply to the ground. To generate force, a tire requires slip, which can occur in both longitudinal and lateral directions. A visualization in the longitudinal direction is given below in Figure 2.2.

Here, slip is calculated as the proportion difference between the hub and patch velocities, and defined as:

$$S = \frac{V_{x,patch} - V_{x,hub}}{max(V_{x,patch}, V_{x,hub})} \tag{2.1}$$

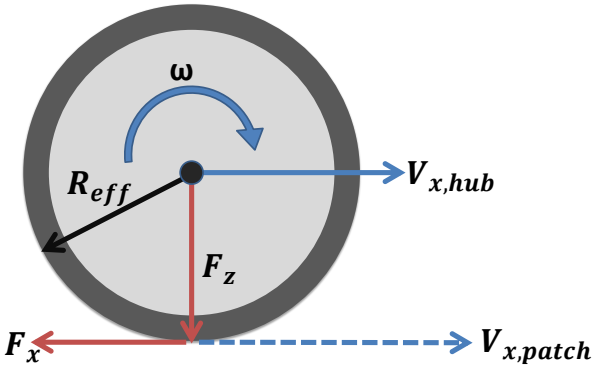


Figure 2.2: Simple Tire Mechanics

where $V_{x,hub}$ would generally require direct measurement or estimation, and $V_{x,patch}$ is a simple rotational relationship, defined as $V_{x,patch} = \omega R_{eff}$. ω is the rotational velocity of the wheel, and R_{eff} represents the effective tire radius, which can marginally change due to rubber compliance.

Meanwhile, longitudinal tire force F_X is a function of both normal forces F_Z, and the road's coefficient of friction μ, the maximum of which is defined as $F_{X,max} = \mu F_Z$. Three longitudinal tire friction curves for differing normal forces are given below in Figure 2.3 for $\mu = 1$, for the platform's Michelin Latitude tires to show their general relationships.

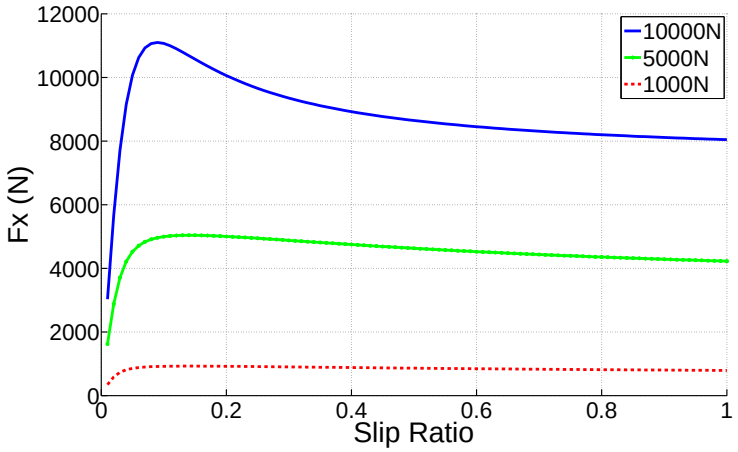


Figure 2.3: Longitudinal Tire Force Curve

It is seen that this tire's response curve is linear for small slip ratios, but plateaus and diminishes with increasing slip. If a tire slips beyond its peak grip threshold, a loss of control is likely since diminishing tire forces lead to yet more slip. Therefore, a goal of ABS, ESC, and other traction control systems is to keep tire slip within the linear region, such that these run-away conditions do not exist and maximum usable tire forces are attained.

In lateral directions, slip is referred to in terms of slip angles, of which there are two. A diagram showing a visualization of this slip occurring in conjunction with a steered angle δ is shown below in Figure 2.4 for a simplified two-wheel bicycle model.

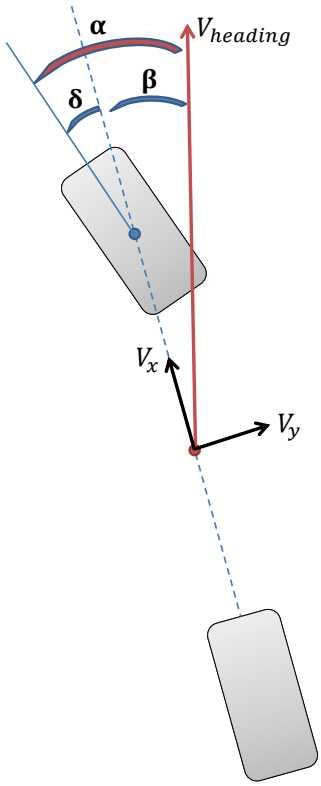


Figure 2.4: Lateral Slip Definitions in a Bicycle Vehicle Model

The first angle is β, which applies to the body and defines the difference between a vehicle's local orientation versus the actual heading of its center mass. Mathematically, it is defined as:

$$\beta = -tan^{-1}\left(\frac{V_y}{V_x}\right) \tag{2.2}$$

where V_x and V_y represent longitudinal and lateral components of vector $V_{heading}$, respectively.

The second angle, α, applies at the wheel level, instead of the body. Assuming the wheel is steerable, it represents the total angle between this steered angle and the vehicle's velocity vector. Mathematically, it is defined as:

$$\alpha = \delta - \beta. \tag{2.3}$$

For cases where a wheel is non-steerable (such as in the rear), δ becomes 0, and $\alpha = \beta$. Ultimately, the lateral slip angle α is used to generate lateral force. Figure 2.5 below shows resultant lateral force curves for the tires used in this thesis, for three differing normal forces.

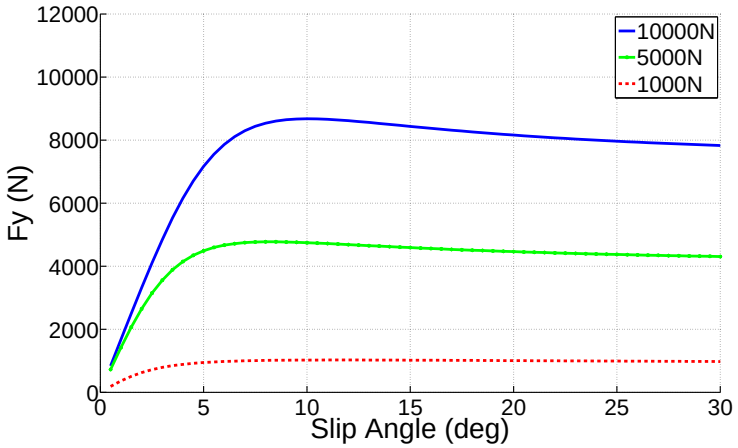


Figure 2.5: Lateral Tire Force Curve

In a relationship similar to the longitudinal case, forces are linear for small slip angles, and eventually plateau and diminish for greater slip. Of note is the X-axis, which substitutes the longitudinal slip ratio with a lateral slip angle.

2.2 Driver Command Interpreter

A vehicle model in the form of a DCI is used to produce a target state for the HCC to follow. The inputs to the DCI are steering wheel angles, driver torque/brake requests, and current vehicle states. In general, the DCI translates these inputs into desired forces, yaw rates and moments about the center of gravity (CG).

A 2 degree of freedom bicycle model as initially shown in Figure 2.4 was chosen for the current HCC implementation, and coupled with a linear tire model to generate an ideal yaw rate. In a linear tire model, forces are assumed to scale linearly with increased slip angles α. This results in the lateral force F_Y, defined as:

$$F_Y = C_\alpha \alpha \tag{2.4}$$

where C_α represents the linear coefficient, also known as cornering stiffness.

Actual tire response, as seen in Figure 2.5, is nonlinear in high-slip conditions, and leads to an error between the simple model and actual feedback. This discrepancy works well as a target for the HCC, as it is assumed that the error would be corrected via the HCC to realize an ideal vehicle response.

Vehicle performance is greatly affected by tuning the cornering stiffness, due to its impact on feed-forward control. A single term known as the understeer coefficient encapsulates the impact of cornering stiffness on a bicycle model, and is defined as:

$$K_{us} = \frac{mb}{LC_{\alpha f}} - \frac{ma}{LC_{\alpha r}} [19] \tag{2.5}$$

where a represents the distance from CG to the front axle, b the distance from CG to the rear axle, and L the total wheel base. m represents the vehicle mass, and independent values for the front and rear cornering stiffness are defined via $C_{\alpha f}$ and $C_{\alpha r}$ respectively. This coefficient plays a major role in the model's desired steady state yaw rate, defined as:

$$r_d = \frac{V_x \delta}{L + K_{us} V_x^2} \tag{2.6}$$

where δ in this case refers to the front steered angle. By relating driver feel to K_{us}, vehicle stability and response may be easily tuned. In general, a negative K_{us} results in an oversteering vehicle, while a positive value results in understeer. As speeds increase, an oversteering vehicle will tend to yaw more in response to a steering input, increasing the likelihood of a spin-out. An understeering vehicle on the other hand, will become less

sensitive, resulting in a safer but less responsive trajectory. Yaw gain is a useful term in explaining the difference between these two situations, and represents the yaw rate output per degree of steered input. Mathematically, this is defined as:

$$\frac{r_d}{\delta} = \frac{V_x}{L + K_{us} V_x^2}. \tag{2.7}$$

A visualization of this term's influence is shown below in Figure 2.6, which plots the vehicle yaw gain for two K_{us} values over a range of speeds. Here, a sample value of 0.001 is used to represent the positive K_{us} case, while a value of -0.001 represents the negative K_{us} case.

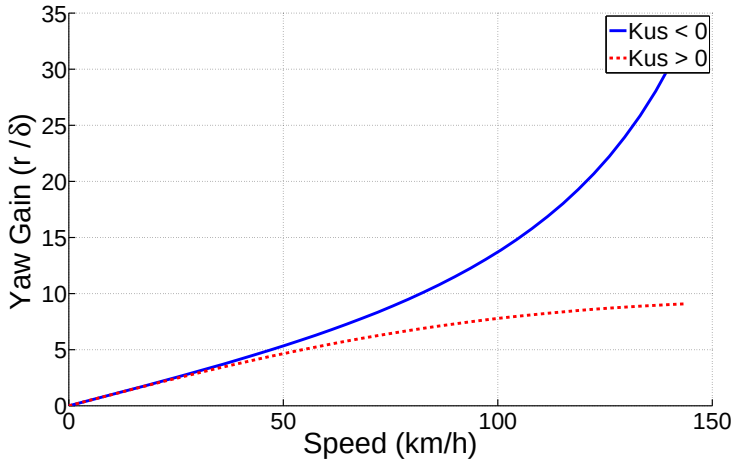


Figure 2.6: Yaw Gain Response Comparison

It is seen that the $K_{us} < 0$ line becomes divergent and unstable as velocities rise. The $K_{us} < 0$ line on the other hand, displays an increasingly dulled response, indicating a safe, understeering car suitable for mass consumption. In the Equinox, the DCI was tuned to match vehicle performance on dry roads as closely as possible, and resulted in a positive K_{us} value as per from the factory.

At this point, it should be noted that yaw rates are output from the model, while the HCC requires yaw moment as the input. As such, the tracking of an ideal yaw rate requires an intermediary controller. There are two major motivations for doing this.

Firstly, the stock commercial IMU only provides yaw rate signals for feedback. Earlier versions of the DCI attempted to use the time derivative of this for moment feedback,

but high-frequency sensor noise causes particularly large spikes in the resulting values. Smoothing filters may be applied, but they incur time delays which lead to laggy response and controller oscillation.

Secondly, a pure moment-based controller is subject to steady-state errors which are only correctable by tracking its integral over time, which is once again the rate. These reasons combined, led to the selection of a rate-based feedback system coupled with a reliable proportional controller to command the HCC. Added to the proportional controller is an adjustable dead-band function centered around zero to further attenuate unwanted actuation due to road noise or small disturbances. The yaw moment output of this filter, G_z, is defined as:

$$G_z = k_g e_{dci} \frac{|e_{dci}|}{k_d + |e_{dci}|}, \text{ where} \tag{2.8}$$

$$e_{dci} = r_d - r_{fb}. \tag{2.9}$$

Here, e_{dci} is the controller error, while r_d and r_{fb} are the desired and measured feedback yaw rates respectively. k_g is the yaw moment controller gain, and k_d is a constant controlling the amount of dead-band.

After physical tuning, a dead-band coefficient $k_d = 0.1$ was applied to Equation (2.8), resulting in the response curve shown below in Figure 2.7. To enhance performance under faulted actuator conditions however, this dead-band is disabled in order to enhance the correction of small yaw deviations, ensuring that the vehicle remains in a straight line.

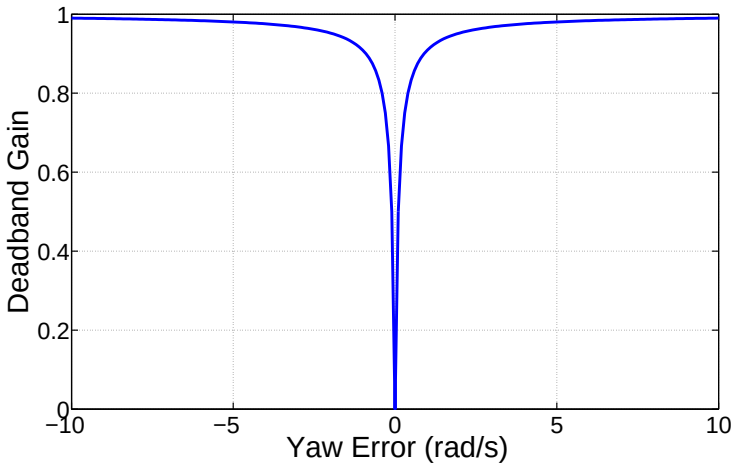


Figure 2.7: DCI Deadband Function

2.3 HCC Formulation

Generally, a ground vehicle has four corners, each of which have longitudinal, lateral and normal forces. Longitudinal forces can be controlled via drive motors and brakes, while lateral forces can be influenced by steering angles or force differentials about the center line (ie. torque vectoring). First, a general vector defining all tire forces may be defined as:

$$f = [F_{x1}, F_{y1}, F_{z1}, F_{x2}, F_{y2}, F_{z2}, \ldots, F_{x4}, F_{y4}, F_{z4}] \tag{2.10}$$

where F_{xi}, F_{yi}, F_{zi} are the longitudinal, lateral and normal tire forces respectively, at each corner $i = \{1, 2, 3, 4\}$, defined by the corner numbering shown below in Figure 2.8.

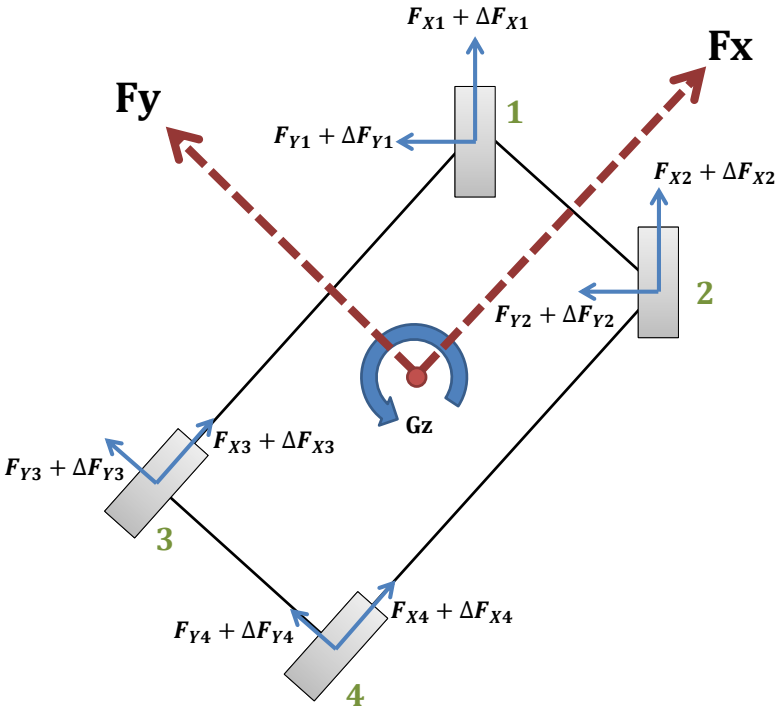


Figure 2.8: Force Conventions

It is implied that longitudinal augmenting forces ΔF_X are provided by motor torque, while lateral forces ΔF_Y are provided via steering angles. Let there be a vector Δf which represents these overlay forces in the same format as vector f, namely:

$$\Delta f = [\Delta F_{x1}, \Delta F_{y1}, \Delta F_{z1}, \Delta F_{x2}, \Delta F_{y2}, \Delta F_{z2}, \ldots, \Delta F_{x4}, \Delta F_{y4}, \Delta F_{z4}]\,. \tag{2.11}$$

The main purpose of the HCC's control strategy is to minimize the error between target and actual vehicle forces. To determine this error, let there be a feedback state vector F, and target vector F^* of the same format for forces at the body level. These take the form:

$$F^* = \left[F_x^*, F_y^*, F_z^*, G_x^*, G_y^*, G_z^*\right] \tag{2.12}$$

$$F = \left[F_x, F_y, F_z, G_x, G_y, G_z\right] \tag{2.13}$$

where F_x, F_x^* are the longitudinal forces, F_y, F_y^* are the lateral forces, G_x, G_x^* are roll moments, G_y, G_y^* are pitch moments, and G_z, G_z^* are yaw moments, all about the CG.

In the current case where the vehicle does not have an active suspension to control pitch and yaw, these vectors may be shortened by removing the elements that these actuators affect, resulting in the forms:

$$F^* = \left[F_x^*, F_y^*, G_z^*\right] \tag{2.14}$$

$$F = \left[F_x, F_y, G_z\right] \tag{2.15}$$

which has the net effect of reducing the formulation's dimensionality, but not the overall format of the optimization procedure.

For feedback, let the error between desired and actual forces be defined as E, where:

$$E = F^* - F. \tag{2.16}$$

Adding the impact of an overlay torque dictated by Δf produces a projected error term E_p, defined as:

$$E_p = F^* - F\left(f + \Delta f\right) \tag{2.17}$$

and approximated via a first order taylor series as:

$$E_p \approx F^* - \left[F + \frac{dF}{df}\Delta f\right] \tag{2.18}$$

$$\approx E - \frac{dF}{df}\Delta f. \tag{2.19}$$

Assuming that a control action vector Δf exists which drives E towards 0, optimal values for each element of Δf may be solved using a cost function P, defined as

$$P = \left[\frac{1}{2}E_p^T W_E E_p\right] + \left[\frac{1}{2}\Delta f^T W_m \Delta f\right] + \left[\frac{1}{2}\left(f + \Delta f\right)^T W_f \left(f + \Delta f\right)\right]. \tag{2.20}$$

Here, three terms are given, each with a different weight. The first term considers the vehicle force tracking error, weighted by W_E. The second term is used to weigh the energy consumption needed to apply Δf, and is weighted by W_m. The third term considers actuator limits, and is weighted by W_f. More detailed information on the weighting structure is given subsequently in Section 2.3.1.

In determining the error term E_p, $\frac{dF(f)}{df}$ is a Jacobian matrix since the body-level force F and the tire-level force f are both vector fields. For simplification, this matrix will be called A_F henceforth. In the general case, it is defined as:

$$A_F = \begin{bmatrix} \frac{dF_x}{dF_{x1}} & \cdots & \frac{dF_x}{dF_{z4}} \\ \vdots & \ddots & \vdots \\ \frac{dG_z}{dF_{x1}} & \cdots & \frac{dG_z}{dF_{z4}} \end{bmatrix} \tag{2.21}$$

where each element can be derived using equations of vehicle motion. These link the relationship between CG and tire forces, and follows the dimension of feedback vectors F and F^*. In the current case where the car is controlled in 2 dimensions, the motion equations are:

$$F_x = \sum_{i=1}^{4} \left[F_{xi} cos(\delta_i) - F_{yi} sin(\delta_i) \right], \tag{2.22}$$

$$F_y = \sum_{i=1}^{4} \left[F_{xi} sin(\delta_i) - F_{yi} cos(\delta_i) \right], \tag{2.23}$$

$$\begin{aligned} G_z = a \sum_{i=1,2} \left[F_{xi} sin(\delta_i) + F_{yi} cos(\delta_i) \right] - b \sum_{i=3,4} \left[F_{xi} sin(\delta_i) + F_{yi} cos(\delta_i) \right] \\ + w \sum_{i=2,4} \left[F_{xi} cos(\delta_i) - F_{yi} sin(\delta_i) \right] - a \sum_{i=1,3} \left[F_{xi} cos(\delta_i) - F_{yi} sin(\delta_i) \right]. \end{aligned} \tag{2.24}$$

By substituting Equations (2.19) and (2.21) into Equaton (2.20), the objective function becomes

$$P = \left[\frac{1}{2} (E - A_F \Delta f)^T W_E (E - A_F \Delta f) \right] + \left[\frac{1}{2} \Delta f^T W_m \Delta f \right] + \left[\frac{1}{2} (f + \Delta f)^T W_f (f + \Delta f) \right]. \tag{2.25}$$

Using this cost function, the optimal Δf is a vector such that the cost function P is minimized. Assuming the weight matrices W_E, W_m, and W_f are selected such that the

objective function P remains positive definite, then P is minimized when:

$$\frac{dP}{d(\Delta f)} = 0. \tag{2.26}$$

Isolating for Δf results in the final analytic solution for solving the required overlay forces, namely:

$$\Delta f = [W_f + W_m + A_F^T W_E A_F]^{-1}[A_F^T W_E E - W_f f]. \tag{2.27}$$

For actuation, these forces may be multiplied against an effective tire radius to determine motor torque requests. The effective tire radius is currently assumed to be a constant value, although minor deflection may be seen under harsh scenarios. In these cases, tire model estimation may be used to determine the dynamic value, and remains as an opportunity for future work.

2.3.1 Weight Selection

The primary HCC formulation denoted in Equation (2.27) contains matrices that are multiplied against the weights W_E, W_m, and W_f. Respectively, these will be referred to as weights for the first, second and third terms.

First Term

The first term, controlled by W_E, applies to the vehicle CG's force tracking error, and is applied against the error signal. In the two dimensional case, this becomes a diagonal matrix,

$$W_E = diag\,[W_{F_X}, W_{F_Y}, W_{G_Z}] \tag{2.28}$$

where each element $W_{F_X}, W_{F_Y}, W_{G_Z}$ weighs the corresponding axis present in the error vector E.

Second Term

The second term in the HCC equation deals with actuator effort, and is governed by W_m. This term attempts to save energy, by preventing excessive use of actuators. The format of this weight is a diagonal matrix, and is dependent upon the dimensionality of force vector f. In the 2 dimensional case, this becomes:

$$W_m = diag\,[W_{x1}, W_{x2}, W_{x3}, W_{x4}] \tag{2.29}$$

where each element corresponds to one of the corner actuators.

Third Term

The third and final term is W_f. In a general sense it prevents control actions past physical limits; in the planar 2D case, it limits tire forces to be within traction limits. Unlike the other weights, W_f is a dynamic function, and in the planar case considers friction capacity. Here, each component of the diagonal matrix W_f is magnified dramatically when requested forces approach limits, such that it functions as a penalty term. W_f follows the size of the tire force vector f, and in the 2D case is a diagonal matrix of the form:

$$W_f = diag[w_f(\rho_1^2), w_f(\rho_2^2), w_f(\rho_3^2) w_f(\rho_4^2)] \tag{2.30}$$

where ρ_i is a normalized indicator of tire utilization, with $i = \{1, 2, 3, 4\}$ acting as an index for each wheel. ρ_i is equal to 1 when fully saturated, and 0 when completely unloaded. The values of ρ_i may be calculated via a friction ellipse concept as:

$$\rho_i^2 = \left(\frac{F_{xi}}{F_{xi,max}}\right)^2 + \left(\frac{F_{yi}}{F_{yi,max}}\right)^2 \tag{2.31}$$

where μ_x and μ_y are the road friction coefficients in the X and Y directions respectively, $F_{xi,max} = \mu_x F_{zi}$, and $F_{yi,max} = \mu_y F_{zi}$.

Figure 2.9 shows a sample function for $w_f(\rho_i^2)$, which shows a desirable steep incline as tire utilization reaches its upper limit. At low utilization, the weighting function should remain small so as to not interfere with other terms in the objective function.

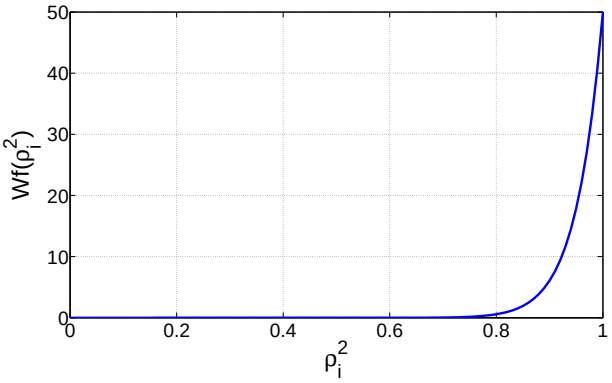


Figure 2.9: Proposed Weighting Function for w_f

2.4 Numerical Extensions

The HCC formulation up until this point is solvable from a purely analytic point of view, and provides an excellent baseline for an integrated torque vectoring application. Recall from Section 2.3.1 that the closed form solution required a dynamic weighting function to penalize actuation as a function of tire utilization. While this works for its intended purpose, one compromise is that an aggressive profile may result in control chatter, while a weak function may not penalize the objective function enough to prevent exceeding traction limits.

A numerical approach to solving the generalized equation on the other hand, offers the advantage of explicitly constraining the solution to stay within allowable friction limits at each wheel, without any interference if the operating point is within bounds. Dynamically setting constraints below the traction limit may also be used for other purposes, such as limiting motor output for energy savings, or dealing with partial actuator failures.

To utilize a numerical solver, the HCC formulation must first be converted into a standard quadratic programming problem [3], of the form:

$$\min_x \frac{1}{2} x^t H x + g^t x, \tag{2.32}$$

where x denotes the unknown tire force control vector, Δf. A full generalized conversion of the HCC formulation then results in the matrices:

$$H = W_f + W_m + (A_F^T W_E) A_F \tag{2.33}$$

$$g = -2 \left[A_F^T (W_E E_p) - W_f f \right] \tag{2.34}$$

where H is the Hessian matrix, and g is a vector containing the remaining linear terms.

As described in Section 2.3.1, weight matrices W_f and W_m correspond to tire friction and actuator effort considerations. In the extended numerical solution, both may be implemented via solution bounds instead, and the terms removed to result in the simplified matrices:

$$H = (A_F^T W_E) A_F \tag{2.35}$$

$$g = -2 \left[A_F^T (W_E E_p) \right] \tag{2.36}$$

where hessian matrix H is positive definite ($H > 0$). This is a convex minimization problem [3], and guarantees the existence of a unique solution. This, in turn can be solved using an iterative procedure [29].

The controller design objective in HCC leads to a quadratic programming problem which has a closed-form solution. Mathematically, a linear constraint is a convex function. Since both the quadratic programming problem and the linear constraints are convex in nature, the quadratic programming problem subject to linear constraints is also a convex problem [3]. Theoretically, such a linearly-constrained quadratic programming (LCQP) problem has a solution and is unique. On the HCC, this becomes:

$$\min_x \frac{1}{2} x^t H x + g^t x \tag{2.37}$$

$$\text{s.t. } B_L \leq x \leq B_U \tag{2.38}$$

where B_L and B_U are $(n \times 1)$ matrices denoting the lower and upper bound on wheel torque limits respectively, where n is the number of actuated wheels.

The experiments in this thesis used an active-set solver, with constraints on the output of each wheel set as an upper and lower bound torque. By constricting or expanding this bound, secondary factors are able to affect the HCC solution within the loop. These include dynamic tire limits, motor output limits, faulted actuator limits, and traction control loops. To use the limits derived from each of the aforementioned modules, the lowest value is selected and pre-existing tire forces are subtracted to determine remaining margins. The bounds become:

$$B_{U,i} = \min\left(F_{xi}^{fault}, F_{xi}^{slip-ub}, F_{xi}^{mot}, F_{xi}^{tire}\right) - F_{xi} \tag{2.39}$$

$$B_{L,i} = \max\left(-F_{xi}^{fault}, F_{xi}^{slip-lb}, -F_{xi}^{mot}, -F_{xi}^{tire}\right) - F_{xi} \tag{2.40}$$

and are now completely defined for Equation (2.37), with F_{xi}^{fault}, F_{xi}^{slip}, F_{xi}^{mot}, and F_{xi}^{tire} representing the limits determined by actuator faults, slip ratios, motor limits, and tire utilization respectively. The derivations of each limit are given in the sections below.

2.4.1 Tire Capacity

The primary consideration for each wheel's torque limit comes from tire capacity, and the need for each wheel to stay within its friction ellipse. At each wheel, the tire's operating limits are:

$$F_{xi}^{max} = \mu_x F_{zi} \tag{2.41}$$

$$F_{yi}^{max} = \mu_y F_{zi} \tag{2.42}$$

where F_{xi}^{max} and F_{yi}^{max} represent the limits along the major X and Y axes respectively. μ_x and μ_y represent the coefficients of friction with the road along each respective axis, and F_{zi} is the vertical tire force at the contact patch. The allowable bounds using an ellipse model then become:

$$F_{xi}^{tire} = F_{xi}^{max}\sqrt{\left|1 - \frac{F_{yi}}{F_{yi}^{max}}^2\right|}. \tag{2.43}$$

2.4.2 Motor Limits

An explicit constraint on requested wheel torque allows for the consideration of electric motor limits, which can vary as a function of speed. In the HCC's analytic form, motor limits are handled by the W_m weight, which penalizes actuation. However, torque availability is not linear in electric motors, as shown below in Figure 2.10.

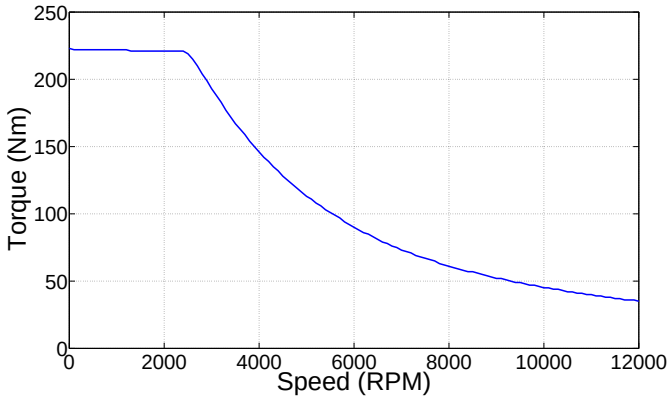


Figure 2.10: Electric Motor Curve

As a consequence, a static weight may result in unattainable requests as speeds rise. Given the numerical formulation however, this limit may simply be expressed as a dynamically changing function of RPM. This is defined as:

$$F_{xi}^{mot} = \frac{T_{max}(rpm)}{R_{eff}} \tag{2.44}$$

where T_{max} represents the peak available motor torque as a function of RPM. In a practical application, this may be attained via a lookup table, or generated via a piece-wise analytic function to account for the nonlinear transition point. Meanwhile, R_{eff} is the effective tire radius.

2.4.3 Drivetrain Reconfiguration and Fault Tolerance

Related to motor torque availability is drivetrain reconfiguration and fault-tolerant torque vectoring, where power delivery may be compromised due to failed or missing actuators. In the analytic HCC solution, actuators are removed as rows and columns from the Jacobian matrix shown in Equation (2.21), which leads to an on/off condition at that corner. Solving the HCC formulation with constraints however, provides a solution for partial actuator failure cases, where diminished torque delivery is still possible.

The detection of failed actuators and their remaining capacity is assumed to be known, and the bound may simply be set to the limit,

$$F_{xi}^{fault} = \frac{T_{i,failed}}{R_{eff}} \tag{2.45}$$

where $T_{i,failed}$ is the remaining capacity on the motor. For total failures, this element would be 0. R_{eff}, meanwhile is the effective tire radius.

To reconfigure the control formulation for FWD cars for example, the rear actuators could simply be given a bound of 0, facilitating easy transport of the controller between chassis configurations.

2.4.4 Traction Control

Anti-lock Brake Systems (ABS), Traction Control Systems (TCS) and Electronic Stability Control (ESC) are examples of less complex control systems that are widely used on cars to prevent wheel slip [25]. In such systems, a simple structure measures signals such as wheel speeds and wheel acceleration to detect slip. Once detected, various forms of power reduction or active braking takes place.

The HCC formulation lowers torque away from saturated wheels to remain within the tire ellipse, which is similar to the function of conventional traction control and stability systems. However, in practice, there may be delays or errors in the estimation of tire forces, which then requires a robust means of correction if limits are surpassed. To accomplish this, feedback from a lower wheel control loop is brought back into the objective function to maintain overall yaw tracking and prevent controller conflicts. The interface for this is through the bound vectors, B_L and B_U from Equation (2.38).

In ABS, brake actuation is pulsed when slip is detected, and uses a bang-bang type approach to utilizing the tire at its limits. This bang-bang approach to control is not

ideal, as the tire is allowed to pass limits before correction is applied. At the same time, the driver feels an unpleasant pulsation. In the HCC, a continuous controller is proposed instead, providing for smoother vehicle response.

Firstly, a velocity differential between the hub and patch speed is defined as a threshold before actuation, defined as:

$$S_{i,limit} = |K_s max(V_{xi,hub}, V_{xi,patch})| \tag{2.46}$$

where K_s is a tunable constant representing the permissible percentage of slip. Typically, values between 0.1 to 0.2 are used as this represents the linear tire region, and is where peak tire force lies. Wheels i exceed the slip threshold when

$$|V_{xi,patch} - V_{xi,hub}| > S_{i,limit}, \tag{2.47}$$

in which case, slip may be calculated as:

$$S_i = \frac{V_{xi,patch} - V_{xi,hub}}{max(V_{xi,patch}, V_{xi,hub})}. \tag{2.48}$$

The maximum and minimum allowable tire force in each direction may then be modified, by implementing a proportional controller to constrict the bounds. The limits then take the form:

$$F_{xi}^{max2} = F_{xi}^{max} - S_i P \tag{2.49}$$

$$F_{xi}^{min2} = -F_{xi}^{max} - S_i P \tag{2.50}$$

where P is a tunable proportional gain and F_{xi}^{max} is the state estimated limit from Equation (2.41) needing adjustment. In theory, a similar approach could be used to constrict tire limits in the lateral direction as well, given a lateral slip angle. However from a practical sense, lateral velocity estimators would be required to derive the slip angle. In deployment, these estimators often incur processing delays or absolute inaccuracies, which can cause unwanted control actions. A purely wheel speed based longitudinal approach, while not optimal, has the benefit of direct measurement feedback via wheel encoders.

Acknowledging this, an ellipse model may be used to determine maximum longitudinal tire forces, defined as:

$$\epsilon_{pos} = F_{xi}^{max2} \sqrt{1 - \frac{F_{yi}}{F_{yi}^{max}}^2} \tag{2.51}$$

$$\epsilon_{neg} = F_{xi}^{min2} \sqrt{1 - \frac{F_{yi}}{F_{yi}^{max}}^2} \tag{2.52}$$

where ϵ_{pos} and ϵ_{neg} define the positive and negative limits respectively.

Depending on the amount of slip, the modified limits may exceed deliverable forces by the motor. To alleviate this effect, a saturation is implemented to cap each bound within motor limits. These bounds are defined as:

$$\epsilon_{pos}^{adj} = max\left[min(\epsilon_{pos}, F_{xi}^{mot}), -F_{xi}^{mot}\right] \tag{2.53}$$
$$\epsilon_{neg}^{adj} = min\left[max(\epsilon_{neg}, -F_{xi}^{mot}), F_{xi}^{mot}\right] \tag{2.54}$$

where ϵ_{pos}^{adj} and ϵ_{neg}^{adj} represent the positive and negative saturated values, respectively.

Finally, a check is applied to ensure that the bounds do not surpass each other in the wrong direction, and the entire range is offset by the feed-forward driver requested force to determine remaining margin. Mathematically, these bounds are defined as:

$$F_{xi}^{slip-ub} = max\left(\epsilon_{pos}^{adj}, \epsilon_{neg}^{adj}\right) - F_{i,driver} \tag{2.55}$$
$$F_{xi}^{slip-lb} = min\left(\epsilon_{pos}^{adj}, \epsilon_{neg}^{adj}\right) - F_{i,driver} \tag{2.56}$$

where $F_{xi}^{slip-ub}$ and $F_{xi}^{slip-lb}$ are the final upper and lower bounds determined from traction control, respectively.

2.5 Energy Management Formulation

A vehicle traveling on the ground is generally subject to several major forces which the motors have to overcome. These are defined as:

$$F_a = \frac{\rho_{air} C_d w (h-c) V_x^2}{2} \tag{2.57}$$
$$F_r = \mu_R m g \tag{2.58}$$
$$F_d = m a_x \tag{2.59}$$
$$F_t = F_a + F_r + F_d \tag{2.60}$$

where F_a is air resistance, F_r is tire rolling resistance, and F_d is the requested acceleration from the driver. At any moment in time, the total force output is F_t, which is the sum of all resistances and accelerations. This total force may be converted to a required torque, defined as:

$$T_t = \frac{F_t}{r_{eff}} \tag{2.61}$$

where (r_{eff}) is the effective tire radius. Let this total torque be split in a 4WD system, such that:

$$T_t = T_f + T_r \tag{2.62}$$

where T_f and T_r represents hypothetical front and rear axle torques respectively. Let the distribution between the front and rear axles be defined as:

$$\nu = \frac{T_r}{T_f + T_r} \tag{2.63}$$

where ν is a set of values which encompass a range of possible front and rear values. Each element in this set is $\nu_n \in \{0, s, 2s, \ldots, 1\}$, with s representing an incremental step size between 0 and 1. Meanwhile, n represents the index of a particular element. The front and rear torques for each index may then be calculated as:

$$T_{f,n} = \frac{\nu_n F_t}{r_{eff}}, \text{ and} \tag{2.64}$$

$$T_{r,n} = \frac{(1-\nu_n) F_t}{r_{eff}}. \tag{2.65}$$

Assuming for symmetry, the per-motor torques are simply half that of each axle. Per-motor efficiency has been demonstrated to vary with requested torque, and may be defined as $\eta_{f,n}$ and $\eta_{r,n}$ for the front and rear motors respectively. The total efficiency, $\eta_{t,n}$ is then defined as:

$$\eta_{t,n} = (T_{f,n} + T_{r,n}) \left(\frac{T_{f,n}}{\eta_{f,n}} + \frac{T_{r,n}}{\eta_{r,n}} \right)^{-1} \tag{2.66}$$

$$= \frac{\eta_{f,n}\eta_{r,n}}{(1-\nu_n)\eta_{r,n} + \nu_n \eta_{f,n}}. \tag{2.67}$$

2.5.1 Energy Management Algorithm

In electric motors, the surface representing motor efficiency is a nonlinear function of torque and rotational velocity, with no guarantee that the objective function would be convex. This is a problem for optimization, as it prevents the use of an analytic solution.

To overcome this, a lookup table is used to accurately represent the surface instead, a common approach in applied engine controls for other areas such as fuel mix ratios or transmission shifting. Multiple efficiencies may be predicted at the motors' instantaneous

operating speed, using a set of possible distribution ratios (ν). Since the number of data points representing motor torque, speed and efficiency are bounded, search times are also bounded and practical for real time control purposes. A profile for the motors used in the Equinox is provided in Figure 2.11.

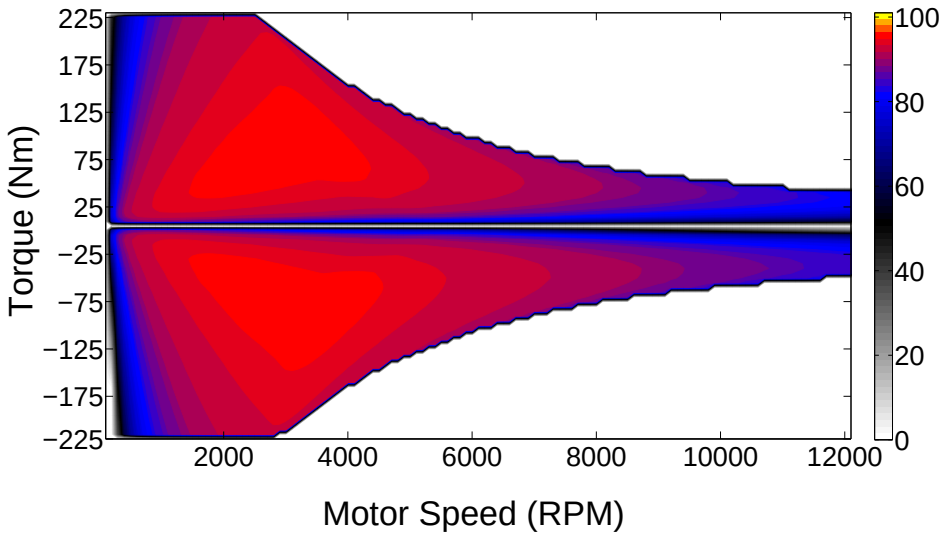


Figure 2.11: Engine Efficiency Map

This efficiency chart is indicative of a typical electric induction motor; the maximum deliverable torque fades at higher speeds, and the efficiency varies over its range of operation. The deviation in this case involves a peak efficiency of 95.5% near the surface centroid, and below 50% at the outer limits of the performance envelope.

In this project's implementation, ν_n is incremented from 0 to 1 in a step size s of 0.05. In theory, any reasonable step size may be used, but this value was chosen after testing showed that smaller step sizes resulted in diminishing returns since increasingly smaller amounts of torque is distributed.

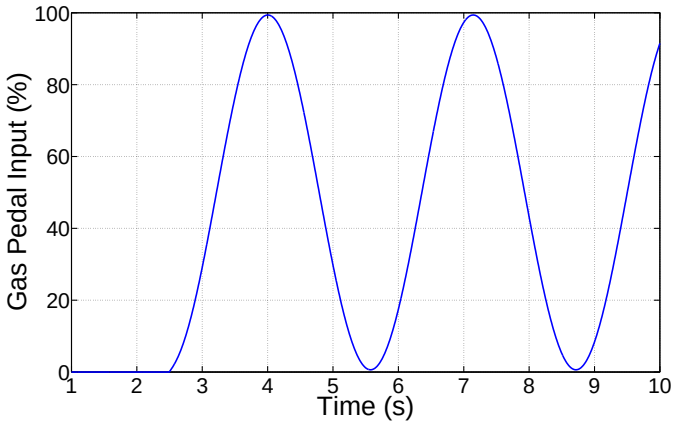
For each ν_n, $\eta_{f,n}$ and $\eta_{r,n}$ are attained using the table shown in Figure 2.11. Equation (2.67) is used to determine the net projected efficiency $\eta_{t,n}$ for each ratio ν_n. The optimal ratio is defined as:

$$\nu^{opt} = \nu_n | max(\eta_{t,n}). \tag{2.68}$$

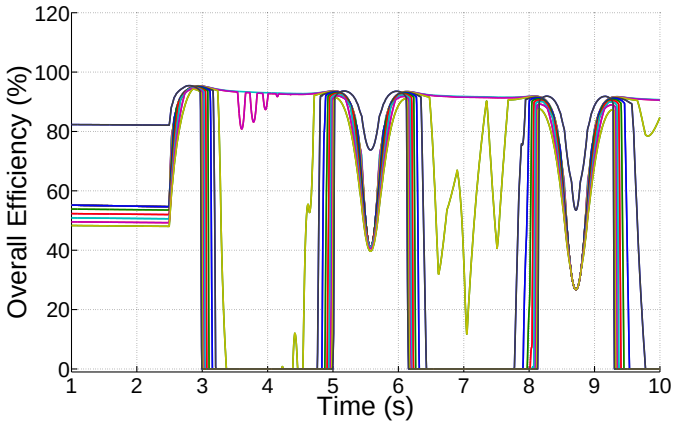
If identical motors are assumed for all corners of the vehicle, there may be cases where a tie occurs between a front or rear-biased ratio, due to a symmetry around $\nu = 0.5$. In such cases, the tie may be broken by placing more torque on the wheel with highest tire

capacity, thereby reducing the amount of slip and thus wasted energy. Parameters useful in determining this include vehicle pitch, longitudinal acceleration, roll angle, or vertical tire forces. In the current implementation, an IMU is used to give a rear-biased preference for acceleration cases, and forward-bias for braking.

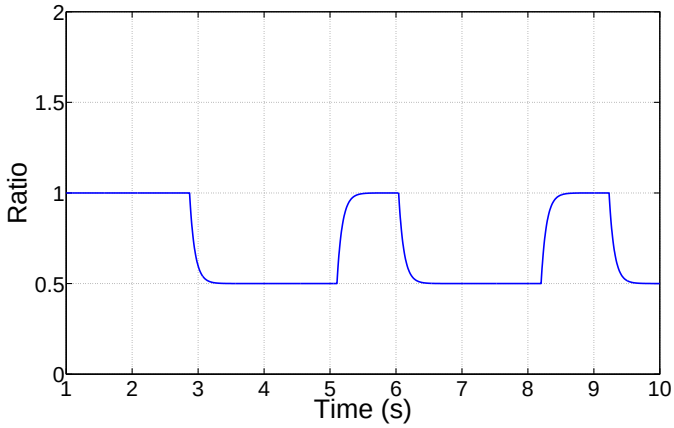
An example is hereby given for a straight-line acceleration, starting with a vehicle speed of 60 kph. The pedal input is sinusoidal to provide a range of requested torques, as shown in Figure 2.12a. As speeds increase, the operating efficiency of the motors vary. On the Equinox, a step size of 0.05 is used to search through the [0-1] range of possible distribution ratios, resulting in 21 indices. For demonstration purposes, the full set of efficiency curves is shown below in Figure 2.12b. Selecting for the most efficient scenario, the optimized ratios are shown in Figure 2.12c. Of note is the fully rear-biased efficiency's drop to 0, which represents the inability of two motors to fully deliver the load requested of a 4WD system. Thus, the solver is forced into a ratio of 0.5 to share the work. When loads are lower, it is seen that the ratio of 1.0 enjoys a higher efficiency at the 5-6s, and 8-9s marks. The system then selects for the rear-biased acceleration. Points between these events may be assumed to lie within the spectrum, leading to a gradual blend between these two ratios.



(a) Pedal Input



(b) Set of Overall Efficiencies for Compared Ratios



(c) Optimal Ratio

Figure 2.12: Sinusoidal Throttle Input and Resulting Efficiencies

2.5.2 Friction considerations

The primary mode of actuation in this algorithm is the redistribution torque to more heavily load one axle, such that the motors operate in higher efficiency region. However, traction limits may at times prevent this torque from being delivered, and instead result in tire slip.

To combat this, the search space may be constricted as a function of tire capacity, which fluctuates. This may be done by checking against current road friction and normal tire force estimators. Knowing this information, the upper bounds on transmissible torque per axle can be determined as:

$$T_f^{bound} = \mu(F_z^{FL} + F_z^{FR})r_{eff} \tag{2.69}$$

$$T_r^{bound} = \mu(F_z^{RL} + F_z^{RR})r_{eff} \tag{2.70}$$

$$\tag{2.71}$$

where T_f^{bound} and T_r^{bound} are the front and rear axle motor limits, respectively. F_z^{FL}, F_z^{FR}, F_z^{RL}, and F_z^{RR} are the front-left, front-right, rear-left, and rear-right normal tire forces respectively. A limiting axle is present, defined as:

$$T_{min}^{bound} = min(T_f^{bound}, T_r^{bound}). \tag{2.72}$$

This limiting axle is used to calculate the proportion of transmissible torque versus total requests, and defined as:

$$W = \frac{2T_{min}^{bound}}{T_t}. \tag{2.73}$$

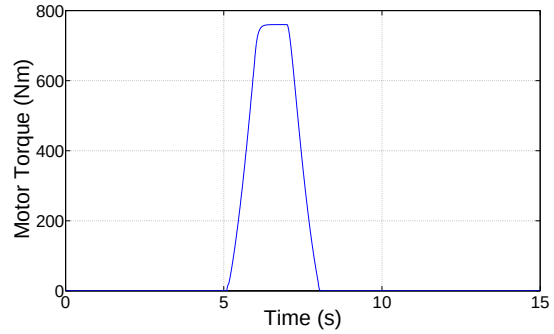
This weight is then used to constrict the search space, resulting in the term:

$$\nu^{adj} = (2W - 1)\nu + (1 - W). \tag{2.74}$$

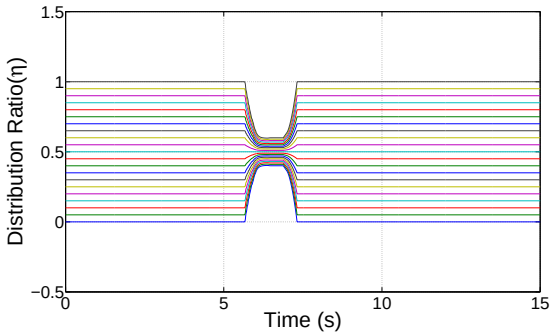
ν^{adj} may then be used in the place of ν to find the optimal value as defined in Equation (2.68). It should be noted that even a constriction to 50% distribution may result in individual tire capacities being surpassed, given sufficiently low friction and high driver requests. One assumption made with this algorithm is that lower-level traction control loops exist to prevent tire slip, and therefore returning the torque distribution to a neutral front/rear bias best prepares the car for peaceful coexistence with such systems.

This friction-based constriction is shown in a following example, whereby a pulsed driver input is applied on to both a slippery and grippy surface. Figure 2.13a shows the requested

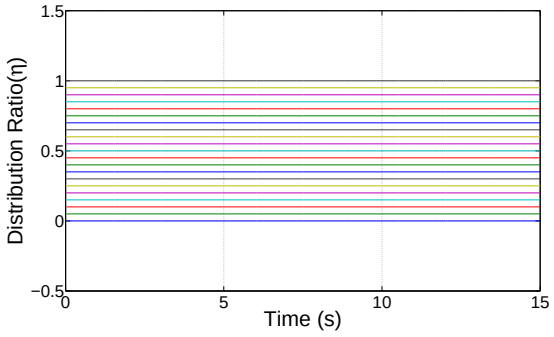
torque at one wheel, while Figures 2.13b and 2.13c show the allowable search spaces for the two cases of $\mu = 0.3$ and $\mu = 1.0$ respectively. It is seen that a slippery road surface of $\mu = 0.3$ is too low to carry the requested torque, causing the algorithm to converge upon the neutral state of $\eta = 0.5$ in preparation for a secondary traction control algorithm.



(a) Requested Motor Torque



(b) At $\mu = 0.3$, Range of η Contracts



(c) At $\mu = 1$, No constrictions

Figure 2.13: Comparison of Allowable Search Indices for Two Differing Surface Frictions

Chapter 3

Platform Design

At the GM collaboration's onset, a specific list of objectives were set for the vehicle test platform. These were:

1. Investigate the specific advantages of independently driven electric 4WD in the SUV class
2. Prepare an electrified platform for various control strategies
3. Benefit from an SUV's larger storage space, and more apparent roll dynamics
4. Keep vehicle as close to production-state as possible

Given these requirements, the Chevrolet Equinox was chosen as an ideal candidate. As a brief background, the Equinox features a unibody construction, MacPherson front struts, multi-link rear suspension, an electric steering rack, and a hydraulic braking system.

The conversion of a production Equinox into an electric research test-bed can then be separated into three main considerations - Mechanical (ie. packaging), Electrical (ie. batteries and power electronics), and software (ie. signaling and controls). Overall, this entails replacing the gasoline engine and tank with electric motors and batteries, and designing a new control system which can interface driver inputs with the new actuators.

3.1 System Layout

Ultimately, two sister SUVs were built, one as RWD and one as 4WD. The primary differences in layout between these are shown below in Figures 3.1 and 3.2 below.

In the RWD Equinox, the batteries are separated into two compartments, one within the front hood and the second in the gas tank cavity. The 4WD Equinox meanwhile, has three battery compartments, due to the installation of motors and inverters in the front hood space. As a result, the front battery was downsized, with the displaced cells relocated to the trunk. A more detailed view of the supporting components is shown below in Figure 3.3 for the 4WD model, while a logical layout of the interconnections between these components is given below in Figure 3.4. Multiple electrical and hydraulic loops exist in the platform. To cool the motors, Automatic Transmission Fluid (ATF) is constantly circulated through each motor, while a separate glycol loop keeps the power electronics

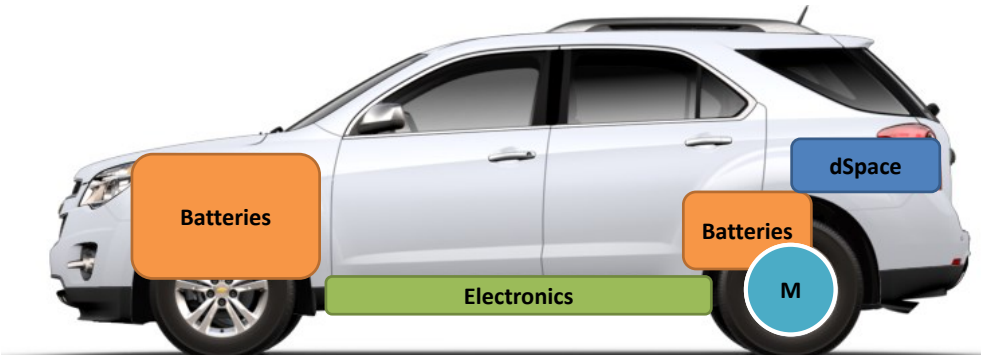


Figure 3.1: RWD Equinox Layout

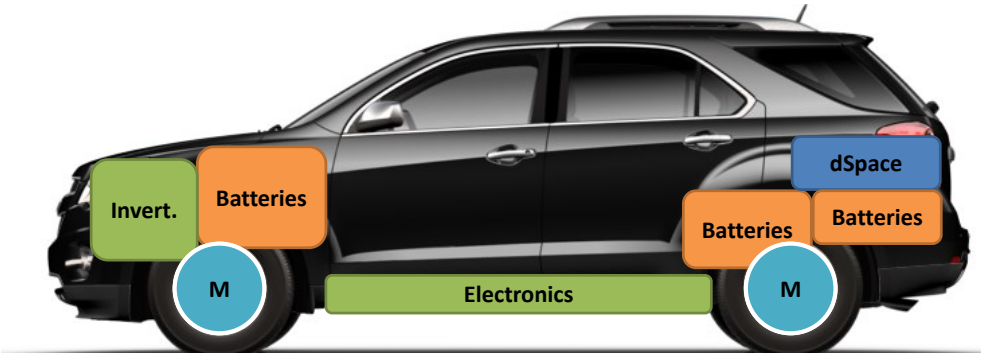


Figure 3.2: 4WD Equinox Layout

(AC inverters, DC-DC converter) within operating limits. Electrically, a high-voltage bus powers the drive motors, while a 12v high-current bus powers regular vehicle electronics, fans, and pumps.

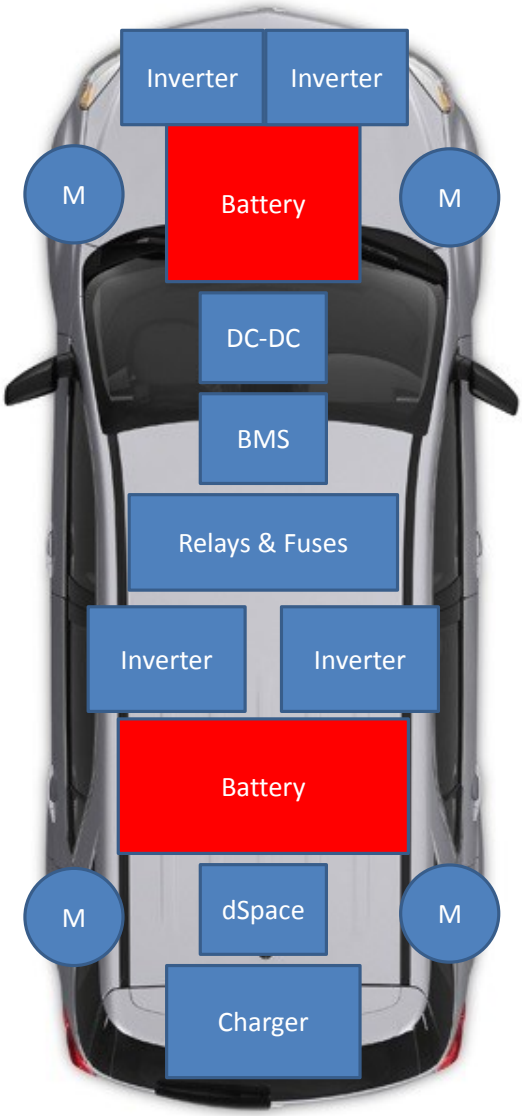


Figure 3.3: Detailed Component Layout (4WD)

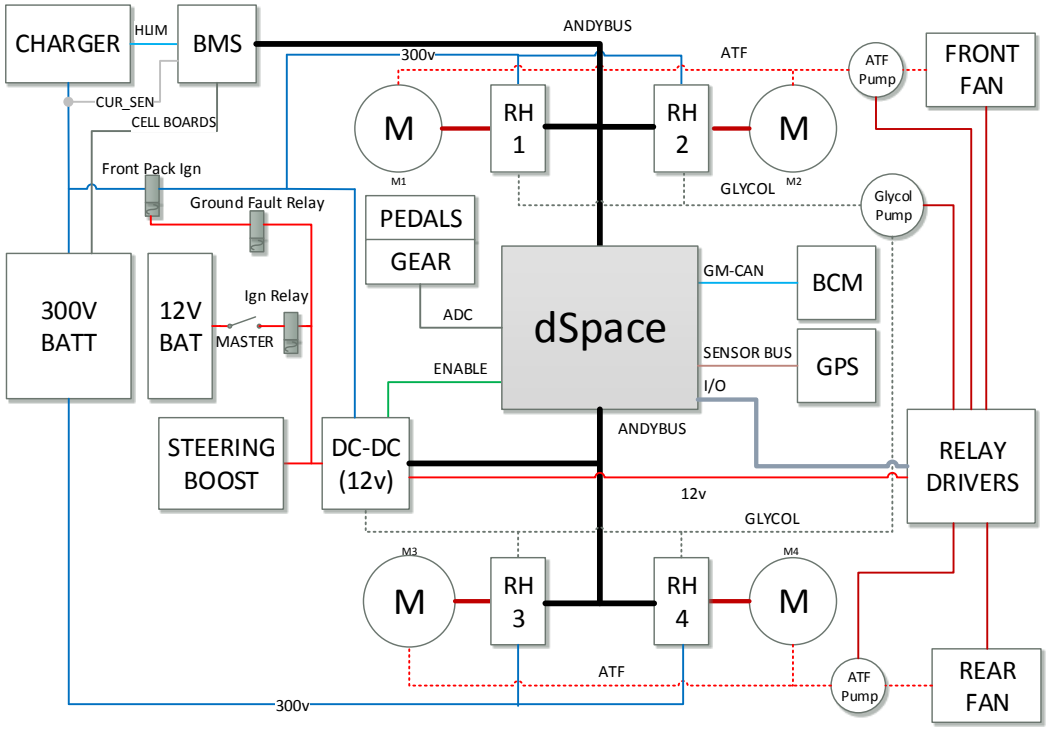


Figure 3.4: Systems Layout

It is seen that a central component to the vehicle's logical layout is the dSpace embedded computer, which performs all processing for the custom code. The specific model used on the platform is the dSpace Micro-Autobox II, which runs a real-time OS with code cross-compiled from a Matlab/Simulink environment. Relevant details of this machine involve a 900 Mhz PowerPC CPU, 4 hardware-driven CAN interfaces, 80 input/output channels, and 16-bit ADCs [10].

The dSpace takes in data from many sources for its computation, and it was decided to centralize all CAN signaling and I/O in the trunk, as shown in Figure 3.5a. Here, the dSpace can be seen in the upper right as a small aluminum box, while surrounding it are industrial automation DIN rails which house connection blocks for I/O.

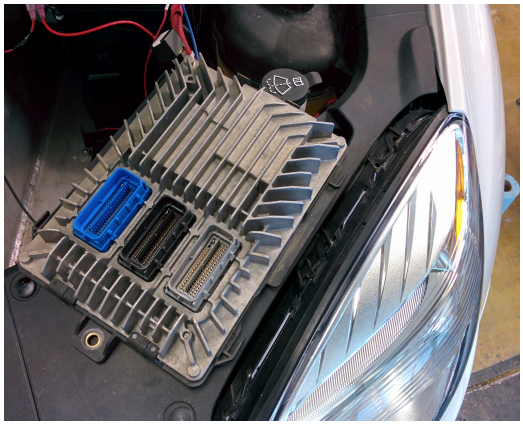
One feature of the dSpace system is its Control Desk software, which enables an Ethernet-connected laptop to interface with the Micro-Autobox to acquire signals, and adjust parameters for tuning. In the Equinox, a laptop is mounted to the passenger seat, as shown in Figure 3.5b.

(a) Trunk I/O layout

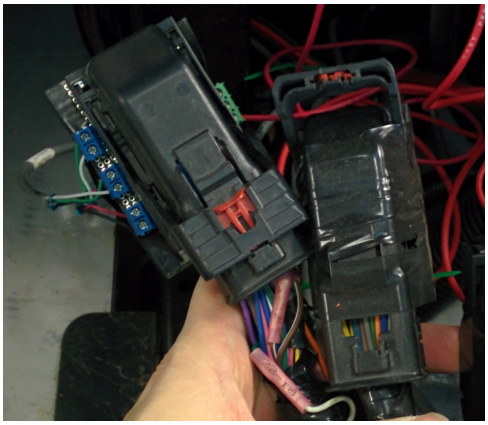
(b) Vehicle Computer Setup

Figure 3.5: dSpace Hardware

At the software level, the Engine Control Module (ECM) was used as the primary injection point for the dSpace to communicate with existing vehicle computers for sensors and outputs. To accomplish this, three sockets were removed from the ECM, and redirected to breakouts for the dSpace. Each of these connectors provide 73 signals normally meant for ECM processing, and provides a convenient means of accessing physical sensors in addition to the software buses. The original unit shown in Figure 3.6a was removed, and replaced with breakout boards that routed relevant signals to the dSpace as shown in Figure 3.6b.



(a) Removed ECU



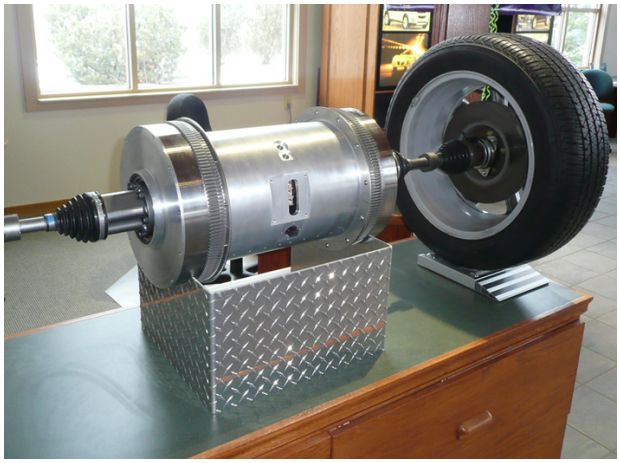
(b) Vehicle-Side ECU Dongles

Figure 3.6: Rewired ECU Dongles

The most important hardware signals from the ECM sockets are the pedal position sensors, which are analog potentiometers. The voltages are passed through ADCs, and translated into percentage values for torque actuation. A system bus wakeup pin is also essential, as this provides dSpace synchronization during vehicle start. Transmission gear positions on the other hand, are directly measured via a retrofit linear potentiometer, since the transmission controller was also removed. A detailed layout of the software using these signals will be provided in Chapter 4.

3.2 Components

The motors used in the Equinox are the HVH-250 models produced by Remy Corporation [22], and were purchased from a company called Amp Motors, which provides custom retrofit vehicles for the consumer market. The purchased modules placed two motors back-to-back in a single cooling jacket, as shown below in Figure 3.7a. Figure 3.7b shows in detail each motor's low-profile planetary gear transmission, which provides a single-stage 8:1 reduction.



(a) AMP Motor Module [6]

(b) Closeup of Motor Gearbox

Figure 3.7: Drivetrain Components

Driving the motors are AC motor inverters positioned near the chassis center line, controlled via CAN signalling. The specific models used are the PM100DX units produced by Rinehart Motion Systems [37], as shown below in Figure 3.8a. These operate at up to 360v input, and up to 300A continuous draw. In the Equinox's case, 356v DC is provided by the battery pack, and each PM100DX's 3-phase AC output is used to drive one motor. A picture showing the inverter casing is shown below in Figure 3.8a.

The platform uses a set of air-cooled Lithium-Ion-Phosphate cells, produced by GBS and shown in Figure 3.8b. 108 cells are used in the Equinox, connected in a single series chain. Each cell provides a nominal voltage of 3.3V, a capacity of 100 Ah, and a peak discharge rate of 10C. This combines to give the total vehicle a nominal pack voltage of 358V, and up to 1000A for the motors. The cells are reinforced with rectangular plastic housings, which makes for easier pack building.

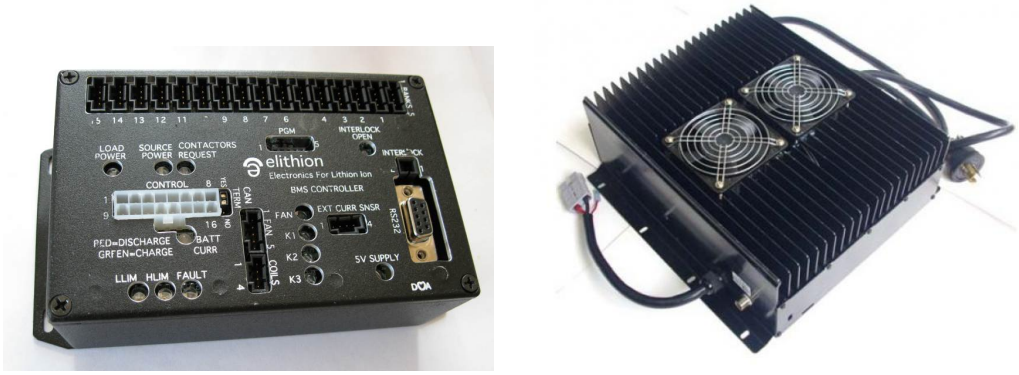
(a) AC Inverter Module [11] (b) GBS 100Ah Cells [27]

Figure 3.8: High Voltage Power Systems

Each cell is connected to a monitoring board, which reports back to an Elithion Battery Management System (BMS), which checks for voltage levels and internal resistance. During charge cycles, these cell boards also have a small resistor to dissipate energy, and are used in cell balancing. A current sensor located at the pack's total output allows the BMS to monitor discharge rates, and use coulomb counting to report usage and report state of charge. The unit used in the Equinox is the Lithiumate Pro, produced by Elithion Inc. A picture of this BMS is shown below in Figure 3.9a. To charge the cells, a power supply set to the pack's charging voltage provides energy to the pack. A relay controlled by the BMS stops current flow during balance cycles, and re-enables during bulk charge. The charger used in this project is the PFC-5000, produced by Elcon. A picture of this unit is provided below in Figure 3.9b.

Much of the production electrical system ran off a 12V lead-acid battery, charged in turn by an alternator driven by the engine. Since the engine was removed, a DC-DC converter is required to produce sufficient 12V for low power systems within the vehicle. In the Equinox, a 2.2 kW model produced by Delphi corporation [12] is used. For starting and initialization purposes, a lead-acid battery is still connected to the vehicle in parallel, to supply power before dSpace initialization completes and turns on the DC-DC converter.

For validation, an Oxford Technical Solutions, RT2500 6-axis GPS/IMU [26] is used to track vehicle forces and locations , and is mounted internally at the SUV's CG as shown in



(a) Elithion Lithiumate Pro [13] (b) Elcon PFC5000 Charger Unit [38]

Figure 3.9: Battery Charging and Management Systems

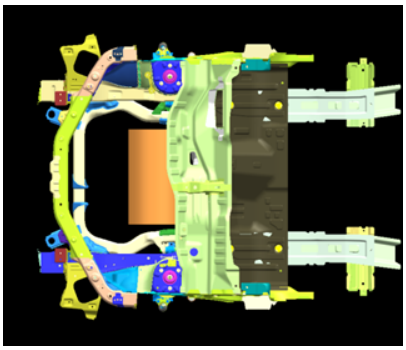
Figure 3.10a. To validate tire forces and moments around the wheel hub, a set of Michigan Scientific MSCLW12.8 [8] load wheels are also mounted to the car, as shown below in Figure 3.10b.

(a) Validation 6-axis GPS/IMU [26] (b) Michigan Scientific Load Wheels

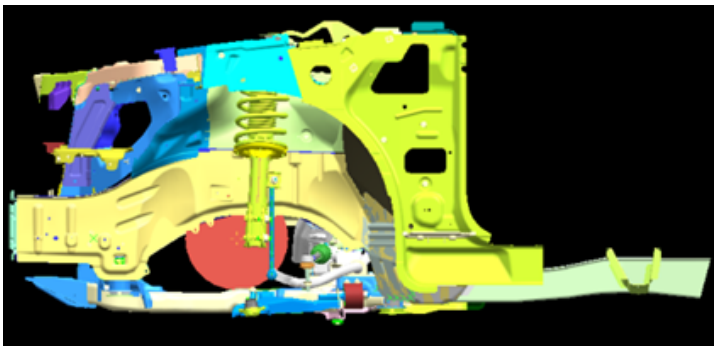
Figure 3.10: Validation Equipment

3.2.1 Physical Integration

The motor modules' compact size enables it to fit within the space of the removed engine components, as seen below in Figure 3.11 for the front sub frame, and Figure 3.12 for the rear.

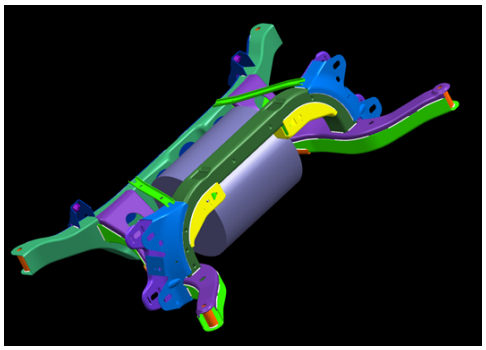


(a) Front Motor Packaging



(b) Side View, Front Motor Placement

Figure 3.11: Front Motor Packaging



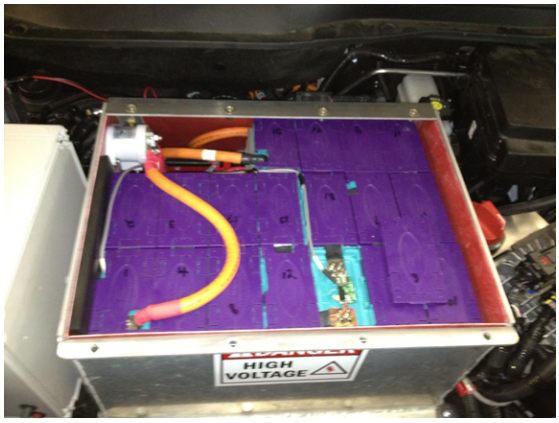
(a) Rear Motor Packaging

(b) Rear Motor Subframe

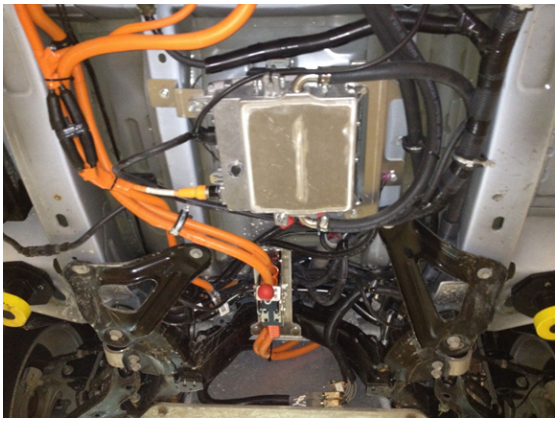
Figure 3.12: Rear Motor Packaging

Powering the motors are the batteries, half of which are stored in the front of the car. Figure 3.13a shows the front pack with the main ignition relay in the upper left, along with one battery cover removed to show the internal wiring of the cell monitoring boards.

Immediately behind the front battery pack, high-voltage power lines are passed through the chassis's transmission tunnel, and routed along the main structural members rear-ward as shown in Figure 3.13b.



(a) Front Ignition Relay



(b) DC-DC Converter Positioning

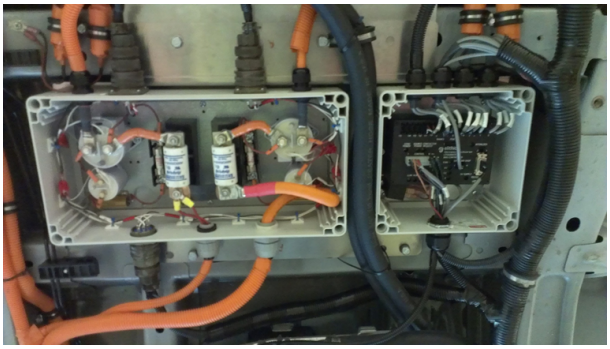
Figure 3.13: Front Vehicle Integration

The silver square in the middle of the picture is the liquid-cooled DC-DC converter unit, with black glycol cooling lines connected to its top and bottom. To the left is a high-voltage input, tapped off the main battery line. In the middle-bottom is a safety disconnect switch, which allows the battery pack to be physically disconnected for vehicle servicing.

Continuing, the high voltage power lines are passed into a relay and fusing section in the middle of the car, shown on the left in Figure 3.14a. These relays are controlled via software, and provides power to the motor inverters. To the right of the picture, the vehicle's BMS is seen. The multiple grey wires leading into the box carry cell-board communications in a channeled fashion, with each physical wire relaying information for 8-10 daisy chained boards. Behind the power relays are the rear motor inverters, as shown in Figure 3.14b.

In the middle, the high-voltage distribution block is seen, which splits the main pack voltage into three outputs for the two motor inverters and DC-DC converter. At the bottom of each blue inverter, 3-phase wiring outputs are seen, which powers each motor. The branched black hosing in this picture also shows the cooling loop, which provides glycol for cooling the power switching electronics.

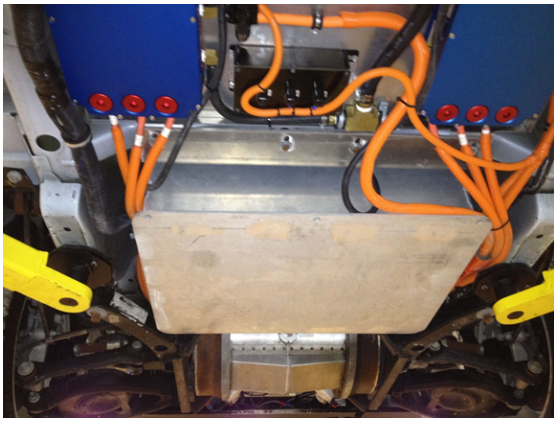
Behind the motor inverters lay the gas-tank battery, which can be seen as the large silver box in Figure 3.15a. The mounted electric motor module can also be seen, as the shiny cylindrical object behind the battery. An exploded view of the rear motor mounting location is shown in Figure 3.15b.



(a) Relays and Fuses

(b) Inverter Layout

Figure 3.14: Vehicle Midsection



(a) Rear Underbody

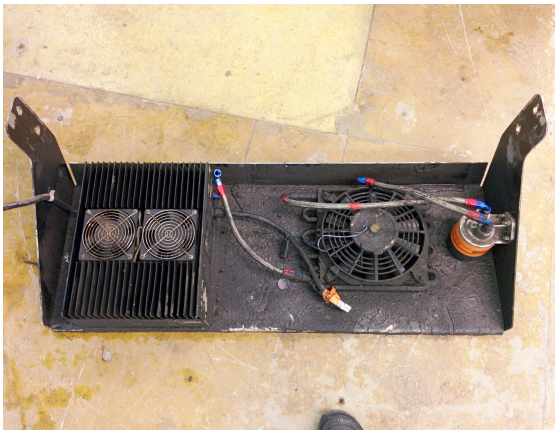
(b) Rear Motor Mounting

Figure 3.15: Vehicle Rear

Finally, the back of the vehicle is shown in Figure 3.16a. An accessory pan was added to provide extra mounting space under the vehicle, and holds the charger unit in addition to a radiator and filter for cooling the motors, as shown in Figure 3.16b.

Despite the major retrofit conversions performed to the underbody, the vehicle's interior cabin was left mostly untouched. The only major modification of note to the driver cockpit is an emergency stop as shown below in Figure 3.17a, which is directly wired to the primary high-voltage battery relay to cut power in an emergency. The 6-axis IMU used for validation purposes is mounted in the vehicle's interior due to CG location as shown in Figure 3.17b, but is completely removable as desired.

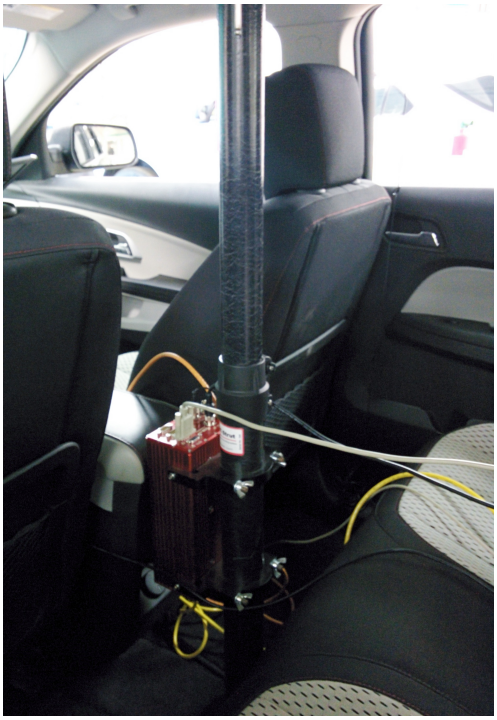
(a) Rear Pan



(b) Inside of Pan

Figure 3.16: Rear Accessory Pan

(a) Front E-Stop



(b) IMU Mounting Location

Figure 3.17: Vehicle Interior

Chapter 4

Software Design

The code was designed to be as modular as possible, to aid collaboration within the group and also to aid in versioning and debugging of logical blocks. There are three main software bundles:

1. Simulink Code for Logical Controls
2. Carsim Simulation Environment for Debugging
3. dSpace Control Desk for Physical Tuning

4.1 Simulink

The control code is created in Matlab-Simulink, cross-compiled to C code for execution on the dSpace host for real time applications. The code is split into two main sections - a component control level consisting of simple lower-level loops, and a high-level logic level which includes the HCC. The HCC requires several supporting function modules, the data flow of which is shown below in Figure 4.1.

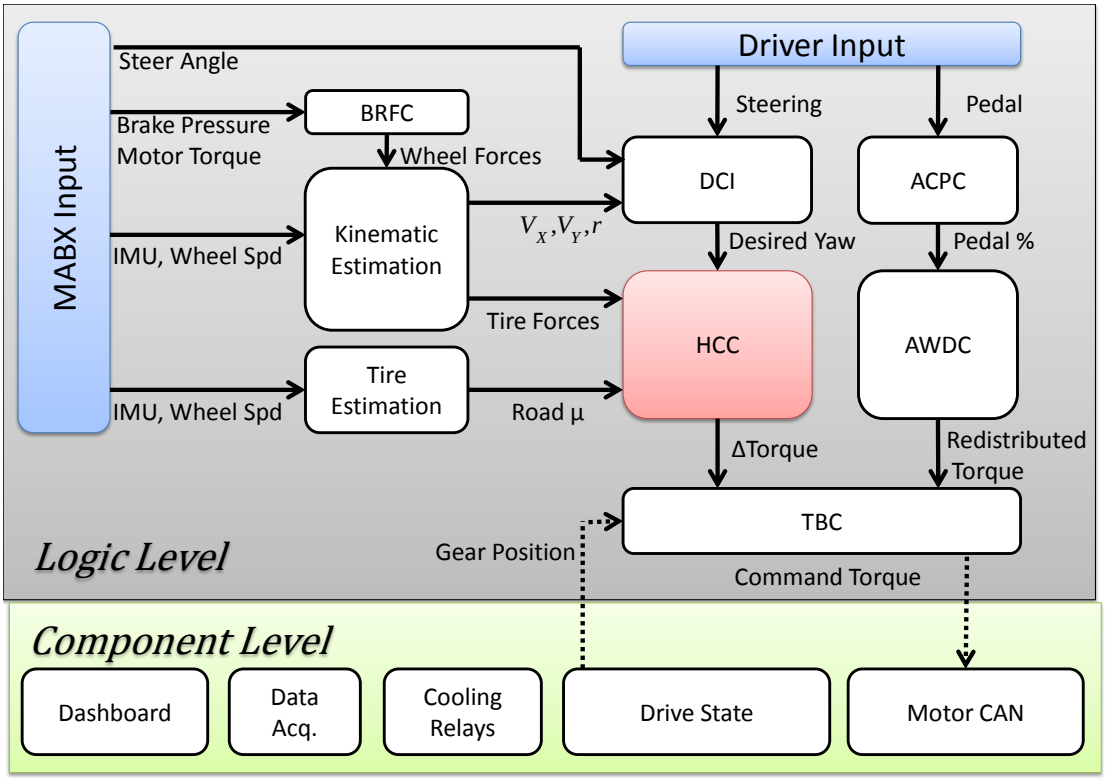


Figure 4.1: Code Logic Flow

It is seen that flow at the logic level is data dependent, and thus requires a function scheduler to trigger each block in a rational order to avoid data starvation. Details outlining each of the logic-level blocks will be given in Section 4.1.1. The component level control layer largely deals with direct hardware access via the dSpace embedded computer, and is covered in Section 4.3.

4.1.1 Controls - Logic Level

This function block houses a bulk of the HCC intelligence, and represents the mathematical processing of the control code.

ACPC (Accelerator Pedal Control)

The purpose of this block is to take in both redundant analog signals of the accelerator pedal position, and compare their values to ensure that a faulty sensor does not result in unintended acceleration. As output, a 0-100% representation of pedal actuation is given.

AWDC (All-Wheel Drive Control)

This module is responsible for the interpretation of desired driver torque, and handles feed-forward distribution to the four wheels. Four functions covered within this module include creep-torque generation, actuator fault tolerance, energy use optimization, and direction setting based on gear selection position. A diagram showing the logic flow is shown below in Figure 4.2.

Creep torque is a feature added to the system, in order to simulate the low-velocity push that is often felt in automatic transmissions. This is done by adding a small pedal percentage to the driver input, to trigger some movement to overcome stiction and create minor movement. A piece-wise function was implemented as:

$$P_C = \begin{cases} P_D + C_L, & \text{if } V_x < V_{T1} \\ P_D + C_L + (C_H - C_L)(V_x - V_{T1})(V_{T2} - V_{T1})^{-1}, & \text{if } V_{T1} < V_x < V_{T2} \\ P_D + C_H, & \text{if } V_x > V_{T2} \\ P_D, & \textit{Otherwise} \end{cases} \tag{4.1}$$

where C_L and C_H are low and high-speed creep constants respectively, V_{T1} is the transition speed at which the C_L begins to ramp down, and V_{T2} is the speed at which C_H is reached. P_D is the original driver pedal input, while P_C is resulting output to the system after accounting for the creep. In cases where braking action is requested, this creep torque is turned off for pedal inputs of greater than 1%.

The main drive torque, meanwhile is a function of the each motor's maximum available torque, multiplied against the gas pedal's percentage request. To enhance driver feel, a non-linear pedal mapping was applied to the user input, as shown below in Figure 4.3a.

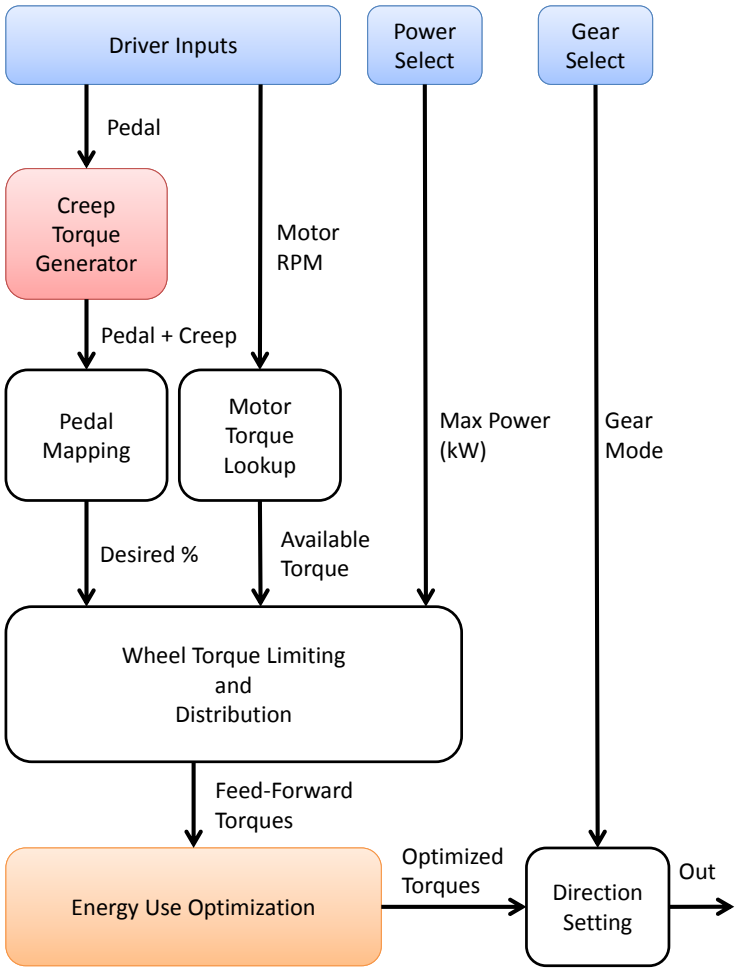
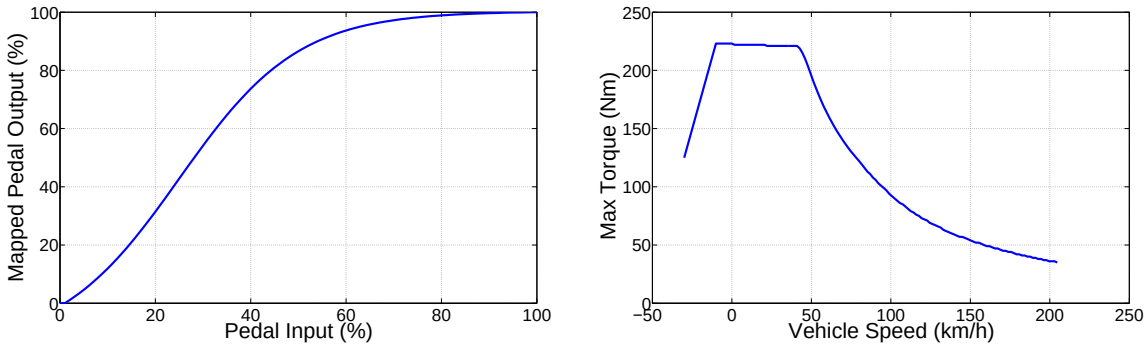


Figure 4.2: AWDC Logic Flow

A sigmoidal function is implemented, which ramps response slowly for low pedal inputs to make the car more controllable. Response improves in the middle of the range, and eventually tapers off for maximum acceleration. This is multiplied against each motor's maximum available torque, as shown in Figure 4.3b for the specific units used in the Equinox.

Provisions for maximum power draw are also included, which compares the requested power against a user-defined limit. In cases where this limit is exceeded, the maximum value is divided against the number of active actuators to achieve saturation. The first stage of fault tolerance also is taken into account by the AWDC, by zeroing torque requests to



(a) Pedal Mapping (b) Maximum Motor Torque Curve

Figure 4.3: AWDC Mappings

failed corners. The total torque request is divided against remaining actuators until motor limits are reached, to satisfy the requested longitudinal acceleration.

This feed-forward torque is then sent to the energy optimization routine covered in Section 2.5, which shifts the front-rear torque bias. Finally, the set of 4 torques are multiplied against direction bits based on the current gear selection before being sent out of the block.

BRFC (Brake Force Control

This block provides a feed-forward predictor for tire torques. It consists of 4 parallel branches representing each wheel, and sums motor torque feedback with a predicted brake torque, using predefined lookup tables as shown below in Figure 4.4.

In the Equinox, there are larger brake calipers in the front as opposed to rear. This is taken into account by using different tables for each axle, the values of which are shown below in Figure 4.5. Of note is the maximum value in the front, which is exceeds maximum motor torques at any speed; this is a good safety feature which guarantees the ability to stop in case of emergency.

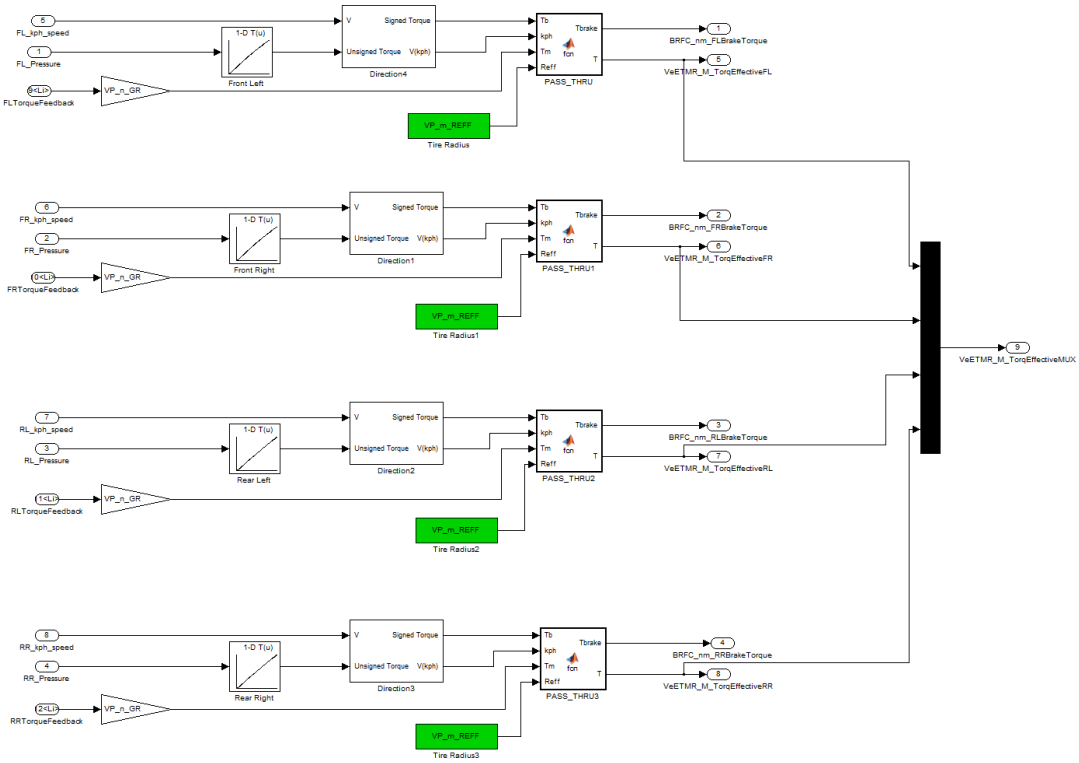


Figure 4.4: BRFC Block

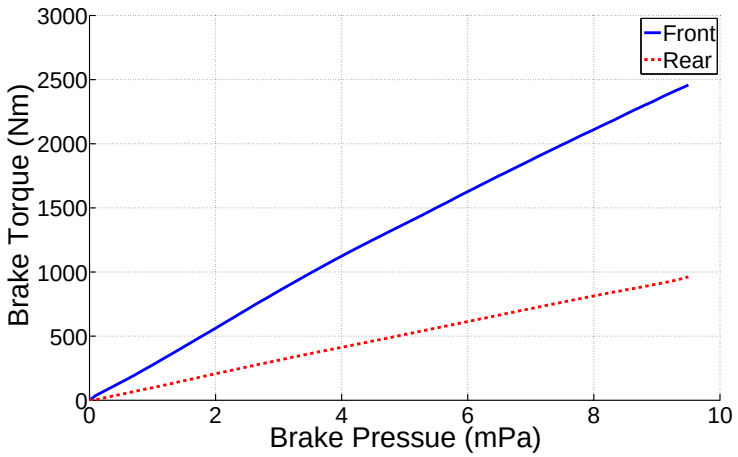


Figure 4.5: Brake Torques of Equinox Model

DCI (Driver Command Interpreter)

The DCI block consists of an embedded matlab function to represent the vehicle model described in 2.2, with external modifiers to mainly affect filtering of the feedback signals as shown below in Figure 4.6.

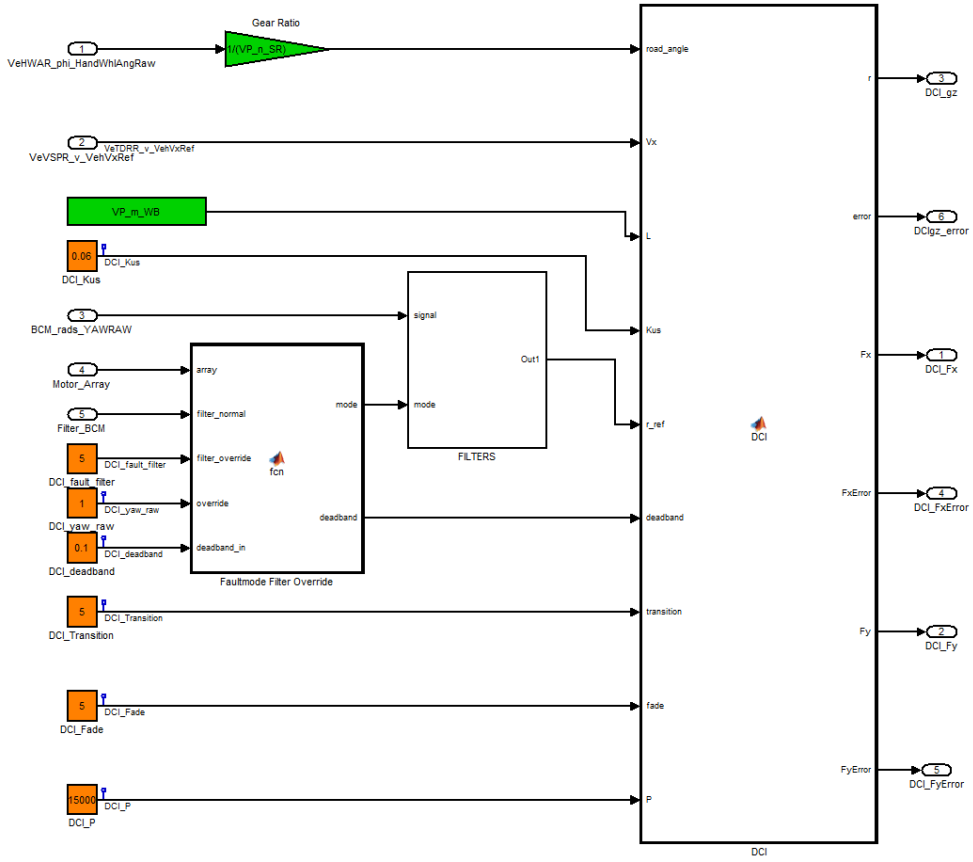


Figure 4.6: DCI block

In the "Faultmode Filter Override" function, the motor status array is monitored for signs of failure. Once detected, a pre-defined backup filter mode is requested, along with a new coefficient to nullify the deadband. The filter mode is passed onto the filter block to apply the corresponding signal, while the dead deadband coefficient is implemented in the main DCI block itself.

As output, the DCI generates values for the F_X, F_Y, and G_Z directions for reference,

as well as errors of all three for control purposes. As input, the orange values define user-selectable parameters such as understeer coefficient, filter modes,creep torque transition points and controller gains. The green blocks meanwhile, represent configuration parameters that are scripted and automatically loaded on a vehicle-specific basis.

Kinematic Estimation

The kinematic estimation blocks provide state feedback to the HCC, in the form of forces and velocities at each wheel corner. Since these are not directly measurable from the contact patch, estimation algorithms are used instead.

This function block is arranged in a cascade fashion, whereby each component is estimated individually and fed forward to generate other estimations. In total, five cascade layers exist, and executed in the order of F_Z, F_X, F_Y, V_X, V_Y.

Inputs to the system include the steering wheel input, IMU forces at the CG, wheel speeds, and torques applied to each corner. Details of the algorithm may be attained in [32] and [40]. In summary, a sprung mass model of the chassis is used to predict F_Z forces under roll and pitch conditions, while a dedicated wheel dynamics model estimates F_X using tire speeds and applied torque. These solved forces are then fed into another model, and an Unscented Kalman Filter is used to predict for F_Y. Finally, V_X is estimated based upon measured wheel speeds, and all the available states are then used to solve for the lateral velocity V_Y.

Tire Estimation

The tire estimation block's main purpose in the HCC is to provide the system with an estimate of road conditions. Of the most interest in literature is the LuGre tire friction model, which replaces the tire surface with a series of small brushes, each with a stiffness and coefficient of friction with the road [5]. As a brush encounters the ground, friction at its tip generates a force that deflects its shaft. Since there is a spring stiffness associated with this deflection, a reaction force is then created on the rim for propulsion. In the event that deflection exceeds available friction, the brush is modeled by a sliding friction force.

By explicitly modeling the rubber stiffness in addition to its coulomb and static friction with the ground, small elements can then be integrated over the contact surface to create

a final grip profile. The formulas governing a LuGre model are:

$$V_r = \omega r - v \tag{4.2}$$

$$F = (\sigma_0 z + \sigma_1 \dot{z} + \sigma_2 V_r) F_n \tag{4.3}$$

where F is the friction force, F_n the normal force, σ_0 the rubber stiffness, σ_1 the rubber longitudinal damping, σ_2 the viscous relative damping, and V_r the relative velocity. z is defined to be the deformation distance of the brush, the derivative of which is:

$$\dot{z} = V_r - \frac{\sigma_0 |V_r|}{g(V_r)} Z, \text{ where} \tag{4.4}$$

$$g(V_r) = \mu_c + (\mu_s - \mu_c) e^{-|V_r/V_s|^\alpha}. \tag{4.5}$$

Here, μ_c and μ_s are the Coulomb and Stribeck friction coefficients respectively, V_s the Stribeck velocity, and α a constant with which to specify the steady-state friction/slip characteristic [5]. Under non-ideal road conditions, a modifying multiplier may be applied to the equation. This takes the form :

$$\tilde{g}(V_r) = \theta g(V_r) \tag{4.6}$$

where θ represents a road classification factor. This is the value used in the HCC for calculating friction limits, and is akin to changing road friction coefficients.

Holistic Cornering Control

The physical coding of the HCC was designed to be as modular as possible, such that the code could be portable between different platforms, and easily interchanged to experiment with new algorithms.

Where possible, vehicle parameters such as mass, wheelbase, etc. are configured in library files and linked upon model initialization. Inputs are grouped by functional purpose into vectors, and passed to the HCC as a predefined MUX signal. A sample picture showing the HCC code layout is shown below in Figure 4.7.

In all cases, input to the system is laid out on the left hand side, where any intermediary signal processing is also performed. The HCC logic is located inside the blue block, which simply outputs a vector of 4 motor torques once computation is complete. Inside the HCC logic block are multiple iterations of the HCC, saved for experimental purposes. At the time of this writing, current research has 9 HCC variants, performing things such as Model

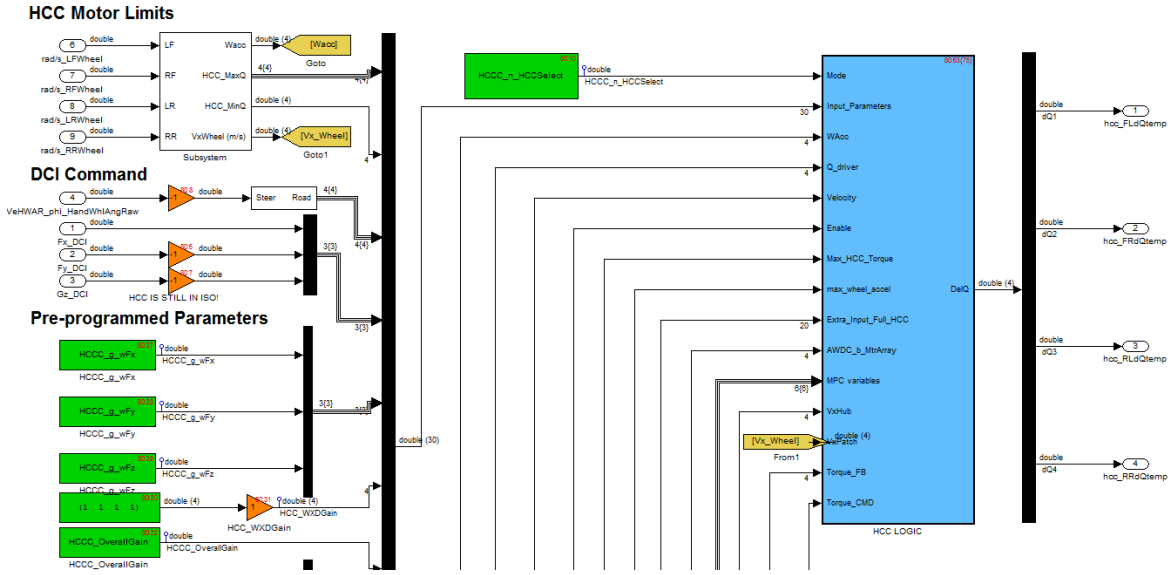


Figure 4.7: HCC Simulink Code

Predictive Control (MPC), simplified analytical solutions, or numerical solver approaches. To facilitate switching between these modes, a cascaded switch structure was implemented, whereby a mode selection number triggers a case activation block to route signals through the desired block as shown below in Figure 4.8.

For safety purposes, cut-offs were also implemented into the HCC. Specifically, a low-speed cutoff of 1 km/h was added to ensure that torque vectoring would not be activated when the vehicle is stopped or parked. Also, a dynamic saturation block was added to limit the amount of torque that any wheel could request, in case of computation errors during test trials. The layout of this block is shown below in Figure 4.9.

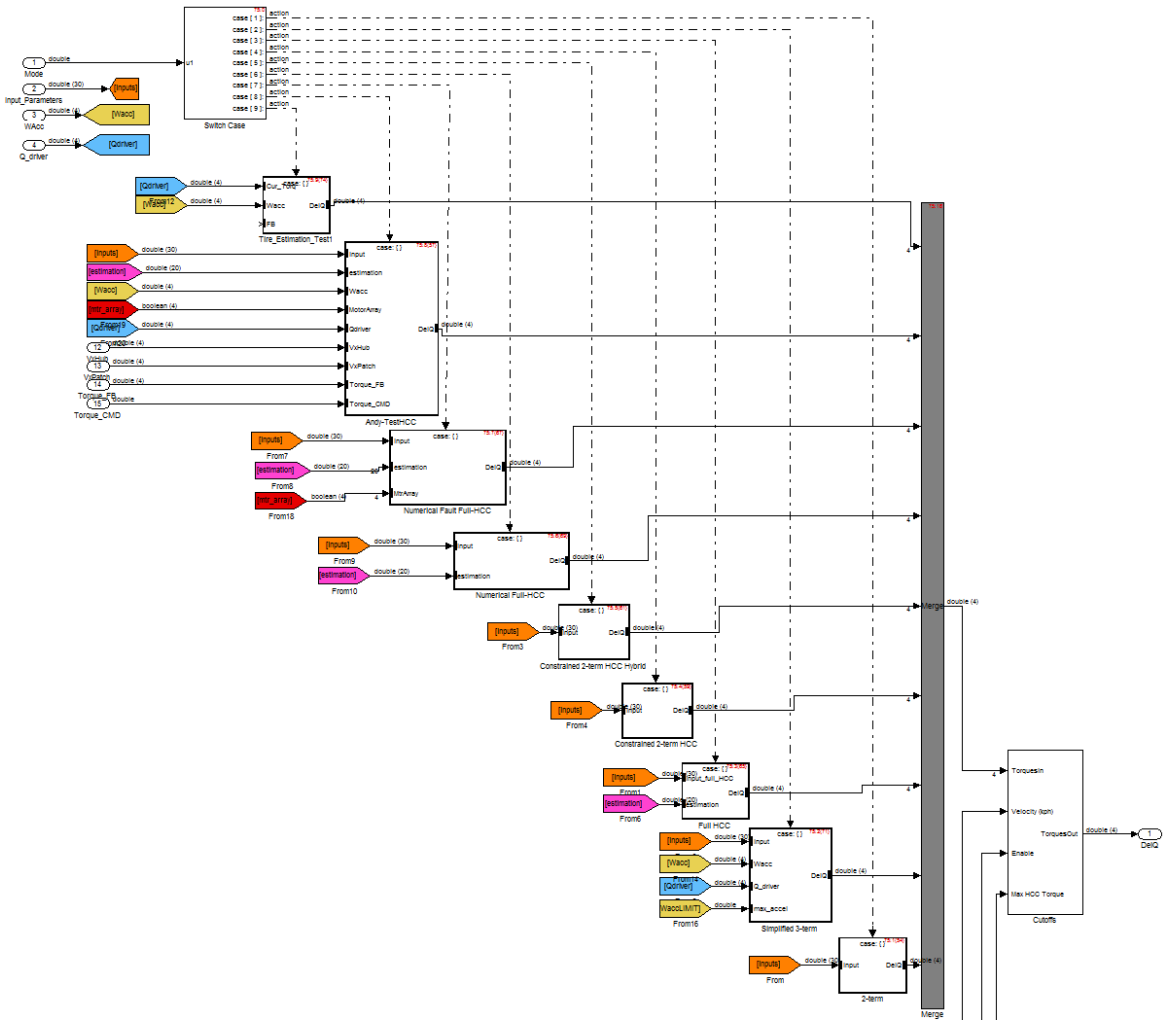


Figure 4.8: Multiple HCC Moding

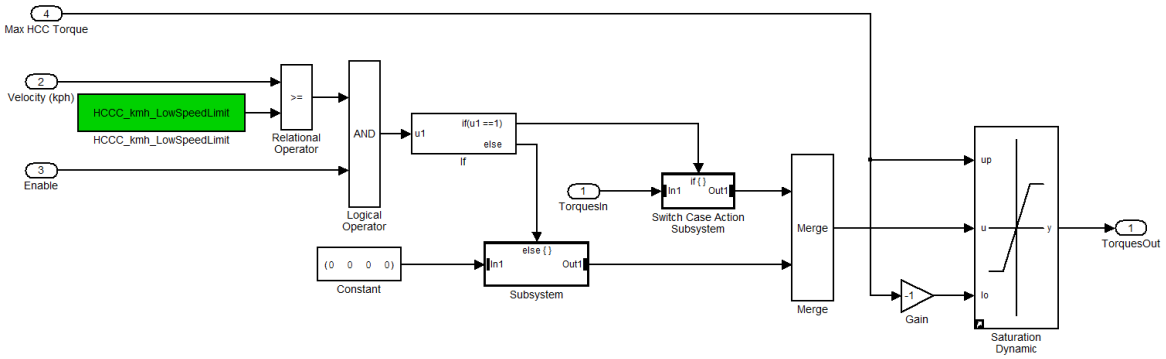


Figure 4.9: HCC Safety Cutoffs

Torque Blending Control

The Torque Blending Controller is the final stage before sending torque requests to the motor, and acts to merge the HCC overlay torques to original driver requests. Three major sets of outputs are generated, namely: 1)motor driver enable bits, 2) motor drive direction bits, and 3) requested motor torques as shown below in Figure 4.10. Inputs to the TBC include the HCC and AWDC torques, in addition to the current gear state for direction.

As a practical implementation, the velocity is also taken into account, whereby the motors are disabled when brakes are applied and the vehicle is stationary ($V_X = 0$). This was done to prevent parasitic energy loss, and the generation of excessive EMF noise of a stalled motor, which can cause CAN communication problems.

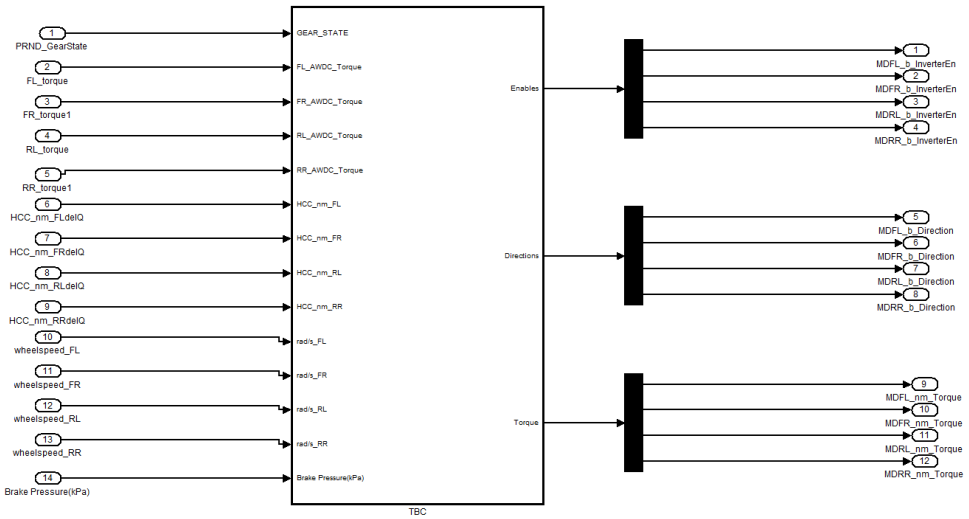


Figure 4.10: TBC block

Function Scheduler

The function scheduler's objective is to serially trigger data-dependent function blocks, such that processing progresses in an expected manner, and input-dependency errors are avoided. The refresh rate of the control loop is also set here, based on the timing of the trigger used. Currently, the function scheduler is set to trigger every 5ms, and the order of essential operations dictated by the function scheduler are as follows:

1. Vehicle Speed Determination
2. Accelerator Pedal Validation
3. AWDC (All-Wheel Drive Control)
4. BRFC (Brake Force Control)
5. DCI (Driver Command Interpreter)
6. Kinematic Estimation
7. Sensor Fault Tolerance
8. Tire Estimation
9. HCC (Holistic Cornering Control)
10. TBC (Torque Blending Control)
11. Motor Controller Send

It should be noted that other code is also executed even if not explicitly triggered by the function scheduler. Certain low-level control loops like the starting of a cooling pump can progress in the background without requiring real-time guarantees, and are thus free-spinning when computation cycles are available. In Simulink, a timed function-call generator block is used as the master, followed by function-call splits to execute subsequent trigger tags in linear sequence. In the current code, trigger tags are grouped into functional groups, as shown below in Figure 4.11.

Figure 4.11: Function Scheduler block

4.2 Carsim Environment

A software simulation environment was created for code debugging and verification, by linking the Matlab code to a CarSim model for dynamics modeling and feedback. Attributes under CarSim's control include a driver model with which to follow velocity-dependent drive cycles, road conditions, and all aspects of vehicle dynamics. By using the simulator to virtualize sensor feedback, identical variable names may be used and the control code may stay unchanged, thus becoming fully portable from the simulation to physical environment with minimal change.

As a brief overview, CarSim is a lookup table-based simulator, which extrapolates vehicle responses using empirical test data of individual components, and known relationships between systems. By avoiding finite-element analysis, CarSim achieves a near real-time simulation speed, which greatly aids debugging and controller tuning. Two sample screens are shown in Figures 4.12 and 4.13 below, which show the vehicle mass and tire modeling, respectively. Physical vehicle information was determined via weight scale and inertia swing testing by technicians at a GM vehicle testing center, while slip ratios with a resolution of 0.01 are derived from Calspan tire testing [7].

Conversion from MABX code to Carsim involves two main additional blocks: 1)virtual driver input, and 2) simulated sensor input into the control stack.

4.2.1 Driver Input

In the simulation environment, requested maneuvers from General Motors were converted into functional blocks of consisting of timed steer/gas/brake signals, and injected into the control code as if they were sensor inputs.

Multiple maneuvers are multiplexed into a switch, and selected manually via a numbered selection as shown below in Figure 4.14. A sample steering input from the specified double lane change maneuver meanwhile, is shown in Figure 4.15

There are cases where a simple feed-forward driver model is insufficient, such as when drive cycle simulations require tracking for target speeds over time. In these instances, a driver model is implemented to keep tests consistent between differing vehicle control algorithms. A PI controller on the CarSim side is then applied, configured to request acceleration as a function of velocity tracking error. An additional proportional gain was

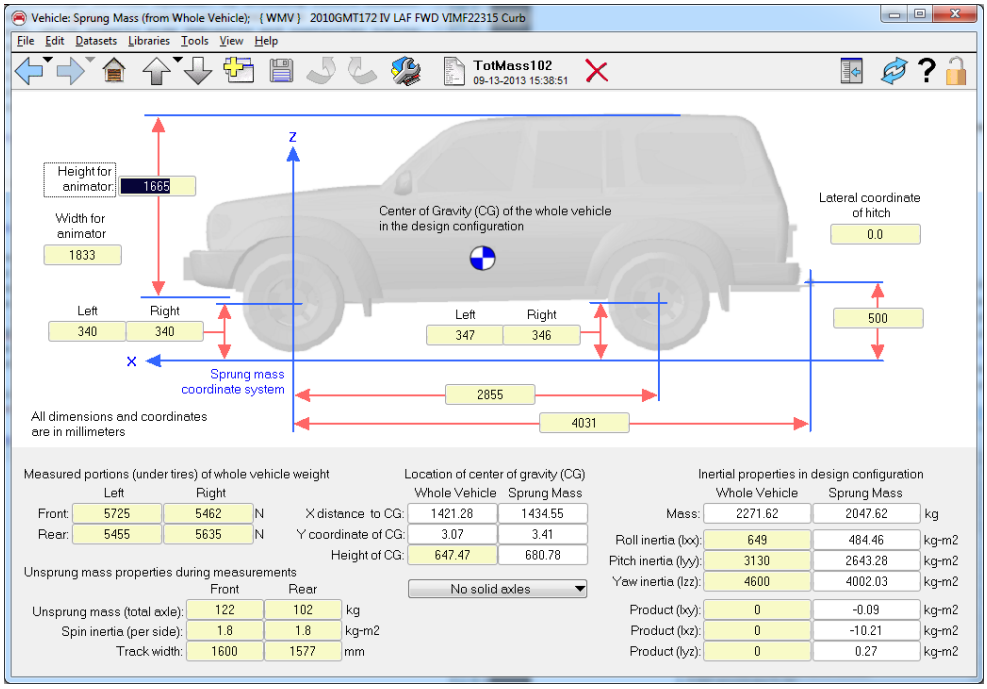


Figure 4.12: Carsim Vehicle Model

added to translate this desired acceleration into pedal responses as follows:

$$P_{gas} = \begin{cases} K_g(K_p * V_{err} + K_i * I_{err}), & \text{if } V_{err} \geq 0 \\ 0, & \text{if } V_{err} < 0 \end{cases}$$

$$P_{brake} = \begin{cases} K_b(K_p * V_{err} + K_i * I_{err}), & \text{if } V_{err} < 0 \\ 0, & \text{if } V_{err} \geq 0 \end{cases}$$

Here, P_{gas} and P_{brake} are the gas and brake pedal positions in percent, K_g and K_b are the gain constants for the gas and brake pedals respectively, K_p is the acceleration P gain, and K_i the integral gain. V_{err} is the measured velocity tracking error, and I_{err} is the integral of V_{err} over time.

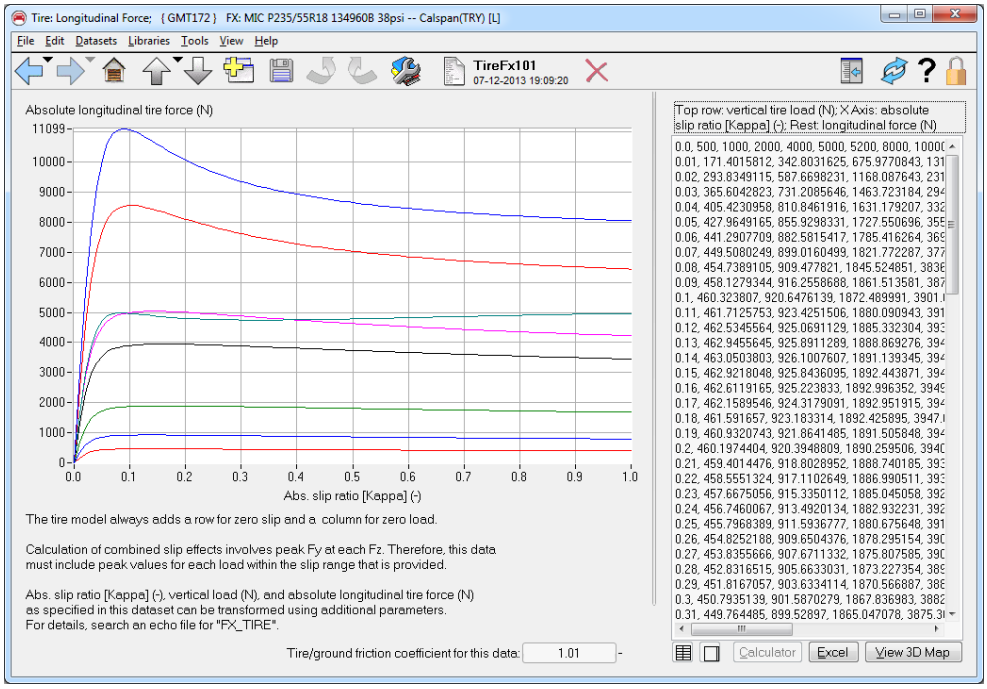


Figure 4.13: Carsim Tire Model

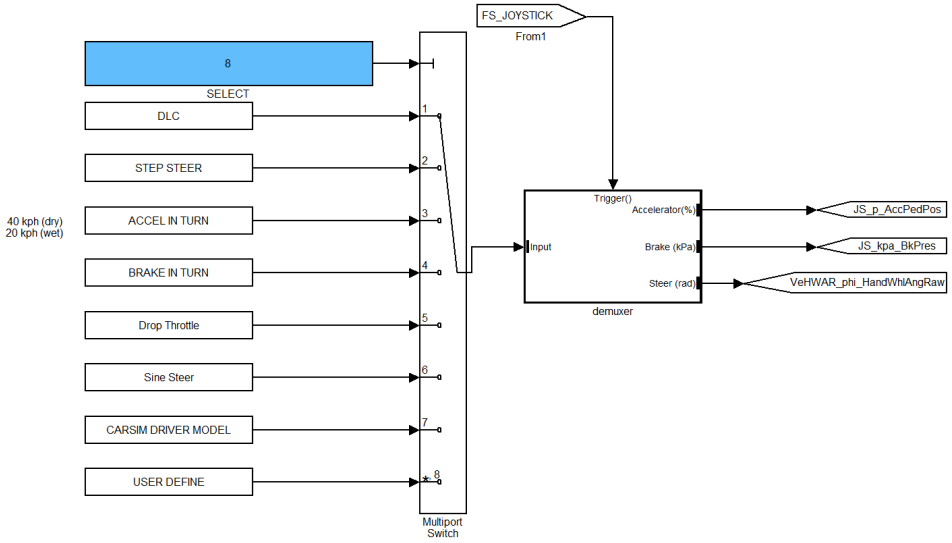


Figure 4.14: Simulated Driver Manoeuvring

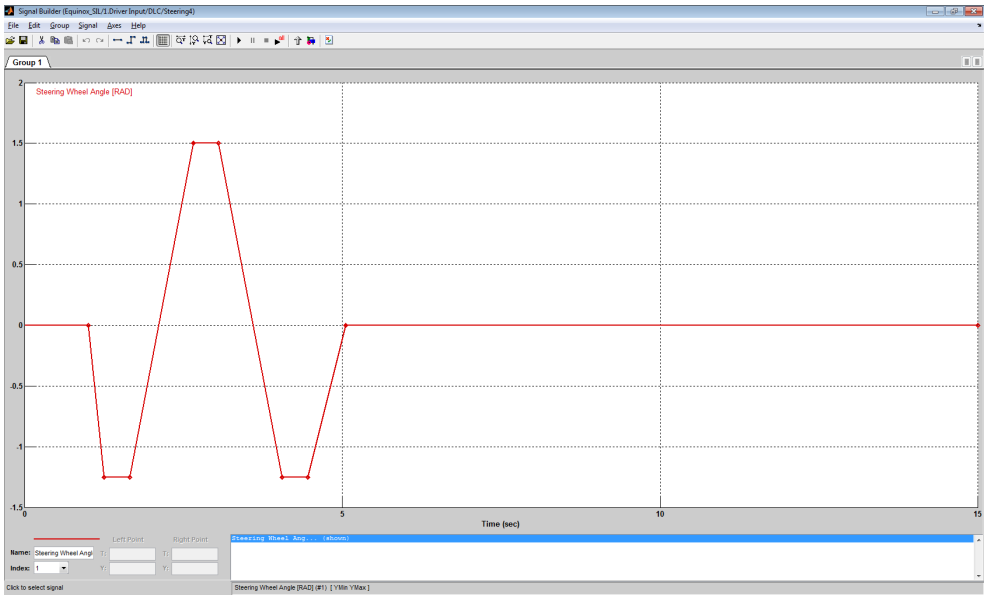


Figure 4.15: Double Lane Change Simulator Input

4.2.2 Simulated Sensor Input

At the code level, CarSim is linked to Simulink via a special s-function block, which performs bi-directional transfer as shown in Figure 4.16. Currently, 10 inputs are fed into Carsim to drive the dynamic model: these involve the driving and braking torques at each wheel, in addition to steering wheel angles and steering torque. 79 outputs are in turn taken from Carsim to attain validation signals and drive the controller code. Primarily, these signals involve simulated velocities, tire forces, CG forces, slip angles, and input requests from the CarSim driver model. The interface is driven by two function-call triggers, which times the reading of vehicle sensor signals at the start of each cycle, and to output requested motor torques at the end.

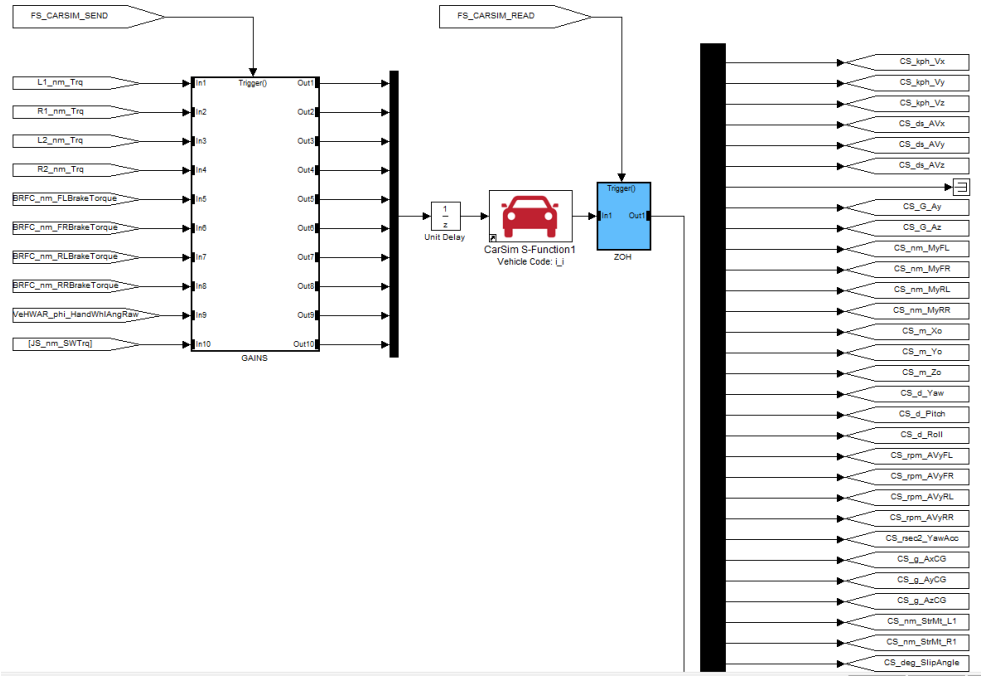


Figure 4.16: Carsim Interface

After attaining raw carsim sensor data, a secondary block gathers all of the interface signals, and converts the coordinate frames or units to match that of the production vehicle sensors as needed. At the same time, filtering or noise generation is added as desired. As output, the tags are renamed to be identical to those coming out of the actual sensors as shown below in Figure 4.17.

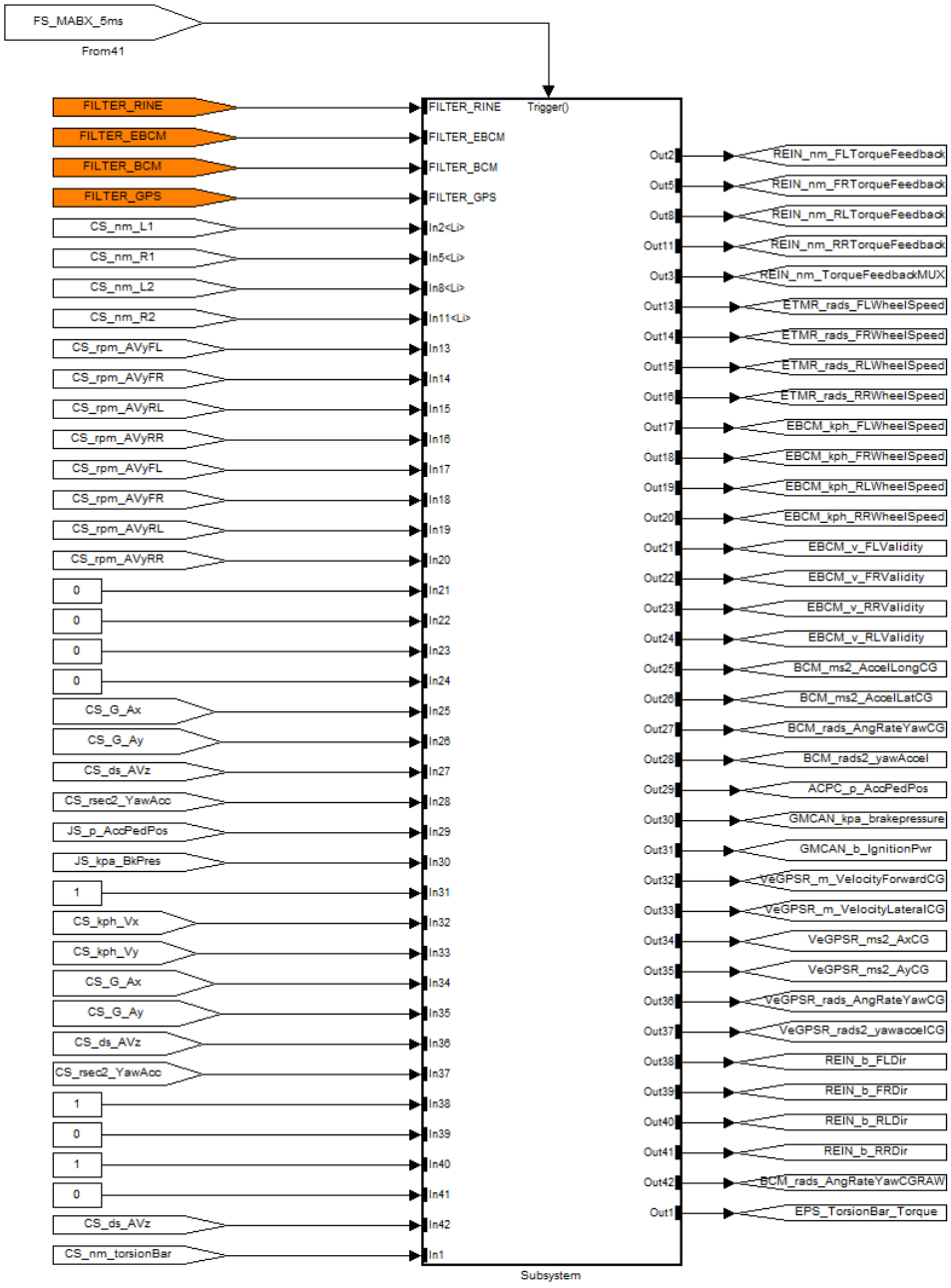


Figure 4.17: MABX Input Simulation

As an example, the inertial measurement signals being internally routed for the BCM are shown below in Figure 4.18. Conversions between $\frac{m}{s^2}$ and G units are shown, as well inversions to some axes to satisfy the SAE standard. At the end, zero-order hold blocks latch the signal values until the next iteration, to keep data consistency within a refresh cycle.

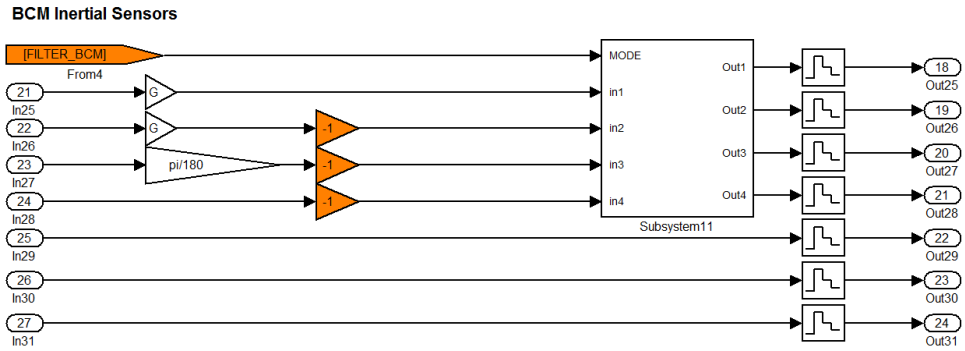


Figure 4.18: BCM Signal Simulation

A tag labeled FILTER_BCM is seen, which is a user-configurable number which routes the signals through one of multiple processing filters, as shown in Figure 4.19.

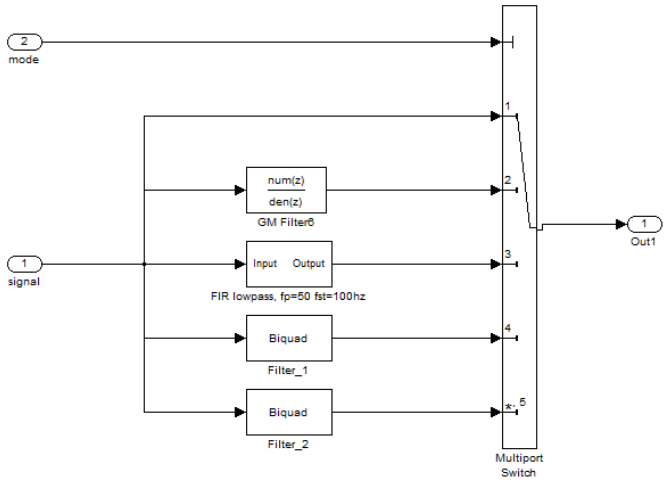


Figure 4.19: Simulated Signal Filtering

4.3 dSpace

By setting the dSpace as a target for compiled code, Matlab is capable of cross-compiling the simulink software into C code, to be run by the dSpace embedded computer. To accomplish this, dSpace-provided RTI blocks are used whenever interfaces to the CANbus are needed. The rest is an automated process, where Matlab accesses the required libraries.

CAN Configuration

Here, Real-Time Interface (RTI-CAN) blocks from the dSpace model set are used to configure each hardware channel, and to assign CAN libraries which define the layout of expected messages on the bus. Three CAN buses exist in the Equinox platform, and their primary purpose is shown below in Figure 4.20.

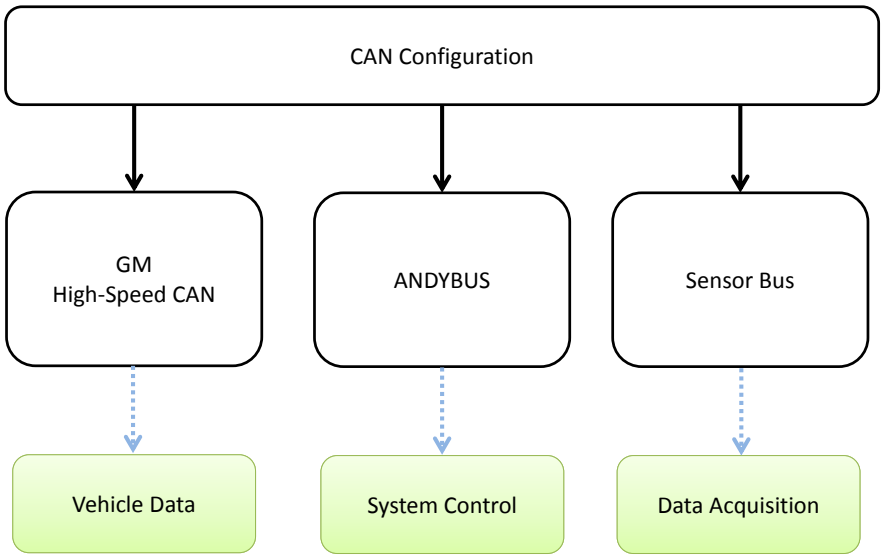


Figure 4.20: CAN Channel Functionality

All three buses are tied together at the dSpace, which coordinates I/O between the devices, and uses information to perform HCC calculations.

The High-Speed CAN is a production vehicle network, and part of GM's Global-A architecture, which standardized the way vehicles communicate throughout different models and platforms. A GM-supplied Global-A CAN database was used to access the vast amount of information located on this bus. Of specific relevance to the HCC are signals involving the Body Control Module (BCM) which provides vehicle IMU data, the Electronic Brake Control Module (EBCM) which provides the brake pressure and wheel speed data, the Electronic Power Steering (EPS) module which provides steering angle and torque information, and the dashboard which outputs indicators for speed and other notifications.

The Andybus, meanwhile is a custom network, meant to control retrofit components such as the DC-DC converter, BMS system, and motor controllers. This was created, since the production GM bus was already at a high utilization rate, and adding throughput involved the risk of data collision, and higher error rates.

The Sensor Bus is second auxiliary channel, meant to transmit high-bandwidth data without over-utilizing either of the primary control buses. This bus carries information from the GPS/IMU, laser speed sensor, and wheel load sensors, which output data at a constant and high rate.

In CAN messaging, each bus is capable of sending messages of a unique ID, and varying length to convey information. Each message is a grouping of related signals, which can be de-multiplexed in Simulink.

An example is given below in Figure 4.21 below, which defines one of the messages for the Rinehart AC Inverter. In messages, the packing order of signals are defined on a per-bit basis, and grouped in the example by color.

In Simulink, the CAN dictionary for this message is called every time a message ID of 0X1AA is encountered, and its signals are parsed to produce the motor driver's statuses as shown in Figure 4.22. In this example, three desirable signals involving the system state, motor direction, enable bit are selected.

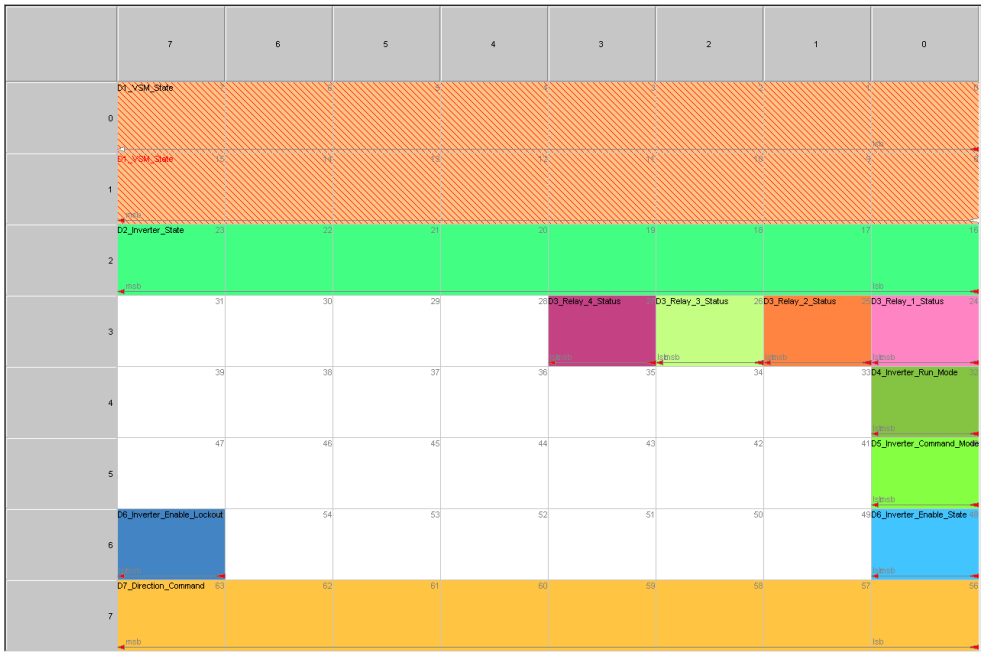


Figure 4.21: Sample CAN Message Layout

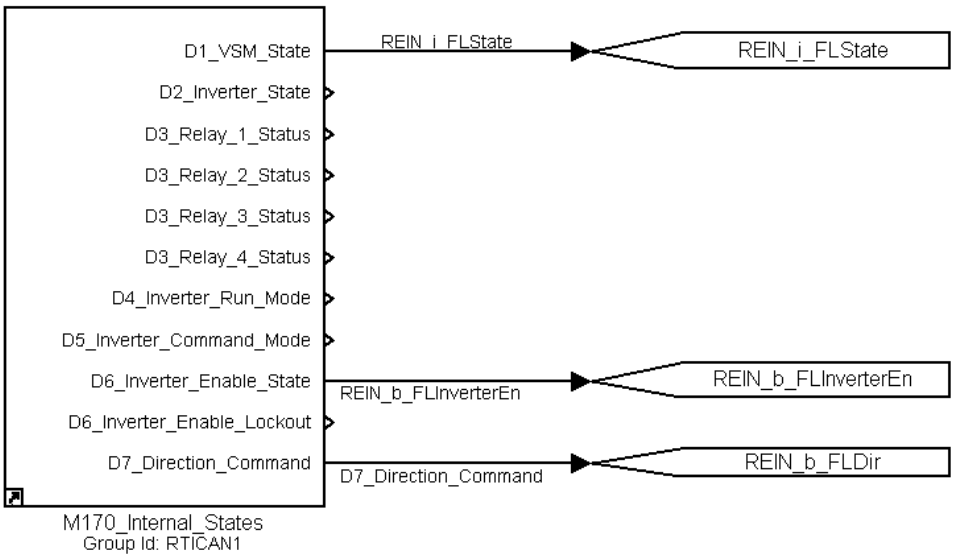


Figure 4.22: Simulink CAN interface

MABX Inputs

Here, MABX is an acronym for "Micro-Autobox", and defines the various input signals which are required for controller operation. For network-based data, CAN libraries described in Section 4.3 are used to define and route incoming signals. Otherwise, analog signals are accessed through the hardware's on-board ADCs. This module is organized into 8 function blocks, logically separated by the source of these signals as shown below in Tables 4.1 and 4.2.

Module	Signal
BCM (Body Control Module)	
	Steering & Gas Pedal Positions)
	IMU Data (CG forces and Yaw Rates)
	Instantaneous Brake Pressure
	Ignition Switch Status
	Miscellaneous System Statuses
BMS (Battery Management System)	
	Battery State of Charge
	Current Draw
	Total Energy In/Out
	Pack Temperature
	BMS temperature
	Highest/Lowest Cell Voltage
	Location of Lowest Cell
	Total Pack Voltage
EPS (Electronic Power Steering Controller)	
	Boost Motor Speed
	Boost Motor Torque
	Torsion Bar Torque
	Steering Angle
	Current Draw
	System Voltage
EBCM (Electronic Brake Control Module)	
	Wheel Speeds
	Sensor Validity

Table 4.1: MABX Inputs from GM-HSCAN

Module	Signal
DC-DC Converter	
	Device Status
	Temperature
GPS/IMU	
	CG Velocity Components
	6-Axis IMU Signals
	GPS Coordinate data
ADC	
	Gear Selector Position
	Accelerator Pedal Position
Motor Controllers	
	Inverter Status
	Temperature
	Torque Feedback
	Input Voltage

Table 4.2: MABX Input Signals for Andybus

The BCM returns a large quantity of chassis-level information, which are not all used. Instead, a total of 18 signals are selected to satisfy the functioning of the HCC. BMS signals are checked mostly for safety purposes. Eventually, many of these statuses are output to a debugging screen to be checked during physical tests, to ensure that the battery does not sustain damage from over-current draws, or excessive SOC depletion. Individual cell voltages, and location numbers are recorded so that imminent failures, or sporadic faults under load could be discovered and resolved before catastrophic cell damage.

The DC-DC converter is a fairly isolated component within the system, and is only monitored for temperature warnings, and activity status to ensure that the low-voltage bus is powered.

The EPS module provides debug information directly from the power steering module, which provides enhanced detailed insight into the front wheel activity. Ultimately, the steering angles and current draw may be used to design controllers for autonomous steering, while the torsion bar torque may be used to deduce road friction conditions.

The Brake controller is used primarily for its wheel-speed encoder, which is used in the Equinox for vehicle velocity estimation, and tire slip control. The signals harvested from the third-party GPS/IMU unit are used mainly for validation purposes, and include a 6-axis accelerometer output in addition to velocity and position readings.

Only three signals are polled through the on-board ADC. The signals are the gear selector position (due to removal of the transmission controller), and two redundant gas pedal position sensors (due to removal of the engine controller). The motor controllers are polled, primarily for validation and safety purposes. Inverter state signals are monitored for fault states, while temperature signals are used to trigger cooling pumps as needed.

Initialization Sequence

One feature of the Global-A from GM's perspective is security; in order to discourage vehicle theft and the sale of stolen parts on illegal markets, the main BCM checks an ID and initialization string of each other controller within its scope. This way, if a component from another vehicle is installed, the vehicle would purposefully shut down. Message IDs are not public information, which further masks the purpose of each signal from potential hackers. This, however presents a problem for an electric retrofit, since the ECM required for ignition would be missing. Even if the ECM were present, the engine and its associated sensors would be also be missing.

To get around this issue, all messages across the CAN network were recorded during ignition, and analyzed using a CAN database to determine dependent signals, modules and timing. This reverse-engineering led to the discovery of critical messages, which could then be spoofed by the dSpace to report status to the security modules. Specifically, these involved links from the ECM and transmission controller.

Critical messages that the BCM expects from the ECM are status updates regarding the engine speed, temperature, fuel level, ignition status, target idle speeds, oil pressure, and various validation signals for engine health. Meanwhile, spoofed transmission controller messages tell the BCM about the current gear selector position, gear number, and engine state.

The timing of these signals are an issue, as there is a time windows with which to complete ignition. Otherwise, modules such as the EPS would not enable, and power steering would be unavailable. To time the sequence of events, the vehicle's wake-up pin is wired into the dSpace, which triggers its wake-up signal. When the key is turned, or the door is opened, the dSpace immediately comes alive and synchronizes to the BCM which wakes up via the same signal.

In the Equinox, a timed sequence of signals mimicking the production network is sent out first. Then, custom devices such as cooling pumps and relays are engaged. Finally, high-voltage relays are engaged, and the motor controllers are powered on for operation.

Low Level Hardware

The remaining items under dSpace control are triggered based on simple outputs.

For instance, the stock Dashboard in the Equinox is driven by wire, and displays an output based on certain CAN messages. This was used to show custom information via remapping. Pack SOC was rerouted to the fuel gauge, motor temperatures were rerouted to the engine temperature, and current drain was routed to the engine RPM meter. Hardware DACs are triggered based on system temperature thresholds; once reached, the code commands driver boards to trigger relays for pumps and fans for cooling.

The remainder of dSpace programming involves graphically drawing Human Machine Interfaces (HMI) for real-time plotting of data and parameter tuning, as shown below in Figures 4.23 and 4.24.

Figure 4.23: Main Demonstration Screen

During testing, HCC modes are selectable via pull-down menus, and specific functions such as simulated motor faults are clickable via buttons. On the tuning screen, parameters affecting the DCI driver feel, HCC gains, and traction control are all available. Real-time motor feedback and HCC outputs are plotted, for visual inspection.

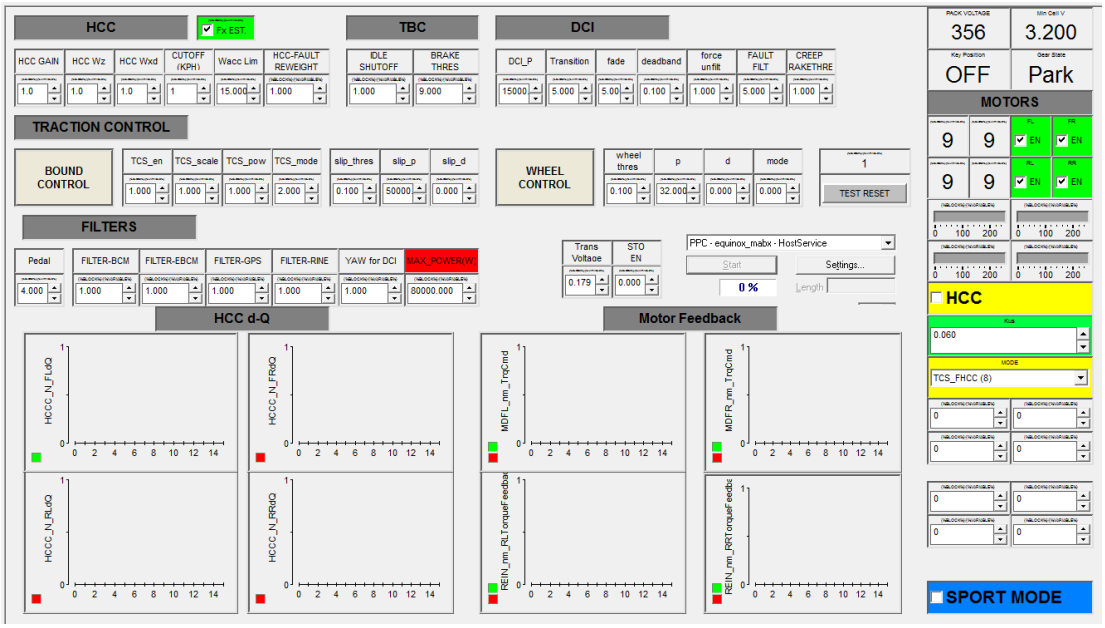


Figure 4.24: Tuning Screen

Chapter 5

Results

The HCC formulation was implemented on a variety of platforms for validation, but the results contained in this chapter are for the Equinox test beds. Simulation data is performed using real-time Simulink code connected to CarSim, while experimental data was acquired from test runs at a local test track designed to train emergency service (police, firetruck) drivers. This test track is an oval shape of relatively narrow width as shown in Figure 5.1a, which generally restricted testing to lane-change and start-stop scenarios. An example of one such maneuver is shown in Figure 5.1b, where the Equinox is undergoing double lane change testing on a course laid out by pylons.

For testing, production sensors are used for steering wheel angles, IMU, and wheel speeds. Stock Michelin Latitude tires were utilized, of P225/60R17 size. In the provided results, the HCC code was implemented in its numerical solver form, with data logged via the embedded dSpace computer.

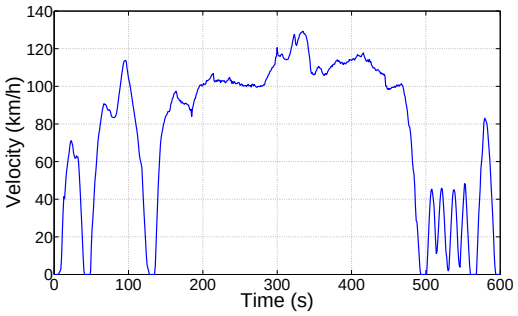
(a) Test Track Layout

(b) Equinox Testing in Action

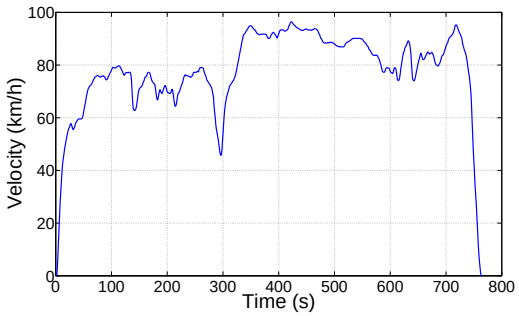
Figure 5.1: Vehicle Testing at UW Fire Research Facility

5.1 Energy Consumption

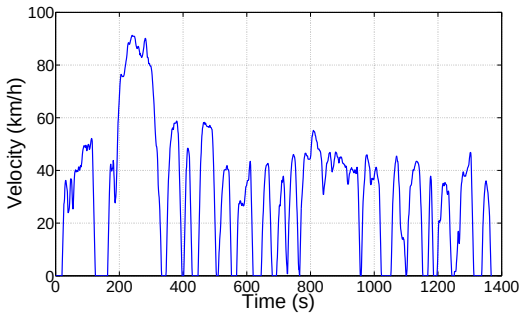
Drive-cycle analyses are performed to determine the total energy consumption over varying, real-life driving conditions. Industry standard EPA drive cycles were used to simulate driver behavior, specifically the aggressive, highway, and urban profiles [14]. The velocity profiles of these are shown in Figure 5.2.



(a) EPA Aggressive Cycle



(b) EPA HWYFET Cycle



(c) EPA urban Cycle

Figure 5.2: EPA Drive Cycles

Two performance metrics were used to determine the effectiveness of the EMS:

1. Time time-average efficiency of the vehicle as shown in Table 5.1
2. Total energy consumption of the drive cycle as shown in Table 5.2

From Table 5.1, it is seen that the EMS is most effective for cases where there is a constant, lower demand for torque such as highway cruising. In such cases, the operating

Table 5.1: Average Motor Efficiency

Cycle	EMS OFF (%)	EMS ON (%)	Improvement (%)
Urban	81.07	90.50	9.43
Hwyfet	57.45	84.75	27.30
Aggressive	73.02	88.60	15.58

Table 5.2: Power Consumption

Cycle	EMS OFF (kWh)	EMS ON (kWh)	Improvement (%)
Urban	36.26	34.29	5.43
Hwyfet	16.29	15.47	5.03
Aggressive	19.49	18.11	7.08

point is closer to the lower fringes of the efficiency curve and has the most to benefit from an upward shift to a single axle.

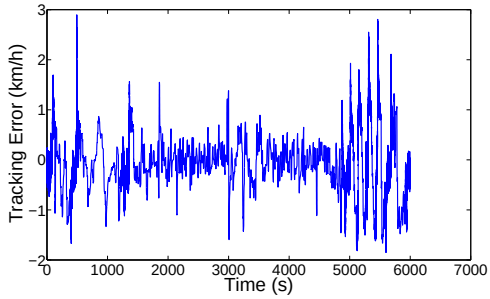
In Table 5.2, it is seen that the order of energy consumption rankings is inverted. The EMS is most effective for aggressive driving which contains high speeds and fast accelerations for this particular motor. In essence, the power consumed during large accelerations exceed that consumed for constant-speed driving and provides the most opportunity for net gains.

To ensure the validity of these results, the driver model was also analyzed to ensure that velocities were being reliably tracked, To accomplish this, the velocity tracking errors were recorded over time, and the RMS values calculated to give an indication of precision. The results are given below in Table 5.3, and visualized in Figure 5.3.

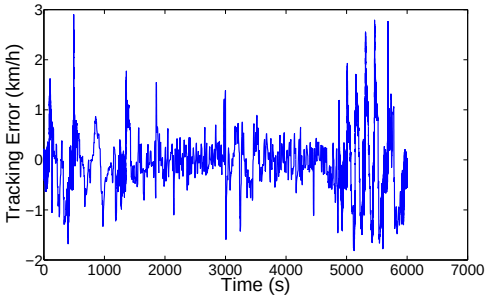
Table 5.3: Driver Model Tracking Error Rates

Cycle	RMS, EMS OFF (kph)	RMS, EMS ON (kph)
Urban	0.368	0.366
HWYFET	0.169	0.171
Aggressive	0.555	0.556

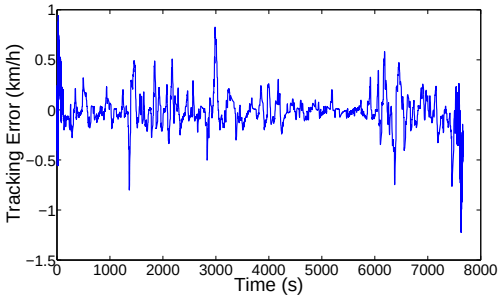
Since all simulations indicated an RMS tracking error of under 1 kph, it was deemed acceptable as the vehicle consistently achieved target speeds.



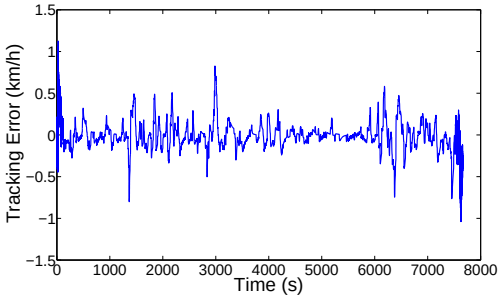
(a) EPA Aggressive Cycle, EMS ON



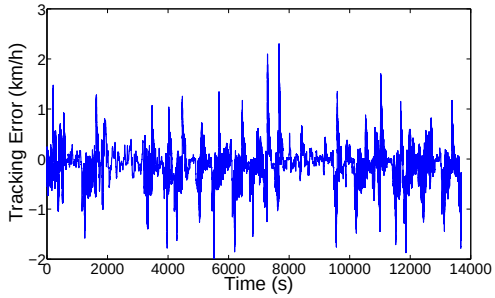
(b) EPA Aggressive Cycle, EMS OFF



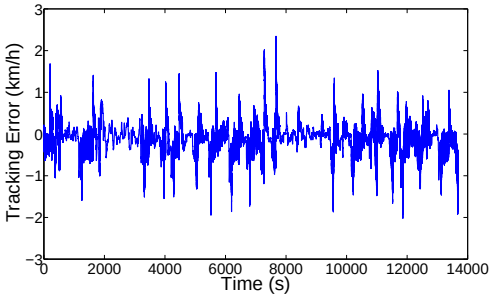
(c) EPA HWYFET Cycle, EMS ON



(d) EPA HWYFET Cycle, EMS OFF



(e) EPA Urban Cycle, EMS ON

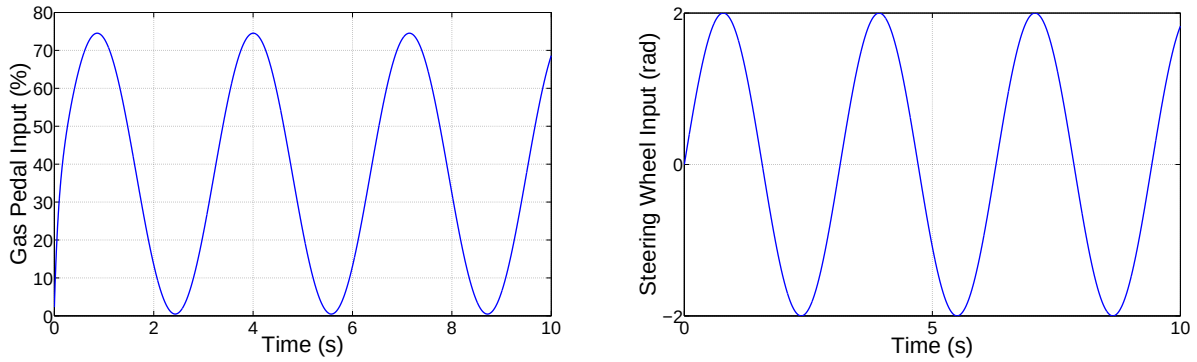


(f) EPA Urban Cycle, EMS OFF

Figure 5.3: Vehicle Tracking Results

5.2 Energy Management Under Dynamic Maneuvers

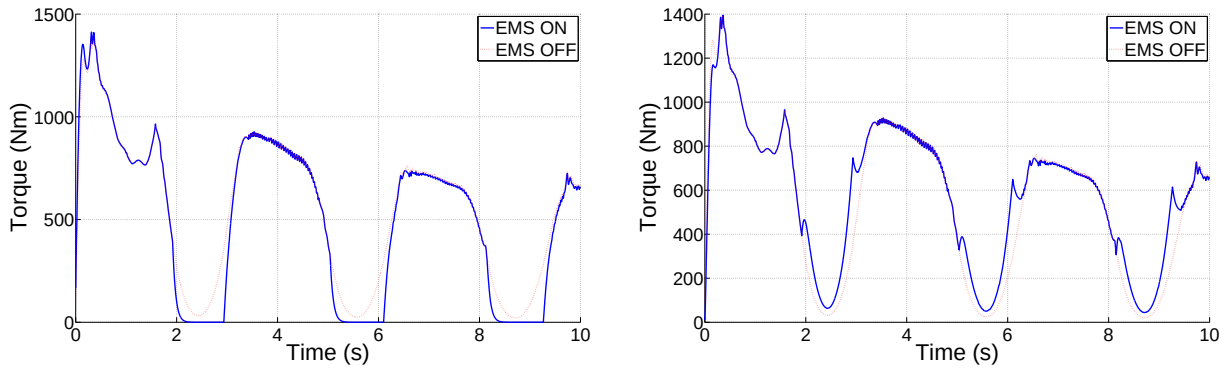
To study the energy management system (EMS)'s possible effects on dynamic response, a sample case is given for a sinusoidal gas pedal and steering wheel input as shown below in Figure 5.4. This input was chosen, such that a range of accelerations are requested while the vehicle undergoes significant dynamic loading. An initial velocity of 30 kph was used.



(a) Pedal Input (b) Steering Input

Figure 5.4: Driver Inputs for Simulated Run

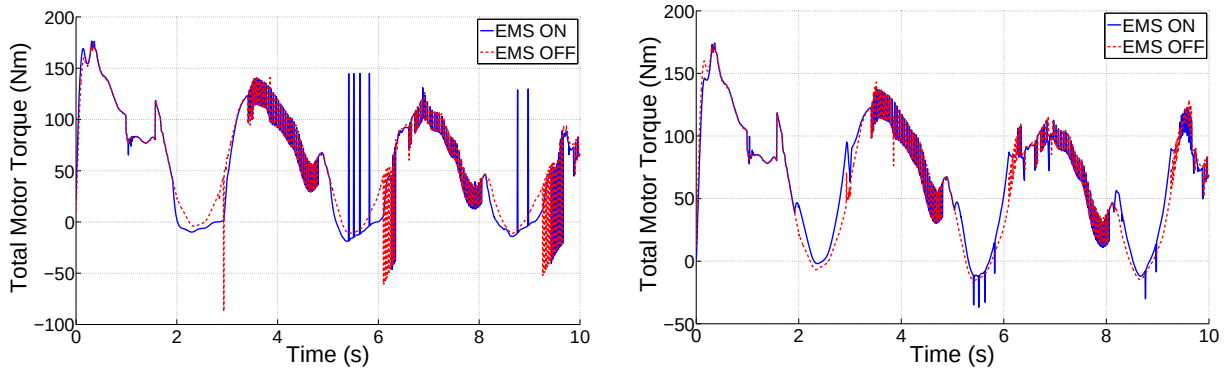
Figure 5.5 below shows the feed-forward torques with and without the EMS, for the left side of the vehicle. Deviations between the two lines are noted, and due to the active redistribution between the front and rear axles.



(a) Motor Torque, LF (b) Motor Torque, LR

Figure 5.5: Comparison of Torques With and Without EMS

After these feed-forward torques are summed with the HCC torque vectoring oututs, total requested motor torques are attained and shown below in Figure 5.6.



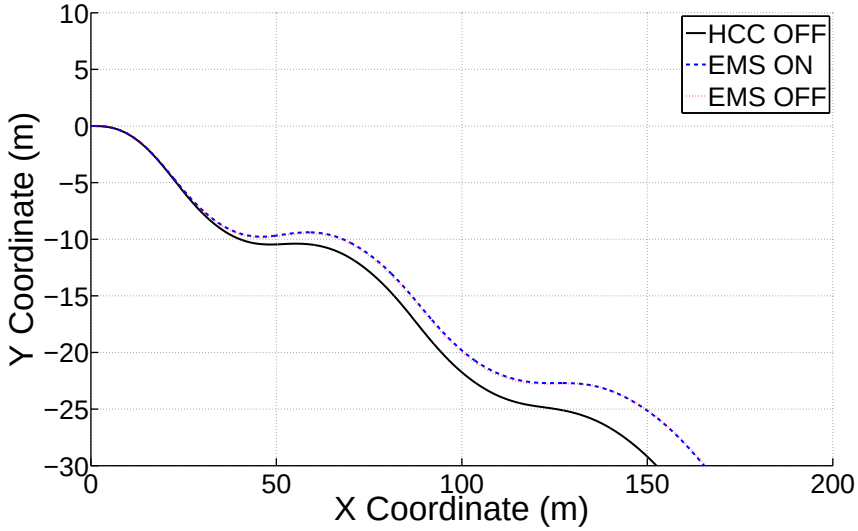
(a) Motor Torque, LF

(b) Motor Torque, LR

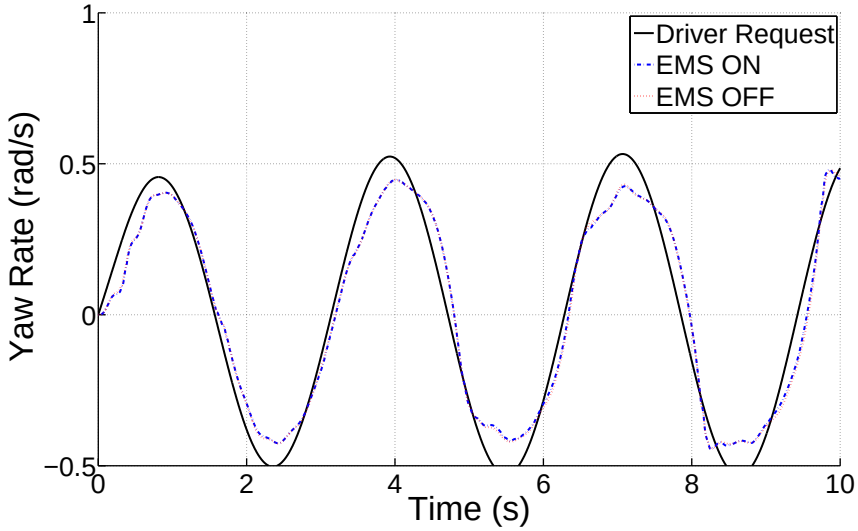
Figure 5.6: Comparison of Torques With and Without EMS

It is seen that there are differences between the EMS off modes, versus on. The high frequency switching areas are regions of active traction control, which indicates that the vehicle has reached handling limits. The jagged nature of the torque curves meanwhile, are due to the HCC overlay torques which are actively maintaining vehicle yaw tracking. It should be noted that the torque vectoring overlay is nearly identical between the two curves, confirming that yaw tracking outputs from the HCC are unaffected by the baseline torque.

Finally, Figure 5.7 below displays the results of the vehicle's path tracking and yaw rate tracking abilities. It is seen that without any active control, the vehicle takes on a deviant path. With the HCC enabled however, the vehicle successfully tracks the DCI generated reference, without interference from the EMS. It is seen that the path and yaw rate tracking with the EMS enabled is virtually identical to the cases where it is disabled, which confirms that no detrimental effects are seen while conserving energy.



(a) Vehicle Path



(b) Yaw Rate

Figure 5.7: Path and Yaw Rate Tracking Comparison With and Without EMS

5.3 Straight Line Acceleration

As explained in Section 2.4.4, some form of traction control is necessary for the HCC to perform stably in the non-linear regions of the tire friction curve.

After empirical tuning, acceptable parameters for the slip-prevention controllers were derived. Namely, a threshold slip ratio of 0.2 at each wheel was used to trigger a proportional controller with $p = 50,000$ which constricted the friction ellipse. In both simulations and empirical testing, this was found to provide satisfactory performance under both wet and dry asphalt conditions.

A simulated case is given below for straight-line acceleration from 40 kph on a slippery surface of $\mu = 0.6$, with an accelerator pulse input as shown below in Figure 5.8. Tire slips with the traction controller turned on are provided in Figure 5.9 below.

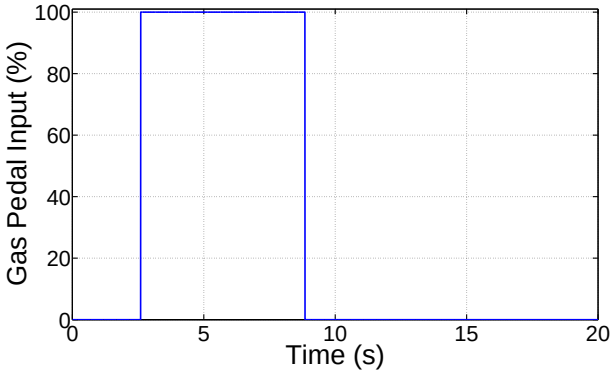
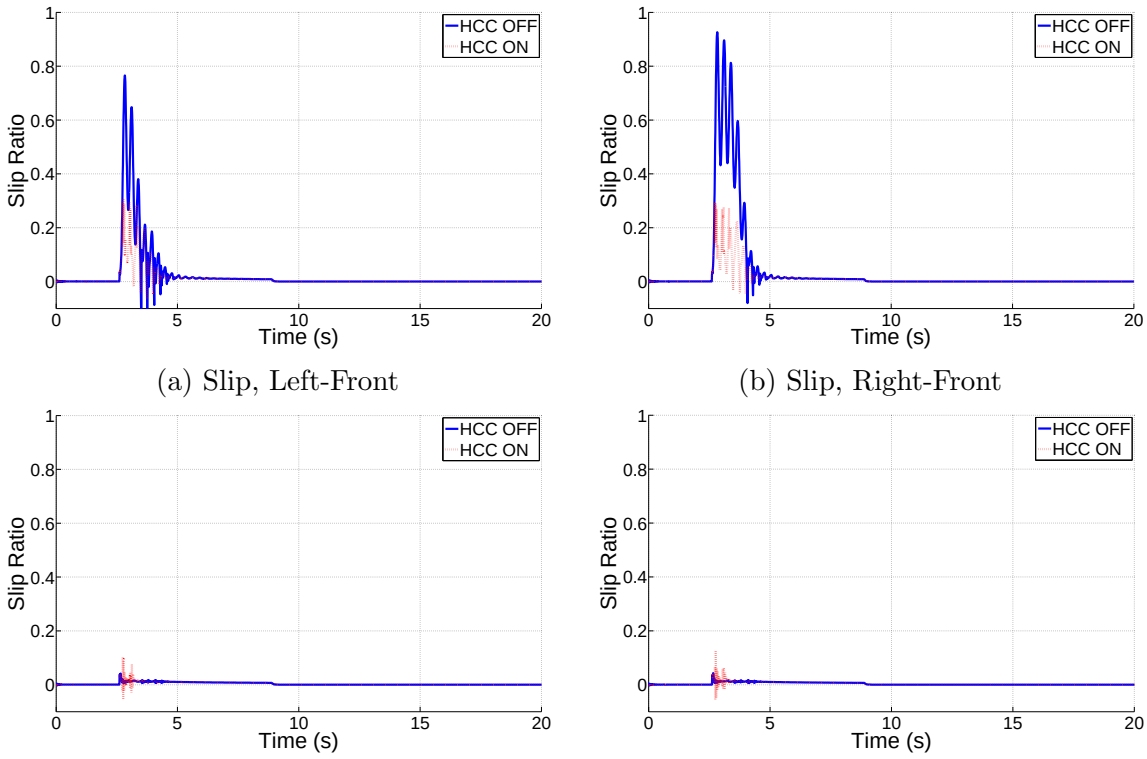


Figure 5.8: Pedal Input for Straight-Line Acceleration Test

In this road condition, it is seen that the front wheels exhibit large slip ratios of up to 90%. There is little slip in the rear, as the coefficient of friction is sufficient to carry the torque given an extra boost to normal forces under accelerative weight shifting to the rear. Slip with the HCC turned on are significantly better, and are held to the 0.2 slip threshold ratio mark. Figure 5.10 shows the final motor torque requests, with HCC augmented torques overlaid upon the original driver requests.

In Figure 5.11, the tire force bounds for the HCC numerical solver are shown. At approximately the 2.5s mark, it is seen that the upper and lower bounds converge at a negative value due to positive tire slip, and forces the system to apply a negative corrective torque. After the slip event is over, the bounds are seen to re-diverge, and give a range of possible values for the HCC to resume torque vectoring. Between the 2 to 8 second mark,



(a) Slip, Left-Front

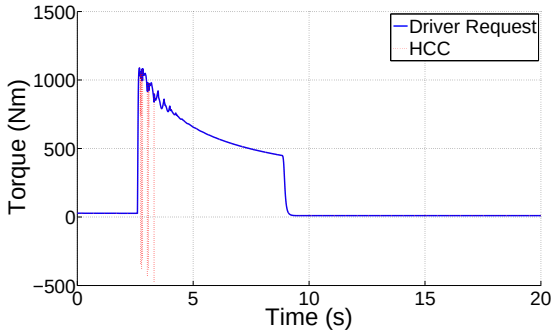
(b) Slip, Right-Front

(c) Slip, Left-Rear

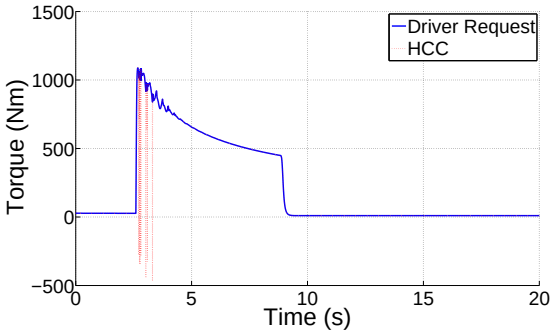
(d) Slip, Right-Rear

Figure 5.9: Slip Ratios, Traction Control On

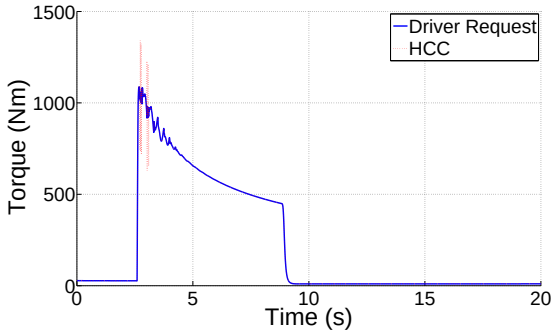
converging slopes can be observed, particularly in the rear wheels. In cases where traction is not the limiting factor, available motor torque becomes the constraining factor. In this experiment's case, vehicle speeds have passed into the motor's constant power/diminishing torque region, thus explaining the contraction.



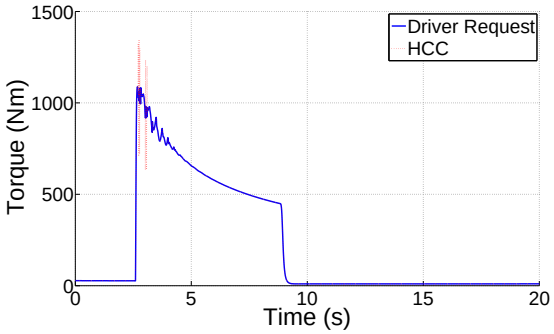
(a) HCC Torque, Left-Front



(b) HCC Torque, Right-Front

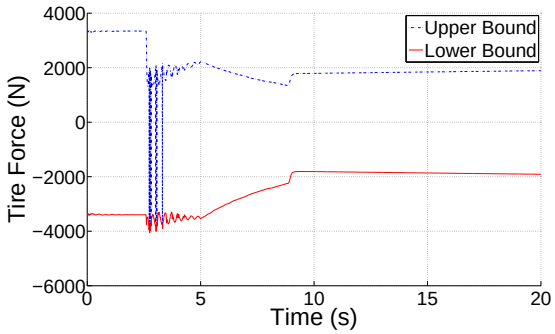


(c) HCC Torque, Left-Rear

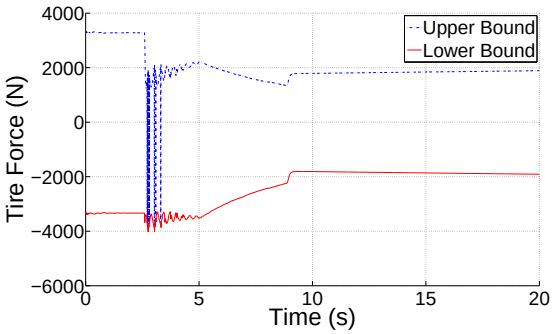


(d) HCC Torque, Right-Rear

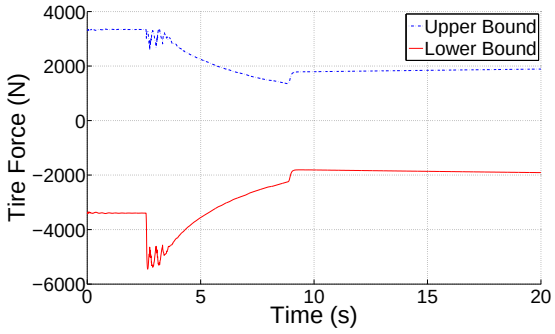
Figure 5.10: Slip Ratios, Traction Control On



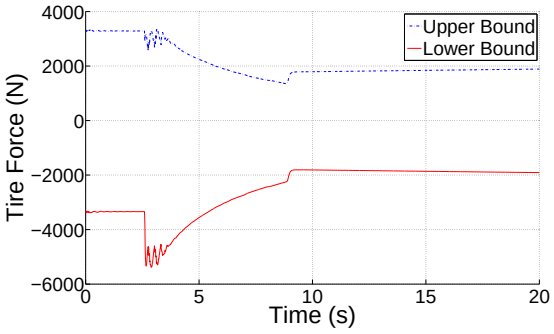
(a) Permissible Tire Force Bounds, Left-Front



(b) Permissible Tire Force Bounds, Right-Front



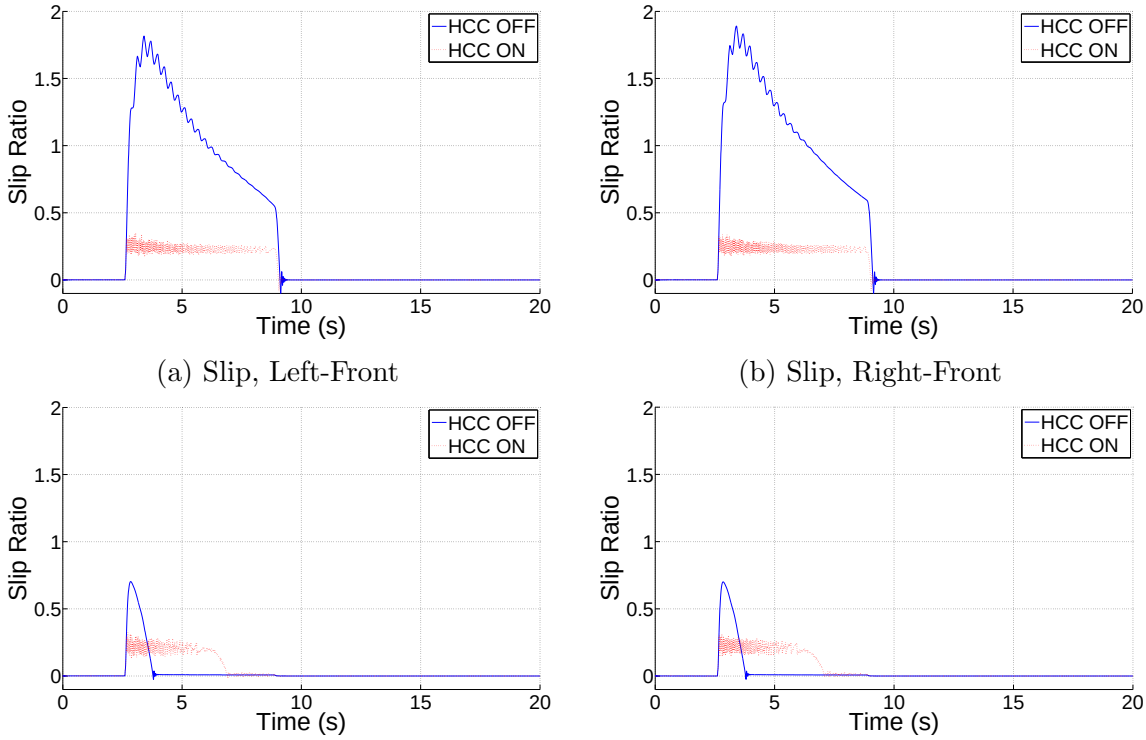
(c) Permissible Tire Force Bounds, Left-Rear



(d) Permissible Tire Force Bounds, Right-Rear

Figure 5.11: Permissible Tire Force Bounds, $\mu = 0.6$

A similar set of results are given in Figures 5.12 and 5.13 below for lower coefficient surface, of $\mu = 0.3$.



(a) Slip, Left-Front

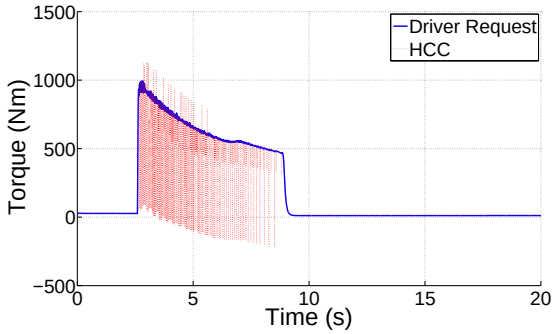
(b) Slip, Right-Front

(c) Slip, Left-Rear

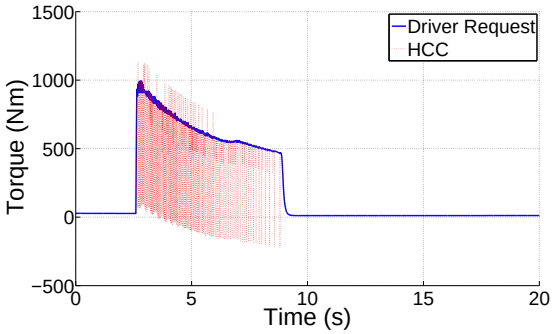
(d) Slip, Right-Rear

Figure 5.12: Slip Ratios, Traction Control On, $\mu = 0.3$

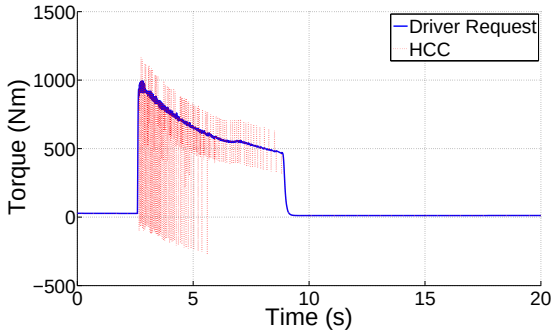
It is seen that slip is very well controlled, not exceeding the target threshold of 0.2.



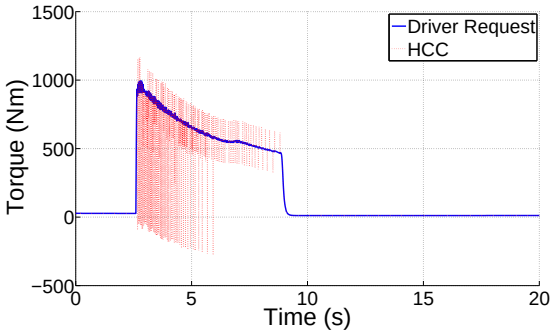
(a) HCC Torque, Left-Front



(b) HCC Torque, Right-Front



(c) HCC Torque, Left-Rear



(d) HCC Torque, Right-Rear

Figure 5.13: Slip Ratios, Traction Control On, $\mu = 0.3$

5.3.1 Experimental Results, Split-Mu, Wet Asphalt

Experimental results are given in this section for the vehicle accelerating in a straight line on a water sprayed split-mu asphalt surface. In this, one side of the vehicle is driven on top of a sealed asphalt surface which provides lower grip, while the other is a regular rough asphalt. The RWD was chosen for this test, as rear wheel drive cars have an increased tendency to spin out under harsh acceleration, which poses a larger problem for the HCC. The pedal inputs for this experiment are shown in Figure 5.14.

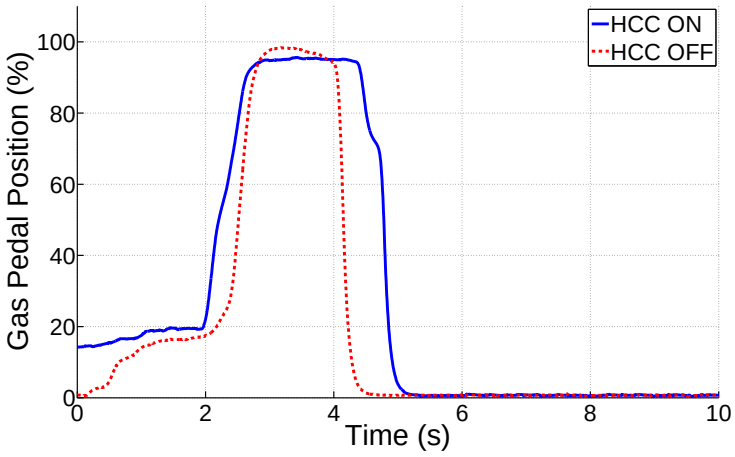
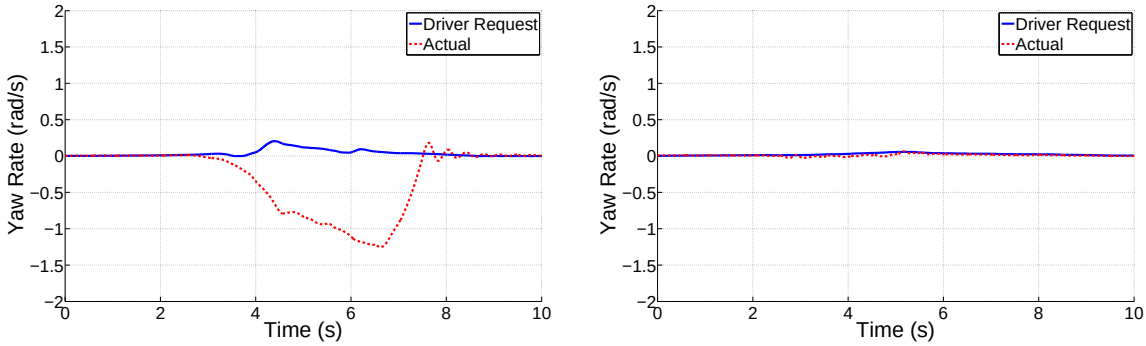


Figure 5.14: Pedal Inputs, Split-Mu Acceleration

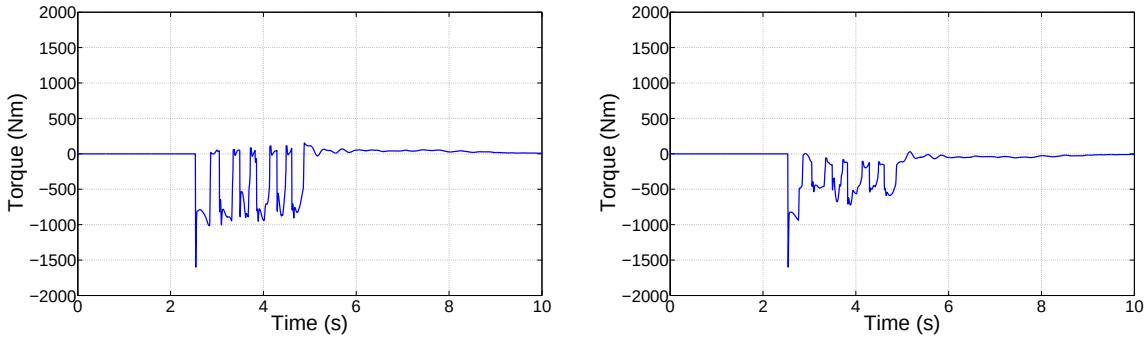
Next, the vehicle yaw responses are shown in Figure 5.15. Here, it is seen that the



(a) HCC OFF (b) HCC ON

Figure 5.15: Comparison of Yaw Rate Responses on a Split-Mu Surface

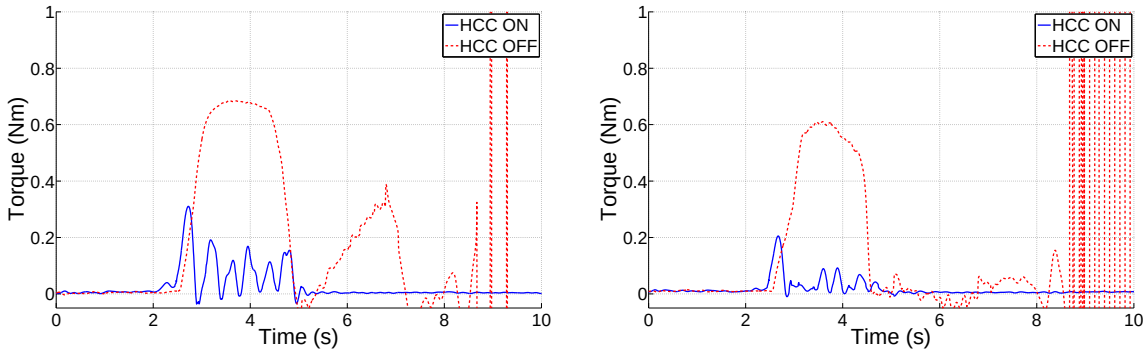
vehicle experiences a spin-out condition without the HCC, as evidenced by the large error in vehicle yaw rate. The controlled system however, is able to maintain a very straight path. The HCC augmentation torques used to achieve this result are shown in Figure5.16.



(a) Rear-Left (b) Rear-Right

Figure 5.16: HCC Correction Torques For Driven Wheels

Finally, a comparison the slips measured at each wheel are shown in Figure 5.17.



(a) Rear-Left (b) Rear-Right

Figure 5.17: Comparison of Slip At Driven Wheels

It is seen that without the traction controller, large magnitudes of slip occur. With the HCC enabled however, slip levels remain controlled.

5.4 Double Lane Change

The main maneuver used to quantify dynamic vehicle performance is the double lane change (DLC) due to its simple nature and ability to scale vehicle accelerations by controlling speed.

The steering input used for the DLC is shown in Figure 5.18.

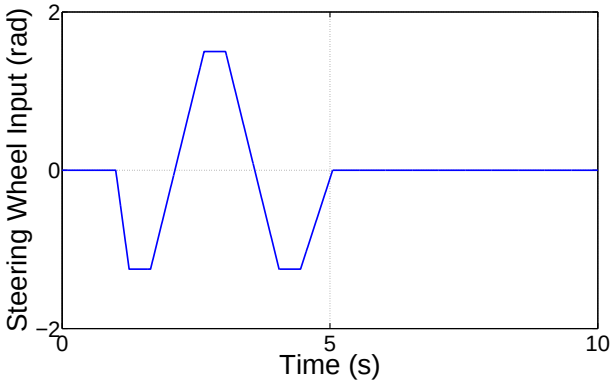
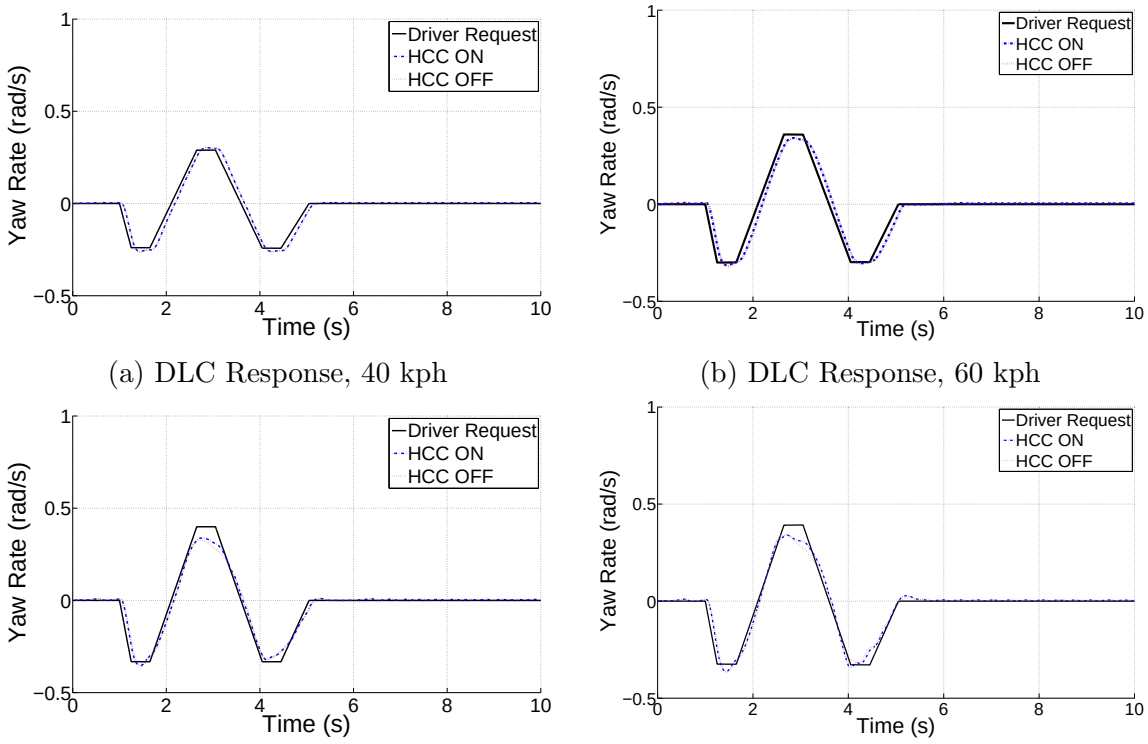


Figure 5.18: Steering Input for Double Lane Change

To validate the DCI, this steering input is given over a range of speeds to determine its accuracy at predicting vehicle behaviour in optimal conditions ($\mu = 1.0$). The ideal scenario is a DCI tuned such that under high-friction conditions, the feed-forward yaw rate response closely matches that of the car without any active control. This guarantees that the HCC would not be activated in normal conditions, and vehicle handling would not be affected. System responses for 40-120 kph trials are shown below in Figure 5.19. In these, "Driver Request" is the feed-forward response generated by the DCI, and used by the HCC-ON mode. In the HCC-OFF mode, both the DCI and HCC are ignored and a pure vehicle response is taken.



(a) DLC Response, 40 kph

(b) DLC Response, 60 kph

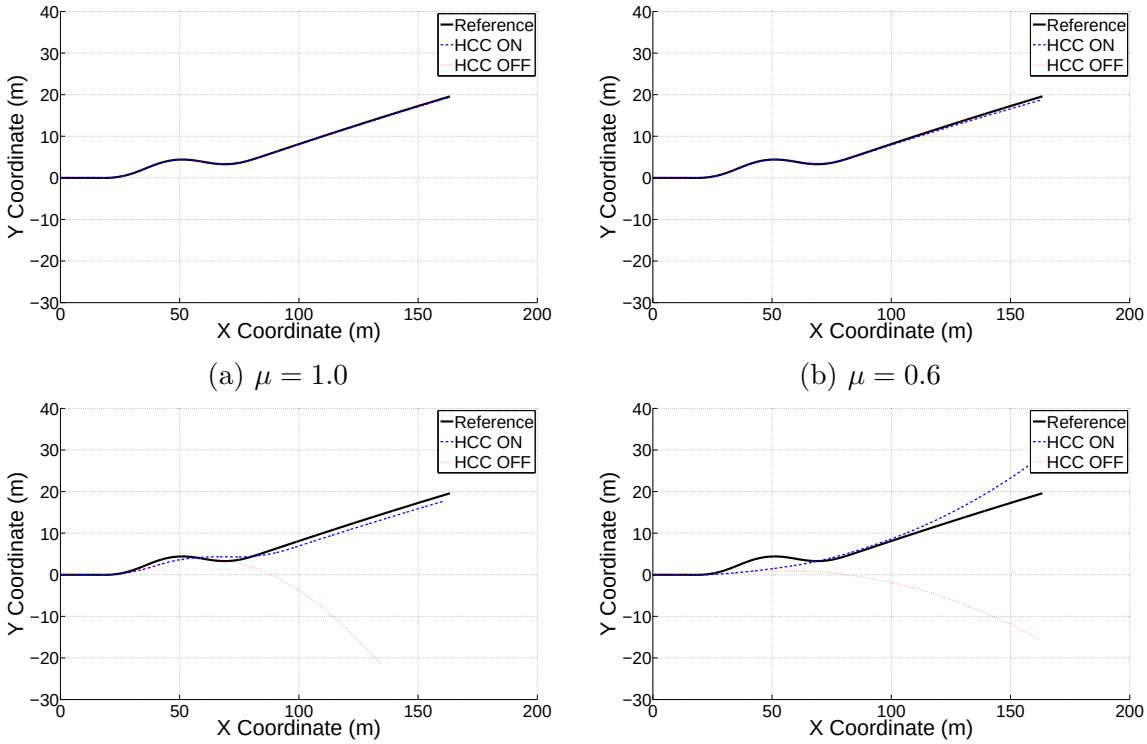
(c) DLC Response, 100 kph

(d) DLC Response, 100 kph

Figure 5.19: Comparison of Yaw Rate Responses at $\mu = 1.0$

It is seen that in general, the DCI generated output closely matches the actual vehicle response. Due to the small amount of error, it is also seen that the HCC enabled system does not significantly stray away from the natural response. In all cases, a minor phase lead is noted between the DCI and system feedback. This may be explained by compliance in the system, such as tire deflections.

By applying the same steering inputs over a range of lower friction surfaces, the HCC's ability to maintain stability may be studied. A nominal entry speed of 60 kph is used in the following examples, and surfaces of $\mu = 0.1, 0.3$, and 0.6 are used to compare against the reference case of $\mu = 1.0$. The path tracking abilities of the car are shown below in Figure 5.20.



(a) $\mu = 1.0$ (b) $\mu = 0.6$

(c) $\mu = 0.3$ (d) $\mu = 0.1$

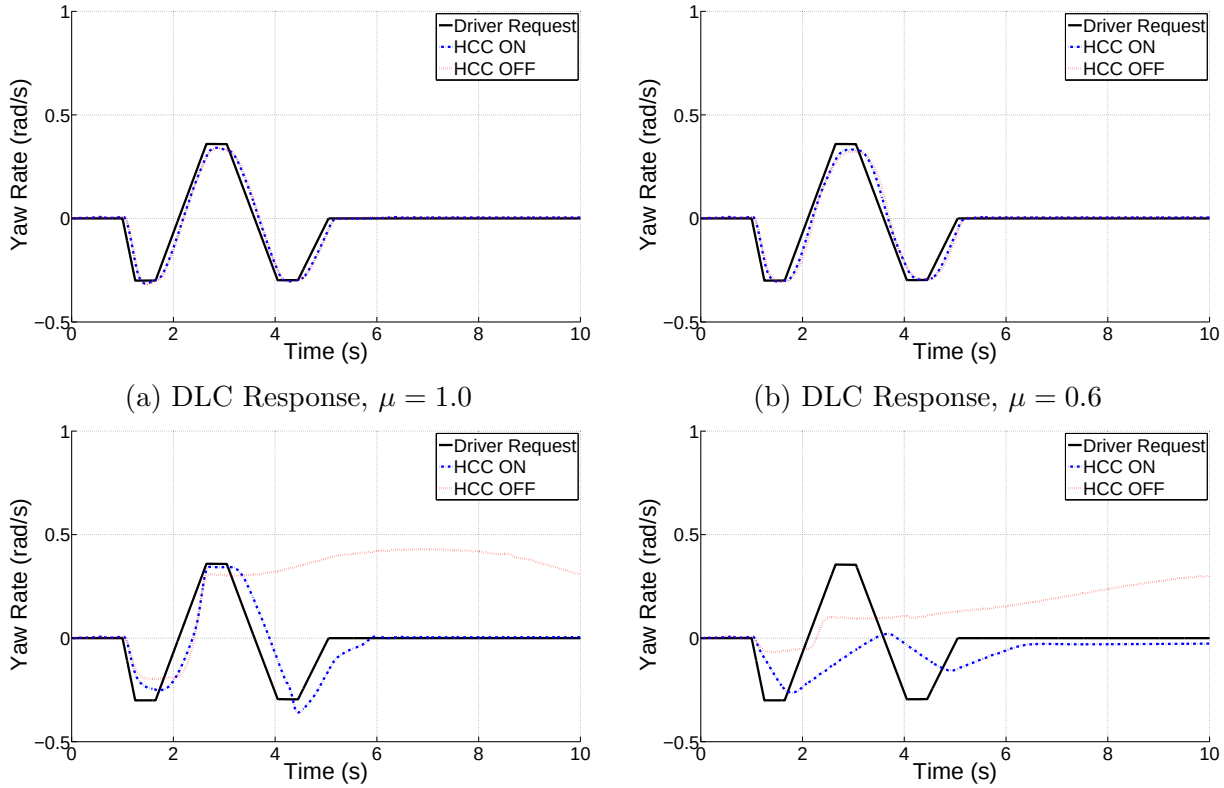
Figure 5.20: Path Following Abilities of HCC Across Various Friction Surfaces

It is seen that in the reference case of $\mu = 1.0$, path tracking of the car is not affected by the HCC, and a natural handling state is achieved. An overall upward trend in the path is noted, despite a symmetric steering input. This is due to the fact that vehicle velocity drops during the maneuver (no gas is applied), and the direction of the first turn incurs more displacement than subsequent counter steers.

The $\mu = 0.6$ case represents the limit of non-assisted handling for this maneuver, and shows that the HCC does not add unwanted actuation to the system unless it is warranted. Starting with the $\mu = 0.3$ case, significant path deviations are seen for cases where the

HCC is off. Performance still very closely resembles the reference command at $\mu = 0.3$, but appreciable error is seen by $\mu = 0.1$.

To gain a better understanding of vehicle heading and the HCC's performance as a yaw controller, rate responses are provided in Figure 5.21.



(a) DLC Response, $\mu = 1.0$

(b) DLC Response, $\mu = 0.6$

(c) DLC Response, $\mu = 0.3$

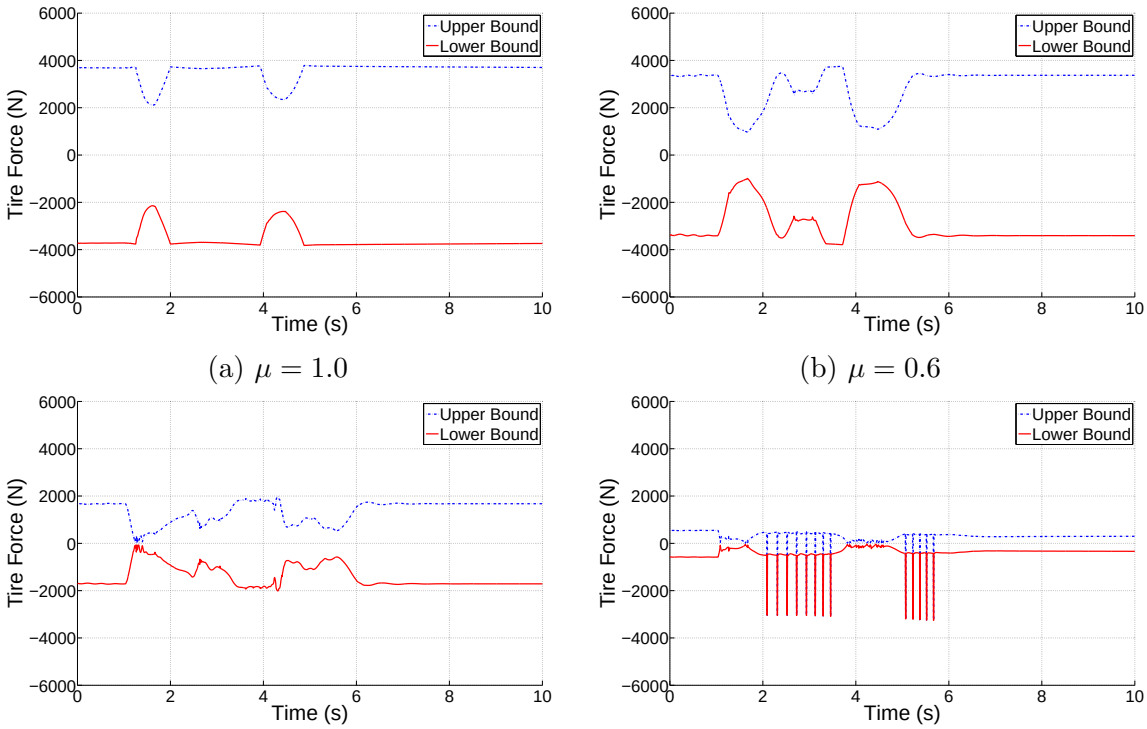
(d) DLC Response, $\mu = 0.1$

Figure 5.21: Comparison of Yaw Rate Responses on Differing Surfaces

As implied from the path tracking results, vehicle stability is maintained with the HCC, although response lags are seen at extremely low coefficients. This is due to lower tire capacities with which to apply corrective yaw moments. Loss of control is clearly shown in the yaw rate charts; without the HCC, divergence is seen after the 2s mark for the $\mu = 0.3$ and $\mu = 0.1$ cases. In these, the vehicle spins with a positive yaw rate despite a steering request of 0. By comparison, the actively controlled system returns to a stable forward motion of $r = 0$ with minimal oscillation or overshoot.

To show how the HCC adapts to differing surfaces, computation bounds for the front-

left tire are shown in Figure 5.22.



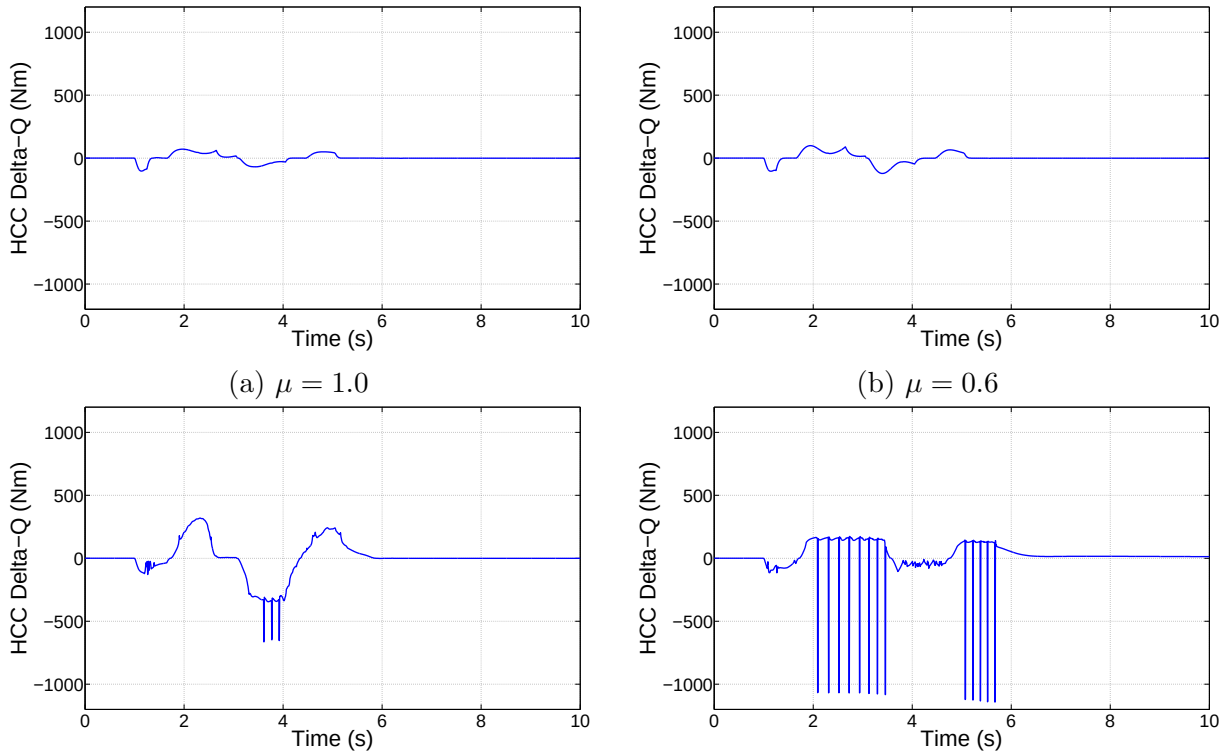
(a) $\mu = 1.0$ (b) $\mu = 0.6$

(c) $\mu = 0.3$ (d) $\mu = 0.1$

Figure 5.22: Permissible Tire Bounds for the Rear-Right Tire Over Various Friction Surfaces

It is seen that the system effectively constricts the search space given diminishing friction values, which works to prevent over-actuation of the tires. In the $\mu = 0.1$ case, a severely constricted search space enables any estimation error to over-torque the wheels to result in slip. Once this happens, the traction control algorithm is activated, which results in the rake-like corrective pulses.

Solving the HCC equation given the tire force constraints results in the final augmentation torques at each wheel. Corresponding values for the same front-left tire are shown below in Figure 5.23.



(a) $\mu = 1.0$

(b) $\mu = 0.6$

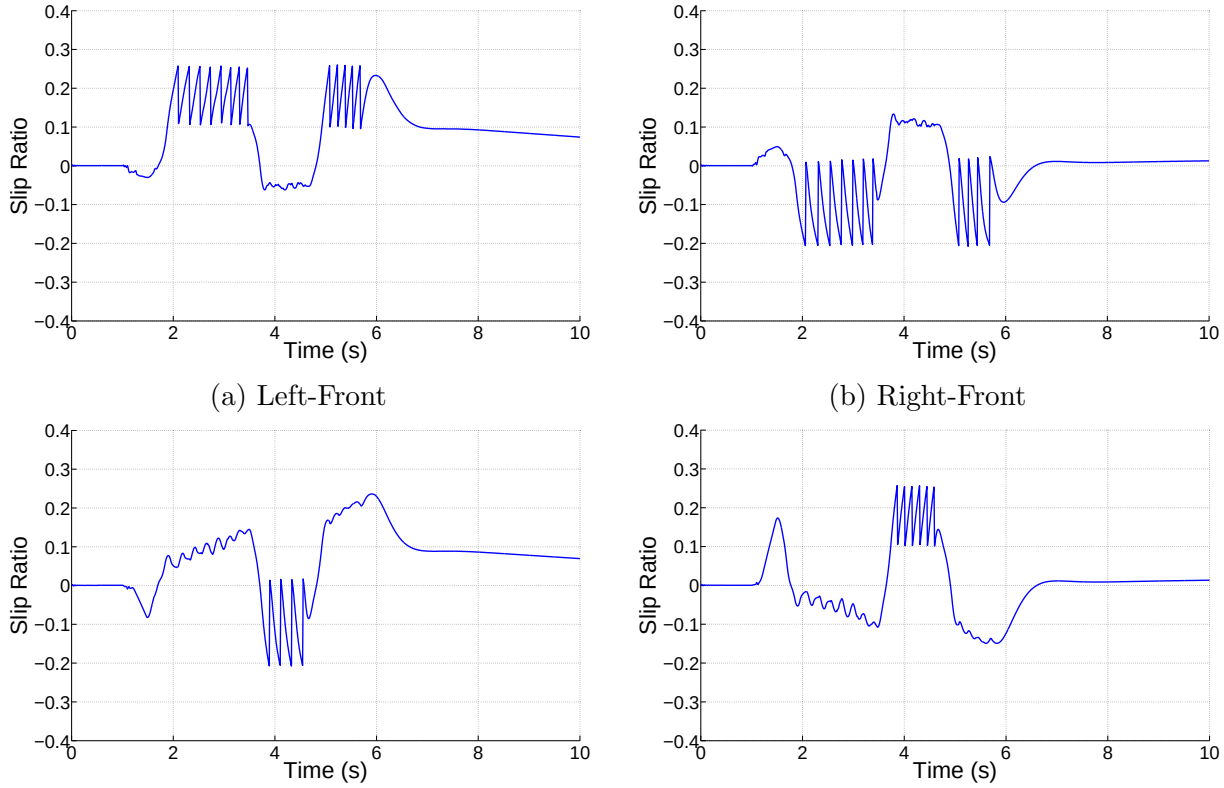
(c) $\mu = 0.3$

(d) $\mu = 0.1$

Figure 5.23: Comparison of HCC Augmentation Torques on Differing Surfaces

It is observed that unless slip limits are exceeded, the HCC results in smooth control signals that scale with increasing yaw rate error. Downward spikes arising from the traction control algorithm are first observed at 4s in the $\mu = 0.3$ case, and eventually comes to dominate control outputs at $\mu = 0.1$.

Finally, Figure 5.24 shows the slip ratios for all four wheels in the extreme $\mu = 0.1$ case, to show effectiveness of the traction controller operating within a dynamic maneuver.



(a) Left-Front

(b) Right-Front

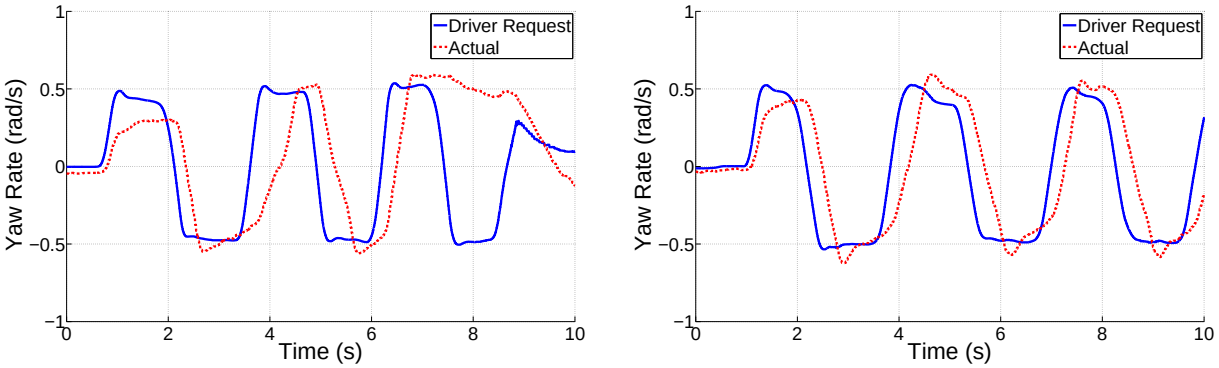
(c) Left-Rear

(d) Right-Rear

Figure 5.24: Slip Ratios at Each Corner, $\mu = 0.1$

5.4.1 Experimental Results, Sine Wave in Snow

In this section, results are provided for the vehicle under a freshly-fallen snow condition, with approximately 1 inch of accumulation. Continuous lane change maneuvers were manually applied, at a speed of approximately 50 kph. Yaw rate responses for the car are provided in Figure 5.25 below, to compare performance with and without the HCC. Vehicle speeds are recorded via GPS for both runs, and provided in Figure 5.26 below to ensure consistency between both tests.



(a) HCC OFF (b) HCC ON

Figure 5.25: Yaw Rate Response Comparison, Snowy Conditions

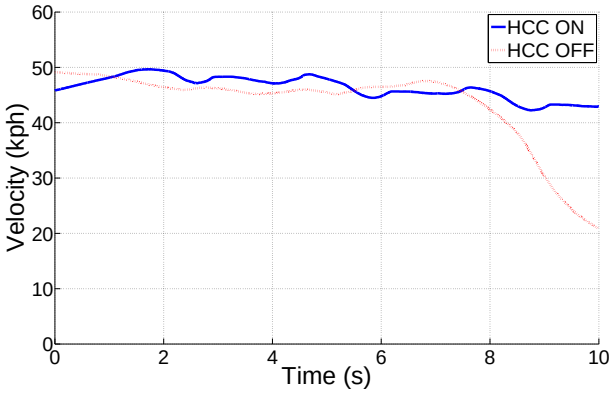
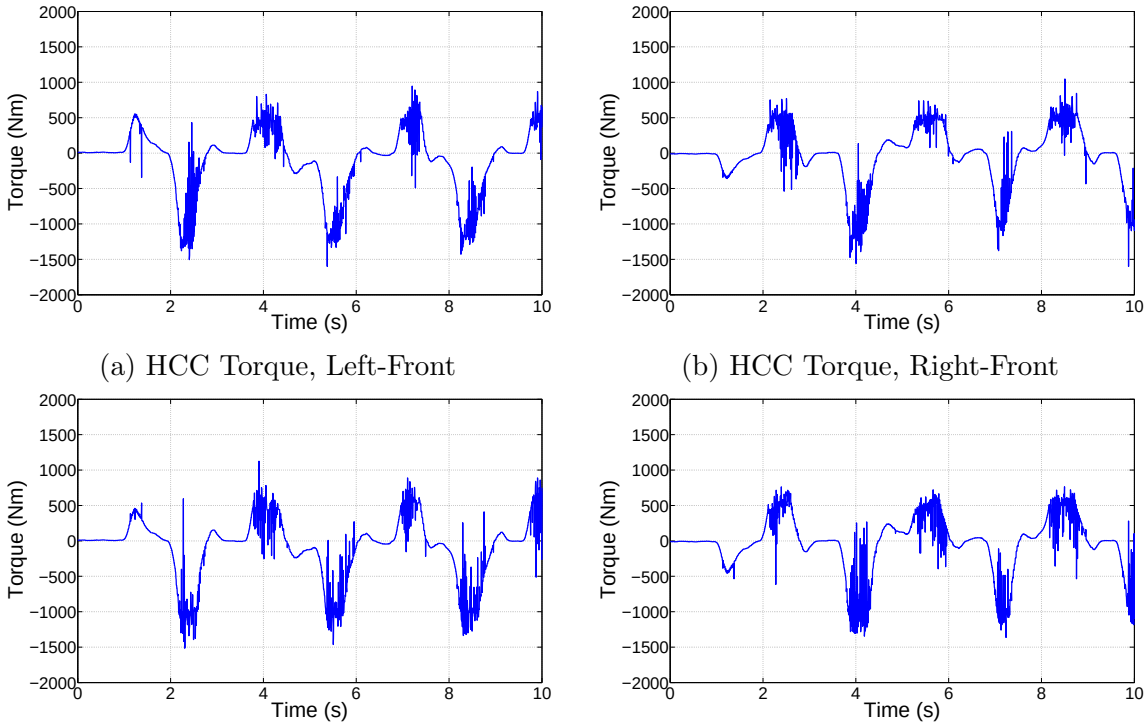


Figure 5.26: Vehicle Speeds During Maneuver

It is seen that without the HCC, major tracking errors are seen between the desired response and actual feedback. Most importantly, a loss of control is encountered at the 7s mark without HCC, while the actively controlled system is able to follow driver inputs.

The HCC augmentation torques used to achieve these results are provided in Figure 5.27 below.



(a) HCC Torque, Left-Front

(b) HCC Torque, Right-Front

(c) HCC Torque, Left-Rear

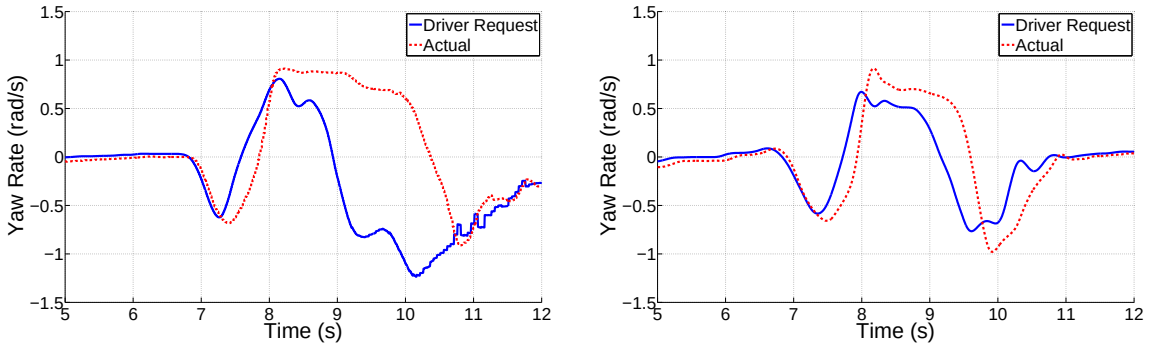
(d) HCC Torque, Right-Rear

Figure 5.27: HCC Torques, Snowy Conditions

The major difference between simulation and experimental results is the high-frequency switching around peak regions, mostly due to real-life sensor noise in the accelerometer and wheel speed signals. Despite this, the vehicle still gains most of the benefit of the torque vectoring, and stability is enhanced.

5.4.2 Experimental Results, Double Lane Change on Asphalt

In this section, vehicle performance on dry asphalt condition are provided. The RWD platform is shown, to offer a view of performance given only have the actuators as seen in snow. A comparison of vehicle yaw rates is shown below in Figure 5.28.



(a) Yaw Rate, HCC OFF

(b) Yaw Rate, HCC ON

Figure 5.28: Performance Comparison, Double Lane Change on Asphalt

Here, a loss of vehicle control is observed at the 8s mark for the vehicle with HCC. An oversteer condition is encountered, and the vehicle continues spinning despite a counter-steer action. A comparison of vehicle speeds through the run is provided below in Figure 5.29, to ensure comparability between tests.

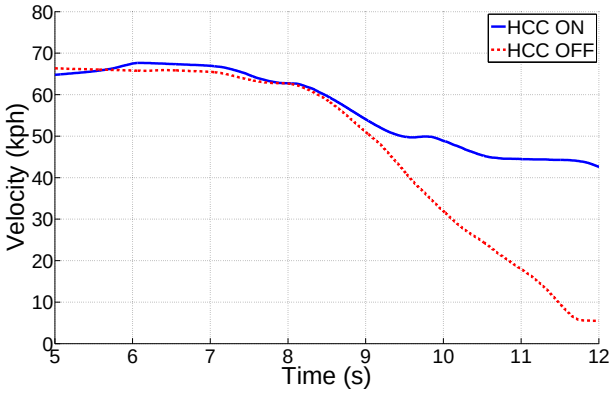
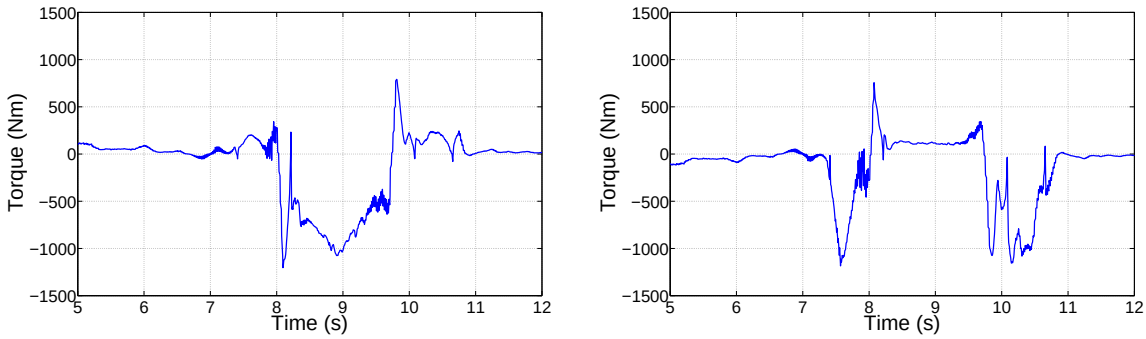


Figure 5.29: Vehicle Speeds During Maneuver

The generated vectoring torques used to achieve these are shown in Figure 5.30.



(a) Rear-Left (b) Rear-Right

Figure 5.30: HCC Vectoring Torques

Chapter 6

Conclusion

It is shown that the vehicle control structure proposed in this thesis was capable of significantly improving vehicle response through a range of conditions. A comprehensive approach to vehicle design was undertaken, with two Equinox SUVs being retrofit with electric motor drives and instrumented for experimental data collection. A simulation environment linking CarSim to Matlab/Simulink was created in parallel for design work and debugging, while compiled code was generated for real-time experimentation on a dSpace embedded computer. In drive cycle simulations, the proposed energy management module conserved between 5-7% energy, with nothing more than software to realize the savings. In experimental testing, a wide range of dynamic maneuvers were successfully executed, including full-acceleration launches to prevent spin-out, sinusoidal steer inputs in snow to test grip, and high-speed lane changes on asphalt to test vehicle stability. Overall, the vehicle controller proposed in this thesis is shown to be a viable production-oriented controller that is highly tunable, modular, and performance enhancing. The goal of this controller, namely the integrated control of stability and energy management controls for an electric vehicle, was successfully achieved.

References

[1] Abtin Athari, Saber Fallah, Bin Li, Amir Khajepour, Shih-Ken Chen, and Baktiar Litkouhi. Optimal torque control for an electric-drive vehicle with in-wheel motors: Implementation and experiments. *SAE International Journal of Commercial Vehicles*, 6(1):82–92, 2013.

[2] Claus Beyer and Peter Dominke. Antilock braking system, January 11 1994. US Patent 5,277,482.

[3] Stephen Poythress Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] G Burgio and P Zegelaar. Integrated vehicle control using steering and brakes. *International Journal of Control*, 79(05):534–541, 2006.

[5] C. Canudas de Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *Automatic Control, IEEE Transactions on*, 40(3):419–425, 1995.

[6] Green Car Congress. Amp unveils electric version of gm equinox; "secret sauce" is the drivetrain, 2010.

[7] Calspan Corporation. Calspan - safer highways.. safer skies, 2014.

[8] Michigan Scientific Corporation. Wheel force transducer system, 2014.

[9] Frédéric Delbos, Jean Charles Gilbert, et al. Global linear convergence of an augmented lagrangian algorithm for solving convex quadratic optimization problems. 2003.

[10] dSpace. Microautobox ii, 2013.

[11] New Eagle. Rinehart pm100dx integrates perfectly with a remy hvh250: New eagle integrates perfectly with your ev/hev project, 2012.

[12] New Eagle. Dc-dc converter data sheet, 2013.

[13] Elithion. Lithiumate pro professional distributed li-ion bms, 2013.

[14] EPA. Dynamometer drive schedules, 2013.

[15] Saber Fallah, Amir Khajepour, Barıs Fidan, Shih-Ken Chen, and Bakhtiar Litkouhi. Vehicle optimal torque vectoring using state-derivative feedback and linear matrix inequality. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 62(4), 2013.

[16] H. Fujimoto and H. Sumiya. Range extension control system of electric vehicle based on optimal torque distribution and cornering resistance minimization. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pages 3858–3863, 2011.

[17] Youssef A Ghoneim, Shih-Ken Chen, Valery Pylypchuk, Nikolai K Moshchuk, and Bakhtiar Brian Litkouhi. Real-time allocation of actuator torque in a vehicle, January 31 2011. US Patent App. 13/017,117.

[18] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. *Numerical Linear Algebra and Optimization*. Basic Books, 1961.

[19] T.D. Gillespie. *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, 1992.

[20] Nick I. M. Gould and Philippe L. Toint. Preprocessing for quadratic programming. *Math. Program.*, 100(1):95–132, 2004.

[21] Yoshikazu Hattori. Optimum vehicle dynamics control based on tire driving and braking forces. *R&D Review of Toyota CRDL*, 38(4):23–29, 2003.

[22] Remy International Inc. Hvh250 series electric motors, 2013.

[23] Daofei Li, Shangqian Du, and Fan Yu. Integrated vehicle chassis control based on direct yaw moment, active steering and active stabiliser. *Vehicle System Dynamics*, 46(S1):341–351, 2008.

[24] Daofei Li and Fan Yu. A novel integrated vehicle chassis controller coordinating direct yaw moment control and active steering. *Training*, 2013:12–09, 2007.

[25] EK Liebemann, K Meder, J Schuh, and G Nenninger. Safety and performance enhancement: the bosch electronic stability control (esp). *SAE Paper*, 20004:21–0060, 2004.

[26] Oxford Technical Solutions Ltd. Rt2000 family, 2013.

[27] Zhejiang GBS Energy Co. LTD. Gbs12v100ah details, 2013.

[28] Jorge J Moré and Danny C Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[29] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[30] Damrongrit Piyabongkarn, Jae Y Lew, Rajesh Rajamani, John A Grogg, and Qinghui Yuan. On the use of torque-biasing systems for electronic stability control: limitations and possibilities. *Control Systems Technology, IEEE Transactions on*, 15(3):581–589, 2007.

[31] Huihuan Qian, Guoqing Xu, Jingyu Yan, Tin Lun Lam, Yangsheng Xu, and Kun Xu. Energy management for four-wheel independent driving vehicle. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5532–5537, 2010.

[32] A Rezaeian, R Zarringhalam, S Fallah, W Melek, et al. Cascaded dual extended kalman filter for combined vehicle state estimation and parameter identification. *Training*, 2005:12–15.

[33] Kaoru SAWASE and Yuichi USHIRODA. Improvement of vehicle dynamics by right-and-left torque vectoring system in various drivetrains. *Mitsubishi Tech. Rev*, 20:14–20, 2008.

[34] Xiaoming Shen and Fan Yu. Study on vehicle chassis control integration based on a main-loop-inner-loop design approach. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 220(11):1491–1502, 2006.

[35] Motoki Shino, Naoya Miyamoto, Y-Q Wang, and Masao Nagai. Traction control of electric vehicles considering vehicle stability. In *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on*, pages 311–316. IEEE, 2000.

[36] Motoki Shino and Masao Nagai. Independent wheel torque control of small-scale electric vehicle for handling and stability improvement. *JsAE Review*, 24(4):449–456, 2003.

[37] Rinehart Motion Systems. Pm family data sheet, 2013.

[38] EV West. Pfc 5000, 2013.

[39] Dejun Yin, Sehoon Oh, and Yoichi Hori. A novel traction control for ev based on maximum transmissible torque estimation. *Industrial Electronics, IEEE Transactions on*, 56(6):2086–2094, 2009.

[40] Reza Zarringhalam, Ayyoub Rezaeian, William Melek, Amir Khajepour, Shih-Ken Chen, and Nikolai Moshchuk. A comparative study on identification of vehicle inertial parameters. In *American Control Conference (ACC), 2012*, pages 3599–3604. IEEE, 2012.