MULTI-INPUT GROUND VEHICLE CONTROL USING QUADRATIC PROGRAMMING BASED CONTROL ALLOCATION TECHNIQUES

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

John Harris Plumlee

Certificate of Approval:

A. Scottedward Hodel
Associate Professor
Department of Electrical and Computer Engineering

David M. Bevly, Chair
Assistant Professor
Department of Mechanical Engineering

George Flowers
Professor
Department of Mechanical Engineering

Stephen L. McFarland
Acting Dean, Graduate School

MULTI-INPUT GROUND VEHICLE CONTROL USING QUADRATIC PROGRAMMING BASED CONTROL ALLOCATION TECHNIQUES

John Harris Plumlee

A Thesis
Submitted to
the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the
Degree of
Master of Science

Auburn, Alabama
August, 5 2004

MULTI-INPUT GROUND VEHICLE CONTROL USING QUADRATIC PROGRAMMING BASED CONTROL ALLOCATION TECHNIQUES

John Harris Plumlee

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date

Copy sent to:

Name	Date

VITA

John was born June 9, 1978, the first child of Pryor and Joan Plumlee. He was born in Montgomery, Alabama, and there he attended St. Bede's Catholic School through the second grade. From third grade until High School Graduation, he attended St. James school, after which, he decided to attend Auburn University, a venture resulting in many wonderful friends, fond memories, a future wife, and eventually a B.S. degree in Mechanical Engineering. As an undergraduate, John learned from his co-op experience at Giles Enterprises that a career in manufacturing is not for him. Therefore, in an effort to avoid such a fate he decided to pursue a graduate degree in the dynamics and controls area of mechanical engineering.

THESIS ABSTRACT

MULTI-INPUT GROUND VEHICLE CONTROL USING QUADRATIC PROGRAMMING BASED CONTROL ALLOCATION TECHNIQUES

John Harris Plumlee

Master of Science, August, 5 2004
(B.S.M.E., Auburn University, 2002)

79 Typed Pages

Directed by David M. Bevly and A. Scottedward Hodel

A quadratic programming based control allocation technique is proposed for the control of multiple inputs to a ground vehicle to track a desired yaw rate trajectory while minimizing vehicle sideslip. Detailed models are derived for three vehicles, each possessing different input capabilities: 1) front steering and front and rear differential braking. 2) front and rear steering and front and rear differential braking. 3) front and rear steering and individual torque control of each wheel. A method for the application of control allocation techniques to a ground vehicle with coupled dynamics is presented. The proposed control strategy uses quadratic programming accompanied by linear quadratic regulator gains to provide intelligent combinations of input commands to a ground vehicle in order to accomplish multiple objectives. Both nominal and failure scenarios are examined and the results are presented along with a discussion of the effects of the proposed controller's design parameters. The control strategy successfully minimizes opposing commands in the presence of conflicting objectives as well as maintains good tracking in the event of a steering failure.

ACKNOWLEDGMENTS

This thesis would not have been possible without the support of many people throughout the last few years. I must first give my thanks to God, for all that He has done and will continue to do in my life.

For their financial and emotional support throughout my life, I would like to thank my parents, Joan and Skip Plumlee. I don't think they could have tried any harder or had better success at being the wonderful role models that they are. Without their continuous love and guidance, I would not be the person I am today. I would also like to thank my brother Ryan for always being there for me no matter what. It is impossible for us to not have a good time when we are together.

I would like to thank Dr. David M. Bevly, my graduate adviser for challenging me in my undergraduate years and showing me that dynamics and controls really are the coolest areas of research. His motivation has helped me to realize that I am capable of accomplishing much more than I had previously thought possible.

I would like to thank Dr. A. Scottedward Hodel, my graduate adviser for taking a chance on an unfamiliar ME student. His inspiration and enthusiasm, along with his efforts to explain things clearly and simply have been invaluable to me throughout this research.

I am blessed to have had not one but two wonderful mentors whose strong Christian values and seemingly infinite knowledge have provided a graduate experience that has far exceeded my expectations.

I would also like to thank Adam Simmons for letting me ask him tons of questions every day and also for listening to me vent my frustrations on occasion. I will always be happy to beat you at tennis as long as you continue to return the favor on the golf course.

Finally, the completion of my Masters degree would have been infinitely more difficult without the love and support of my best friend and wife Emily. I am forever indebted to her for the sacrifices she has made over the last two years and it is to her that I dedicate this thesis.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliography follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package TeX (specifically LaTeX) together with the departmental style-file `aums.sty`.

TABLE OF CONTENTS

LIST OF FIGURES xi

1 INTRODUCTION 1

2 CONTROL ALLOCATION 4
- 2.1 Introduction 4
- 2.2 The Control Allocation Problem 5
- 2.3 Direct Control Allocation and Generalized Inverse Solutions 6
- 2.4 Daisy Chaining 7
- 2.5 Optimization Based Control Allocation Methods 8
  - 2.5.1 Least Squares 9
  - 2.5.2 Linear Programming 10
  - 2.5.3 Quadratic Programming 11
  - 2.5.4 Model Predictive Control 12

3 VEHICLE MODEL 13
- 3.1 Nonlinear Vehicle Model 13
- 3.2 Nonlinear Tire Model 18
- 3.3 Linear Vehicle Model 21
- 3.4 Model Verification 23

4 CONTROLLER DESIGN 25
- 4.1 Control Law 26
- 4.2 Command Generation 27

5 SIMULATION AND RESULTS 31
- 5.1 Effector Limitations 31
- 5.2 Effector Failures 32
- 5.3 The Effects of Noise 33
- 5.4 Results 35
  - 5.4.1 Nominal Case 37
  - 5.4.2 Failure Case 40

6 CONCLUSION 48

A COMPUTER CODE 51
A.1 Initialization and Simulation Routine 51
A.2 S-function Implementation of the Vehicle Model 54
A.3 Implementation of Nonlinear Tire Model and Friction Circle 59
A.4 Implementation of Quadratic Programming Based Control Allocation . . 63

BIBLIOGRAPHY 66

List of Figures

3.1 Free body diagram of vehicle model 13

3.2 Free body diagram of vehicle's front view 16

3.3 Pacejka tire model lateral force curves 18

3.4 Free body diagram of tire. 19

3.5 Friction Circle for $F_z = 3000N$ 20

3.6 Step Response at 55mph for Front Steering input of $\delta_f = 1^\circ$ 23

3.7 Step Response at 55mph for Rear Steering input of $\delta_r = 1^\circ$ 24

3.8 Step Response at 55mph for Front Differential Braking input of $\Delta F_{xf} = 2000N$ 24

5.1 Controller performance in the presence of noise. 3-input vehicle model at 55mph, $Q_\nu = 1e7$. 34

5.2 Nominal case braking commands at 55mph for different values of Q_ν . . . 38

5.3 Nominal case sideslip angle tracking at 55mph. 39

5.4 Poor yaw rate tracking due to rear steering command. $Q_{\delta_r} = 1$ 40

5.5 Nominal Case 3-Input: 55mph using $Q_\nu = 1e7$ 42

5.6 Nominal Case 4-Input: 55mph using $Q_\nu = 1e7$ 43

5.7 Nominal Case 6-Input: 55mph using $Q_\nu = 1e7$ 44

5.8 Steering Failure, 3-Input: 55mph using $Q_\nu = 1e3$ 45

5.9 Steering Failure 4-Input: 55mph using $Q_\nu = 1e3$ 46

5.10 Steering Failure 6-Input: 55mph using $Q_\nu = 1e3$ 47

Chapter 1

Introduction

Ground vehicles have become an almost essential part of modern life where they are depended upon daily to provide services such as transportation for people and/or cargo. For this reason, much research has been devoted to the overall advancement of ground vehicle technology. Modern science has allowed for the production of ground vehicles that can operate autonomously or semi-autonomously saving both time and money. Even the military has come to rely on intelligent unmanned vehicles for perfomming routine and/or potentially dangerous tasks. Saftey is undoubtedly a major source of motivation behind the increacing performance demands of manned ground vehicles. Steer/brake/throttle-by-wire technology has allowed for passenger vehicle safety systems such as driver assisted automated lane keeping [30] and stability control systems [36] to become realizable. Performance requirements must be satisfied with complex vehicle designs that incorporate by-wire technology and additional control inputs such as differential braking and rear steering. The addition of differential braking as a control parameter has proven to be successful in maintaining vehicle stability during drastic maneuvers [34] and in rollover prevention [12], [13]. Much research has been conducted on the control of multi-input ground vehicles and many methods have been developed based on linear quadratic regulator/state feedback design [32, 35]. However, in order to fully exploit the performance potential of a multi-input vehicle, a well designed *control allocation* law may be required.

The term *control allocation* in the general sense, refers to the distribution of a control effort among the available effectors of a physical system in order to achieve a desired objective. Any input parameter that can be used in a controlled manner, e.g., the steering angle on a car, to manipulate system behavior may be thought of as an effector. The need for control allocation arises in many different physical systems such as vehicle control, chemical process control, and even production planning of manufacturing facilities.

In vehicle control system design, the problem of command generation is made even more difficult by the fact that the input effectors have physical limitations that must be considered. Traditionally, control allocation (CA) techniques have been used for aerospace and underwater vehicle applications due to their ability to satisfy multiple control objectives (e.g. torques in all three axes). However, CA techniques may also be applied to a multi-input ground vehicle in order to accomplish more than one goal (e.g. yaw rate tracking and sideslip minimization).

In the past, solutions to the CA problem of generating input commands that accomplish multiple objectives have been found with simple methods such as generalized inverse or least squares approaches. These methods can provide fast online solutions but fail to consider effector limitations. By posing the CA problem as a quadratic programming (QP) problem however, the physical constraints of the effectors may be incorporated into the solution. Although QP has been disregarded as a CA technique in the past due to the computational time required to solve such a problem, increased processor speed and the development of faster QP algorithms have allowed for QP to become a feasible CA solution method.

Another advantage of CA is its ability to account for failed or damaged effectors and maintain the best possible desired performance with the remaining effectors. If the dynamics of the vehicle model are coupled then a CA problem could result in which conflicting objectives must be satisfied. Opposing effector commands will be produced and poor tracking of the desired responses will result. This problem can be avoided by setting up the QP problem in such a way that the dynamic coupling is effectively reduced so that the most important objective is made a priority.

In this thesis, a method for the application of CA techniques to a ground vehicle with coupled dynamics is presented. The proposed control strategy provides intelligent combinations of input commands to a ground vehicle in order to accomplish multiple objectives. The control strategy successfully minimizes opposing commands in the presence of conflicting objectives as well as maintains good tracking in the event of an effector failure.

The remainder of this thesis is organized as follows. An introduction to the control allocation problem and a review of control allocation techniques are presented in Chapter 2. A derivation of the vehicle models used for this research is presented in Chapter 3. The design of the vehicle controller is discussed in Chapter 4. Simulation tests and results are presented in Chapter 5. And finally, conclusions are drawn in Chapter 6.

CHAPTER 2

CONTROL ALLOCATION

2.1 Introduction

In this chapter, existing methods and approaches for *control allocation* are discussed. *Control allocation* (CA) refers to the calculation of multiple input commands that will cause a desired dynamic response in a physical system. Control allocation is of particular interest to the vehicle control community where it plays a necessary role in the control system design of overactuated vehicles. An overactuated vehicle is one in which different combinations of effector commands can produce the same result. In many cases the number of effectors available exceeds the number of states being controlled. For example, a car with steering and differential braking capabilities can turn a corner either by steering in the desired direction, by braking the wheels on the side of the vehicle corresponding to the desired direction, or by using some combination of these two inputs. The control allocation problem is defined as the selection of which set of inputs to use and the corresponding input command values.

A key feature of control allocation is that of reconfiguration. In the event an effector failure occurs and is detected, the control effort is redistributed among the remaining active effectors to minimize the tracking error. Therefore, the advantages of such multiply redundant control systems can be realized not only through increased maneuverability but also through the additional safety aspects they can provide. It is not surprising that much research has been devoted to this subject and many different methods for finding a solution to CA have been developed. Different methods of control allocation have been developed for aerospace vehicles [20], [17], marine vessels [24], and other areas where this type of problem arises.

2.2 The Control Allocation Problem

Recent advances in technology have enabled the design of overactuated vehicles that in turn lead to increasingly complex CA design requirements. Consider, in this chapter, a vehicle with n actuators with respective command values u_1,...,u_n. These commands shall be assumed to be physically limited in the range $u_i^- \leq u_i \leq u_i^+$, or, in vector form $u^- \leq u \leq u^+$, where $u \triangleq \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}^T$. The general control allocation problem is stated as the generation of a set of effector commands u that will match the actual effect to some desired effect $\bar{u}$ as closely as possible while minimizing the control effort and obeying the position and rate constraints of the effectors. In other words, given a desired response $\bar{u}$, solve

$$Bu = \bar{u} \tag{2.1a}$$

$$u^- \leq u \leq u^+ \tag{2.1b}$$

for the unknown command vector u where B is a $m \times n$ matrix of rank m defining the effectiveness of the actuators. If multiple solutions exist, then choose the solution that will minimize a predetermined cost function $J(x)$. If there are no solutions u satisfying the constraint set (2.1b), then the problem (2.1) is infeasible, and we instead compute a command vector u such that Bu provides the best possible approximation of $\bar{u}$.

2.3 Direct Control Allocation and Generalized Inverse Solutions

Direct control allocation is based on the concept of the *attainable moment set* (AMS) of a vehicle as proposed by Duram in [15], [16]. The AMS is defined as the set of all moment vectors (control effects $\bar{u}$) that can be achieved within the available control space $\Omega = \{u \in \mathrm{I\!R}^m : u^- \leq u \leq u^+\}$. The solution to the direct control allocation problem is then to calculate a matrix G that maps the set of commands u to the resulting control effect $\bar{u}$. If the desired control effect vector $\bar{u}$ lies outside the AMS, it is clipped at the boundary and the command vector which produces the maximum effect in the direction of the original vector is used. The generalized inverse solution is described in [8] as any matrix $G : \mathrm{I\!R}^m \rightarrow \mathrm{I\!R}^n$ which satisfies $BG = I_m$. The most widely used generalized inverse solution is the pseudo-inverse $G = B^T[BB^T]^{-1}$ due to the fact that it will yield the minimum 2-norm solution (or minimum control energy) [15]. The generalized inverse solution is then found through multiplication of the pseudo-inverse $B^\dagger$ of the effectiveness matrix B by the desired control effect.

$$u = B^T[BB^T]^{-1}\bar{u} \tag{2.2}$$

An alternative to the generalized inverse method is the weighted pseudo-inverse which uses a weighting matrix Q in the pseudo-inverse calculation. The weighted pseudo-inverse solution is calculated as

$$u = Q^{-1}B^T[BQ^{-1}B^T]^{-1}\bar{u} \tag{2.3}$$

where the matrix Q may be selected to emphasize or de-emphasize the use of certain effectors [8], [26].

The cascaded generalized inverse (CGI) method has an advantage over the previous methods in that it addresses actuator saturation. The CGI method solves the weighted pseudo-inverse problem repeatedly. The effects of any saturated effectors are subtracted from the desired effect and removed from the problem and the weighted pseudo-inverse is solved again. This procedure is repeated until either the desired effect is achieved, all effectors saturate, or fewer effectors remain than desired effects [26].

2.4 Daisy Chaining

The method of daisy chaining to solve the CA problem involves dividing the control vector u and effectiveness matrix B into two or more groups. These groups are then used successively to produce the desired effect $\bar{u}$. In mathematical notation, for k groups of effectors

$$Bu = \begin{bmatrix} B_1 & \cdots & B_k \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_k \end{bmatrix} = B_1 u_1 + \cdots + B_k u_k \tag{2.4}$$

where B_i, $i = 1, \ldots, k$ are considered full rank and invertible. The idea is then to use a generalized inverse method to try and satisfy the control effect with the first group of actuators while the other groups are held constant. If one or more effectors saturate then the second group is employed to make up for the saturation while all other groups are held constant. This process is repeated until either the desired effect is met or all groups are saturated [15]. The disadvantage of daisy chaining lies in the fact that the effectiveness of a group of effectors is limited by the most constrained effector in that group. Therefore, control authority is lost when effectors with similar physical limitations are not grouped together (not all attainable moments are accessible) [8].

2.5 Optimization Based Control Allocation Methods

As stated in Section 2.2, the control allocation problem involves the selection of a control vector u for a vehicle that 1) obeys the vehicle's physical limits (2.1b) and 2) provides the best possible approximation of the desired effect $\bar{u}$. Difficulty arises when an approximate solution must be found because an exact solution is not possible, or when a unique solution must be found and there are an infinite number of equally valid solutions. Therefore, the CA problem (2.1) may be presented as the optimization problem

$$\begin{aligned} \min_{u} \quad & \|W_1(Bu-\bar{u})\| + \gamma \|W_2(u-u_p)\| \\ \text{subject to} \quad & u^- \leq u \leq u^+ \end{aligned} \tag{2.5}$$

where u_p is a vector of preferred commands, and $\|\cdot\|$ represents the chosen norm by which the accuracy of the solution u is measured (by which the distance between the computed and desired solution is minimized). The $\|Bu-\bar{u}\|$ term serves to minimize the allocation error, the $\|u-u_p\|$ term serves to minimize the total control effort based on the selection of u_p, and γ is a weighting factor used to dictate which term is more important. These concepts are appropriately referred to by Bodson [7] as the Error Minimization and Control Minimization Problems, respectively.

2.5.1 Least Squares

Traditional CA approaches are centered around a simple least squares approach. The least squares method

$$\begin{aligned} \min_{u} \quad & \frac{1}{2}u^T Q u \\ \text{subject to} \quad & Bu = \bar{u} \end{aligned} \tag{2.6}$$

is a minimization based on a weighted 2-norm (sum of the squared error) cost criteria subject to linear equality constraints. For control allocation purposes, u is the set of effector commands and Q is a weighting matrix chosen to prioritize effector usage. The input effectiveness matrix B from the linear model (3.15) of a vehicle is incorporated into the equality constraint to ensure that the solution vector u matches the desired control effect vector $\bar{u}$. A computationally simple solution u to (2.6) may be found using Lagrange multiplier techniques and is equivalent to the computation of a weighted pseudo-inverse of a reference model. Although this method is easily implemented and computationally efficient, it does not consider effector command limitations $u^- \leq u \leq u^+$ [7]. The constraints $u^- \leq u \leq u^+$ exist in practically all vehicle environments and can drastically limit the control authority of a vehicle. Because the simplest of the control allocation techniques do not take into account these important restrictions, more complex mathematical optimization methods similar to LS (i.e. Quadratic Programming, Linear Programming, Model Predictive Control, etc.) have been the most obvious solutions.

2.5.2 Linear Programming

Linear programming (LP) is an optimization technique involving the minimization of a linear cost function subject to linear equality and inequality constraints. By using a 1-norm cost criteria (sum of the absolute values of error), the CA problem (2.1) can be written in standard LP form as demonstrated by Enns [18]. The standard form of a linear program is

$$\begin{aligned} \min_{u} \quad & c^T u \\ \text{subject to} \quad & Bu = \bar{u} \\ & u^- \leq u \leq u^+ \end{aligned} \tag{2.7}$$

The most popular methods for solving an LP problem are simplex methods [25] and interior point methods [4]. The LP problem is based on the assumption that the solution lies within the space $\mathcal{U}$ defined by the inequality constraints. The idea behind the simplex method is to move along the boundary of the constraint set $\mathcal{U}$ from one extreme point to another until a minimum cost is found where as interior point methods approach a minimum from the interior of $\mathcal{U}$ rather than along the edges. The fact that LP solution methods such as the simplex algorithm are easy to code and implement make LP an attractive optimization technique. Hodel and Shtessel [22], and Callahan [11] proposed the use of multiple LPs to provide a fast online estimate of the AMS of an aerospace vehicle. Descriptions and evaluations of LP methods for control allocation are presented in [7], [18], [23], [26]. It is shown in [18] and [23] that the LP solution uses as few effectors as possible to match the desired effect $\bar{u}$. The drawback however, is that the use of exact inequality constraints requires more unknowns to be introduced into the problem making the iterative solution methods required to solve a LP computationally expensive.

2.5.3 Quadratic Programming

Quadratic Programming (QP), like LS, is based on a weighted 2-norm optimization. The general form of the QP problem is the same as that of the LP with the addition of a quadratic term in the objective function.

$$\begin{aligned} \min_{u} \quad & \tfrac{1}{2}u^T Q u + c^T u \\ \text{subject to} \quad & Bu = \bar{u} \\ & u^- \leq u \leq u^+ \end{aligned} \tag{2.8}$$

The Q and c terms are weights placed respectively on the quadratic and linear parts of the cost function. These weights can be chosen to favor certain effectors and/or weight the frequency content of the effector commands over time. The incorporation of inequality constraints ensures that the set of commands u will always be inside the attainable operating ranges of the effectors whether they be position limits, rate limits, or any other limiting factor associated with the effectors. The 2-norm cost criteria causes the feasible region $\mathcal{U}$ defined by the inequality constraints to be approximated by an ellipsoid. This approximation prevents the solution from being exact but provides the advantage of numerical simplicity. Unlike the LP solution, the QP solution will distribute the control effort among all available effectors. The disadvantage is that a QP problem is generally harder to solve than a LP problem. Many different methods have been devised for a fast online solution to a QP problem for CA purposes [7], [10], [20], [18].

2.5.4 Model Predictive Control

Model predictive control (MPC) has recently gained popularity in the vehicle control community due to advancements that significantly reduce the computational time required to solve this type of optimization. MPC uses a parameterized linear vehicle model to solve a finite horizon trajectory planning optimization. This problem can be converted to QP notation and solved using multi-parametric quadratic programming techniques similar to those described in [27]. The disadvantage of this technique is that it requires a much larger QP problem than the one described in Section 2.5.3 to be solved. MPC has been developed significantly in the chemical industry where plant dynamics allow for sufficient computational time. Recent advancements in MPC, however, allow for a faster online solution by shifting some of the computational burden off-line [5]. This has been proven to be an effective CA technique for rollover prevention of ground vehicles [12] but still possesses significant computational complexity along with a trade off between simplicity of online solution and memory to store off-line computed solutions.

Chapter 3

Vehicle Model

3.1 Nonlinear Vehicle Model

In this chapter, the dynamic equations of motion for a ground vehicle model are derived for a vehicle that possesses both front and rear steering capabilities as well as individual torque control of each wheel. This general model is easily adapted to simpler vehicles, e.g., vehicles with only front wheel steering, by the assignment of a constant control input of zero to the appropriate (unavailable) control effectors. The free body diagram (FBD) for the vehicle under consideration is shown in Figure 3.1. A right handed coordinate system is used for the derivation of the equations of motion. The body fixed reference frame is labeled on the FBD in Figure 3.1 with its origin at the vehicle's center of gravity. The z axis is pointing down toward the ground, the x axis is

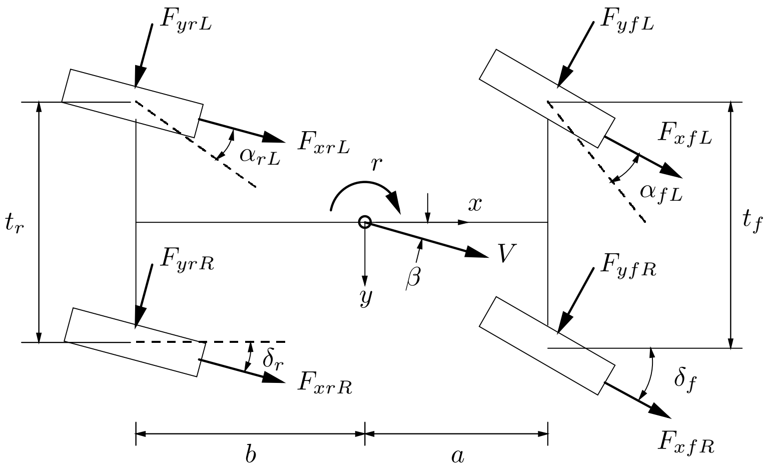


Figure 3.1: Free body diagram of vehicle model

pointing toward the front of the vehicle, and the y axis is pointing out to the vehicle's right side. In the figure, r is the yaw rate, $V = \begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix}^T$ is the velocity vector acting at the vehicle's center of gravity, β represents sideslip angle, δ is the steering angle, t is the track width, and a and b are distances from the vehicle center of gravity to the front and rear axles, respectively. Subscripts f, r, R, L denote front, rear, right, and left sides of the vehicle, respectively. For clarity, only the left slip angles (α_{fL} and α_{rL}) and the right steer angles (δ_{fR} and δ_{rR}) are shown on the FBD. This model assumes that left and right steer angles are the same for front and rear axles. Note that all forces F and slip angles α are drawn in the positive direction such that lateral force $F_y = -C_\alpha \alpha$ where C_α represents tire cornering stiffness.

Newton's laws of motion are applied to the FBD (Fig. 3.1). The sum of the moments about the vehicle's center of gravity is written as

$$\begin{aligned} \sum M_{cg} = I_z\ddot{\psi} = \quad & a(F_{yfL}\cos\delta_f + F_{yfR}\cos\delta_f - F_{xfL}\sin\delta_f - F_{xfR}\sin\delta_f) \\ & -\tfrac{t_f}{2}(F_{yfL}\sin\delta_f - F_{yfR}\sin\delta_f - F_{xfL}\cos\delta_f + F_{xfR}\cos\delta_f) \\ & -b(F_{yrL}\cos\delta_r + F_{yrR}\cos\delta_r + F_{xrL}\sin\delta_r + F_{xrR}\sin\delta_r) \\ & -\tfrac{t_r}{2}(F_{yrL}\sin\delta_r - F_{yrR}\sin\delta_r - F_{xrL}\cos\delta_r + F_{xrR}\cos\delta_r) \end{aligned} \tag{3.1}$$

where I_z is the moment of inertia about the yaw axis. The solution of (3.1) for $\ddot{\psi}$ yields

$$\begin{aligned} \ddot{\psi} = \dot{r} = \quad & [a((F_{yfL} + F_{yfR})\cos\delta_f - (F_{xfL} + F_{xfR})\sin\delta_f) \\ & -\tfrac{t_f}{2}((F_{yfL} - F_{yfR})\sin\delta_f - (F_{xfL} - F_{xfR})\cos\delta_f) \\ & -b((F_{yrL} + F_{yrR})\cos\delta_r + (F_{xrL} + F_{xrR})\sin\delta_r) \\ & -\tfrac{t_r}{2}((F_{yrL} - F_{yrR})\sin\delta_r - (F_{xrL} - F_{xrR})\cos\delta_r)]/I_z \end{aligned} \tag{3.2}$$

which describes the nonlinear yaw dynamics of the vehicle.

The sideslip angle β is the angle between the vehicle's actual velocity vector V and its longitudinal velocity component V_x. Therefore, the vehicle fixed velocities and corresponding accelerations due to sideslip are

$$\begin{aligned} V_x &= V\cos\beta \qquad \dot{V}_x = \dot{V}\cos\beta - V\dot{\beta}\sin\beta \\ V_y &= V\sin\beta \qquad \dot{V}_y = \dot{V}\sin\beta + V\dot{\beta}\cos\beta \end{aligned} \tag{3.3}$$

The effects of yaw rate are then included to give the complete expressions for acceleration in the vehicle fixed reference frame

$$\ddot{x} = \dot{V}\cos\beta - V\dot{\beta}\sin\beta - rV\sin\beta \tag{3.4}$$

$$\ddot{y} = \dot{V}\sin\beta + V\dot{\beta}\cos\beta + rV\cos\beta \tag{3.5}$$

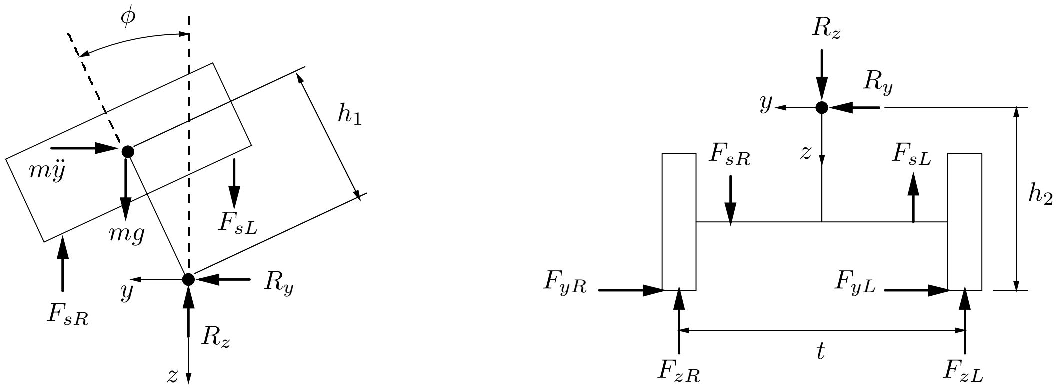
The summation of the forces in the y axis yields

$$\begin{aligned} \sum F_y = m\ddot{y} = \;& (F_{yfL} + F_{yfR})\cos\delta_f + (F_{xfL} + F_{xfR})\sin\delta_f \\ & +(F_{yrL} + F_{yrR})\cos\delta_r + (F_{xrL} + F_{xrR})\sin\delta_r \end{aligned} \tag{3.6}$$

Equation (3.5) is substituted into (3.6) and solved for $\dot{\beta}$ to obtain the equation of motion for sideslip

$$\begin{aligned} \dot{\beta} = \;& [(F_{yfL} + F_{yfR})\cos\delta_f + (F_{xfL} + F_{xfR})\sin\delta_f + (F_{yrL} + F_{yrR})\cos\delta_r \\ & +(F_{xrL} + F_{xrR})\sin\delta_r]/mV\cos\beta - \dot{V}\tan\beta/V - r \end{aligned} \tag{3.7}$$

The roll angle ϕ is the amount of rotation of the vehicle's unsprung mass about its roll axis x. In reality, the roll center of a vehicle with independent suspension is not stationary during transient maneuvers, but since the work done in this thesis does not rely heavily on extremely accurate roll dynamics a stationary roll center assumption is made to simplify modeling. The FBD of the front view of the vehicle is split into sprung and unsprung mass at the roll center as shown in Figure 3.2. A positive roll angle ϕ is shown, R_y and R_z are the reaction forces acting at the roll center, F_s represents the combined force of the suspension coils and shock absorbers acting on the vehicle body due to roll, h_1 is the distance between the center of mass of the sprung mass and the roll axis, and h_2 is the distance between the roll axis and the ground. Summation of the moments about the roll center of the sprung mass (Fig. 3.2(a)) yields



(a) sprung mass (b) unsprung mass

Figure 3.2: Free body diagram of vehicle's front view

$$\sum M_{rc} = I_x\ddot{\phi} = -B_\phi\dot{\phi} - K_\phi\phi - m\ddot{y}h_1\cos\phi + mgh_1\sin\phi \tag{3.8}$$

where I_x is the moment of inertia about the roll axis, and B_ϕ and K_ϕ represent the total roll damping and roll stiffness constants, respectively, due to the suspension forces F_s. The solution of Equation (3.5) for $\ddot{\phi}$ yields the equation for the nonlinear roll dynamics of the vehicle.

$$\begin{aligned}\ddot{\phi} = \; & [-B_\phi \dot{\phi} - K_\phi \phi - m(\dot{V}\sin\beta + V\dot{\beta}\cos\beta + rV\cos\beta)h_1\cos\phi \\ & + mgh_1\sin\phi]/I_x\end{aligned} \tag{3.9}$$

Expressions for the vertical forces at the tires are found by summing the moments about the roll center of the unsprung mass (Fig. 3.2(b))

$$\sum M_{rc} = 0 = \frac{t}{2}(F_{zL} - F_{zR}) + h_2(F_{yL} + F_{yR}) + B_\phi\dot{\phi} + K_\phi\phi \tag{3.10}$$

The summation of the forces in the y direction for both sprung and unsprung mass FBDs and equating the reaction forces at the roll center gives the following relation $-m\ddot{y} = -(F_{yL} + F_{yR})$ which is substituted into (3.10) to obtain

$$t\Delta F_z = \frac{t}{2}(F_{zR} - F_{zL}) = h_2 m\ddot{y} + B_\phi\dot{\phi} + K_\phi\phi \tag{3.11}$$

Through the substitution of the expression for $\ddot{y}$ (3.5) into the above equation (3.11), equations for the vertical load difference at each axle can now be written:

$$\begin{aligned}\Delta F_{zf} &= [B_{\phi f}\dot{\phi} + K_{\phi f}\phi + h_f m(\dot{V}\sin\beta + V\dot{\beta}\cos\beta + rV\cos\beta)]/t_f \\ \Delta F_{zr} &= [B_{\phi r}\dot{\phi} + K_{\phi r}\phi + h_r m(\dot{V}\sin\beta + V\dot{\beta}\cos\beta + rV\cos\beta)]/t_r\end{aligned} \tag{3.12}$$

Finally, the work in this thesis assumes the vehicle travels at a constant velocity. Therefore, longitudinal weight transfer is not included in the expressions for the vertical load at each tire.

$$\begin{aligned} F_{zrL} &= F_{zr} - \Delta F_{zr} \qquad & F_{zfL} &= F_{zf} - \Delta F_{zf} \\ F_{zrR} &= F_{zr} + \Delta F_{zr} \qquad & F_{zfR} &= F_{zf} + \Delta F_{zf} \end{aligned} \tag{3.13}$$

3.2 Nonlinear Tire Model

A Pacejka tire model [3] is used to model the behavior of the tires. This nonlinear tire model uses tire slip angle and vertical force to approximate the lateral force acting on the tire. Figure 3.3 below shows lateral force plotted against tire slip angle for several different vertical loads. The slip angle of the tire is calculated as $\alpha = \tan^{-1}(\frac{V_{y'}}{V_{x'}})$. From

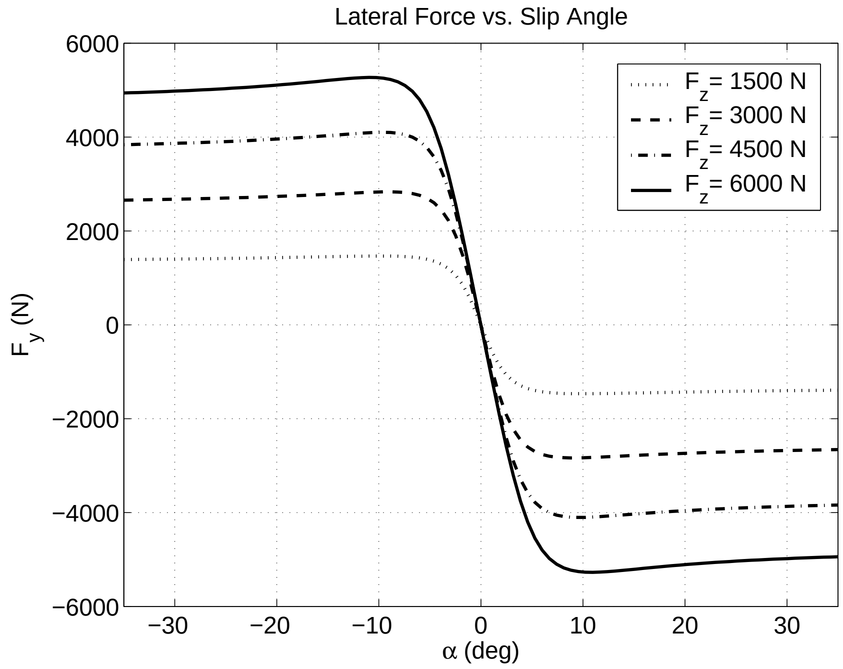


Figure 3.3: Pacejka tire model lateral force curves

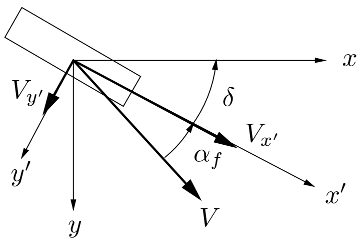


Figure 3.4: Free body diagram of tire.

Figures 3.1 and 3.2 the velocity components for each tire is deduced and the equations for tire slip angle are written as follows

$$\begin{aligned} \alpha_{rL} &= \tan^{-1}\left[\frac{V\sin\beta - rb}{V\cos\beta + r\frac{t_r}{2}}\right] - \delta_r & \qquad \alpha_{fL} &= \tan^{-1}\left[\frac{V\sin\beta + ra}{V\cos\beta + r\frac{t_f}{2}}\right] - \delta_f \\ \alpha_{rR} &= \tan^{-1}\left[\frac{V\sin\beta - rb}{V\cos\beta - r\frac{t_r}{2}}\right] - \delta_r & \qquad \alpha_{fR} &= \tan^{-1}\left[\frac{V\sin\beta + ra}{V\cos\beta - r\frac{t_f}{2}}\right] - \delta_f \end{aligned} \tag{3.14}$$

The maximum traction limit of each tire is assumed to be the product of the vertical load F_z and the coefficient of friction μ between the tire and the road, i.e., $F_t = F_z\mu$. The friction circle (or friction ellipse) concept [19] requires the vector total of the lateral and longitudinal tire forces to lie within this maximum friction limit of the tire $F_t > \sqrt{F_y^2 + F_x^2}$. A plot of the friction circle is shown in Figure 3.5, from which one may observe how the acceleration and braking force limits depend upon the amount of lateral acceleration being experienced by the tire. Assumptions about the friction coefficient and the role the friction circle plays in the CA problem are discussed in Section 5.1.

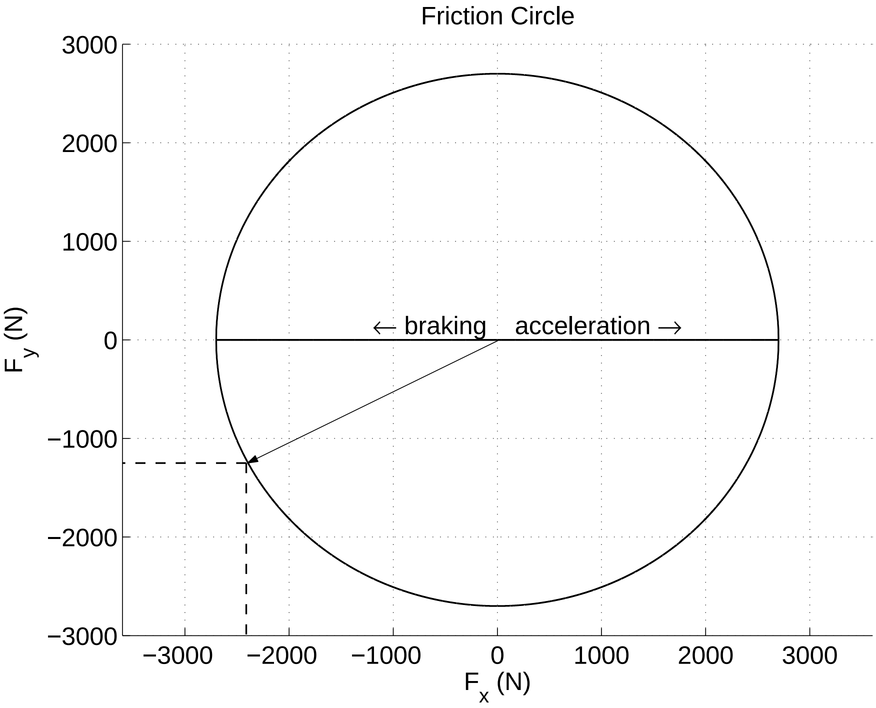


Figure 3.5: Friction Circle for $F_z = 3000N$

3.3 Linear Vehicle Model

The proposed control law relies on a linear approximation of the vehicle model to produce meaningful commands. The two states being controlled are yaw rate and sideslip. Therefore, Equations (3.2) and (3.7) are linearized around a constant velocity. Through the use of small angle approximations, the linear vehicle model takes the state space form

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_0}{mV} & -\frac{C_1}{mV^2} - 1 \\ -\frac{C_1}{I_z} & -\frac{C_2}{VI_z} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + Bu \tag{3.15}$$

$$C_0 = C_{\alpha f} + C_{\alpha r}$$

$$C_1 = aC_{\alpha f} - bC_{\alpha r}$$

$$C_2 = a^2 C_{\alpha f} + b^2 C_{\alpha r}$$

where $C_{\alpha f}$ and $C_{\alpha r}$ are constants representing the front and rear linearized tire cornering stiffness values (per axle), B represents the input effectiveness matrix, and u is the vector of corresponding effector commands. Equation (3.15) is also known as the linearized "bicycle model" [14] because the effects of lateral and longitudinal weight transfer are neglected. This model also assumes that steering and tire slip angles are the same for right and left sides.

The control law proposed in this work is tested for several different vehicle effector suites that require different input effectiveness matrices. The vehicle configurations and their corresponding linear input effectiveness matrices are as follows:

1) effector suite: front steer, front differential braking, and rear differential braking.

$$Bu = \begin{bmatrix} \frac{C_{\alpha f}}{mV} & 0 & 0 \\ a\frac{C_{\alpha f}}{I_z} & \frac{t_f}{2I_z} & \frac{t_r}{2I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ \Delta F_{xf} \\ \Delta F_{xr} \end{bmatrix} \tag{3.16}$$

2) effector suite: front steer, rear steer, front differential braking, and rear differential braking.

$$Bu = \begin{bmatrix} \frac{C_{\alpha f}}{mV} & \frac{C_{\alpha r}}{mV} & 0 & 0 \\ a\frac{C_{\alpha f}}{I_z} & -b\frac{C_{\alpha r}}{I_z} & \frac{t_f}{2I_z} & \frac{t_r}{2I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ \delta_r \\ \Delta F_{xf} \\ \Delta F_{xr} \end{bmatrix} \tag{3.17}$$

3) effector suite: front steer, rear steer, and individual torque control of each wheel. The individual torque (at the wheel) is represented by a road/tire interface force.

$$Bu = \begin{bmatrix} \frac{C_{\alpha f}}{mV} & \frac{C_{\alpha r}}{mV} & 0 & 0 & 0 & 0 \\ a\frac{C_{\alpha f}}{I_z} & -b\frac{C_{\alpha r}}{I_z} & -\frac{t_f}{2I_z} & \frac{t_f}{2I_z} & -\frac{t_r}{2I_z} & \frac{t_r}{2I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ \delta_r \\ F_{xfR} \\ F_{xfL} \\ F_{xrR} \\ F_{xrL} \end{bmatrix} \tag{3.18}$$

3.4 Model Verification

The accuracy of the linear vehicle model is confirmed by comparing the step responses to those of the nonlinear model derived in Sections 3.1 and 3.2. The eigenvalues of the linear vehicle model (3.15) at 55mph are $\lambda_\beta = -5.8218 + 3.7249i$ and $\lambda_r = -5.8218 - 3.7249i$. For the maneuvers considered in this research, the steering inputs δ are never required to exceed an angle of 1.5 degrees, therefore a step input of 1 degree is used to validate the front and rear steering responses. The state outputs for inputs of $\delta_f = 1^\circ$ and $\delta_r = 1^\circ$ are shown in Figures 3.6 and 3.7 respectively.

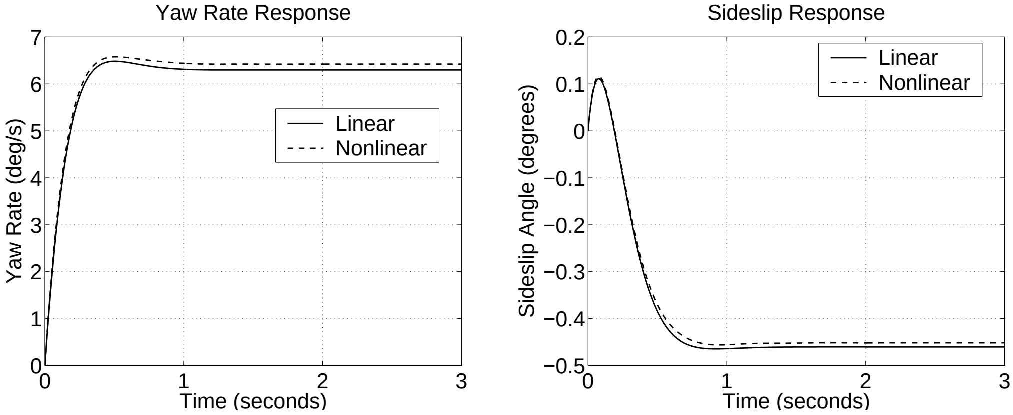


Figure 3.6: Step Response at 55mph for Front Steering input of $\delta_f = 1^\circ$

Observe, from the linear vehicle models (3.16)-(3.18), that the differential braking inputs for models (3.16) and (3.17) have the same effect as the individual wheel torque inputs for model (3.18). Therefore, an accurate differential braking step response is sufficient for verification of the individual wheel torque inputs. Unlike the steering inputs δ, the differential braking/wheel torque inputs ΔF_x are expected to perform at their limits, therefore a step input of $2000N$ is appropriate for model verification. The state outputs for an input of $\Delta F_x f = 2000N$ is shown in Figure 3.8.

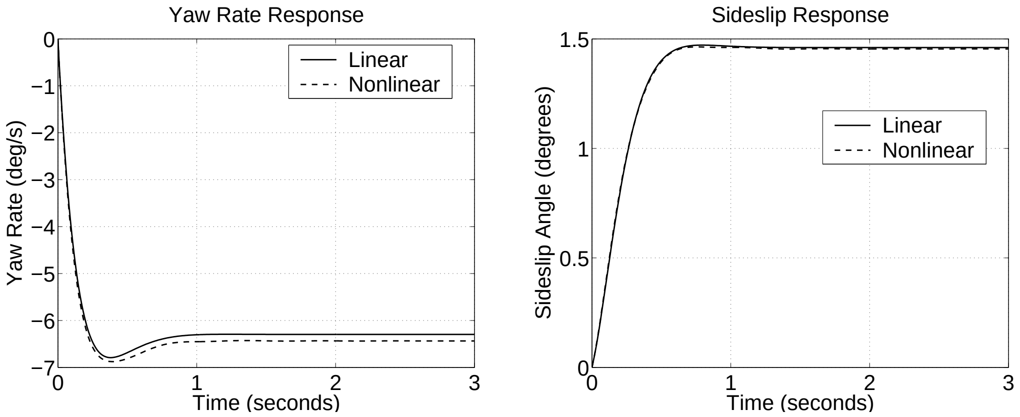


Figure 3.7: Step Response at 55mph for Rear Steering input of $\delta_r = 1°$

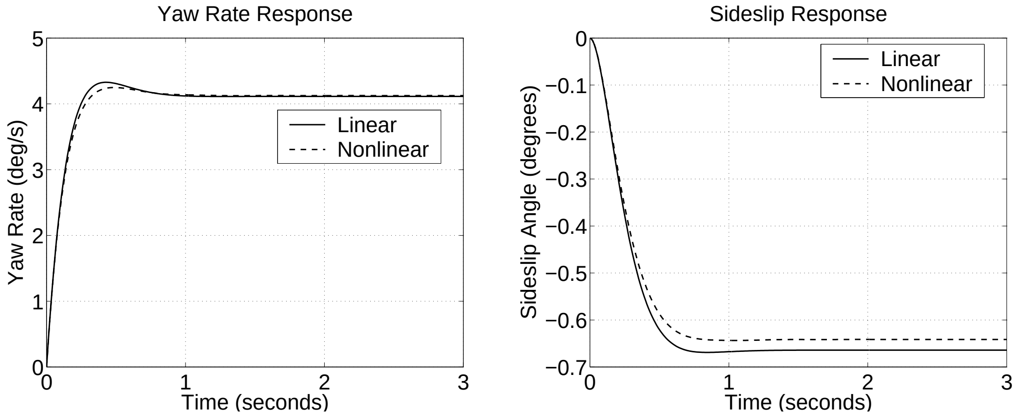


Figure 3.8: Step Response at 55mph for Front Differential Braking input of $\Delta F_{xf} = 2000N$

The yaw rate and sideslip responses for the linear and nonlinear vehicle models follow each other closely for the step inputs examined. Therefore, the linear model can be confirmed to be an accurate representation of the nonlinear model for these conditions.

CHAPTER 4

CONTROLLER DESIGN

In this chapter the design procedure for a quadratic programming (QP) based control allocation (CA) law for a ground vehicle is presented. Recall from Chapter 3 that the linearized model of the vehicle takes the form

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_0}{mV} & -\frac{C_1}{mV^2} - 1 \\ -\frac{C_1}{I_z} & -\frac{C_2}{VI_z} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + Bu \tag{4.1}$$

where the input effectiveness matrix B has 3, 4, or 6 columns depending on the available effector suite. The details of the 3, 4, and 6 input vehicle models will be presented in Chapter 5. Observe that, for a given desired derivative vector $\dot{x}_{des} = \begin{bmatrix} \dot{\beta} & \dot{r} \end{bmatrix}^T$, there is more than one possible corresponding input vector u. Therefore, the control law design is reduced to the selection of a virtual input $\bar{u}$ that is chosen to achieve a desired derivative vector $\dot{x}_{des} = \begin{bmatrix} \dot{\beta} & \dot{r} \end{bmatrix}^T$. Then a QP-based CA algorithm is used to compute corresponding input commands that, when possible, match the desired derivative vector while obeying effector constraints. The design of the proposed QP-based CA algorithm is predicated on the assumption that measurements or estimates of the vehicle's yaw rate r and sideslip angle β are available. These values may be estimated by traditional Kalman filtering [34] or by fusion of information from Global Positioning Systems (GPS) and inertial sensors [6, 31]. If available estimates are not sufficiently accurate, the QP law may require the use of a robust nominal CA law such as that presented in [21]. Performance of the proposed controller in the presence of noisy state estimates is discussed further in Chapter 5.

4.1 Control Law

Linear quadratic regulator (LQR) gains designed for a modified linear vehicle model are used to produce the desired control effect, $\bar{u}$. The modified system assumes a perfect input matrix $B = I_{2x2}$ so that two ideal control effects are produced which represent the yaw rate and sideslip objectives. The model is augmented with a yaw rate integrator in order to improve asymptotic tracking of the desired yaw angle and to de-emphasize sideslip angle β behavior in the control law optimization. The resulting state space model used for the LQR control design is

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \\ r \end{bmatrix} = \begin{bmatrix} -\frac{C_0}{mV} & -\frac{C_1}{mV^2} - 1 & 0 \\ -\frac{C_1}{I_z} & -\frac{C_2}{VI_z} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{u}_\beta \\ \bar{u}_r \end{bmatrix} \tag{4.2}$$

Since any practical vehicle control law will be implemented in discrete time, a linear quadratic regulator (LQR) design is performed based on a discretized model of (4.2) with a sampling time of $T = 0.01s$. The desired tracking control effect $\bar{u}$ is found using state error feedback

$$\begin{bmatrix} \bar{u}_\beta \\ \bar{u}_r \end{bmatrix} = -K \begin{bmatrix} \beta \\ \dot{e}_r \\ e_r \end{bmatrix} \tag{4.3}$$

where $\dot{e}_r = r - r_{des}$ and e_r is simply $\int \dot{e}_r$. The controller gains K serve to drive the vector of signals $\begin{bmatrix} \beta & \dot{e}_r & e_r \end{bmatrix}^T$ to zero, and $\begin{bmatrix} \bar{u}_\beta & \bar{u}_r \end{bmatrix}^T$ are the resulting sideslip and yaw rate control effects the control allocator must try to match.

4.2 Command Generation

The task of a control allocation law is to map the ideal command $\bar{u}$ described above to a set of physical vehicle actuator commands, taking into account actuator failures and physical limitations such as position and/or rate limits. This task is addressed through the use of a quadratic programming problem. Recall that the general form of the quadratic programming problem is

$$\begin{aligned} \min_{u} \quad & \tfrac{1}{2}u^T Q u + c^T u \\ \text{subject to} \quad & Bu = \bar{u} \\ & u^- \leq u \leq u^+ \end{aligned} \tag{4.4}$$

where the input effectiveness matrix B from the linear model (4.1) of the vehicle is incorporated into the equality constraint to ensure that the solution vector u matches the desired control effect vector $\bar{u}$.

The nature of the input matrix B in model (4.1) implies that the steering inputs have the most control authority over both states while the differential braking and wheel torque inputs can only affect vehicle sideslip indirectly through the coupling between the two states. Due to this coupling, the QP optimization may produce effector commands that work against each other in an effort to satisfy both objectives. An effector failure would result in an even greater compromise between objectives yielding larger tracking errors. Therefore, a virtual input ν was added to the original linear vehicle model (4.1) to address this problem. The new state space model is

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_0}{mV} & -\frac{C_1}{mV^2} - 1 \\ -\frac{C_1}{I_z} & -\frac{C_2}{VI_z} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} B & \vdots & \begin{matrix} \kappa_\beta \\ \kappa_r \end{matrix} \end{bmatrix} \begin{bmatrix} u \\ \dots \\ \nu \end{bmatrix} \tag{4.5}$$

The effectiveness of ν through the variables κ_β and κ_r is selected such that the DC gain from r to ν is zero. This relaxes the equality constraint on the sideslip and removes some of the control responsibility from the steering angle without affecting the yaw rate. In other words, the QP algorithm will be allowed to use the virtual effector ν to match the infeasible portion of the sideslip objective $\bar{u}_\beta$ while the main goal of tracking the yaw rate is left to the real effectors.

The first step in the selection of κ_β and κ_r is to find the transfer function from r to ν. The virtual input ν affects the states by the following relations

$$\dot{\beta} = -\frac{C_0}{mV}\beta - \left(\frac{C_1}{mV^2} + 1\right) r + \kappa_\beta \nu \tag{4.6}$$

$$\dot{r} = -\frac{C_1}{I_z}\beta - \frac{C_2}{VI_z} r + \kappa_r \nu \tag{4.7}$$

The frequency domain representations of these equations are obtained from the Laplace transform

$$B(s)s = -\frac{C_0}{mV}B(s) - \left(\frac{C_1}{mV^2} + 1\right) R(s) + \kappa_\beta N(s) \tag{4.8}$$

$$R(s)s = -\frac{C_1}{I_z}B(s) - \frac{C_2}{VI_z}R(s) + \kappa_r N(s) \tag{4.9}$$

In order to account for the coupling effect of ν on r, Equation (4.8) is solved for $B(s)$

$$B(s) = \frac{\kappa_\beta N(s) - \left(\frac{C_1}{mV^2} + 1\right) R(s)}{s + \frac{C_0}{mV}} \tag{4.10}$$

Equation (4.10) is then substituted into (4.9) to yield an expression solely in terms of $R(s)$ and $N(s)$

$$R(s)s = -\frac{C_1}{I_z}\left(\frac{\kappa_\beta N(s) - \left(\frac{C_1}{mV^2} + 1\right) R(s)}{s + \frac{C_0}{mV}}\right) - \frac{C_2}{VI_z}R(s) + \kappa_r N(s) \tag{4.11}$$

The transfer function representing the complete relationship between ν and r can now be calculated from (4.11).

$$\frac{R(s)}{N(s)} = \frac{\kappa_r m I_z V^2 s + \kappa_r I_z V C_0 - \kappa_\beta C_1 m V^2}{m I_z V^2 s^2 + I_z V C_0 s - C_1 m V^2 - C_1^2} \tag{4.12}$$

Now, the appropriate values for κ_β and κ_r may be selected. The transfer function (4.12) will have zero DC gain under the following condition

$$\kappa_r I_z V C_0 - \kappa_\beta C_1 m V^2 = 0 \tag{4.13}$$

For simplicity, a value of 1 is assigned to κ_β. The corresponding value of κ_r that satisfies (4.13) is then

$$\kappa_r = \frac{C_1 m V}{I_z C_0} \tag{4.14}$$

The resulting modified constraints on the QP problem (4.4) are

$$\begin{bmatrix} B & \vdots & \begin{matrix} 1 \\ \frac{C_1 mV}{I_z C_0} \end{matrix} \end{bmatrix} \begin{bmatrix} u \\ \nu \end{bmatrix} = \begin{bmatrix} \bar{u}_\beta \\ \bar{u}_r \end{bmatrix}$$

$$\begin{bmatrix} u^- \\ \nu^- \end{bmatrix} \leq \begin{bmatrix} u \\ \nu \end{bmatrix} \leq \begin{bmatrix} u^+ \\ \nu^+ \end{bmatrix} \tag{4.15}$$

In order to prevent an infeasible QP problem caused by tight inequality constraints on ν, sufficiently large inequality constraints were chosen so that they never become active. However, since the virtual effector ν has no physical meaning, it should generally be assigned a large quadratic penalty to reduce its use and leave most of the control responsibility up to the real effectors. The magnitude of the quadratic penalty Q_ν on virtual effector ν does have a significant affect on the control action making it an important design parameter. The selection of an appropriate value for Q_ν is discussed further in Chapter 5.

Chapter 5

Simulation and Results

Results of the control and control allocation laws tested in simulation on several vehicles are presented in this chapter. The simulation uses the full nonlinear vehicle model (3.1)-(3.13) developed in Chapter 3 with the nonlinear tire model (3.14). The CA laws were tested on vehicles with 3, 4, and 6 input capabilities under nominal operation as well as a failure scenario. Recall from Chapter 4 that the control action is significantly affected by the choice of the quadratic penalty Q_ν placed on the virtual effector ν. To offer insight as to how the over all performance is affected, each case was tested for two different weights; $Q_\nu = 1e3$ and $Q_\nu = 1e7$. Simulations were performed in the Matlab/Simulink simulation environment. A single sine wave oscillation corresponding to a double lane change maneuver was computed off-line and used as the desired yaw rate trajectory. For each scenario the vehicle was simulated at constant velocities of 45, 55, and 65mph. This assumes a separate controller regulates fuel flow to the engine during differential braking commands to maintain a constant speed.

5.1 Effector Limitations

A constant position limit of $\pm 0.5\text{rad}(\approx 30^\circ)$ was placed on the steering angle of the front and rear tires. The limits placed on the differential braking commands were calculated online in accordance to the friction circle concept discussed in Section 3.2. In reality, the peak friction coefficient μ between the tire and road is a function of the veritcal load F_z. However this work assumes a constant value of $\mu = 0.8$ for simplicity. The maximum tractional force for each tire is calculated as $F_t = \mu F_z$. Then the maximum braking/acceleration force that can be applied without producing slippage is $F_x^{\pm} = \pm\sqrt{F_t^2 - F_y^2}$. Researchers have shown that online estimates of vertical force and road friction coefficient are possible through the use of extended Kalman-Bucy filtering and Bayesian hypothesis selection [29]. Alternatively, the underestimation of road friction coefficient has been shown to provide conservative approximations of maximum braking force available to the vehicle [1].

5.2 Effector Failures

Any reduction in the effectiveness of an effector is considered a failure. Failures may be caused by hardware malfunction or by damage inflicted upon the vehicle. Regardless of the cause, an effector failure always results in decreased maneuverability jeopardizing the safety of the vehicle and its passengers. During a detected failure situation, control allocation has the ability to reconfigure the vehicle's remaining effectors in an effort to maintain the least amount of tracking error possible.

In order to demonstrate the reconfigurable abilities of the controller, a steering failure is examined in which the steering angle of the front tires is stuck at its current position. For example, this could represent the failure of the drive motor on a steer-by-wire vehicle. The front steering input is available to all of the vehicle models and provides more control authority over yaw rate and sideslip than any of the other effectors. Therefore, a comparison of effector suites can be made in addition to an evaluation of the controller's performance during a steering failure.

Failures were implemented by scaling the columns of the input effectiveness matrix B from Equation (3.15) corresponding to the failed effector. Then the corresponding inequality constraints were set equal to the previous command. Actual implementation would require the controller to be alerted of a failure through an on-board vehicle diagnostic system such as the one demonstrated in [28]. Also, online calculation of input effectiveness matrix B would be necessary due to its dependence on both failure mode and velocity.

5.3 The Effects of Noise

The job of the QP based CA routine is to match the virtual command $\bar{u}$ obtained from the LQR control law, therefore any noise in the virtual command will be tracked by the resulting effector commands. Perfect state measurements or estimates are desirable in nearly all cases of closed loop control. In reality however, sensor noise, process noise, or other disturbances sometimes prevent sufficiently clean signals from being available. Fortunately, much research is devoted to this subject and new techniques are constantly being developed which allow for cleaner and more accurate estimates. In [2], the authors propose the use of GPS measurements and a model based Kalman filter to estimate vehicle yaw rate and sideslip angle. Provided with a perfect vehicle model and known process noise covariance, the state estimation error for this method at a 50Hz update rate is predicted to be 0.2012 deg/s for yaw rate and 0.1101 deg for sideslip. These values were used as guidelines for implementing appropriate amounts of noise in the simulation. This was done for the sole purpose of demonstrating the proposed controller's performance under realistic operating conditions. Figure 5.1 shows the resulting commands and corresponding tracking performance for the 3-input vehicle model in the presence of noisy state estimates. The vehicle dynamics filter out most of the noisy inputs and the yaw rate is still tracked with very little error. The noisy input commands are still unfavorable due to wear and tear on the hardware and/or ride discomfort. A QP based CA routine has the ability to address this issue by weighting the frequency content of the

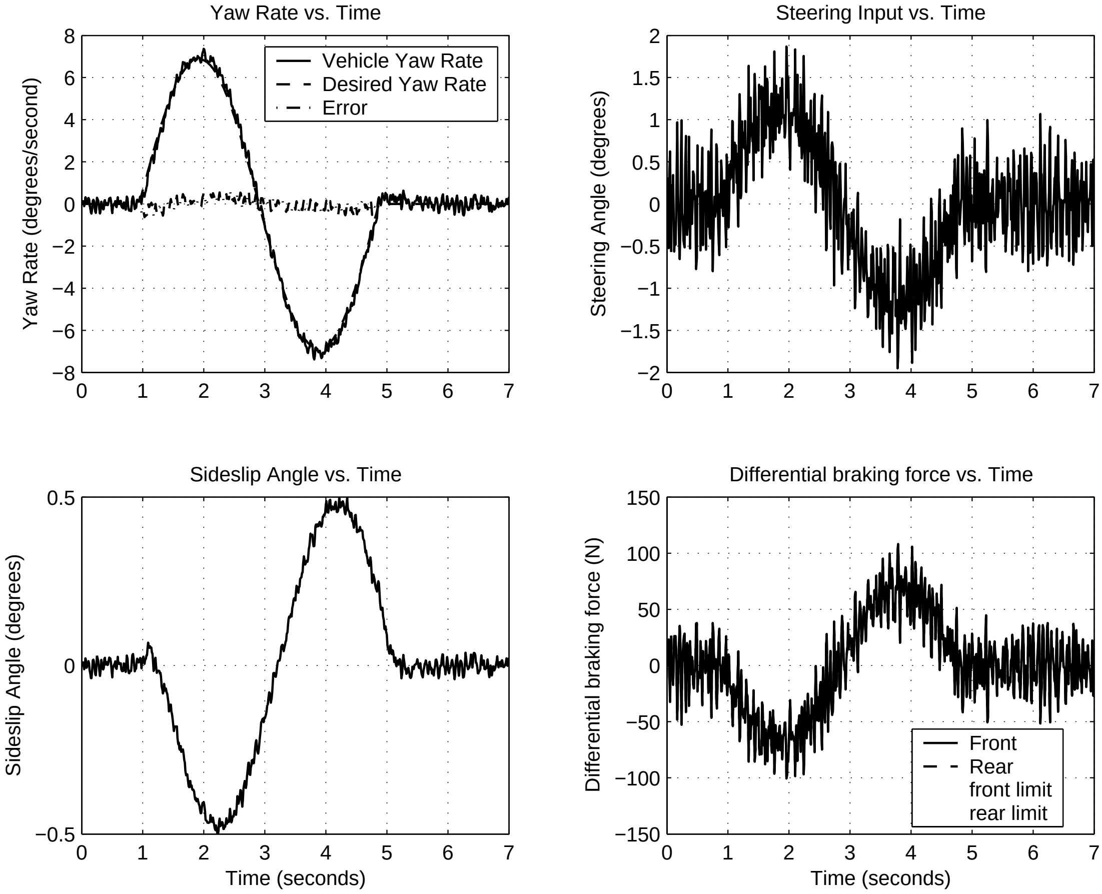


Figure 5.1: Controller performance in the presence of noise. 3-input vehicle model at 55mph, $Q_\nu = 1e7$.

output commands. The frequency weighted sign preserving formulation of the QP problem proposed in [33] has been shown to successfully cut out some of the high frequency content of the effector commands.

5.4 Results

The main focus of this research is the development and evaluation of a control allocation scheme for ground vehicle applications and therefore neither disturbances nor sensor noise were simulated in the following experiments. The results presented here are for relative comparison only and are not a measure of true performance (qualitative not quantitative). The root mean squared (RMS) errors of yaw rate and sideslip tracking are used to compare algorithms and assess the effectiveness of the controller in each scenario.

Vehicle Model - 3 Inputs

The 3-input vehicle model has front steering and differential braking capabilities for the front and rear axles.

$$Bu = \begin{bmatrix} \frac{C_{\alpha f}}{mV} & 0 & 0 \\ a\frac{C_{\alpha f}}{I_z} & \frac{t_f}{2I_z} & \frac{t_r}{2I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ \Delta F_{xf} \\ \Delta F_{xr} \end{bmatrix} \tag{5.1}$$

Vehicle Model - 4 Inputs

The 4-input vehicle model possesses the same effector suite as the 3-input model with the addition of rear steering δ_r.

$$Bu = \begin{bmatrix} \frac{C_{\alpha f}}{mV} & \frac{C_{\alpha r}}{mV} & 0 & 0 \\ a\frac{C_{\alpha f}}{I_z} & -b\frac{C_{\alpha r}}{I_z} & \frac{t_f}{2I_z} & \frac{t_r}{2I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ \delta_r \\ \Delta F_{xf} \\ \Delta F_{xr} \end{bmatrix} \tag{5.2}$$

The addition of rear steering changes the control problem such that the yaw rate no longer dictates the trajectory. A vehicle with four wheel steering can change lanes without inducing any yaw by steering all four tires in the same direction. For this reason, rear steering has made its way into production automobiles under claims of increased high speed stability [9].

Vehicle Model - 6 Inputs

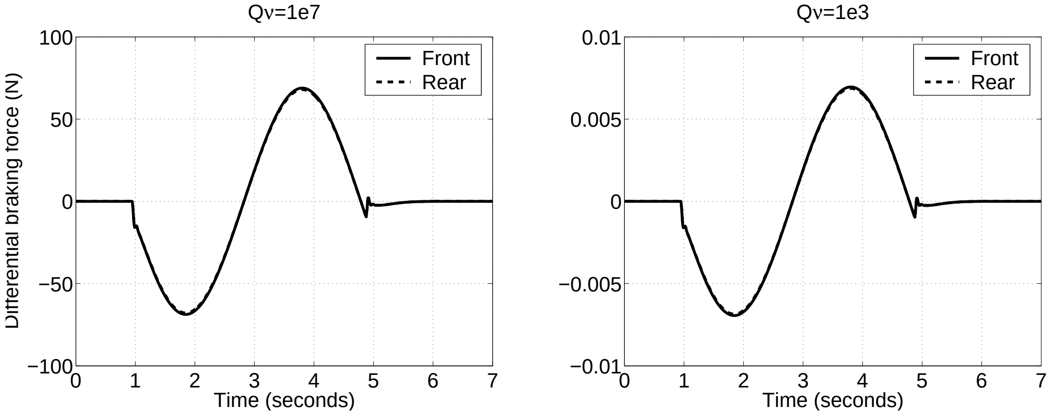
The 6-input vehicle model has front and rear steering capabilities along with individual torque control of each wheel.

$$Bu = \begin{bmatrix} \frac{C_{\alpha f}}{mV} & \frac{C_{\alpha r}}{mV} & 0 & 0 & 0 & 0 \\ a\frac{C_{\alpha f}}{I_z} & -b\frac{C_{\alpha r}}{I_z} & -\frac{t_f}{2I_z} & \frac{t_f}{2I_z} & -\frac{t_r}{2I_z} & \frac{t_r}{2I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ \delta_r \\ F_{xfR} \\ F_{xfL} \\ F_{xrR} \\ F_{xrL} \end{bmatrix} \tag{5.3}$$

This model resembles a vehicle with separate electric motors at each wheel that can provide independent braking or acceleration forces generated by torques at each wheel. Differential braking can only provide a yaw torque by providing a braking force to one side of the vehicle. Individual torque control however, has the ability to induce the same amount of yaw torque by applying less braking force to one side of the vehicle while accelerating the wheels on the opposite side by an equal amount. Therefore, individual torque control of each wheel provides much more maneuverability than differential braking alone and can maintain vehicle speed.

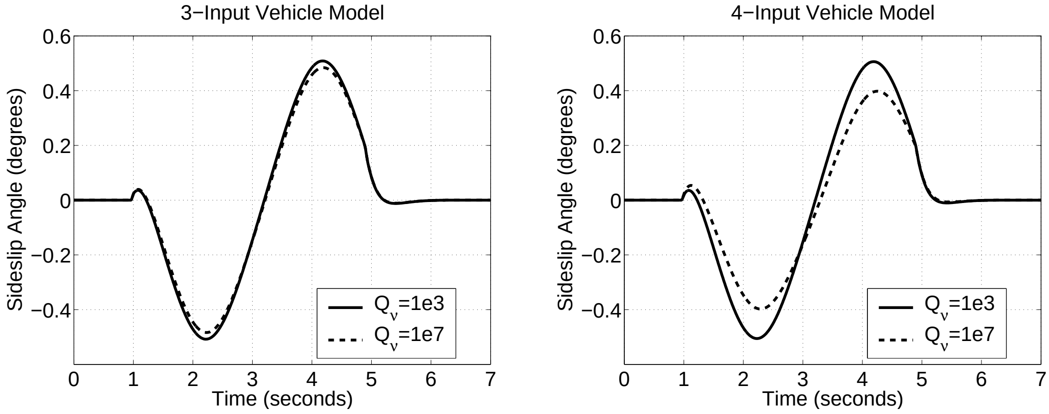
5.4.1 Nominal Case

The front steering angle has the most control authority in all three of the considered vehicle models. Therefore, it is the primary effector used to minimize the main objective of tracking the yaw rate. Due to the coupling of the yaw rate and sideslip equations of motion (Equations (3.2) and (3.7) respectively), the QP optimization results in commands for all other available effectors that oppose the front steering command in an effort to keep the sideslip angle small. The importance of the sideslip objective can be manipulated by the choice of the quadratic penalty on the virtual effector. This is shown by the differential braking commands for the 3-input vehicle model. A quadratic penalty of $Q_\nu = 1e3$ results in minimal use of the differential braking commands as the controller relies heavily on the virtual effector to minimize sideslip. Less braking force opposing the steering angle allows for better yaw rate tracking at lower speeds. The larger penalty of $Q_\nu = 1e7$ makes the sideslip objective more of a priority for the real effectors. Therefore, $Q_\nu = 1e7$ provides slightly better sideslip minimization for all vehicle models under nominal operation. The difference in magnitude of the braking commands for the 3-input vehicle is shown in Figure 5.2 with plots of the commanded braking force for different values of Q_ν. The corresponding difference in sideslip tracking appears to be small for the 3-input vehicle model (Figure 5.3a) while the difference for the 4-input model is more significant (Figure 5.3b).



(a) (b)

Figure 5.2: Nominal case braking commands at 55mph for different values of Q_ν



(a)
(b)

Figure 5.3: Nominal case sideslip angle tracking at 55mph.

The results for each of the mentioned virtual effector weights $Q_\nu = 1e3$ and $Q_\nu = 1e7$ are included in the tabulated results at the end of the chapter. For the sake of consistency all of the figures for the nominal case are results from simulations run with a quadratic penalty of $Q_\nu = 1e7$ on the virtual effector.

The 3-input vehicle model has the least amount of effectors to oppose the front steering input and is able to provide lower RMS yaw rate tracking error than the other two vehicle models. Similar to the front steering input, rear steering also has a direct effect on both states. The linear model (5.2) reveals that the front and rear steering inputs affect sideslip in the same manner but have an opposite effect on the yaw rate. The resulting commands will steer all tires in the same direction. The front wheels must be commanded a larger angle than the rear however, to produce a yaw rate. These commands will result in a lane change maneuver but the yaw rate tracking will be poor. Figure 5.4 shows the large rear steering command and poor yaw rate tracking that results when a nominal weight of $Q_{\delta_r} = 1$ is used. Since the rear steering input has the ability to directly affect both states, the rear steering commands are unaffected by the choice of the virtual

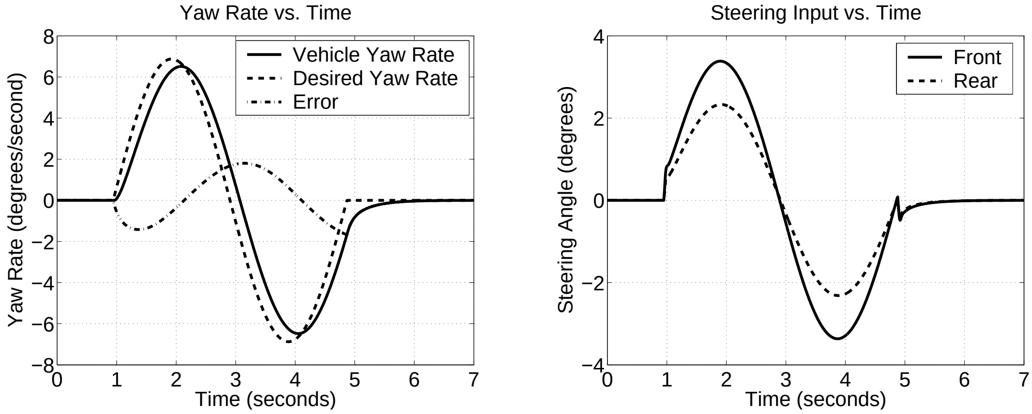


Figure 5.4: Poor yaw rate tracking due to rear steering command. $Q_{\delta_r} = 1$

effector penalty Q_ν. A large quadratic weight of $1e10$ was therefore assigned to δ_r to reduce the undesirable commands and ensure good yaw rate tracking. Figure 5.6 shows the rear steering command for the new weighting matrix Q along with the improved yaw rate tracking. The design goal of the controller is to track a desired yaw rate, therefore all of the results generated for this thesis utilize a quadratic penalty of $1e10$ for the rear steering input. The 6-input vehicle model provides the same performance as the 4-input model for the nominal case when $Q_\nu = 1e3$.

5.4.2 Failure Case

In the event of a front steering failure, the controller must cancel the effect caused by the front wheels in addition to maintaining good yaw rate tracking. This type of failure is difficult to recover from, even for the double lane change maneuver considered. The results in Table 5.2 show that a lower quadratic penalty of $Q_\nu = 1e3$ gave the lowest tracking error for both yaw rate and sideslip for all of the vehicle models.

Figure 5.8 displays the controller's ability to maintain best possible yaw rate tracking through reconfiguration while respecting the limitations of the remaining effectors. Both of the differential braking commands hit their limits but are still unable to provide enough force to track the desired yaw rate. The 4-input vehicle model is able to provide good yaw rate tracking even during a serious failure. Figure 5.9 shows that the rear steering and the differential braking are necessary to make up for the jammed front wheels. The 6-input vehicle model provides slightly better performance than the 4-input model in the failure case due to its ability to keep the wheel torque commands further from their limits than the differential braking.

Table 5.1: Nominal Case - no failures

		3-Input		**4-Input**		**6-Input**	
Vel (mph)	Q_ν ($1e*$)	RMS r (deg/s)	RMS β (deg)	RMS r (deg/s)	RMS β (deg)	RMS r (deg/s)	RMS β (deg)
45	3	0.0778	0.0222	0.1296	0.0220	0.1296	0.0220
45	7	0.0837	0.0170	0.1678	0.0081	0.1747	0.0085
55	3	0.0381	0.1145	0.0485	0.1135	0.0485	0.1135
55	7	0.0376	0.1039	0.0587	0.0700	0.0612	0.0630
65	3	0.1452	0.2826	0.1111	0.2781	0.1111	0.2781
65	7	0.1375	0.2685	0.0921	0.2203	0.0885	0.2093

Table 5.2: Front Steering Failure at 2.25s

		3-Input		**4-Input**		**6-Input**	
Vel (mph)	Q_ν ($1e*$)	RMS r (deg/s)	RMS β (deg)	RMS r (deg/s)	RMS β (deg)	RMS r (deg/s)	RMS β (deg)
45	3	4.5795	0.5797	0.3920	1.4248	0.3867	1.2255
45	7	4.8792	0.5983	0.5167	1.6312	0.5449	1.4367
55	3	3.0645	0.7366	0.2744	1.5617	0.2645	1.3770
55	7	3.5522	0.7408	0.3232	1.6971	0.3295	1.5233
65	3	1.9753	0.9872	0.5794	1.8633	0.5418	1.6785
65	7	2.3252	0.9797	0.5869	1.9694	0.5553	1.7940

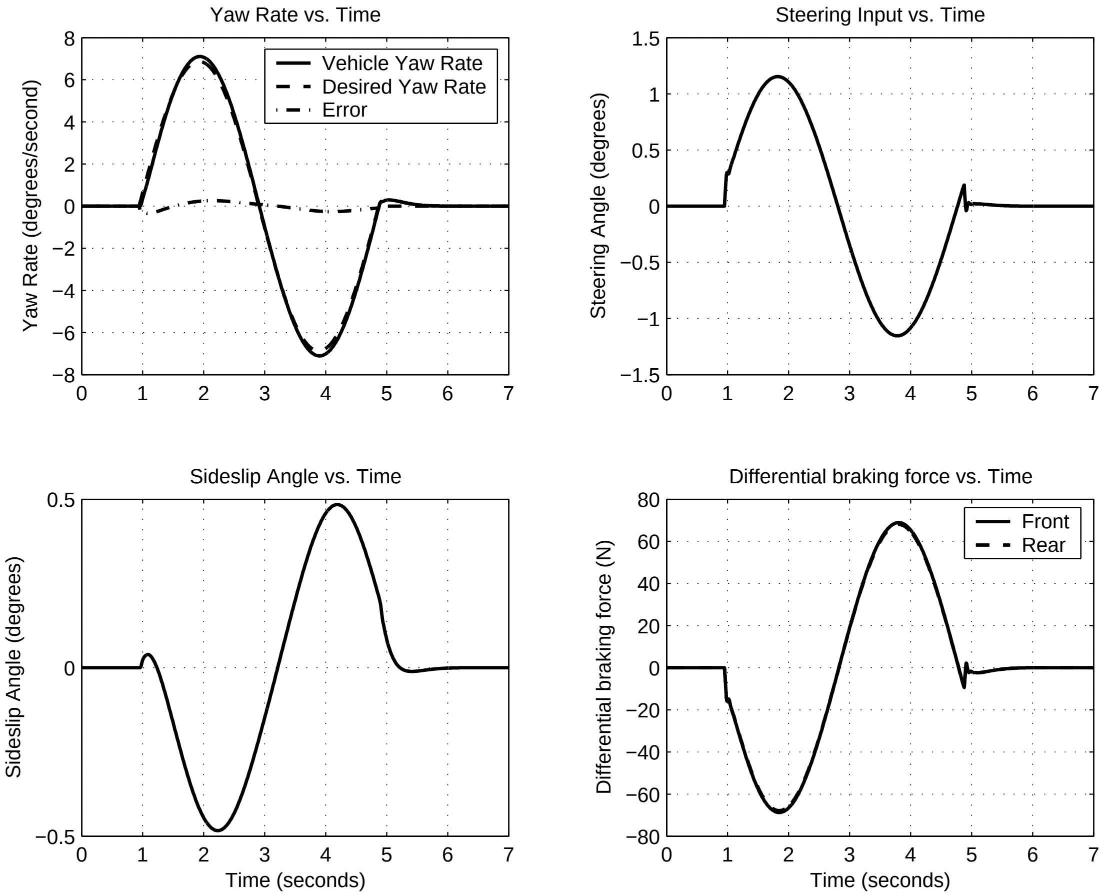


Figure 5.5: Nominal Case 3-Input: 55mph using $Q_\nu = 1e7$

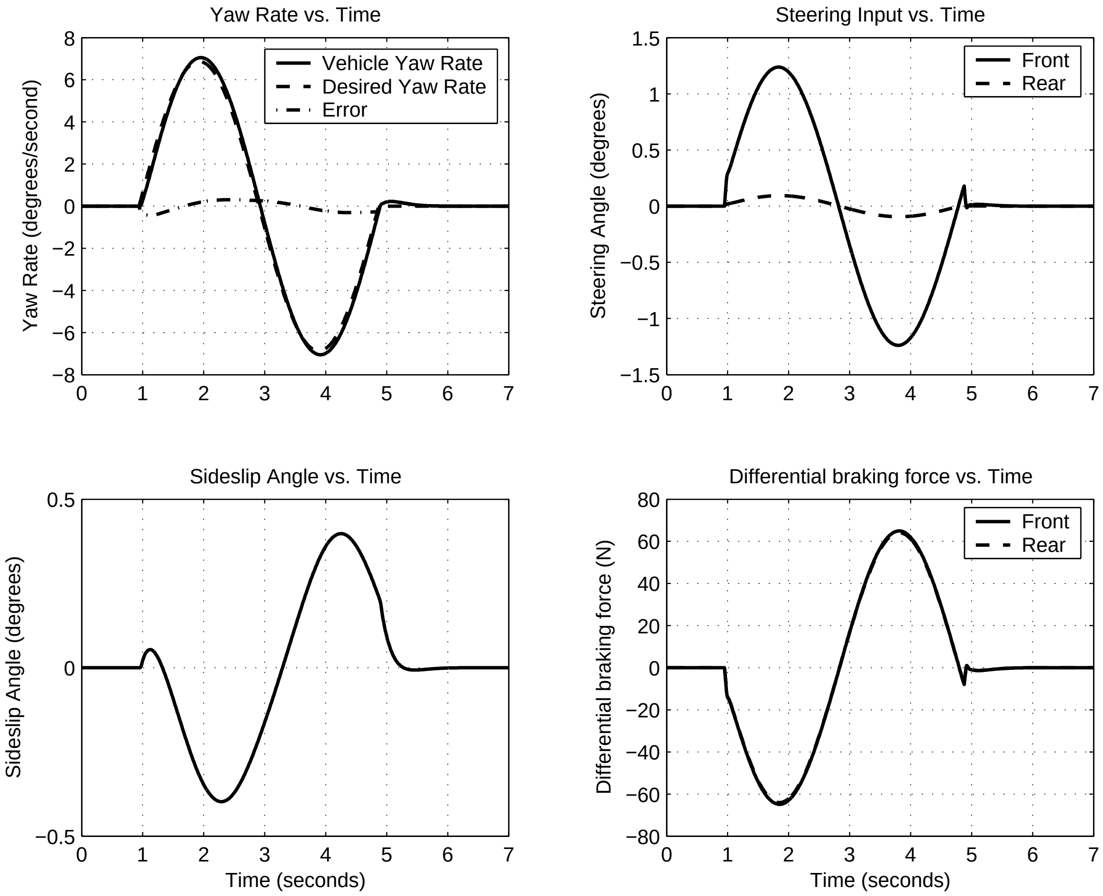


Figure 5.6: Nominal Case 4-Input: 55mph using $Q_\nu = 1e7$

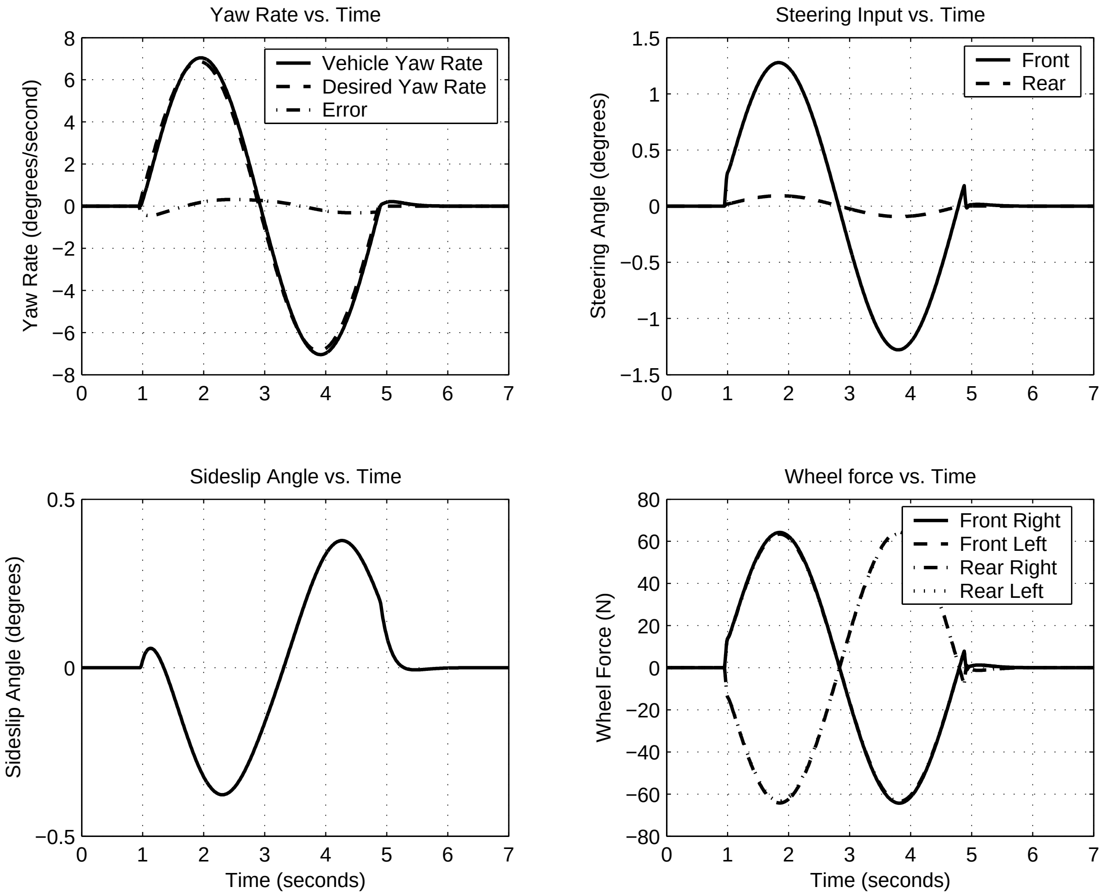


Figure 5.7: Nominal Case 6-Input: 55mph using $Q_\nu = 1e7$

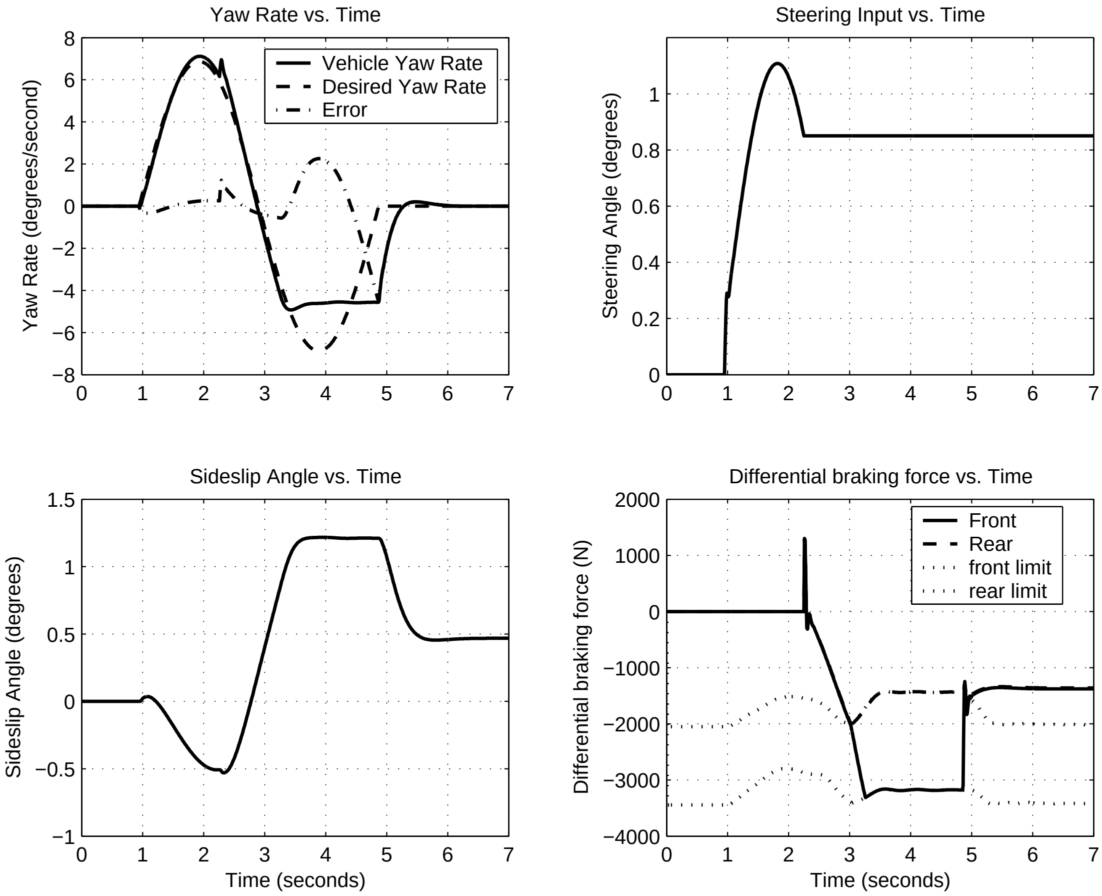


Figure 5.8: Steering Failure, 3-Input: 55mph using $Q_\nu = 1e3$

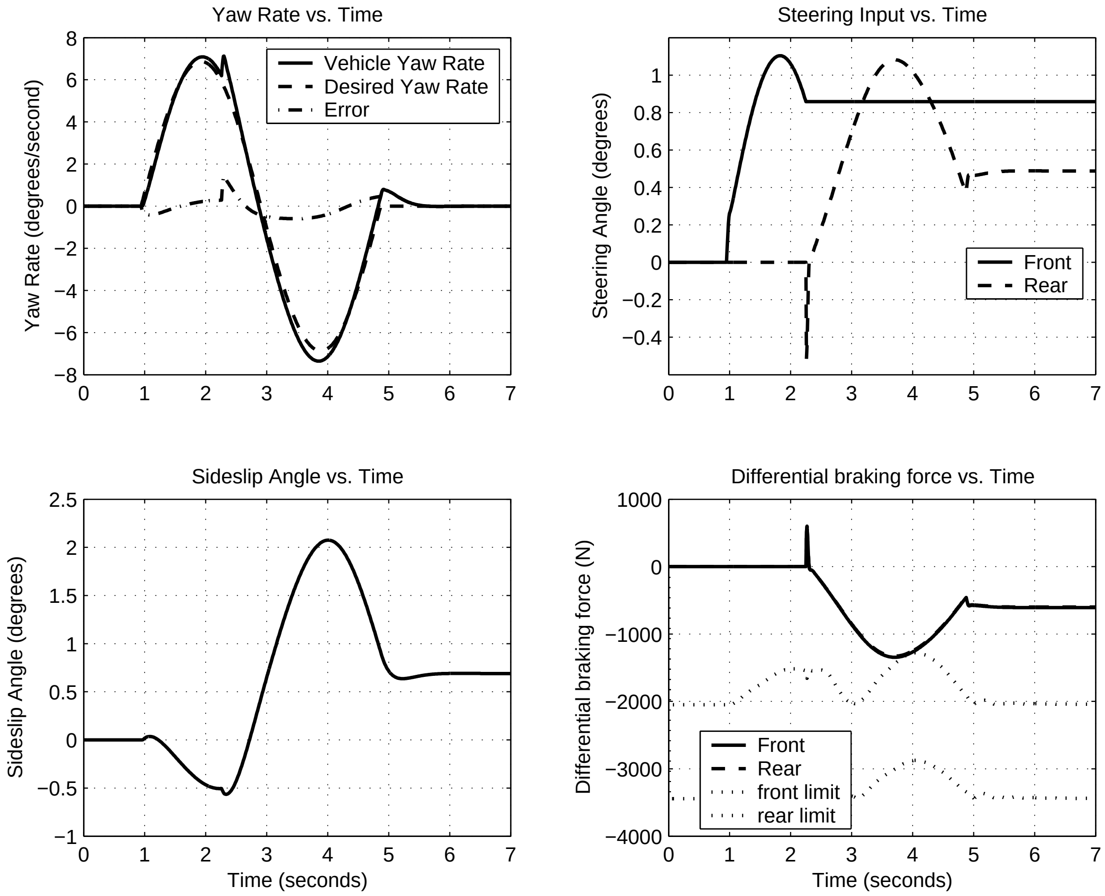
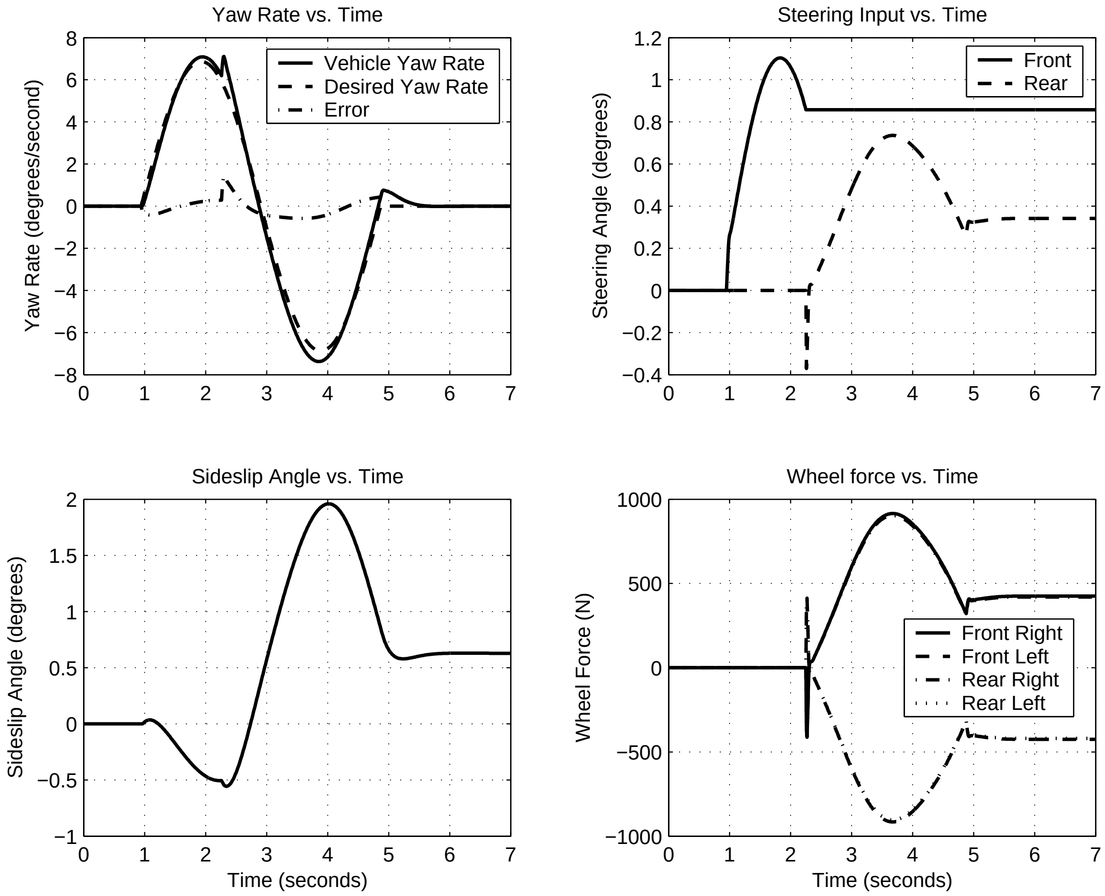


Figure 5.9: Steering Failure 4-Input: 55mph using $Q_\nu = 1e3$



Yaw Rate vs. Time

Steering Input vs. Time

Sideslip Angle vs. Time

Wheel force vs. Time

Figure 5.10: Steering Failure 6-Input: 55mph using $Q_\nu = 1e3$

CHAPTER 6

CONCLUSION

In this thesis, a quadratic programming based control allocation procedure has been shown to provide intelligent distribution of control effort among the effectors in both nominal and failure cases. The addition of the virtual effector ν provided a decoupling effect in the QP optimization so that opposing commands would be minimized in the presence of conflicting objectives. The resulting QP problem setup produces commands which do their best to keep the sideslip angle small while maintaining accurate tracking of the desired yaw rate. The simulation results show that the selection of the quadratic weight matrix Q, the virtual effector weight Q_ν in particular, significantly affects the output dynamics. The choice of $Q_\nu = 1e7$ gave good sideslip minimization in the nominal case while yaw rate tracking was only better for speeds of 55 and 65mph. $Q_\nu = 1e3$ gave the best tracking for both objectives during a steering failure. The extra maneuverability provided by additional effectors is a clear advantage in the failure case. The 3-input vehicle model could not track the desired yaw rate but the 4-input and 6-input models both gave very good results. The 6-input vehicle slightly out performed the 4-input vehicle during the failure case for $Q_\nu = 1e3$. The proposed control strategy offers different degrees of design freedom which should be taken into account depending on the specific design goal.

The majority of control allocation research has been published for aerospace and underwater applications where overactuated vehicles are necessary for good maneuverability in three dimensional space. Overactuated ground vehicles are becoming more popular however, as performance demands increase with the advancement of technology. The overactuated ground vehicle models presented in this thesis have the ability to satisfy multiple control objectives which necessitate the development of a good control scheme. Control allocation and its reconfiguration abilities prove to be very useful in dealing with overactuated systems such as the proposed ground vehicle models. Although the steering failure presented is not likely in today's mechanically steered automobiles, the fear of such a failure has prevented steer-by-wire systems from being implemented in production vehicles. The steer-by-wire concept has drawn a lot of attention in the area of vehicle stability control systems, and the proposed controller's ability to recover from a steering failure could be of significant value in this area of research. Overall, for basic control allocation purposes the proposed QP based method provides fast on-line commands yielding good results.

The results presented here are very promising and naturally lead to areas of future investigation. The simulation of maneuvers other than a double lane change should offer more insight into the controller's behavior and allow for the development of an intelligent strategy for tuning the weighting matrices (Q and c) of the QP problem. Also, an evaluation of driver-in-the-loop performance would be necessary if such a control strategy were to be used in a vehicle stability control system. Another logical step in continuing this research is actual implementation and testing, but a couple of issues must first be addressed:

1. The controller is only able to reconfigure the effector commands if it is alerted of a failure, therefore a suitable fault detection system must be incorporated into the control system design.

2. In Chapter 5, it was shown that noisy state estimates will cause the controller to produce high frequency effector commands. Future work will include investigating new and existing ways to alter the frequency content of the QP solution in order to achieve smoother commands.

The CA problem is a relatively small scale problem and the solution does not change a great deal from one time step to the next. Most QP software that is currently available is for finding an accurate solution to large scale problems, and is therefore not very efficient for smaller problems. Future work also involves the development of a fast QP solver geared toward finding an approximate solution to small scale problems.

Appendix A

Computer Code

A.1 Initialization and Simulation Routine

```
% main.m - This mfile is the main file.  It sets up the desired yaw
% rate trajectory and initializes the constant vehicle parameters and
% linear vehicle model used by the controller.  It also calculates the
% LQR gains and sets up the desired failure case.  The simulink
% simulation file car4.mdl is then run one time for each velocity
% specified in the vector 'vel' and for each quadratic weight on the
% virtual effector specified in the vector 'wBeta'.

clear all
close all
clc

% set up desired yaw profile============================================
lanes = 2;     %how many lanes to change(setup desired position vector)
psi_des(1) = 0;%lane change ~4m = 157.5in
rdes(1) = 0;
tt = 0;
dt = 0.01;     %time step
in = 0;
for ii = 1:1200
    if ii<396
        psi_des(ii) = 0;
        rdes(ii) = 0;
        pos_des(ii) = 0;
    end
    if ((ii>395) & (ii<788))
        psi_des(ii) = 0.075*(1-cos(1.60*tt(ii-1)));  % desired heading
        rdes(ii) = (psi_des(ii) - psi_des(ii-1))/dt; % desired yaw rate
        pos_des(ii) = in+(psi_des(ii-1)+psi_des(ii))*dt/2;
        in = pos_des(ii);
    end
    if ii>=788
        psi_des(ii) = 0;
        rdes(ii) = 0;
        pos_des(ii) = in;
    end
    tt(ii) = dt*(ii-1);
end
r_des = [tt' rdes'];
psi_des = [tt' psi_des'];
pos_des = [tt' lanes*13.58*pos_des'];

% Project parameters ====================================================
g = 9.81;               % gravity (m/s^2)
lb_N = 4.448;           % converts lbs to N
rad_d = 180/pi;         % converts radians to degrees

% car parameters %
L = 2.715;              % wheelbase of car (m)
T_f = 1.554;            % track width of front axle (m)
T_r = 1.534;            % trach width of rear axle (m)
W = 13735.424;          % curb weight of car + 200lb driver(N)
a = 1.013;              % distance from front wheel to cg (m)
b = L-a;                % distance from rear wheel to cg (m)
Wf = W*b/L;             % weight of front 1/2 car (N)
Wr = W*a/L;             % weight of rear 1/2 car (N)
mass = W/g;             % mass of car (kg)
Ixx =(0.18*3088-150) *32.174*0.04214011; %roll moment of inertia
Iyy =(0.99*3088-1149)*32.174*0.04214011; %pitch moment of inertia
Izz =(1.03*3088-1206)*32.174*0.04214011; %yaw moment of inertia(kg*m^2)
hcg = 0.4*1.4554;       %height of center of gravity(m)(40%roof height)
hf = 0.127; hr = 0.127; % height of roll center in front & rear (m)
del_h = hr-hf;          % difference in roll center heights
if del_h == 0           % calc distance between cg and roll axis
    h1 = hcg-hf;
else
    h2 = a*del_h/(a+b);
    h1 = hcg-hf-h2;
end

Ca_f = 1975*180/pi; % tire stiffness x2 front axl (N/rad)
Ca_r = 1575*180/pi; % tire stiffness x2 rear axl (N/rad)

% Roll stiffness for Front and Rear %
Kphf = 750*180/pi;     % total stiffness fr axl (N*m/rad)
Kphr = 650*180/pi;     % total stiffness rr axl (N*m/rad)
```

```
Kus=Wf/(Ca_f*pi/180)-Wr/(Ca_r*pi/180) % understeer gradient

count = 1;
vel = [45, 55, 65];      % simulate at these velocities (mph)
for ii = 1:length(vel)
    % Linear Car Model ==============================================
    V = vel(ii)*0.44704;             % mi/hr x0.44704 =(m/s)

    C0 = Ca_f + Ca_r;
    C1 = a*Ca_f - b*Ca_r;
    C2 = a^2 * Ca_f + b^2 * Ca_r;

    A = [-C0/(mass*V),  -C1/(mass*V^2)-1 ; %[beta]
         -C1/Izz,        -C2/(V*Izz)    ]; %[ r  ]

    B = [Ca_f/(mass*V), Ca_r/(mass*V),     0,       0,     1;%[delt_f]
         a*Ca_f/Izz,    -b*Ca_r/Izz,   T_f/(2*Izz),          %[delt_r]
         T_r/(2*Izz), C1*mass*V/(Izz*C0)];                   %[delFxf]
                                                             %[delFxr]
                                                             %[virEff]
    C = [1 0; 0 1];
    D = zeros(size(B));
    [mm,nn]=size(B);     % Quad_Prog s-function parameters

    % Controller Gains ==============================================
    [Ad, Bd] = c2d(A,B,0.01); % discrete B is used by QP (Bd*u = ubar)

    % add integrators to sys to minimize error
    Ap = [A, zeros(2,1);0, 1, 0];
    Bp = [1 0; 0 1; 0 0];

    [Ad, Bpd] = c2d(Ap,Bp,0.01); %gains computed assuming perfect input
    [K,S,E] = dlqr(Ad,Bpd,0.5*eye(3),eye(2)); % compute LQR gains

    algName = {'QP'};%,'SP'};%,'LS'};
    algFlg = [1];
    for jj = 1:length(algFlg)

        % Run Simulation ==============================================
        mu = 0.8; %approx coef of static friction between tire and road
        algorithm = algFlg(jj); % 1: QP,  3: SPQP,  0: LS
        wBeta = [1e3 1e7];      %quadratic weight on "virtual-effector"
        Beta = [3 7];

        % implement failures
        fault1 = diag([1, 1, 1, 1, 1]); % condition before failure
        thresh = 5.25;                  % time of failure
        fault2 = diag([0, 1, 1, 1, 1]); % condition after failure
        FailName = 'Fsteer';            % name of the failure

        for kk = 1:length(wBeta)
            wB = wBeta(kk);

            % save all simulation data
            cmd = sprintf('save results\\%s_%s_B%i_%i.mat yawRateR
                           sideslipR rollRateR rollAngR yawErrR yawDes
                           posDes Cmds psiDes X Y heading sigma V
                           algorithm Bd fLim rLim ubar', FailName,
                           algName{jj}, Beta(kk), vel(ii));
            fprintf('running simulation %i of %i ... ', count,...
                     length(vel)*length(algFlg)*length(wBeta));

            sim('car4',[3, 10]);      % run the simulation
            fprintf('done\n');
            eval(cmd);
            count = count + 1;
        end
    end
end

analysis(vel,algName,Beta,FailName);  % function to analyze sim results
% plots; % generate plots
```

A.2 S-function Implementation of the Vehicle Model

```
% John Plumlee
% car_eom.m - This s-function simulates the nonlinear differential
% equations of motion for the 4 wheeled car model with front and rear
% steering capabilities.

function [sys,x0,str,ts] = car_eom(t,x,u,flag)

switch flag,
  case 0, [sys,x0,str,ts]=mdlInitializeSizes;
  case 1, sys=mdlDerivatives(t,x,u);
  case 3, sys=mdlOutputs(t,x,u);
  case { 2, 4, 9 }
     sys = []; % Unused flags
  otherwise error(['Unhandled flag = ',num2str(flag)]);
end

%=======================================================================
function [sys,x0,str,ts]=mdlInitializeSizes
global betadot
betadot = 0;           % used by dFz, must be global
sizes = simsizes;
sizes.NumContStates  = 8;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 8;
sizes.NumInputs      = 7;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;   % at least one sample time is needed
sys = simsizes(sizes);
x0  = [0 0 0 0 0 0 0 0];    % initial conditions
str = [];                   % str is always an empty matrix
ts  = [0 0];                % initialize the array of sample times
return

%=======================================================================
function xdot=mdlDerivatives(t,x,u)
global betadot
% car parameters %
g = 9.81;              % accelleration due to gravity (m/s^2)
mu = 0.85;             % apprx cf of static friction b/w tire and road
L = 2.715;             % wheelbase of car (m)
T_f = 1.554;           % track width of front axle (m)
T_r = 1.534;           % trach width of rear axle (m)
W = 13735.424;         % curb weight of car + 200lb driver(N)
a = 1.013;             % distance from front wheel to cg (m)
b = L-a;               % distance from rear wheel to cg (m)
Wf = W*b/L;            % weight of front 1/2 car (N)
Wr = W*a/L;            % weight of rear 1/2 car (N)
mass = W/g;            % mass of car (kg)
Ixx = (0.18*3088-150)*32.174*0.04214011;  %roll moment of inertia
Izz = (1.03*3088-1206)*32.174*0.04214011; %yaw moment of inertia(kg*m^2)
hcg = 0.4*1.4554;      % height of center of gravity (m(40%roof height)
hf = 0.127; hr = 0.127; % height of roll center in front & rear (m)
del_h = hr-hf;          % difference in roll center heights
if del_h == 0           % calc distance between cg and roll axis
    h1 = hcg-hf;
else
    h2 = a*del_h/(a+b)+hf;
    h1 = hcg-h2;
end
% Roll stiffness and damping coeffs %
Kphf = 750*180/pi;       % total stiffness front axle (N*m/rad)
Kphr = 650*180/pi;       % total stiffness rear axle (N*m/rad)
Bphf = 900;              % total damping front axle (N*m*s/rad)
Bphr = 850;              % total damping rear axle (N*m*s/rad)

% Inputs
delt_f = u(1);           % steer angle of inner tire (rad)
delt_r = u(2);           % steer angle of outer tire
DFx_f  = u(3);           % differential braking command front axle (N)
DFx_r  = u(4);           % differential braking command front axle
Fxf    = u(5);           % front axle drive force? (N)
Vdot   = u(6);           % acceleration (m/s^2)
V      = u(7);           % Velocity (m/s)

%States
beta   = x(1);           % sideslip angle (rad)
psidot = x(2);           % yaw rate (rad/s)
psi    = x(3);           % yaw/heading angle (rad)
phidot = x(4);           % roll rate (rad/s)
phi    = x(5);           % roll angle (rad)

% vertical force difference across axle, front and rear
dFzf = (Kphf*phi+Bphf*phidot + Wf/g*hf*(Vdot*sin(beta)...
       +V*betadot*cos(beta) + psidot*V*cos(beta)))/T_f;
dFzr = (Kphr*phi+Bphr*phidot + Wr/g*hr*(Vdot*sin(beta)...
       +V*betadot*cos(beta) + psidot*V*cos(beta)))/T_r;

Fzf = Wf/2;              % normal static load on front tires
Fzr = Wr/2;              % normal static load on rear tires
if dFzf > Fzf            % restrict 2 wheel lift-off
    dFzf = Fzf;end
if dFzr > Fzr
    dFzr = Fzr;end
```

```
FzRf = (Fzf+dFzf);       % norm force on right front tire (N)
FzLf = (Fzf-dFzf);       % norm force on left front tire
FzRr = (Fzr+dFzr);       % norm force on right rear tire
FzLr = (Fzr-dFzr);       % norm force on left rear tire
if FzRf <= 0             % restrict 2 wheel lift-off
    FzRf = 1e-6;end
if FzRr <= 0
    FzRr = 1e-6;end
if FzLf <= 0
    FzLf = 1e-6;end
if FzLr <= 0
    FzLr = 1e-6;end

% longitudinal forces at tires due to braking command (N)
if DFx_f>0
    FxRf = Fxf - DFx_f;
    FxLf = 0;
else
    FxRf = 0;
    FxLf = Fxf + DFx_f;
end
if DFx_r>0
    FxRr = 0 - DFx_r;
    FxLr = 0;
else
    FxRr = 0;
    FxLr = 0 + DFx_r;
end

% tire slip angles (rad)
alph_Rf =atan((V*sin(beta)+psidot*a)/(V*cos(beta)-psidot*T_f/2))-delt_f;
alph_Lf =atan((V*sin(beta)+psidot*a)/(V*cos(beta)+psidot*T_f/2))-delt_f;
alph_Rr =atan((V*sin(beta)-psidot*b)/(V*cos(beta)-psidot*T_r/2))-delt_r;
alph_Lr =atan((V*sin(beta)-psidot*b)/(V*cos(beta)+psidot*T_r/2))-delt_r;

% lateral forces at tires (N)
FyRf = calc_Fy(FzRf/1000, alph_Rf*180/pi);
FyLf = calc_Fy(FzLf/1000, alph_Lf*180/pi);
FyRr = calc_Fy(FzRr/1000, alph_Rr*180/pi);
FyLr = calc_Fy(FzLr/1000, alph_Lr*180/pi);

% calculate the derivatives
```

```
betadot = ((FyRf+FyLf)*cos(delt_f) + (FxRf+FxLf)*sin(delt_f)  ...
          +(FyRr+FyLr)*cos(delt_r) + (FxRr+FxLr)*sin(delt_r)) ...
          /(mass*V*cos(beta)) - (Vdot*tan(beta))/V - psidot;

rdot = (a*((FyRf+FyLf)*cos(delt_f) + (FxRf+FxLf)*sin(delt_f)) ...
       -b*((FyRr+FyLr)*cos(delt_r) + (FxRr+FxLr)*sin(delt_r)) ...
       +T_f/2*((FyRf-FyLf)*sin(delt_f) + (FxLf-FxRf)*cos(delt_f)) ...
       +T_r/2*((FyRr-FyLr)*sin(delt_r) + (FxLr-FxRr)*cos(delt_r)))/Izz;

phiddot = (mass*h1*g*sin(phi) - (Kphf+Kphr)*phi - (Bphf+Bphr)*phidot ...
          -mass*h1*(Vdot*sin(beta) + V*betadot*cos(beta) ...
          +psidot*V*cos(beta))*cos(phi))/Ixx;

% position, space fixed frame
Xdot = V*(cos(beta)*cos(psi)-sin(beta)*sin(psi));
Ydot = V*(cos(beta)*sin(psi)+sin(beta)*cos(psi));

Fxr = (FyLf+FyRf)*sin(delt_f) - (FxLf+FxRf)*cos(delt_f) ...
      + mass*Vdot*cos(beta) - mass*V*(betadot+psidot)*sin(beta);

xdot = [betadot, rdot, psidot, phiddot, phidot, Xdot, Ydot, Fxr]';

return

%========================================================================
function yy=mdlOutputs(t,x,u);
yy = [x];
return

%========================================================================
function Fy = calc_Fy(Fz, alph)
% Nonlinear tire force model
%   input: vertical force Fz in kN, slip angle alph in deg
%   output: lateral force Fy in N

% Pacejka coefficients of a real data set SAE Paper #870421
a1 = -22.1;   %3.6;
a2 = 1011;    %1200;
a3 = 1078;
a4 = 1.82;
a5 = 0.208;   %0.25;
a6 = 0;
```

```
a7 = -0.354;
a8 = 0.707;
a9 = 0.028;
a10 = 0;
a11 = 14.8;
a12 = 0.022;
a13 = 0;
C = 1.30;

D = a1*Fz^2+a2*Fz;              % represents the peak side force
B = (a3*sin(a4*atan(a5*Fz)))/(C*D);
E = a6*Fz^2+a7*Fz+a8;
BCD = a3*sin(a4*(atan(a5*Fz)));  % represents cornering stiffness

Phi = (1-E)*alph+(E/B)*(atan(B*alph));
Fy = -D*sin(C*(atan(B*Phi)));  % calc lateral force on the inner tires
```

A.3 Implementation of Nonlinear Tire Model and Friction Circle

```
% John Plumlee 3/12/04
% fricCirc.m - This s-function calculates the max force avaliable in the
% x direction for each tire of a four wheeled ground vehicle based on
% the fricion circle concept. The Pacejka tire model is used to simulate
% nonlinear tire behavior.

function [sys,x0,str,ts] = fricCirc(t,x,u,flag,mu)

switch flag,
  case 0, [sys,x0,str,ts]=mdlInitializeSizes;
  case 3, sys=mdlOutputs(t,x,u,mu);
  case { 1, 2, 4, 9 }
     sys = []; % Unused flags
  otherwise error(['Unhandled flag = ',num2str(flag)]);
end

%=========================================================================
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 7;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;   % at least one sample time is needed
sys = simsizes(sizes);
x0  = [];           % initial conditions
str = [];                   % str is always an empty matrix
ts  = [0.01 0];                % initialize the array of sample times
return

%=========================================================================
function yy=mdlOutputs(t,x,u,mu);
global betadot

% car parameters %
g = 9.81;                % gravity (in/s^2)
L = 2.715;               % wheelbase of car (m)
T_f = 1.554;             % track width of front axle (m)
T_r = 1.534;             % trach width of rear axle (m)
W = 13735.424;           % curb weight of car + 200lb driver(N)
a = 1.013;               % distance from front wheel to cg (m)
b = L-a;                 % distance from rear wheel to cg (m)
Wf = W*b/L;              % weight of front 1/2 car (N)
Wr = W*a/L;              % weight of rear 1/2 car (N)
mass = W/g;              % mass of car (kg)
Izz = (1.03*3088-1206)*32.174*0.04214011;%yaw moment of inertia (kg*m^2)
hcg = 0.4*1.4554;        % height of center of gravity(m)(40%roof height)
hf = 0.127; hr = 0.127; % height of roll center in front & rear (m)
del_h = hr-hf;           % difference in roll center heights
if del_h == 0            % calc distance between cg and roll axis
    h1 = hcg-hf;
else
    h2 = a*del_h/(a+b)+hf;
    h1 = hcg-h2;
end
% Roll stiffness and damping coeffs %
Kphf = 750*180/pi;        % total stiffness front axle (N*m/rad)
Kphr = 650*180/pi;        % total stiffness rear axle (N*m/rad)
Bphf = 900;               % total damping front axle
Bphr = 850;               % total damping rear axle

delt_f = u(1);            % steer angle of inner tire
delt_r = u(2);            % steer angle of outer tire
V      = u(3);            % Velocity (m/s)
```

```
Vdot = 0;
beta   = u(4);           % sideslip angle (rad)
psidot = u(5);           % yaw rate (rad/s)
phidot = u(6);           % roll rate (rad/s)
phi    = u(7);           % roll angle (rad)

% vertical force difference across axle, front and rear
dFzf = (Kphf*phi+Bphf*phidot + Wf/g*hf*(Vdot*sin(beta)...
       +V*betadot*cos(beta) + psidot*V*cos(beta)))/T_f;
dFzr = (Kphr*phi+Bphr*phidot + Wr/g*hr*(Vdot*sin(beta)...
       +V*betadot*cos(beta) + psidot*V*cos(beta)))/T_r;
Fzf = Wf/2;              % normal static load on fr tires
Fzr = Wr/2;              % normal static load on rr tires
if dFzf > Fzf
    dFzf = Fzf;end
if dFzr > Fzr
    dFzr = Fzr;end
FzRf = (Fzf+dFzf);       % norm force on right front tire (N)
FzLf = (Fzf-dFzf);       % norm force on left front tire
FzRr = (Fzr+dFzr);       % norm force on right rear tire
FzLr = (Fzr-dFzr);       % norm force on left rear tire
if FzRf <= 0             % restrict 2 wheel lift-off
    FzRf = 1e-6;end
if FzRr <= 0
    FzRr = 1e-6;end
if FzLf <= 0
    FzLf = 1e-6;end
if FzLr <= 0
    FzLr = 1e-6;end

% calculate tire slip angles (rad)
alph_Rf =atan((V*sin(beta)+psidot*a)/(V*cos(beta)-psidot*T_f/2))-delt_f;
alph_Lf =atan((V*sin(beta)+psidot*a)/(V*cos(beta)+psidot*T_f/2))-delt_f;
alph_Rr =atan((V*sin(beta)-psidot*b)/(V*cos(beta)-psidot*T_r/2))-delt_r;
alph_Lr =atan((V*sin(beta)-psidot*b)/(V*cos(beta)+psidot*T_r/2))-delt_r;

% lateral forces at tires (N)
FyRf = calc_Fy(FzRf/1000, alph_Rf*180/pi);
FyLf = calc_Fy(FzLf/1000, alph_Lf*180/pi);
FyRr = calc_Fy(FzRr/1000, alph_Rr*180/pi);
FyLr = calc_Fy(FzLr/1000, alph_Lr*180/pi);
```

```
% Max longitudinal force allowed by the friction circle
FxRf = sqrt(abs((FzRf*mu)^2 - FyRf^2));
FxLf = sqrt(abs((FzLf*mu)^2 - FyLf^2));
FxRr = sqrt(abs((FzRr*mu)^2 - FyRr^2));
FxLr = sqrt(abs((FzLr*mu)^2 - FyLr^2));

front = min(FxRf,FxLf);
rear  = min(FxRr,FxLr);

yy = [front;rear];
return

%=======================================================================
function Fy = calc_Fy(Fz, alph)
% input: vertical force Fz in kN, slip angle alph in deg
% output: lateral force Fy in N
% Pacejka coefficients of a real data set SAE Paper #870421
a1 = -22.1;   %3.6;
a2 = 1011;    %1200;
a3 = 1078;
a4 = 1.82;
a5 = 0.208;   %0.25;
a6 = 0;
a7 = -0.354;
a8 = 0.707;
a9 = 0.028;
a10 = 0;
a11 = 14.8;
a12 = 0.022;
a13 = 0;
C = 1.30;

D = a1*Fz^2+a2*Fz;              % represents the peak side force
B = (a3*sin(a4*atan(a5*Fz)))/(C*D);
E = a6*Fz^2+a7*Fz+a8;
BCD = a3*sin(a4*(atan(a5*Fz)));  % represents cornering stiffness

Phi = (1-E)*alph+(E/B)*(atan(B*alph));
Fy = -D*sin(C*(atan(B*Phi)));  % calc lateral force on the inner tires
```

A.4 Implementation of Quadratic Programming Based Control Allocation

```
% Quad_Prog.m - This S-function is the quadratic programming based
% control allocation routine used by the vehicle model.  It takes as
% inputs the discretized input effectiveness matrix from the linear car
% model, the desired virtual effect calculated by the LQR gains and the
% front and rear braking limits calculated by fricCirc.m.  The outputs
% are the computled command vector and allocation error.

function [sys,x0,str,ts] = QuadProg(t,x,u,flag,algorithm,mm,nn,wB)

switch flag,
  case 0, [sys,x0,str,ts]=mdlInitializeSizes(mm,nn);
  case 2, sys=mdlUpdate(t,x,u,algorithm,mm,nn,wB);
  case 3, sys=mdlOutputs(t,x,u,mm,nn);
  case { 1, 4, 9 }
     sys = []; % Unused flags
  otherwise error(['Unhandled flag = ',num2str(flag)]);
end
%======================================================================
function [sys,x0,str,ts]=mdlInitializeSizes(mm,nn)
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = nn*4+3;
sizes.NumOutputs     = nn+3;
sizes.NumInputs      = mm*nn+mm+2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [zeros(nn,1); 0; 1; 1; zeros(nn*3,1)];%initial conditions for sys
str = [];            % str is always an empty matrix
ts  = [0.01 0];      % initialize the array of sample times
return

%======================================================================
function sys=mdlUpdate(t,x,u,algorithm,mm,nn,wB)

% Inputs
B = reshape(u(1:mm*nn,1),mm,nn); % Effectiveness Matrix
ubar = u(mm*nn+(1:mm),1);        % desired control effect

c = zeros(nn,1);         % linear weight matrix
```

```
Q = [1 1e10 1 1 wB];     % quadratic weight matrix (Diagonalized later)

limIdx = mm*nn+mm+1;     % index of braking limits

% position limits of effectors (inequality constraints)
xhi =  [ 0.5  0.5  u(limIdx)  u(limIdx+1)  100]';
xlo =  [-0.5 -0.5 -u(limIdx) -u(limIdx+1) -100]';

% Uncomment these for steering failures
% if t>5.24 % front steering failure, stuck-at fault
%     xhi =  [ x(1)  0.5  u(limIdx)  u(limIdx+1)  100]';
%     xlo =  [ x(1) -0.5 -u(limIdx) -u(limIdx+1) -100]';
% end
% if t>5.49 % rear steering failure, stuck-at fault
%     xhi =  [  0.5  x(2)  u(limIdx)  u(limIdx+1)  100]';
%     xlo =  [ -0.5  x(2) -u(limIdx) -u(limIdx+1) -100]';
% end

% Command History
xxk1 = reshape(x([1:nn]),nn,1);      % cmd from last time step (k-1)
xxk2 = reshape(x(nn+3+[1:nn]),nn,1); % cmd (k-2)
xxk3 = reshape(x(2*nn+3+[1:nn]),nn,1);
xxk4 = reshape(x(3*nn+3+[1:nn]),nn,1);
xxh = [xxk1, xxk2, xxk3, xxk4];

c = -1e-3*eye(nn)./(0.01)*xxk1;        % setup to penalize cmd velocity
c = c + 1e-5*eye(nn)./(0.01^2)*(xxk2-2*xxk1);% penalize cmd vel & accel

% Matlab's quadratic programming solver
ops = optimset('Display','off'); % set up options
warning off;
[uinput,fval,exflg] = quadprog(diag(Q),c,[],[],B,ubar,xlo,xhi,[],ops);

if isempty(uinput)
    uinput = xx(nn,1);
    fprintf('uinput is empty QP solution is not good\n');
end

% Allocation Error
cErr = ubar - B(:,:)*reshape(uinput,nn,1);

% save CA error and command history
```

```
output4 = [norm(cErr);1;1];
output5 = reshape(xxh(:,1:3),nn*3,1);

sys = [uinput;output4;output5];
return

%======================================================================
function y=mdlOutputs(t,x,u,mm,nn);
y = x(1:nn+3);
return
```

BIBLIOGRAPHY

[1] L. Alvarez, J. Yi, R. Horowitz, and L. Olmos. Emergency braking control in automated highway systems with underestimation of friction coefficient. In *Proceedings of the American Control Conference*, pages 574–579. IEEE, June 2000.

[2] Rusty Anderson and David M. Bevly. Estimation of slip angles using a model based estimator and gps. In *Proceedings of the American Control Conference*, Boston, MA, June 2004. IEEE.

[3] Egbert Bakker, Lars Nyborg, and Hans B. Pacejka. Tyre modelling for use in vehicle dynamic studies. Technical Report 870421, Society of Automotive Engineers, Febuary 1987.

[4] J. E. Beasley, editor. *Advances in Linear and Integer Programming.* Clarendon Press, Oxford, 1996.

[5] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

[6] D. M. Bevly, J. C. Gerdes, and C. Wilson. The use of gps based velocity measurements for measurement of sideslip and wheel slip. *Vehicle System Dynamics*, 38:127–147, 2002.

[7] Marc Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, July-August 2002.

[8] Kenneth A Bordington and Wayne C Durham. Closed-form solutions to constrained control allocation problem. *Journal of Guidance, Control, and Dynamics*, 18(5):1000–1006, September-October 1995.

[9] Jean L. Broge. Quadrasteer debuts in 2002. Tech briefs, SAE International Online, http://www.sae.org/automag/techbriefs/01-2001/techb2.htm, February 2001.

[10] John J. Burken, Ping Lu, Zhenglu Wu, and Cathy Bahm. Two reconfigurable flight–control design methods: Robust servomechanism and control allocation. *Journal of Guidance, Control, and Dynamics*, 24(3):482–493, May-June 2001.

[11] Ronnie R. Callahan. Dynamic control allocation and autonomous reconfigurable control allocation for aerospace vehicles. Master's thesis, Auburn University, May 2002.

[12] Christopher R. Carlson and J. Christian Gerdes. Optimal rollover prevention with steer by wire and differential braking. In *Proceedings of IMECE 2003.* ASME, November 2003.

[13] B. Chen and H. Peng. Differrential-braking-based rollover prevention for sport utility vehicles with human-in-the-loop evaluations. *Vehicle System Dynamics*, 36(4-5):359–389, 2001.

[14] John C. Dixon. *Tires, Suspension, and Handling.* Society of Automotive Engineers Inc., second edition, 1996.

[15] Wayne C. Durham. Constrained control allocation. *Journal of Guidance, Control, and Dynamics*, 16(4):717–725, August 1993.

[16] Wayne C. Durham. Constrained control allocation: Three-moment problem. *Journal of Guidance, Control, and Dynamics*, 17(2):330–337, March-April 1994.

[17] Wayne C. Durham. Computationally efficient control allocation. *Journal of Guidance, Control, and Dynamics*, 24(3):519–524, May-June 2001.

[18] Dale Enns. Control allocation approaches. In *Proceeding of AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 98–108, Boston, MA, 1998. AIAA.

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

[20] Ola Harkegard. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 1295–1300. IEEE, December 2002.

[21] A. S. Hodel. Robust inversion and data compression in control allocation. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, August 2000. AIAA.

[22] A. S. Hodel and Yuri B. Shtessel. On-line computation of a local attainable moment set for reusable launch vehicles. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit.* AIAA, August 2002.

[23] Yutaka Ikeda and Mark Hood. An application of l1 optimization to control allocation. In *Proceeding of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, August 2000. AIAA.

[24] Tor A. Johansen, Thor I. Fossen, and Svein P. Berge. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *IEEE Trans. Control Systems Technology*, 12, 2004.

[25] David G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA, 1984.

[26] Anthony B Page and Marc L Steinberg. A closed-loop comparison of control allocation methods. In *Proceeding of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, August 2000. AIAA Paper 2000-4538.

[27] C. Van De Panne. *Methods for Linear and Quadratic Programming*, volume 17. North-Holland publishing Co., 1974.

[28] R. Rajamani, A. S. Howell, C. Chen, J. K. Hendrick, and M. Tomizuka. A complete fault diagnostic system for automated vehicles operating in a platoon. *IEEE Transactions on Control Systems Technology*, 9(4):553–564, July 2001.

[29] Laura R. Ray. Nonlinear tire force estimation and road friction identification: Simulation and experiments. *Automatica*, 33(10):1819–1833, 1997.

[30] Eric J. Rossetter, J. P. Swikes, and J. C. Gerdes. A gentle nudge towards safety: Experimental validation of the potential field driver assistance system. In *Proceedings of the American Control Conference*, Denver, CO, June 2003. IEEE.

[31] J. Ryu, E. J. Rossetter, and J. C. Gerdes. Vehicle sideslip and roll parameter estimation using gps. In *AVEC 2002 6th International Symposium of Advanced Vehicle Control*, 2002.

[32] Arvin R. Savkoor and C.T. Chou. Application of aerodynamic actuators to improve vehicle handling. *Vehicle System Dynamics*, 32:345–374, 1999.

[33] Adam T. Simmons and A. Scottedward Hodel. Control allocation for the x-33 using existing and novel quadratic programming techniques. In *Proceedings of the American Control Conference*, Boston, MA, June 2004. IEEE.

[34] Anton T. van Zanten. Evolution of electronic control systems for improving the vehicle dynamic behavior. In *Proceedings of the 6th International Symposium on Advanced Vehicle Control*, 2002.

[35] Anthony B. Will and Stanislaw H. Zak. Modelling and control of an automated vehicle. *Vehicle System Dynamics*, 27:131–155, 1997.

[36] Anton T. Van Zanten, Rainer Erhardt, Georg Pfaff, Friedrich Kost, Uwe Hartmann, and Thomas Ehret. Control aspects of the bosch-vdc. In *International Symposium on Advanced Vehicle Control*, pages 573–605, June 1996.