Automatic Steering Methods for Autonomous Automobile Path Tracking

Jarrod M. Snider

February 2009

Robotics Institute

Carnegie Mellon University

Pittsburgh, Pennsylvania

Abstract

This research derives, implements, tunes and compares selected path tracking methods for controlling a car-like robot along a predetermined path. The scope includes commonly used methods found in practice as well as some theoretical methods found in various literature from other areas of research. This work reviews literature and identifies important path tracking models and control algorithms from the vast background and resources. This paper augments the literature with a comprehensive collection of important path tracking ideas, a guide to their implementations and, most importantly, an independent and realistic comparison of the performance of these various approaches. This document does not catalog all of the work in vehicle modeling and control; only a selection that is perceived to be important ideas when considering practical system identification, ease of implementation/tuning and computational efficiency. There are several other methods that meet this criteria, however they are deemed similar to one or more of the approaches presented and are not included. The performance results, analysis and comparison of tracking methods ultimately reveal that none of the approaches work well in all applications and that they have some complementary characteristics. These complementary characteristics lead to an idea that a combination of methods may be useful for more general applications. Additionally, applications for which the methods in this paper do not provide adequate solutions are identified.

Acknowledgements

Contents

List of Figures

1 Introduction

A significant portion of Robotics research involves developing autonomous car-like robots. This research is often at the forefront of innovation and technology in many areas. However, it is often common practice to use relatively simple and sometimes naive control strategies and/or system models for vehicle control, even on some well known and successful autonomous vehicle projects [18, 17, 16, 4].

Figure 1: Sandstorm

Figure 2: Stanley

Figure 3: Boss

Figure 1 is Sandstorm, the autonomous vehicle that placed second in the DARPA Grand Challenge using a very simple steering control law based on a geometric vehicle model. Figure 2 is Stanley, the autonomous vehicle that won the DARPA Grand Challenge using an intuitive steering control law based on a simple kinematic vehicle model. Figure 3 is Boss, the autonomous vehicle that won the DARPA Urban Challenge. Boss uses a much more sophisticated model predictive control strategy to perform vehicle control. However, a very simple kinematic model of the vehicle, a time delay and rate limits on steering is all that is included in the optimization of the steering controls. This does

not mean, however, there has not been extensive research in this area, and a great deal of literature on the topic exists. These observations lead to the following questions:

- How good are these simple methods commonly found in practice?

- Can improvements in performance be achieved using existing methods found in other theoretical research?

- Can good matches between methods and applications be identified?

- What are the limitations and potential for future breakthroughs?

It is these observations and questions that motivate the research found in this paper. This paper endeavors to collect scattered sources and provide an independent and realistic comparison of the performance of several classes of vehicle controllers along with advice on how to implement them.

The family of vehicle controllers of interest are called path trackers. Path tracking refers to a vehicle executing a globally defined geometric path by applying appropriate steering motions that guide the vehicle along the path. The goal of a path tracking controller is to minimize the lateral distance between the vehicle and the defined path, minimize the difference in the vehicle's heading and the defined path's heading, and limit steering inputs to smooth motions while maintaining stability.

For each class of path trackers presented in this paper, an underlying system model will be developed before the algorithm is presented. The algorithms themselves will be presented in a way to minimize the complexity of performance tuning, and the effects of the parameters will be illustrated using three representative courses. Chapter 2 will concentrate on methods that exploit geometric relationships between the vehicle and the path to design control laws [2, 16] that are simple, robust and achieve accurate path tracking in a limited set of driving scenarios and can provide moderate tracking in a much larger set of scenarios. Chapter 3 moves on to more control theory based techniques and uses a simple kinematic model of a vehicle [6] to show that accurate path tracking can be achieved with this simple model in a limited set of driving scenarios. Chapter 4 introduces a dynamic vehicle model and techniques such as Optimal Control and Optimal Preview Control [11, 10, 14] to demonstrate that accurate path tracking can be achieved over a wider range of driving scenarios when the dynamics of the vehicle are considered. Chapter 5 then turns to a head-to-head performance comparison of these algorithms which illustrates why relatively primitive control techniques are commonly used successfully as well as highlighting the need for more advanced techniques as Robotics moves forward in the development of precision path tracking for vehicles operating at higher speeds and with new objectives.

2

1.1 Experimental Design
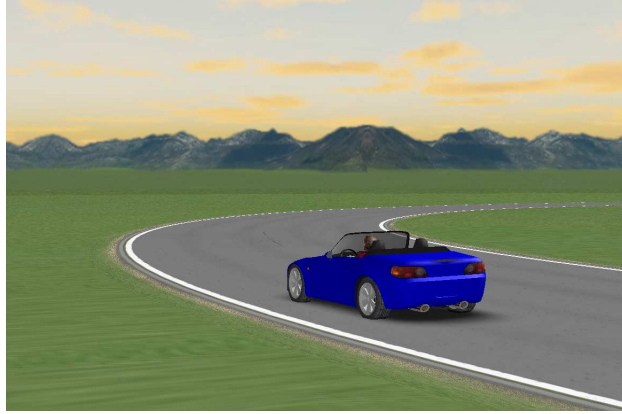
Figure 4: Screen shot from a CarSim animation

To provide a valid evaluation and comparison, the path tracking controllers implemented for this paper are tested with a high fidelity vehicle simulator, CarSim. CarSim quickly and accurately simulates the dynamic behavior of vehicles and is used in the automotive industry as the standard by which vehicle handling and dynamics are tested [1]. Figure 4 is a screen shot from a CarSim animation. The standard CarSim simulator provides a complete model of the vehicle system and the environment [7]. Additionally, rate limits and delays associated with the steering actuator are added to the standard model. The controllers are implemented in C and communicate with CarSim through the available API [8].

Three driving courses are chosen to perform the various experiments found in this paper. The courses are designed to test attributes of the controllers and provide insight into their relative advantages and disadvantages.

1.1.1 Lane Change Course

The *Lane Change Course* is a straight section of two lane road in which the vehicle is required to perform a single lane change maneuver. The lane change maneuver is a common test for vehicle handling as it represents an essential collision avoidance maneuver. The *Lane Change Course* is chosen to demonstrate the tracking capability on a straight path as well as the response to a quick, yet (position and curvature) continuous, transient section. Experiments on this course are performed at constant velocities of $5m/s$, $10m/s$, $15m/s$ and $20m/s$. Figure 5 illustrates the *Lane Change Course*.
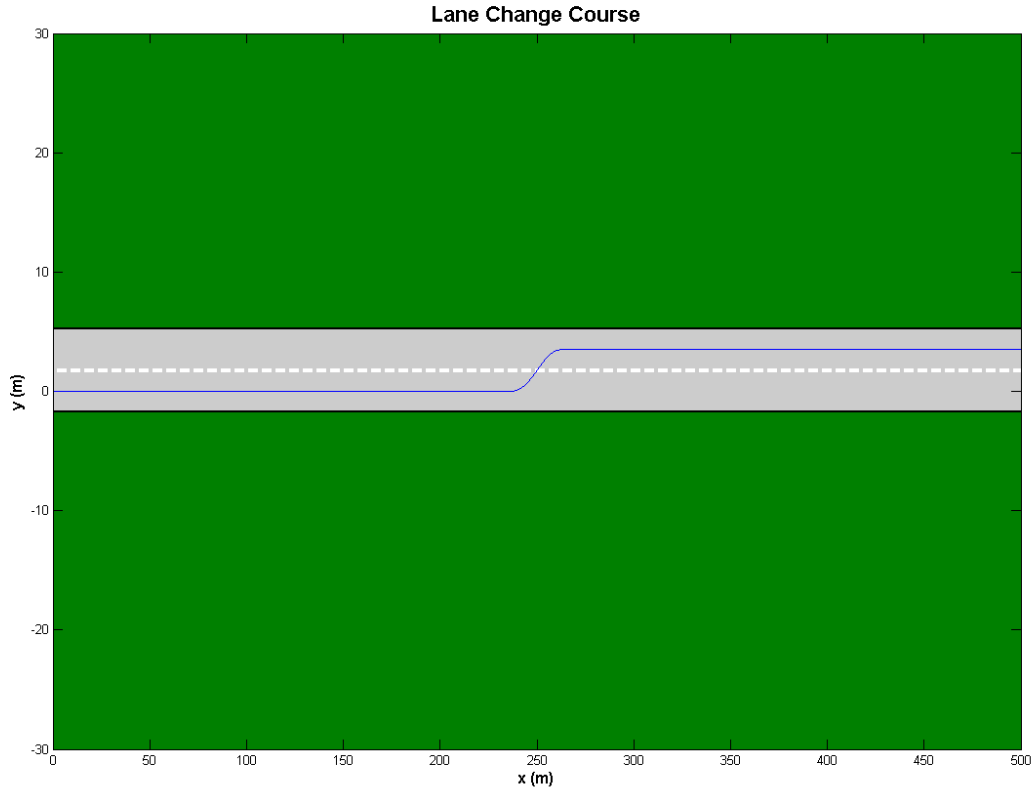
Figure 5: Lane Change Course

1.1.2 Figure Eight Course

The *Figure Eight Course* consists of two circular paths that intersect at a tangent point. This course is not commonly found in everyday driving. However, it can provide valuable insight into the handling of a vehicle as well as some important characteristics of a controller. This course was chosen to illustrate the steady state characteristics of the controllers while executing a constant nonzero curvature path. This course also includes a point with discontinuous curvature where vehicles transition from one circle to an other. While it is possible to require that all paths be generated with continuous curvature, a controller's response to this discontinuity provides insight into it's robustness. Experiments on this course are performed at constant velocities of $5m/s$, $10m/s$, $15m/s$ and $20m/s$. Figure 6 illustrates the *Figure Eight Course*.

4
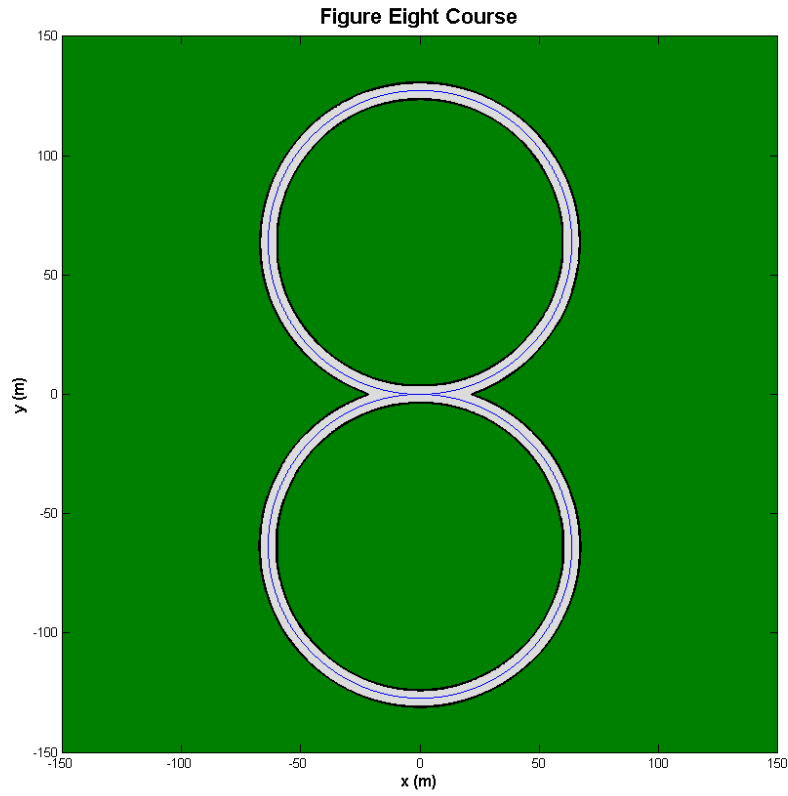
Figure 6: Figure Eight Course

1.1.3 Road Course

The *Road Course* captures a variety of driving scenarios representative of real-world driving. The path is generated to minimize lateral acceleration while staying on the road surface. The path is continuous and the speed varies as a function of the path. The *Road Course* facilitates a general performance comparison of the various tracking methods. Figure 7 illustrates the *Road Course*.
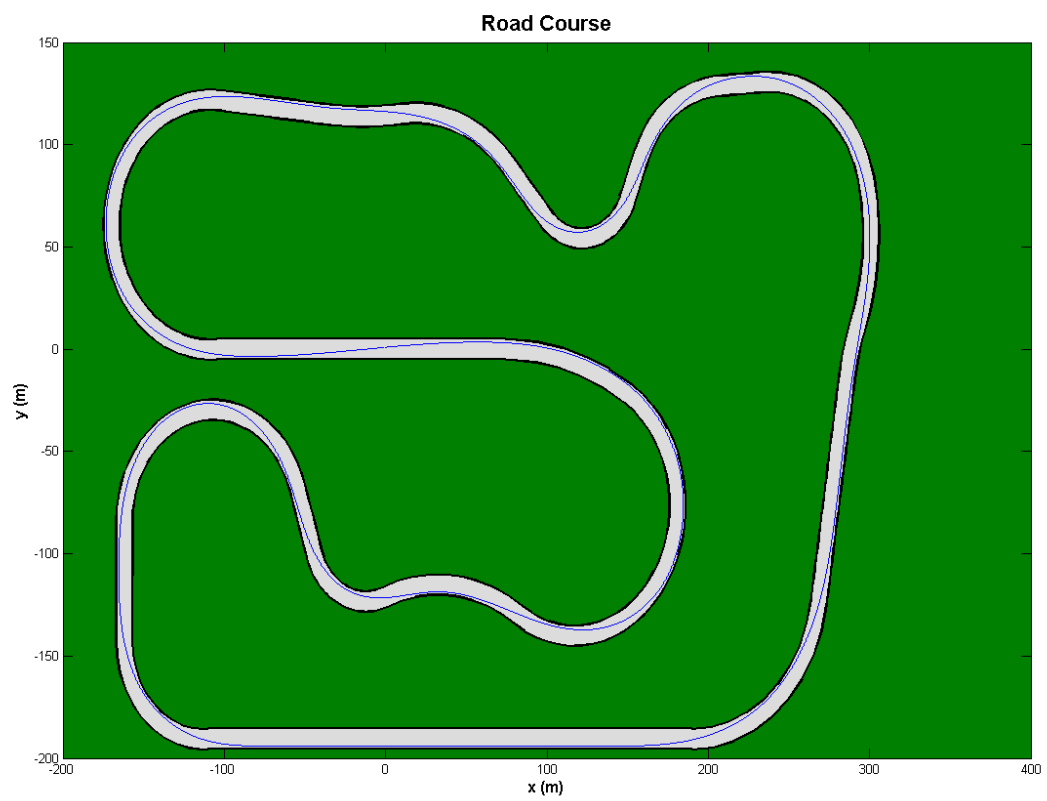
Figure 7: Road Course

Experiments on this course are performed at three different velocity profiles. The velocity profiles are generated to limit the longitudinal acceleration to within a range a vehicle could achieve while also limiting the theoretical kinematic centripetal acceleration of the vehicle. The longitudinal acceleration is limited to be between -4 m/s^2 and 3 m/s^2 for all experiments. The three velocity profiles are generated with kinematic centripetal accelerations limits of $0.1g$, $0.25g$ and $0.5g$. Despite these kinematic limitations, the actual lateral acceleration of the vehicle can be much greater in magnitude. The velocity profiles for this course are illustrated in Figure 8.
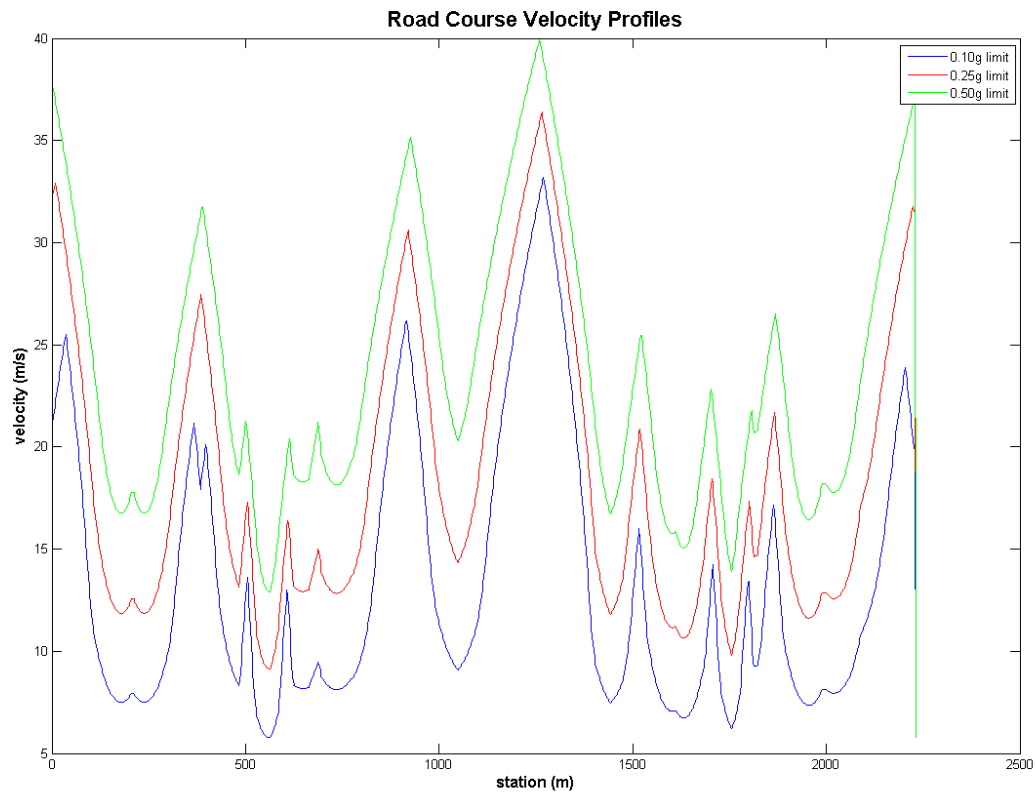
Figure 8: Velocity profiles used on the *Road Course*

2 Geometric Path Tracking

One of the most popular classes of path tracking methods found in robotics is that of geometric path trackers. These methods exploit geometric relationships between the vehicle and the path resulting in control law solutions to the path tracking problem. These techniques often make use of a look ahead distance to measure error ahead of the vehicle and can extend from simple circular arc calculations to more complicated calculations involving screw theory [19]. This section will describe the geometric vehicle model most commonly used by these methods and two of these methods: Pure Pursuit and the Stanley Method.

2.1 Geometric Vehicle Model
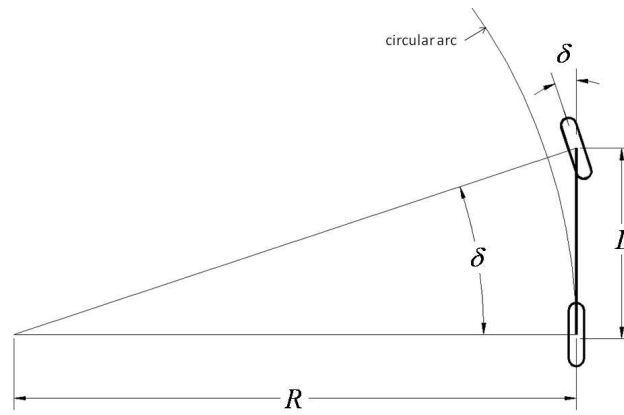
Figure 9: Geometric Bicycle Model

A common simplification of an Ackerman steered vehicle used for geometric path tracking is the bicycle model. Section 3.1 includes a detailed discussion of Ackerman steering and the kinematics of the bicycle model. For the purpose of geometric path tracking, it is only necessary to state that the bicycle model simplifies the four wheel car by combining the two front wheels together and the two rear wheels together to form a two wheeled model, like a bicycle. The second simplification is that the vehicle can only move on a plane. These simplifications result in a simple geometric relationship between the front wheel steering angle and the curvature that the rear axle will follow. As shown in Figure 9, this simple geometric relationship can be written as

$$\tan(\delta) = \frac{L}{R}, \tag{1}$$

8

where δ is the steering angle of the front wheel, L is the distance between the front axle and rear axle (wheelbase) and R is the radius of the circle that the rear axle will travel along at the given steering angle. This model approximates the motion of a car reasonably well at low speeds and moderate steering angles.

2.2 Pure Pursuit
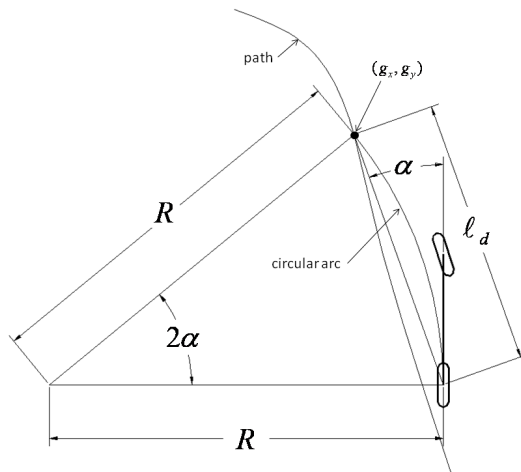
Figure 10: Pure Pursuit geometry

The pure pursuit [2] method and variations of it are among the most common approaches to the path tracking problem for mobile robots. The pure pursuit method consists of geometrically calculating the curvature of a circular arc that connects the rear axle location to a goal point on the path ahead of the vehicle. The goal point is determined from a look-ahead distance ℓ_d from the current rear axle position to the desired path. The goal point (g_x, g_y) is illustrated in Figure 10. The vehicle's steering angle δ can be determined using only the goal point location and the angle α between the vehicle's heading vector and the look-ahead vector. Applying the law of sines to Figure 10 results in

$$\frac{\ell_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)}$$

$$\frac{\ell_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R}{\cos(\alpha)}$$

$$\frac{\ell_d}{\sin(\alpha)} = 2R$$

9

or

$$\kappa = \frac{2\sin(\alpha)}{\ell_d},$$ (2)

where κ is the curvature of the circular arc. Using the simple geometric bicycle model of an Ackerman steered vehicle from Section 2.1, the steering angle can be written as

$$\delta = \tan^{-1}(\kappa L).$$ (3)

Using Eq. 2 and 3, the pure pursuit control law is given as

$$\delta(t) = \tan^{-1}\left(\frac{2L\sin(\alpha(t))}{\ell_d}\right).$$

A better understanding of this control law can be gained by defining a new variable, e_{ℓ_d} to be the lateral distance between the heading vector and the goal point resulting in the equation

$$\sin(\alpha) = \frac{e_{\ell_d}}{\ell_d}.$$

Eq. 2 can then be rewritten as

$$\kappa = \frac{2}{\ell_d^2}e_{\ell_d}.$$ (4)

Eq. 4 demonstrates that pure pursuit is a proportional controller of the steering angle operating on a cross track error some look-ahead distance in front of the vehicle and having a gain of $2/\ell_d^2$. In practice the gain (look-ahead distance) is independently tuned to be stable at several constant speeds, resulting in ℓ_d being assigned as a function of vehicle speed.

2.2.1 Tuning the Pure Pursuit Controller

To simplify tuning, the control law can be rewritten, scaling the look-ahead distance with the longitudinal velocity of the vehicle. Scaling the look-ahead distance in this manner is a common practice. Addionally, the look-ahead distance is commonly saturated at a minimum and maximum value. In this paper these value are set to 3m and 25m respectively. This results in

$$\delta(t) = \tan^{-1}\left(\frac{2L\sin(\alpha)}{kv_x(t)}\right).$$

Experiments are conducted on the *Lane Change Course*. Figure 11 illustrates the effects of the tuning parameter on tracking performance during these experiments. The tracking results are what one might expect. As k increases
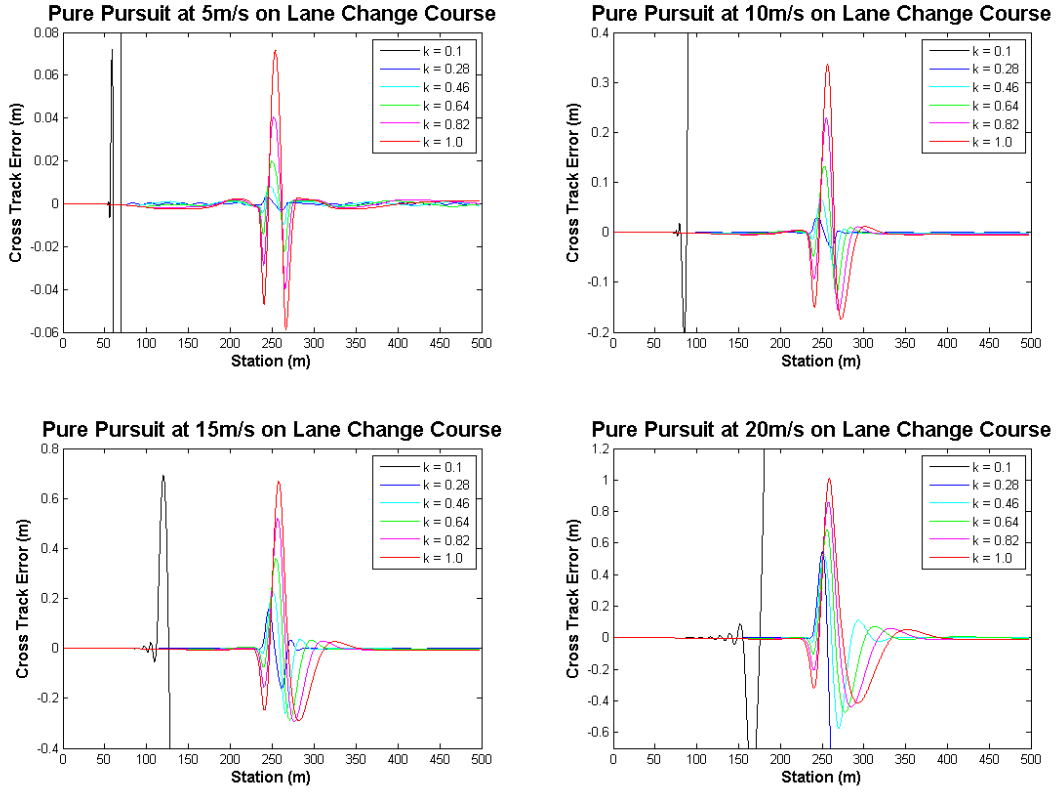
Figure 11: Pure Pursuit at multiple speeds and various gains on the lane change course

the look-ahead distance is increased and the tracking becomes less and less oscillatory. A short look-ahead distance provides more accurate tracking while a longer distance provides smoother tracking. It is clear that a k value that is too small will cause instability and a k value that is too large will cause poor tracking. Another characteristic of Pure Pursuit is that a sufficient look-ahead distance will result in "cutting corners" while executing turns on the path. The trade off between stability and tracking performance is difficult to balance with Pure Pursuit and will begin to seem course dependent. This is in part due to the fact that the Pure Pursuit method ignores the curvature of the path. Intuition would leave one to believe that the curvature of the path should somehow influence the look-ahead distance as well as the velocity (and perhaps even the current local cross track error). The effects of this will be seen in further tests. Pure Pursuit demonstrates a high level of robustness to the quick transient section of this test, even at a fairly high speed in the final test that would not be typical of most driving scenarios. This robustness is an important quality

of Pure Pursuit.

Experiments are conducted on the *Figure Eight Course*. Figure 12 illustrates the effects of the tuning parameter on tracking performance during these experiments. Again, similar results are obtained. The good characteristic to
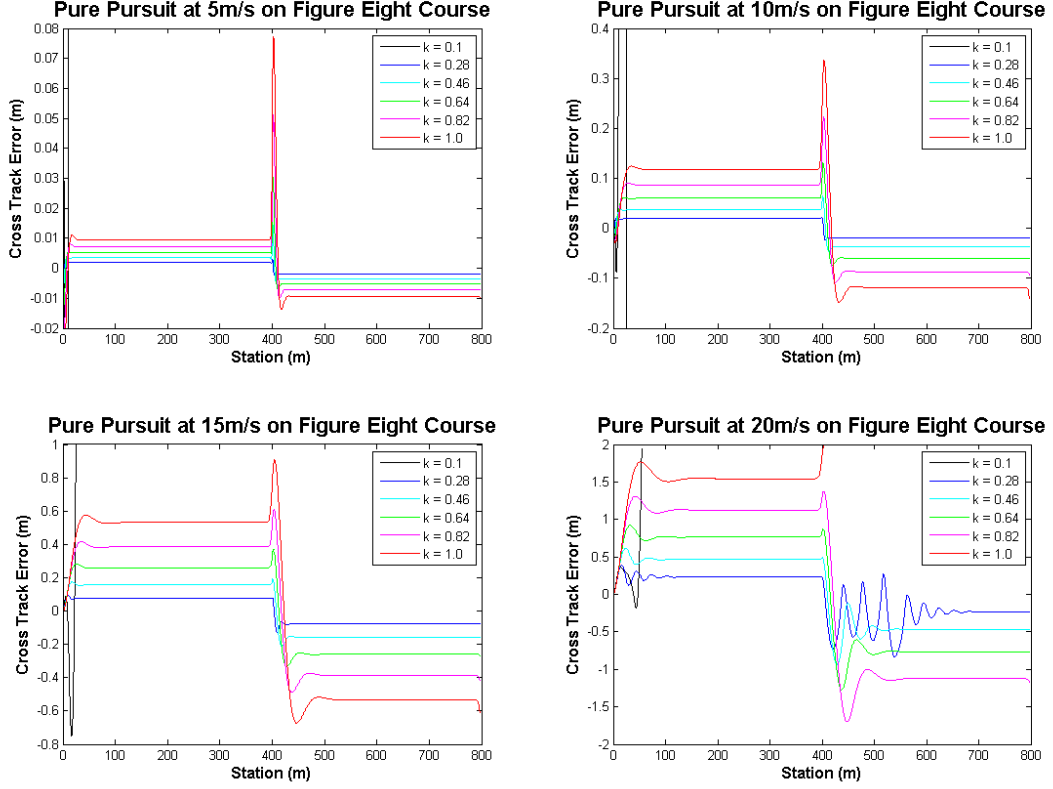
Figure 12: Pure Pursuit at multiple speeds and various gains on the figure eight course

point out during this test is that the Pure Pursuit method is robust to the discontinuity in the path. It is clear to see that on a constant curvature path a value of k can always be chosen that will perform well on that curvature, but could fail miserably when presented with a different curvature or different speed. This is because Pure Pursuit is simply calculating a circular arc based on the simple geometric model of the vehicle. In this case there is clearly a circular arc the vehicle would have to travel to stay on the path, but the vehicle travels a different circular arc than what the model would predict. This discrepancy comes from the model ignoring the vehicle's lateral dynamic characteristics that are more and more influential as the speed and/or curvature increases. It is easy to imagine that in constant curvature and constant speed tests this dynamic effect can be compensated for by increasing k until the circular arc that is computed is tighter than the circular arc of the path at a proportion that would cancel out the dynamic side slip of the vehicle.

This is a bad characteristic for tuning the tracker. One must be careful not to over tune on a course and test a variety of courses and speeds to find a k that can perform well over the operating space of the vehicle. This usually results in giving up accuracy to insure stability. Again, the final test is not typical of of most driving scenarios.

Experiments are conducted on the *Road Course*. Figure 13 illustrates the effects of the tuning parameter on tracking performance during these experiments. The "cutting corners" is apparent again in this test. However, Pure Pursuit can
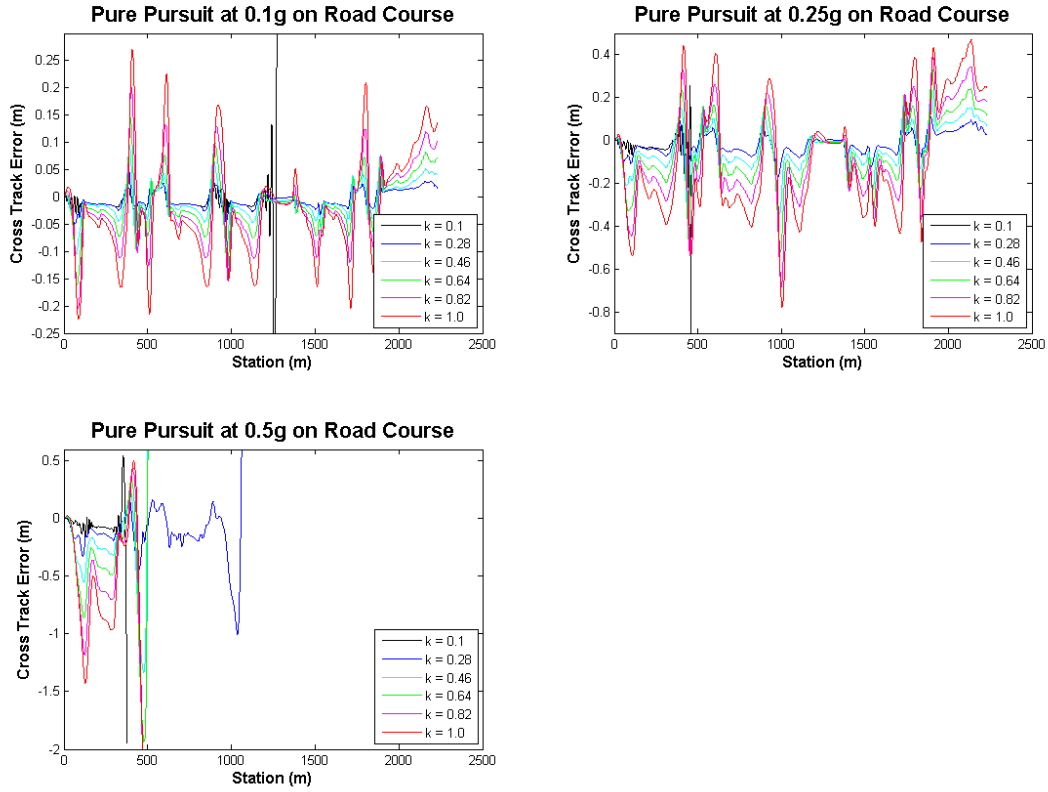
Figure 13: Pure Pursuit at multiple velocity profiles and various gains on the road course

be tuned to perform reasonably well on this course. The 0.5g test is simply an analysis tool and may not be a test that a tracker should be able to complete. It is valuable to know how and under what conditions a method will fail.

The characteristics of Pure Pursuit during tuning can be summarized as follows: Decreasing the look-ahead distance results in higher precision tracking and eventually oscillation, and increasing the look-ahead distance results in lower precision tracking and eventually stability.

2.3 Stanley Method
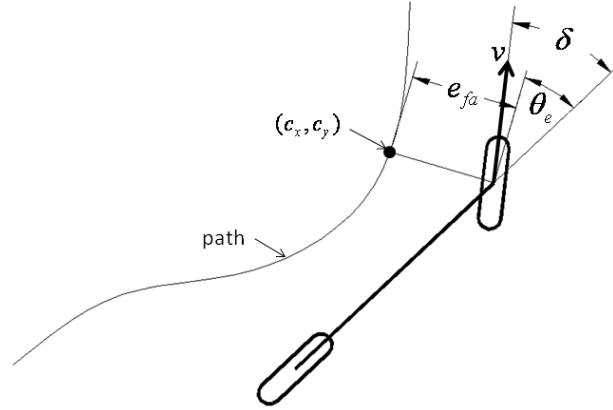
Figure 14: Stanley method geometry

The Stanley method [16] is the path tracking approach used by Stanford University's autonomous vehicle entry in the DARPA Grand Challenge, Stanley. The Stanley method is a nonlinear feedback function of the cross track error e_{fa}, measured from the center of the front axle to the nearest path point (c_x, c_y), for which exponential convergence can be shown [16]. Co-locating the point of control with the steered front wheels allows for an intuitive control law, where the first term simply keeps the wheels aligned with the given path by setting the steering angle δ equal to the heading error

$$\theta_e = \theta - \theta_p,$$

where θ is the heading of the vehicle and θ_p is the heading of the path at (c_x, c_y). When e_{fa} is non-zero, the second term adjusts δ such that the intended trajectory intersects the path tangent from (c_x, c_y) at $kv(t)$ units from the front axle. Figure 14 illustrates the geometric relationship of the control parameters. The resulting steering control law is given as

$$\delta(t) = \theta_e(t) + \tan^{-1}\left(\frac{ke_{fa}(t)}{v_x(t)}\right), \tag{5}$$

where k is a gain parameter. It is clear that the desired effect is achieved with this control law: As e_{fa} increases, the wheels are steered further towards the path.

14

2.3.1 Tuning the Stanley Controller

Experiments are conducted on the *Lane Change Course*. Figure 15 illustrates the effects of the tuning parameter on tracking performance during these experiments. The tracking results are what one might expect. As k is increased the
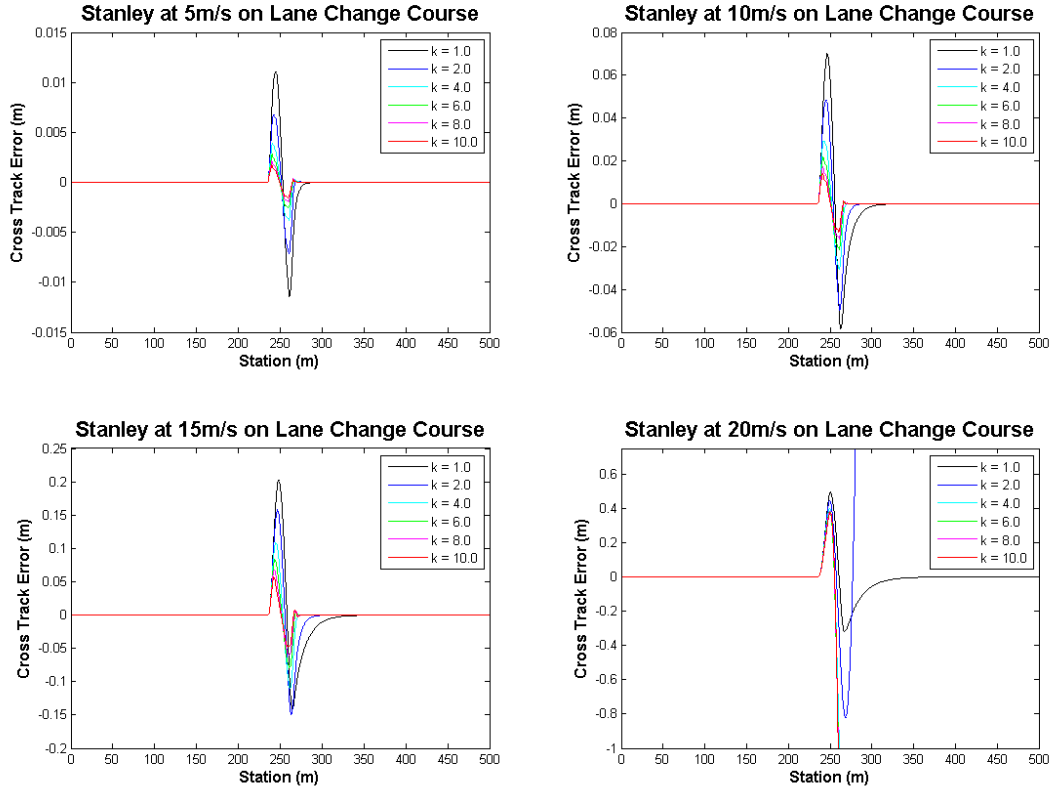
Figure 15: Stanley controller at multiple speeds and various gains on the lane change course

tracking performance improves. An upper limit exists to guarantee stability. It also appears that this method is not as robust to the lane change as Pure Pursuit and only values in the lower k values tested should be considered. However, keep in mind that the final test is an extreme maneuver.

Experiments are conducted on the *Figure Eight Course*. Figure 16 illustrates the effects of the tuning parameter on tracking performance during these experiments. In this case, a similar dynamic effect compensation from increasing
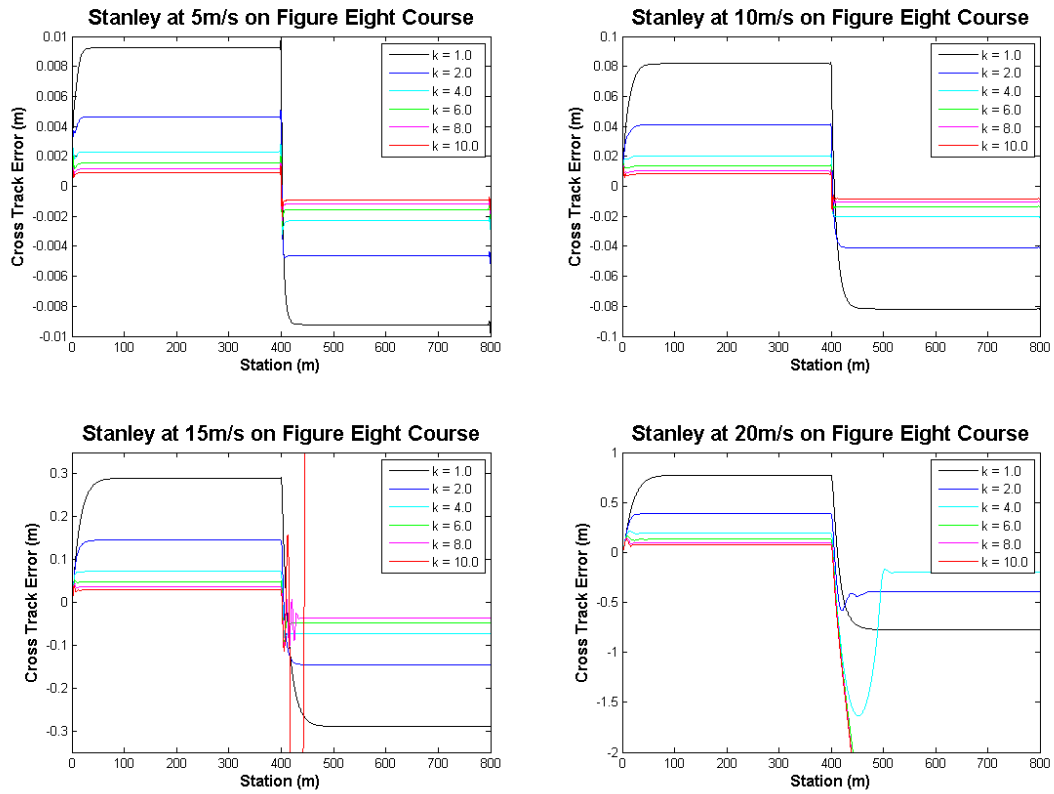
Figure 16: Stanley controller at multiple speeds and various gains on the figure eight course

the gain as in the Pure Pursuit test is seen, so care must be taken not to over tune to a course. Additionally, it is clear that the Stanley method has some trouble with the discontinuity of the path. This may not be a problem in practice since one can guarantee smooth planned paths.

Experiments are conducted on the *Road Course*. Figure 17 illustrates the effects of the tuning parameter on tracking performance during these experiments. It is clear that this method works quite well under varying normal
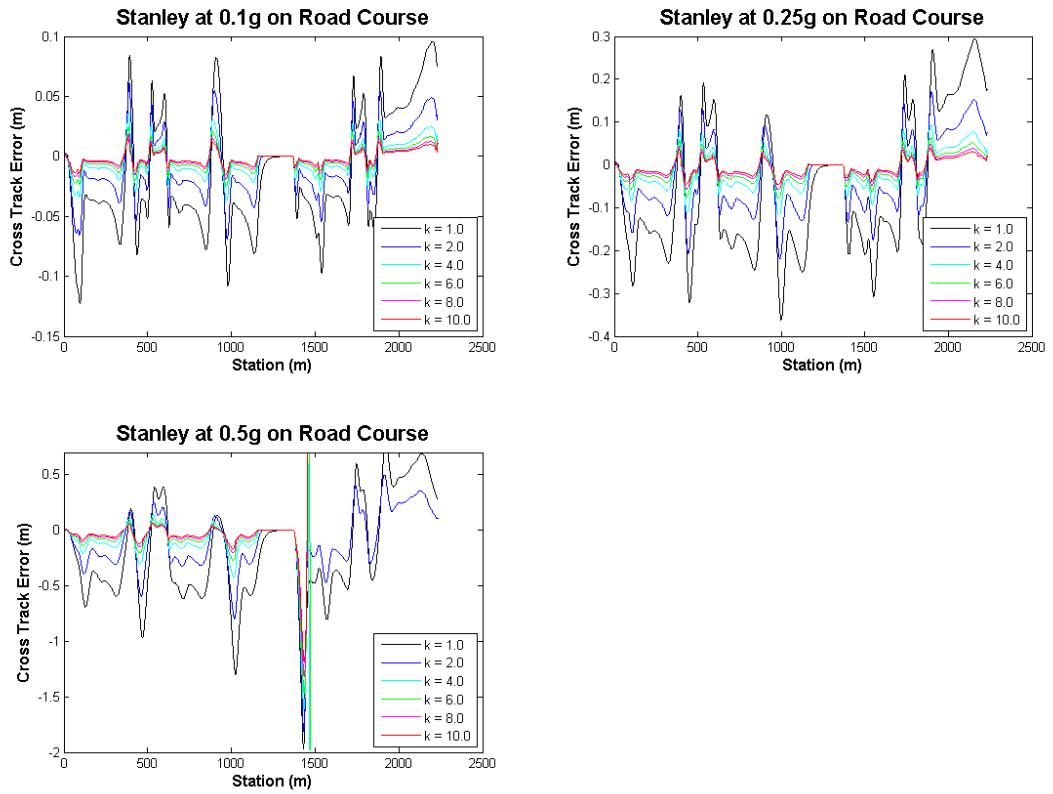
Figure 17: Stanley controller at multiple velocity profiles and various gains on the road course

driving scenarios. The final test illustrates that this method is well suited for higher speed driving when compared to Pure Pursuit.

17

3 Path Tracking Using a Kinematic Model

Simplifying the vehicle system model to a kinematic bicycle model is a common approximation used for robot motion planning, simple vehicle analysis and (as for the geometric methods) deriving intuitive control laws. This chapter presents the kinematic equations of motion for such a model. In addition, an important method from control theory for chained systems is applied to reformulate the equations of motion and provide a solution to the path tracking problem. The application of this theory results in the ability to use well known control theory tools for the design of controllers and stability analysis.

3.1 Kinematic Bicycle Model
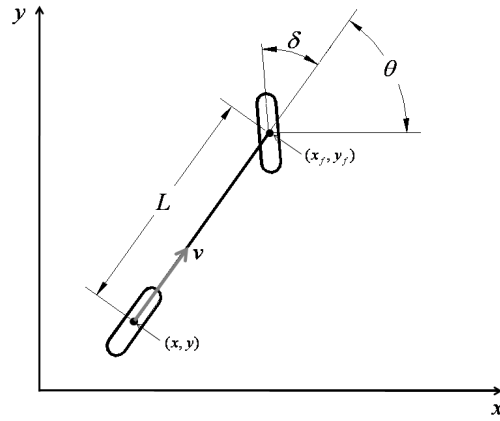
Figure 18: Kinematic bicycle model

The equations of motion for the kinematic bicycle model of a car is readily available in the literature [5, 11, 6, 4]. A derivation of the model is included here for completeness. The kinematic bicycle model collapses the left and right wheels into a pair of single wheels at the center of the front and rear axles as shown in Figure 18. The wheels are assumed to have no lateral slip and only the front wheel is steerable. Restricting the model to motion in a plane, the nonholonomic constraint equations for the front and rear wheels are:

$$\dot{x}_f \sin(\theta + \delta) - \dot{y}_f \cos(\theta + \delta) = 0 \qquad (6)$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \qquad (7)$$

where (x, y) is the global coordinate of the rear wheel, (x_f, y_f) is the global coordinate of the front wheel, θ is the orientation of the vehicle in the global frame, and δ is the steering angle in the body frame. As the front wheel is located at distance L from the rear wheel along the orientation of the vehicle, (x_f, y_f) may be expressed as:

$$x_f = x + L\cos(\theta)$$

$$y_f = y + L\sin(\theta)$$

Eliminating (x_f, y_f) from Eq. 6:

$$
\begin{aligned}
0 &= \frac{d(x + L\cos(\theta))}{dt}\sin(\theta + \delta) - \frac{d(y + L\sin(\theta))}{dt}\cos(\theta + \delta) \\
&= (\dot{x} - \dot{\theta}L\sin(\theta))\sin(\theta + \delta) - (\dot{y} + \dot{\theta}L\cos(\theta))\cos(\theta + \delta) \\
&= \dot{x}\sin(\theta + \delta) - \dot{y}\cos(\theta + \delta) \\
&\quad - \dot{\theta}L\sin(\theta)(\sin(\theta)\cos(\delta) + \cos(\theta)\sin(\delta)) \\
&\quad - \dot{\theta}L\cos(\theta)(\cos(\theta)\cos(\delta) - \sin(\theta)\sin(\delta)) \\
&= \dot{x}\sin(\theta + \delta) - \dot{y}\cos(\theta + \delta) \\
&\quad - \dot{\theta}L\sin^2(\theta)\cos(\delta) - \dot{\theta}L\cos^2(\theta)\cos(\delta) \\
&\quad - \dot{\theta}L\sin(\theta)\cos(\theta)\sin(\delta) + \dot{\theta}L\cos(\theta)\sin(\theta)\sin(\delta) \\
&= \dot{x}\sin(\theta + \delta) - \dot{y}\cos(\theta + \delta) - \dot{\theta}L(\sin^2(\theta) + \cos^2(\theta))\cos(\delta) \\
&= \dot{x}\sin(\theta + \delta) - \dot{y}\cos(\theta + \delta) - \dot{\theta}L\cos(\delta)
\end{aligned}
$$

The nonholonomic constraint on the rear wheel, Eq. 7, is satisfied by $\dot{x} = \cos(\theta)$ and $\dot{y} = \sin(\theta)$ and any scalar multiple thereof. This scalar corresponds to the longitudinal velocity v, such that

$$\dot{x} = v\cos(\theta) \tag{8}$$

$$\dot{y} = v\sin(\theta). \tag{9}$$

Applying this to the constraint on the front wheel, Eq. 6, yields a solution for $\dot{\theta}$

$$
\begin{aligned}
\dot{\theta} &= \frac{\dot{x}\sin(\theta + \delta) - \dot{y}\cos(\theta + \delta)}{L\cos(\delta)} \\
&= \frac{v\cos(\theta)(\sin(\theta)cos(\delta) + \cos(\theta)\sin(\delta))}{L\cos(\delta)}
\end{aligned}
$$

19

$$-\frac{v\sin(\theta)(\cos(\theta)\cos(\delta)-\sin(\theta)\sin(\delta))}{L\cos(\delta)}$$
$$=\frac{v(\cos^2(\theta)+\sin^2(\theta))\sin(\delta)}{L\cos(\delta)}$$
$$=\frac{v\tan(\delta)}{L} \tag{10}$$

The instantaneous radius of curvature R of the vehicle determined from v and $\dot{\theta}$ leads to the previously introduced Eq. 1:

$$R=\frac{v}{\dot{\theta}}$$
$$\frac{v\tan(\delta)}{L}=\frac{v}{R}$$
$$\tan(\delta)=\frac{L}{R}$$

For the purpose of control it is useful to write the kinematic model from Eqs. 8, 9 and 10 in the two-input driftless form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \left(\frac{\tan(\delta)}{L}\right) \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\delta}, \tag{11}$$

where v and $\dot{\delta}$ are the longitudinal velocity and the angular velocity of the steered wheel respectively.

3.1.1 Path Coordinates

For path tracking, it is useful to express the bicycle model with respect to the path as in [6]. Defining the path as a function of its length s, let $\theta_p(s)$ denote the angle between the path tangent at s and the global x axis. Orientation error θ_e of the vehicle with respect to the path is defined as

$$\theta_e = \theta - \theta_p(s).$$

The curvature along the path is defined as

$$\kappa(s) = \frac{e_{ra}\theta_p(s)}{ds}.$$
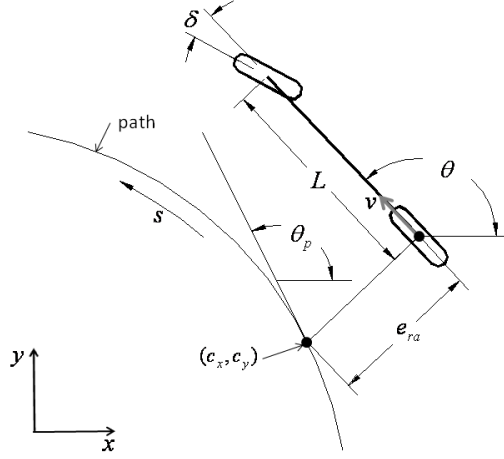
Figure 19: Kinematic bicycle model in path coordinates

Multiplying both sides by $\dot{s}$

$$\dot{\theta}_p(s) = \kappa(s)\dot{s}.$$

Given e_{ra} as the orthogonal distance from the center of the rear axle to the path, $\dot{s}$ and $\dot{e}_{ra}$ are

$$\dot{s} = v_1 \cos\left(\theta_e\right) + \dot{\theta}_p e_{ra}$$

$$\dot{e}_{ra} = v_1 \sin\left(\theta_e\right).$$

Substituting (s, e_{ra}, θ_e) for (x, y, θ) in Eq. 11 defines the kinematic model in path coordinates as

$$
\begin{bmatrix} \dot{s} \\ \dot{e}_{ra} \\ \dot{\theta}_e \\ \dot{\delta} \end{bmatrix}
=
\begin{bmatrix}
\left(\frac{\cos(\theta_e)}{1 - \dot{e}_{ra}\kappa(s)} \right) \\
\sin\left(\theta_e\right) \\
\left(\frac{\tan(\delta)}{L} - \frac{\kappa(s)\cos(\theta_e)}{1 - \dot{e}_{ra}\kappa(s)} \right) \\
0
\end{bmatrix}
v
+
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
\dot{\delta}.
\tag{12}
$$

Figure 19 illustrates the kinematic model in path coordinates.

3.2 Kinematic Controller

An interesting and useful method for controlling kinematic models of nonholonomic systems can be found in [6]. The method is applied to car-like robots, however it can be used to control many kinematic models describing a variety

of wheeled mobile robots and is easily extended to control n trailers attached to the system. This section repeats the controller design found in [6] using nomenclature that is consistent with this paper for completeness.

3.2.1 Chained Form

Canonical forms for kinematic models of nonhololonomic systems are commonly used in controller design. One of these canonical structures is the chained form. The general two-input driftless control system

$$\dot{x}_1 = u_1 \qquad (13)$$
$$\dot{x}_2 = u_2$$
$$\dot{x}_3 = x_2 u_1$$
$$\vdots$$
$$\dot{x}_n = x_{n-1} u_1$$

is called $(2, n)$ single-chained form. It turns out that this type of nonlinear chained system has a strong underlying linear structure. This linear structure allows a designer to take advantage of some well known tools in control theory. The underlying linear structure clearly appears when u_1 is assigned as a function of time, and is not considered as a control variable. Under these circumstances, system 13 becomes a single-input, time-varying, linear system that has been shown to be controllable.

The conditions for converting a two-input system like 13 into chained form have been given in [9]. This conversion consists of a change of coordinates $x = \phi(q)$, and an invertible input transformation $v = \beta(q)u$. It has been shown that this conversion can always be done successfully for nonholonomic systems with $m = 2$ inputs and $n = 3$ or 4 generalized coordinates. The kinematic model 12 can be put in chained canonical form by using the following change of coordinates

$$x_1 = s$$

$$x_2 = -\kappa'(s)e_{ra}\tan\left(\theta_e\right) - \kappa(s)(1 - e_{ra}\kappa(s))\frac{1 + \sin^2\left(\theta_e\right)}{\cos^2\left(\theta_e\right)} + \frac{(1 - e_{ra}\kappa(s))^2\tan\left(\delta\right)}{L\cos^3\left(\theta_e\right)}$$

$$x_3 = (1 - e_{ra}\kappa(s))\tan\left(\theta_e\right)$$

$$x_4 = e_{ra}$$

with the input transformation

$$v = \frac{1 - e_{ra}\kappa(s)}{\cos\left(\theta_e\right)}u_1$$

$$\dot{\delta} = \alpha_2(u_2 - \alpha_1 u_1)$$

where α_1 and α_2 are defined as

$$\alpha_1 = \frac{\partial x_2}{\partial s} + \frac{\partial x_2}{\partial e_{ra}}(1 - e_{ra}\kappa(s))\tan(\theta_e) + \frac{\partial x_2}{\partial \theta_e}\left(\frac{\tan(\delta)(1 - e_{ra}\kappa(s))}{L\cos(\theta_e)} - \kappa(s)\right)$$

$$\alpha_2 = \frac{L\cos^3(\theta_e)\cos^2(\delta)}{(1 - e_{ra}\kappa(s))^2}$$

3.2.2 Smooth Time-Varying Feedback Control

The following smooth feedback stabilization method was originally proposed in [13] and its application to vehicle path tracking was further developed in [6]. This method takes advantage of the internal structure of chained systems so as to break the solution into two design phases. The first phase assumes that one control input (that satisfies some technical requirements) is given, while the additional control input is used to stabilize the remaining sub-vector of the system state. The second phase simply consists of specifying the first control input so as to guarantee convergence while maintaining stability.

For convenience, the variables of the chained form are reordered by letting

$$\chi = (\chi_1, \chi_2, \chi_3, \chi_4) = (x_1, x_4, x_3, x_2)$$

so that the chained form system can be written as

$$\dot{\chi}_1 = u_1 \tag{14}$$
$$\dot{\chi}_2 = \chi_3 u_1$$
$$\dot{\chi}_3 = \chi_4 u_1$$
$$\dot{\chi}_4 = u_2$$

The reordering exchanges x_2 and x_4 so the position of the rear axle is (χ_1, χ_2). Now let $\chi = (\chi_1, \chi_2)$, where $\chi_2 = (\chi_2, \chi_3, \chi_4)$. The goal of the controller is to stabilize χ_2 to zero.

23

3.2.3 Input Scaling

If u_1 is assigned as a function of time, the chained system 14 can be written as

$$\dot{\tilde{\chi}}_1 = 0 \qquad (15)$$

$$\dot{\chi}_2 = \begin{bmatrix} 0 & u_1(t) & 0 \\ 0 & 0 & u_1(t) \\ 0 & 0 & 0 \end{bmatrix} \chi_2 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2,$$

with

$$\tilde{\chi}_1 = \chi_1 - \int_0^t u_1(t)dt.$$

The first equation in 15 is not controllable when u_1 is assigned a priori. However, the structure of the differential equations for χ_2 is interesting. This structure looks like the familiar controllable canonical form for linear systems. Another interesting observation is that system 15 becomes time-invariant when u_1 is constant and nonzero. Under this condition, the second part of system 15 becomes controllable. More importantly, it will always be controllable whenever $u_1(t)$ is a piecewise-continuous, bounded, and strictly positive (or negative) function. Under these assumptions, x_1 varies monotonically with time and differentiation with respect to time can be replaced by differentiation with respect to χ_1, meaning

$$\frac{d}{dt} = \frac{d}{d\chi_1}\dot{\chi}_1 = \frac{d}{d\chi_1}u_1,$$

and thus

$$sign(u_1)\frac{d}{d\chi_1} = \frac{1}{|u_1|} \cdot \frac{d}{dt}.$$

This change of variable is equivalent to an input scaling procedure [6]. This allows the second part to be rewritten as

$$\chi_2^{[1]} = sign(u_1)\chi_3 \qquad (16)$$

$$\chi_3^{[1]} = sign(u_1)\chi_4$$

$$\chi_4^{[1]} = sign(u_1)u_2',$$

with the definitions

$$\chi_i^{[j]} = sign(u_1)\frac{d^j\chi_i}{d\chi_1^j}$$

24

and

$$u_2' = \frac{u_2}{u_1}.$$

System 16 is linear and time-invariant. It has an equivalent input-output representation of

$$\chi_2^{[n-1]} = sign(u_1)^{n-1} u_2'.$$

Such a system is controllable and admits an exponentially stable linear feedback in the form

$$u_2'(\chi_2) = -sign(u_1)^{n-1} \sum_{i=1}^{n-1} k_i \chi_2^{[i-1]}, \tag{17}$$

where the gains $k_i > 0$ are chosen so as to satisfy the Hurwitz stability criterion. Hence, the time-varying control

$$u_2(\chi_2, t) = u_1(t) u_2'(\chi_2)$$

globally asymptotically stabilizes the origin $\chi_2 = 0$.

This approach leads to a solution to the path tracking problem for Ackerman steered vehicles. By transforming system 11 into path coordinates and reordering the variables as shown, χ_1 represents the arc length s along the path, χ_2 is the distance e_{ra} between the center of the rear axle and the path, while χ_3 and χ_4 are related to the steering angle δ and to the relative orientation θ_e between the path and the vehicle. Path tracking consists of zeroing the χ_2, χ_3 and χ_4 states independently from χ_1. Then, for any piecewise-continuous, bounded, and strictly positive (or negative) u_1, equation 17 can be written as

$$u_2'(\chi_2, \chi_3, \chi_4) = -sign(u_1)[k_1\chi_2 + k_2 sign(u_1)\chi_3 + k_3\chi_4]. \tag{18}$$

Using equation 18, the final path tracking feedback control law is obtained as

$$u_2(\chi_2, \chi_3, \chi_4, t) = -k_1 |u_1(t)| \chi_2 - k_2 u_1(t)\chi_3 - k_3 |u_1(t)| \chi_4.$$

3.2.4 Tuning the Kinematic Controller

It has been shown in [6] that stability can be obtained by choosing the gains based on the following relationships

$$k_1 = k^3$$

25

$$k_2 = 3k^2$$

$$k_3 = 3k.$$

This relationship gives a single gain parameter to adjust, resulting in a manageable means to appropriately tune the controller.

Experiments are conducted on the *Lane Change Course*. Figure 20 illustrates the effects of the tuning parameter on tracking performance during these experiments. It can be seen that the tracking can be improved by increasing
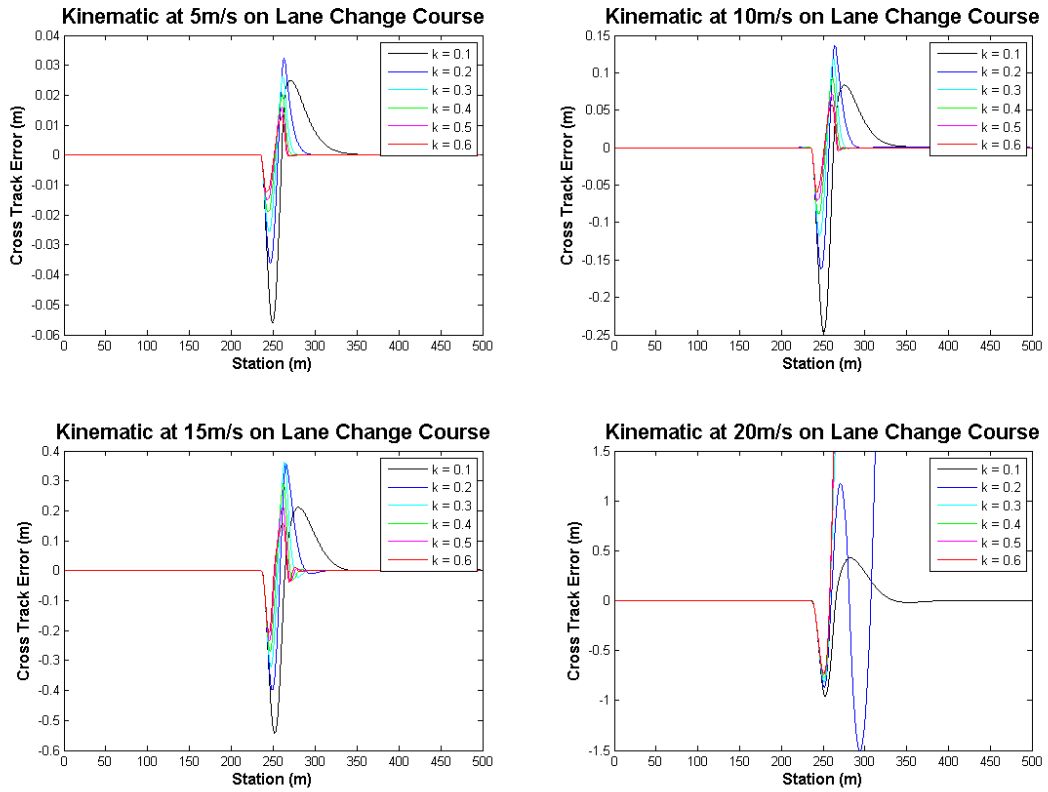
Figure 20: Kinematic controller at multiple speeds and various gains on the lane change course

k. The same kind of trade off between stability and performance is apparent as before. This method tracks the path accurately at low speeds, but has problems at higher speeds due to the dynamics being neglected. It can be seen that the robustness to the lane change is not as good as Pure Pursuit and is close to that of the Stanley method.

Experiments are conducted on the *Figure Eight Course*. Figure 21 illustrates the effects of the tuning parameter on tracking performance during these experiments. Again, it can be seen that the performance drops off as speed is
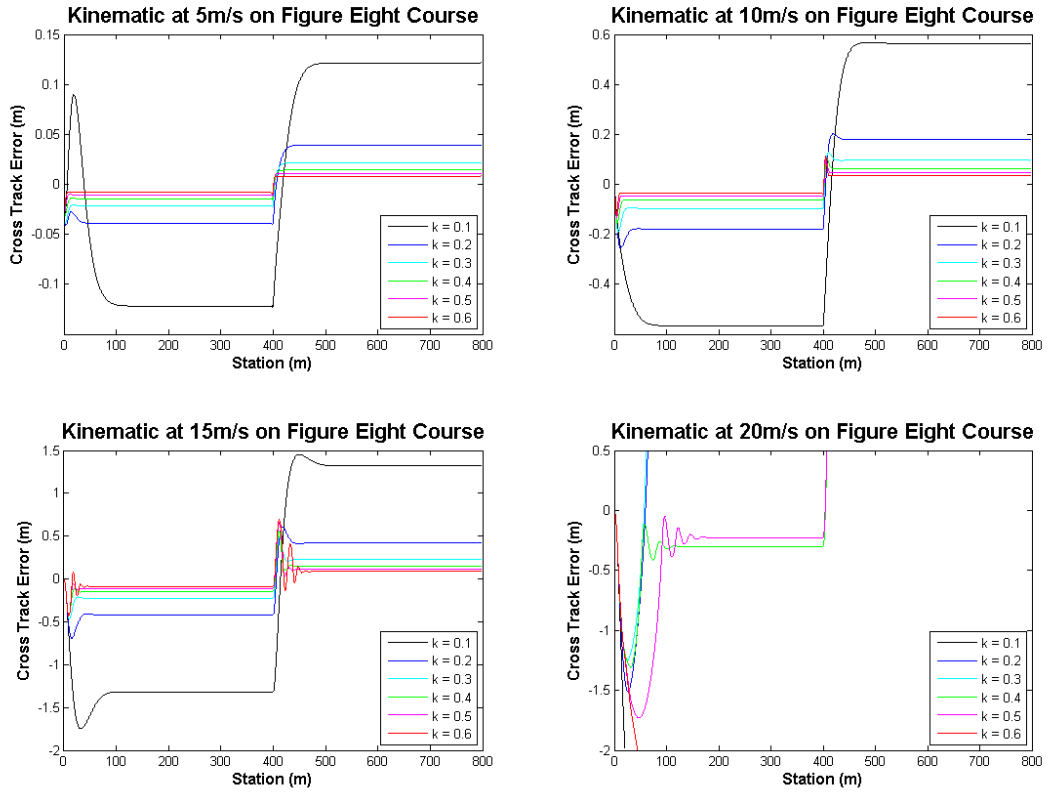
Figure 21: Kinematic controller at multiple speeds and various gains on the figure eight course

increased. The kinematic method is considerably effected by the discontinuity in a similar way as the Stanley method.

Experiments are conducted on the *Road Course*. Figure 22 illustrates the effects of the tuning parameter on tracking performance during these experiments. This method works quite well under varying normal driving scenarios. The
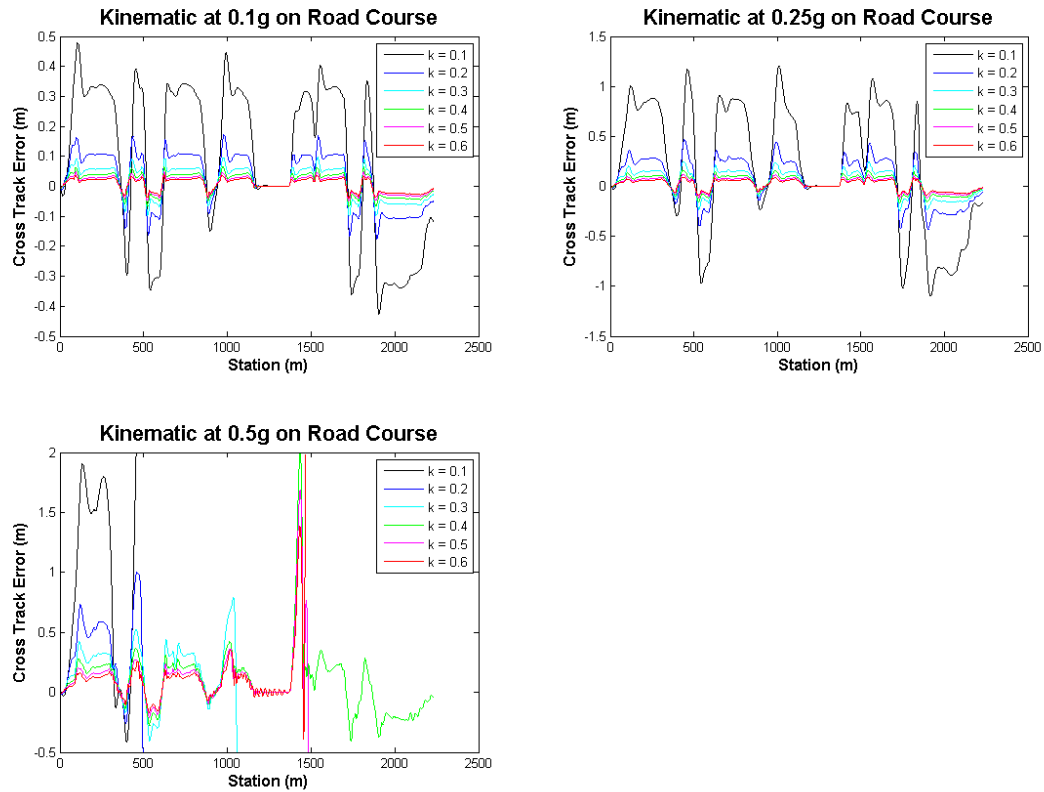
Figure 22: Kinematic controller at multiple velocity profiles and various gains on the road course

final test illustrates that this method is similar, although not as good, to the Stanley method in higher speed driving. An interesting point in the final test is that it was not the extremes of the range of k that was able to complete the course, rather it was in the middle.

4   Path Tracking Control Using a Dynamic Model

It is clear that neglecting vehicle dynamics in the previous models has a negative impact on tracking performance as speeds are increased and path curvature varies. The dynamics of a car are very complicated and high fidelity models are very non-linear, discontinuous and computationally expensive. This chapter derives a simple lateral dynamics model that approximates the dynamic effects and enables the design of linear path tracking controllers.

28

4.1 Dynamic Vehicle Model
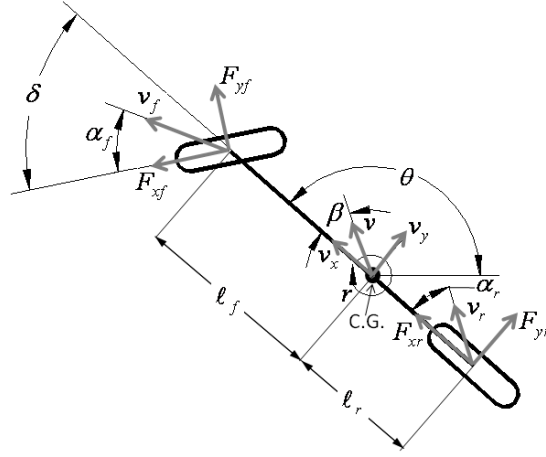
Figure 23: Dynamic Bicycle Model

The bicycle model introduced in the last section can be extended to include dynamics. Lateral forces on the vehicle are of primary concern as path tracking is an exercise in lateral control. For this section, longitudinal velocity is assumed to be controlled separately. Summing the lateral forces illustrated in Figure 23, given vehicle mass m, yields

$$F_{yf} \cos(\delta) - F_{xf} \sin(\delta) + F_{yr} = m(\dot{v}_y + v_x r).$$
(19)

Considering only motion in the plane, a center of gravity C.G. along the center line of the vehicle, and yaw inertia I_z, balancing the yaw moments gives

$$\ell_f (F_{yf} \cos(\delta)) - \ell_r (F_{yr} - F_{xf} \sin(\delta)) = I_z \dot{r}.$$
(20)

where r is the angular rate about the yaw axis. Without the constraint on lateral slip from the last section, the slip angles of the tires are given as

$$\alpha_f = \tan^{-1} \left(\frac{v_y + \ell_f r}{v_x} \right) - \delta$$

$$\alpha_r = \tan^{-1} \left(\frac{v_y - \ell_r r}{v_x} \right).$$

29

Modelling the force generated by the wheels as linearly proportional to the slip angle, the lateral forces are defined as

$$F_{yf} = -c_f \alpha_f \tag{21}$$

$$F_{yr} = -c_r \alpha_r. $$

Assuming a constant longitudinal velocity, $\dot{v}_x = 0$, allows the simplification

$$F_{xf} = 0. \tag{22}$$

Substituting Eqs. 21 and 22 into Eqs. 19 and 20 and solving for for $\dot{v}_y$ and $\dot{r}$

$$\dot{v}_y = \frac{-c_f \left[\tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right) - \delta \right] \cos(\delta) - c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)}{m} - v_x r \tag{23}$$

$$\dot{r} = \frac{-\ell_f c_f \left[\tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right) - \delta \right] \cos(\delta) + \ell_r c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)}{I_z} \tag{24}$$

gives the dynamic bicycle model.

4.1.1   Linearized Dynamic Bicycle Model

To apply linear control methods to the dynamic bicycle model, the model must be linearized. Applying small angle assumptions to Eqs. 23 and 24 gives

$$\dot{v}_y = \frac{-c_f v_y - c_f \ell_f r}{m v_x} + \frac{c_f \delta}{m} + \frac{-c_r v_y + c_r \ell_r r}{m v_x} - v_x r$$

$$\dot{r} = \frac{-\ell_f c_f v_y - \ell_f^2 c_f r}{I_z v_x} + \frac{\ell_f c_f \delta}{I_z} + \frac{\ell_r c_r v_y - \ell_r^2 c_r r}{I_z v_x}.$$

Collecting terms results in

$$\dot{v}_y = \frac{-(c_f + c_r)}{m v_x} v_y + \left[\frac{(\ell_r c_r - \ell_f c_f)}{m v_x} - v_x \right] r + \frac{c_f}{m} \delta \tag{25}$$

$$\dot{r} = \frac{\ell_r c_r - \ell_f c_f}{I_z v_x} v_y + \frac{-\left(\ell_f^2 c_f + \ell_r^2 c_r\right)}{I_z v_x} r + \frac{\ell_f c_f}{m} \delta. \tag{26}$$

Finally, the linear dynamic bicycle model can be written in state space form as

$$
\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{-(c_f+c_r)}{mv_x} & \frac{\ell_r c_r - \ell_f c_f}{mv_x} - v_x \\ \frac{\ell_r c_r - \ell_f c_f}{I_z v_x} & \frac{-\left(\ell_f^2 c_f + \ell_r^2 c_r\right)}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{\ell_f c_f}{m} \end{bmatrix} \delta.
$$

4.1.2 Path Coordinates
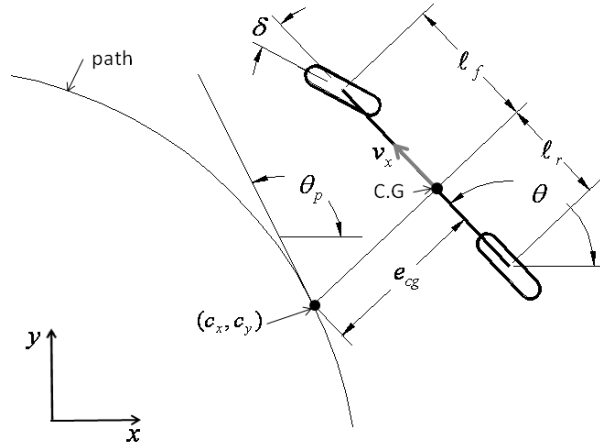
Figure 24: Dynamic Bicycle Model in path coordinates

As with the kinematic bicycle model, it is useful to express the dynamic bicycle model with respect to the path. With the constant longitudinal velocity assumption, the yaw rate derived from the path $r(s)$ is defined as

$$
r(s) = \kappa(s)v_x.
$$

Path derived lateral acceleration $\dot{v}_y(s)$ follows as

$$
\dot{v}_y(s) = \kappa(s)v_x^2.
$$

Letting e_{cg} be the orthogonal distance of the C.G. to the path,

$$
\begin{aligned}
\ddot{e}_{cg} &= (\dot{v}_y + v_x r) - \dot{v}_y(s) \\
&= \dot{v}_y + v_x(r - r(s)) \\
&= \dot{v}_y + v_x \dot{\theta}_e.
\end{aligned}
\tag{27}
$$

31

and

$$\dot{e}_{cg} = v_y + v_x \sin(\theta_e) \tag{28}$$

where θ_e was previously defined as $\theta - \theta_p(s)$. Substituting (e_{cg}, θ_p) into Eqs. 25 and 26 yields

$$
\ddot{e}_{cg} - v_x\dot{\theta}_e = \frac{-(c_f + c_r)}{mv_x}(\dot{e}_{cg} - v_x\theta_e)
$$
$$
+ \left[\frac{\ell_r c_r - \ell_f c_f}{mv_x} - v_x\right](\dot{\theta}_e + r(s)) + \frac{c_f}{m}\delta
$$
$$
\ddot{e}_{cg} = \frac{-(c_f + c_r)}{mv_x}\dot{e}_{cg} + \frac{c_f + c_r}{m}\theta_e
$$
$$
+ \frac{\ell_r c_r - \ell_f c_f}{mv_x}\dot{\theta}_e + \left[\frac{\ell_r c_r - \ell_f c_f}{mv_x} - v_x\right]r(s) + \frac{c_f}{m}\delta
$$
$$
\ddot{\theta}_e + \dot{r}(s) = \frac{\ell_r c_r - \ell_f c_f}{I_z v_x}(\dot{e}_{cg} - v_x\theta_e)
$$
$$
+ \frac{-\left(\ell_f^2 c_f + \ell_r^2 c_r\right)}{I_z v_x}(\dot{\theta}_e + r(s)) + \frac{\ell_f c_f}{m}\delta
$$
$$
\ddot{\theta}_e = \frac{\ell_r c_r - \ell_f c_f}{I_z v_x}\dot{e}_{cg} + \frac{\ell_f c_f - \ell_r c_r}{I_z}\theta_e
$$
$$
+ \frac{-\left(\ell_f^2 c_f + \ell_r^2 c_r\right)}{I_z v_x}(\dot{\theta}_e + r(s)) + \frac{\ell_f c_f}{m}\delta - \dot{r}(s).
$$

The state space model in tracking error variables is therefore given by

$$
\begin{bmatrix} \dot{e}_{cg} \\ \ddot{e}_{cg} \\ \dot{\theta}_e \\ \ddot{\theta}_e \end{bmatrix}
=
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & \frac{-(c_f + c_r)}{mv_x} & \frac{c_f + c_r}{m} & \frac{\ell_f c_f - \ell_f c_f}{mv_x} \\
0 & 0 & 0 & 1 \\
0 & \frac{\ell_r c_r - \ell_f c_f}{I_z v_x} & \frac{\ell_f c_f - \ell_r c_r}{I_z} & \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x}
\end{bmatrix}
\begin{bmatrix} e_{cg} \\ \dot{e}_{cg} \\ \theta_e \\ \dot{\theta}_e \end{bmatrix}
\tag{29}
$$

$$
+
\begin{bmatrix} 0 \\ \frac{c_f}{m} \\ 0 \\ \frac{\ell_f c_f}{I_z} \end{bmatrix}\delta
+
\begin{bmatrix} 0 \\ \frac{\ell_r c_r - \ell_f c_f}{mv_x} - v_x \\ 0 \\ \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} \end{bmatrix}r(s)
$$

4.1.3   Model Parameter Identification

Unlike the previous models described in Chapters 2 and 3, the dynamic bicycle model has parameters that are not as convenient to directly measure. However, a workable estimate can be obtained using commonly available tools.

The measurement of the vehicle's split mass, utilizing four scales under each wheel, is necessary to estimate the vehicle's total mass, C.G. location and moment of inertia. The total mass of the vehicle is simply the sum of the measurements at front and rear, left and right corners.

$$m = m_{fr} + m_{fl} + m_{rr} + m_{rl}$$

Given the modelling of the front and rear wheels as a single wheel at the center of each axle, an assumption is also made that the vehicle's mass is laterally evenly distributed. Then a useful intermediate quantity is the sum of the measurements at the front and rear,

$$m_f = m_{fr} + m_{fl}$$

and

$$m_r = m_{rr} + m_{rl}.$$

From this and a measurement of the wheel base L, the location of the vehicle's C.G., described by distances ℓ_f and ℓ_r from the front and rear axles along the center line can be estimated as

$$\ell_f = L \left(1 - \frac{m_f}{m} \right)$$

and

$$\ell_r = L \left(1 - \frac{m_r}{m} \right).$$

The vehicle's moment of inertia is approximated by treating the vehicle as two point masses joined by a mass-less rod. The moment of inertia for the vehicle is then given as

$$I_z = \ell_f \ell_r (m_f + m_r)$$

Finally, the cornering stiffness parameters c_f and c_r must be identified. These parameters can be obtained from data sheets produced for the specific tire, if available. Figure 25 is an example of what this tire data looks like. Notice that the slip angle of the tire changes nonlinearly as the lateral force on the tire changes and as the normal
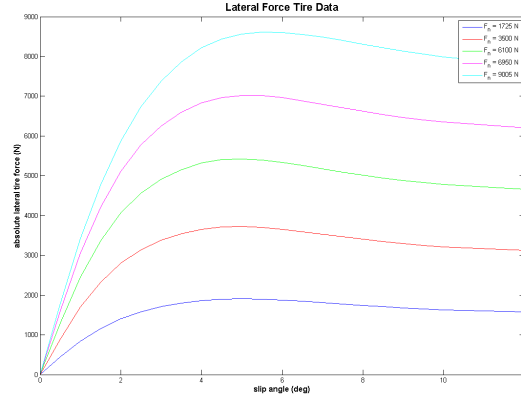
Figure 25: Example of lateral force tire data
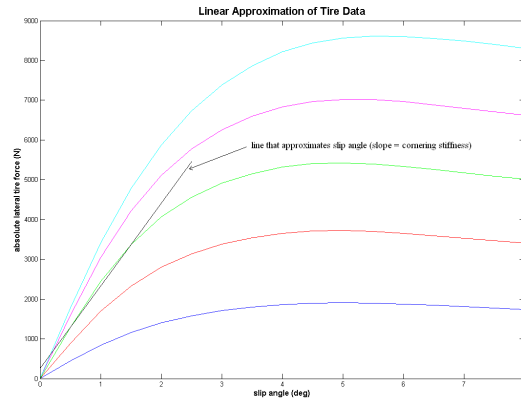
Figure 26: Linear approximation of the lateral force tire data

force on the tire varies. If a data sheet is used to obtain the cornering stiffness values, a constant value describing the slope in the most linear region of the data at a nominal normal force must be chosen. Figure 26 illustrates this linear approximation. Understanding this approximation is important to understanding the limitations of the dynamic bicycle model as a whole. Note that the cornering stiffness values from the dynamic bicycle model are double the value from a data sheet since the two tires are being treated as one.

If this data is not readily available, a method to estimate it is required. Starting with Eqs. 23 and 24, repeated here for convenience

$$\dot{v}_y = \frac{-c_f \left[\tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right) - \delta\right]\cos\left(\delta\right) - c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)}{m} - v_x r$$

$$\dot{r} = \frac{-\ell_f c_f \left[\tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right) - \delta\right]\cos(\delta) + \ell_r c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)}{I_z}$$

Eq. 23 can be rewritten as

$$\dot{v}_y + v_x r = \left(\frac{-\ell_f r - v_y}{m v_x} + \frac{\delta}{m}\right) c_{\alpha f} + \left(\frac{\ell_r r - v_y}{m v_x}\right) c_{\alpha r} \tag{30}$$

and similarly Eq. 24 as

$$\dot{r} = \left(\frac{-(\ell_f v_y + \ell_f^2 r)}{I_z v_x} + \frac{\ell_f \delta}{I_z}\right) c_{\alpha f} + \left(\frac{\ell_r v_y - \ell_r^2 r}{I_z v_x}\right) c_{\alpha r} \tag{31}$$

Writing these equations as functions of the cornering stiffness lends itself to the formulation of a least squares problem. Assuming that $\dot{v}_y$ and $\dot{r}$ are not directly measured, and that v and r are available, Eqs. 30 and 31 are rewritten in a discrete form using the Euler method as

$$v_y(k+1) - v_y(k) + v_x(k) r(k) \Delta t = \left(\frac{-\ell_f r(k) - v_y(k)}{m v_x(k)} + \frac{\delta(k)}{m}\right) \Delta t c_{\alpha f}$$
$$+ \left(\frac{\ell_r r(k) - v_y(k)}{m v_x(k)}\right) \Delta t c_{\alpha r}$$

$$r(k+1) - r(k) = \left(\frac{-(\ell_f v_y(k) + \ell_f^2 r(k))}{I_z v_x(k)} + \frac{\ell_f \delta(k)}{I_z}\right) \Delta t c_{\alpha f}$$
$$+ \left(\frac{\ell_r v_y(k) - \ell_r^2 r(k)}{I_z v_x(k)}\right) \Delta t c_{\alpha r}$$

where k is an index of the measurement set and Δt is the time between measurement sets. The least squares formulation then follows as

$$A = B \begin{bmatrix} c_{\alpha f} \\ c_{\alpha r} \end{bmatrix}, \tag{32}$$

where

$$A = \begin{bmatrix} v_y(2) - v_y(1) + v_x(1) r(1) \Delta t \\ r(2) - r(1) \\ \vdots \\ v_y(n) - v_y(n-1) + v_x(n-1) r(n-1) \Delta t \\ r(n) - r(n-1) \end{bmatrix}$$

$$B = \begin{bmatrix} \left(\frac{-\ell_f r(1) - v_y(1)}{m v_x(1)} + \frac{\delta(1)}{m}\right)\Delta t & \left(\frac{\ell_r r(1) - v_y(1)}{m v_x(1)}\right)\Delta t \\[2ex] \left(\frac{-(\ell_f v_y(1) + \ell_f^2 r(1))}{I_z v_x(1)} + \frac{\ell_f \delta(1)}{I_z}\right)\Delta t & \left(\frac{\ell_r v_y(1) - \ell_r^2 r(1)}{I_z v_x(1)}\right)\Delta t \\[2ex] \vdots & \vdots \\[2ex] \left(\frac{-\ell_f r(n-1) - v_y(n-1)}{m v_x(n-1)} + \frac{\delta(n-1)}{m}\right)\Delta t & \left(\frac{\ell_r r(n-1) - v_y(n-1)}{m v_x(n-1)}\right)\Delta t \\[2ex] \left(\frac{-(\ell_f v_y(n-1) + \ell_f^2 r(n-1))}{I_z v_x(n-1)} + \frac{\ell_f \delta(n-1)}{I_z}\right)\Delta t & \left(\frac{\ell_r v_y(n-1) - \ell_r^2 r(n-1)}{I_z v_x(n-1)}\right)\Delta t \end{bmatrix}$$

This procedure has been used and works well in practice. However, special care needs to be taken in collecting the vehicle data. The dynamics of the vehicle must be sufficiently excited. The best results are obtained when the lateral force on the tires are moderate (without exciting the suspension too much) and continuously varying throughout the data collection. In [12] the effects of various frequencies for the steering input are studied in making these estimates. Other methods for obtaining these values both offline and online can be found in [12] and [20]. In addition to the above procedure, an optimization algorithm can be used to further fit the model to the available data. Using the above estimates to initialize an optimization algorithm that allows the inertia and cornering stiffness to change, minimizing the model error, can improve the identification.

4.2   Optimal Control

As derived in section 4.1.2, the state space model for the lateral dynamics [11] can be written as

$$\dot{x} = Ax + B_1\delta + B_2\dot{r}_{des}$$

where

$$x = \begin{pmatrix} e_{cg} & \dot{e}_{cg} & \theta_e & \dot{\theta}_e \end{pmatrix}^T,$$

e_{cg} is the lateral distance from the C.G. to the path, θ_e is the heading error of the vehicle with respect to the path, δ is the steering angle input, and $\dot{r}_{des}$ is the desired yaw rate determined from the current path curvature and vehicle speed.

The following values of vehicle parameters were identified using the procedure in section 4.1.3 and will be used for the control design:

$$m = 1140.0kg$$

$$I_z = 1436.24kgm^2$$

$$\ell_f = 1.165m$$

$$\ell_r = 1.165m$$

$$c_f = 155494.663N/rad$$

$$c_r = 155494.663N/rad$$

The eigenvalues of the open-loop matrix A are $(0, -231.878, 0, -231.878)^T$. The two eigenvalues at the origin reveals that the system is not stable and must be stabilized by feedback. The controllability matrix $[B_1, AB_1, A^2B_1, A^3B_1]$ has full rank, therefore using the full state feedback law

$$\delta = -Kx = -k_1 e_{cg} - k_2 \dot{e}_{cg} - k_3 \theta_e - k_4 \dot{\theta}_e$$

allows the eigenvalues of the closed-loop matrix $(A - B_1K)$ to be placed at any desired location.

Rather than placing the eigenvalues manually, a discrete infinite-horizon Linear Quadratic Regulator (LQR) from optimal control theory is utilized. Let A_d and B_d refer to the discrete-time versions of A and B_1 respectively. The discretization method used in the conversion is zero-order hold on the inputs with a sample time of 0.01s. Given that the pair (A_d, B_d) is controllable, the optimal steering command can be written as

$$\delta^*(k) = -Kx(k)$$

where

$$K = \left(R + B_d^T P B_d\right)^{-1} B_d^T P A_d,$$

the objective cost function to be minimized by the control is

$$J = \sum_{k=0}^{\infty} x^{\delta}(k) Qx(k) + \delta(k) R\delta(k),$$

where P satisfies the matrix difference Riccati equation,

$$P = A_d^T P A_d - A_d^T P B_d \left(R + B_d^T P B_d\right)^{-1} B_d^T P A_d + Q,$$

Q is a diagonal weighting matrix with an entry for each state corresponding to the performance aspects contributing to the cost function and R is weighting value corresponding to the control effort contributing to the cost function.

The solution to the matrix difference Riccati equation is omitted here since it is not specific to path tracking. The

details to solving optimal control problems can be found in books such as [15] and many others. There are also many software packages available to perform the task.

4.2.1 Tuning the Optimal Controller

The tuning parameters for the LQR optimal control solution are

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix}$$

and R is chosen as

$$R = 1.$$

The tuning is further reduced by setting

$$q_2 = q_3 = q_4 = 0$$

leaving a single parameter to adjust for tuning purposes. Setting these values to zero simply means that we are only weighting cross track error against control effort.

Experiments are conducted on the *Lane Change Course*. Figure 27 illustrates the effects of the tuning parameter on tracking performance during these experiments. Again, similar characteristics to the previous trackers can be seen
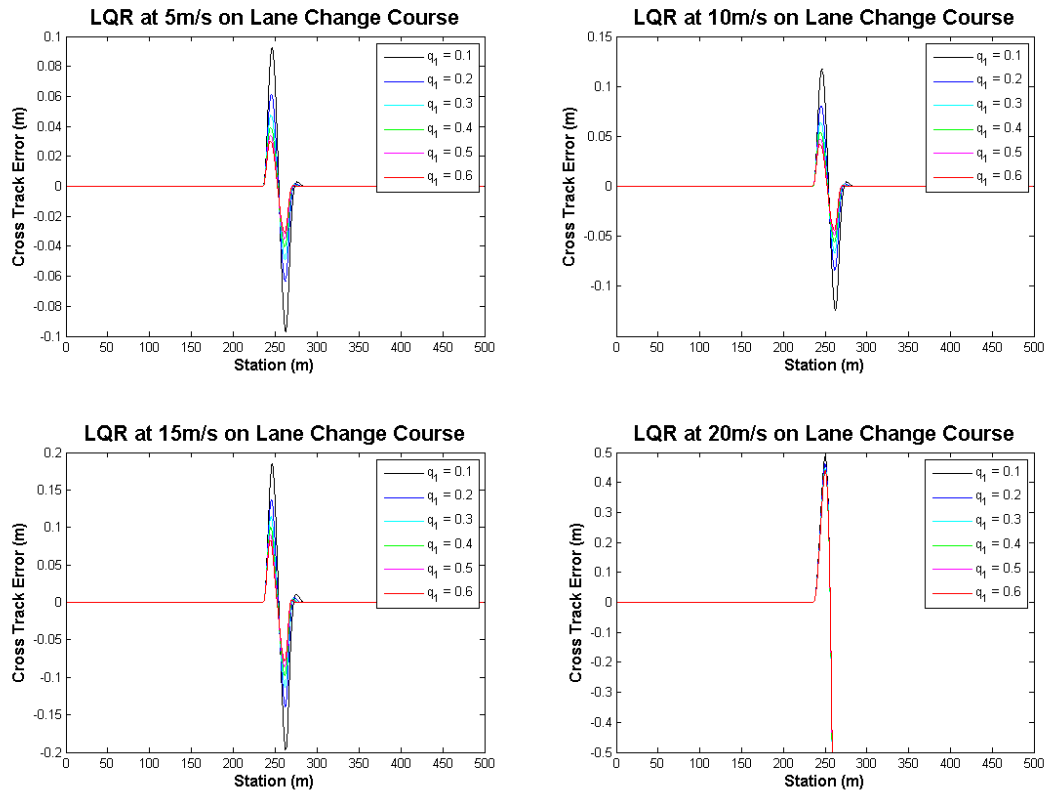
Figure 27: LQR controller at multiple speeds and various gains on the lane change course

as the gain is adjusted. Even though dynamics of the vehicle are included, this method fails the maneuver entirely at high speed. This is because information about the path is no longer included. The linear requirement for the vehicle model doesn't allow for the non-linear path dynamics.

Experiments are conducted on the *Figure Eight Course*. Figure 28 illustrates the effects of the tuning parameter on tracking performance during these experiments. Similar problems with the discontinuity as the previous method are
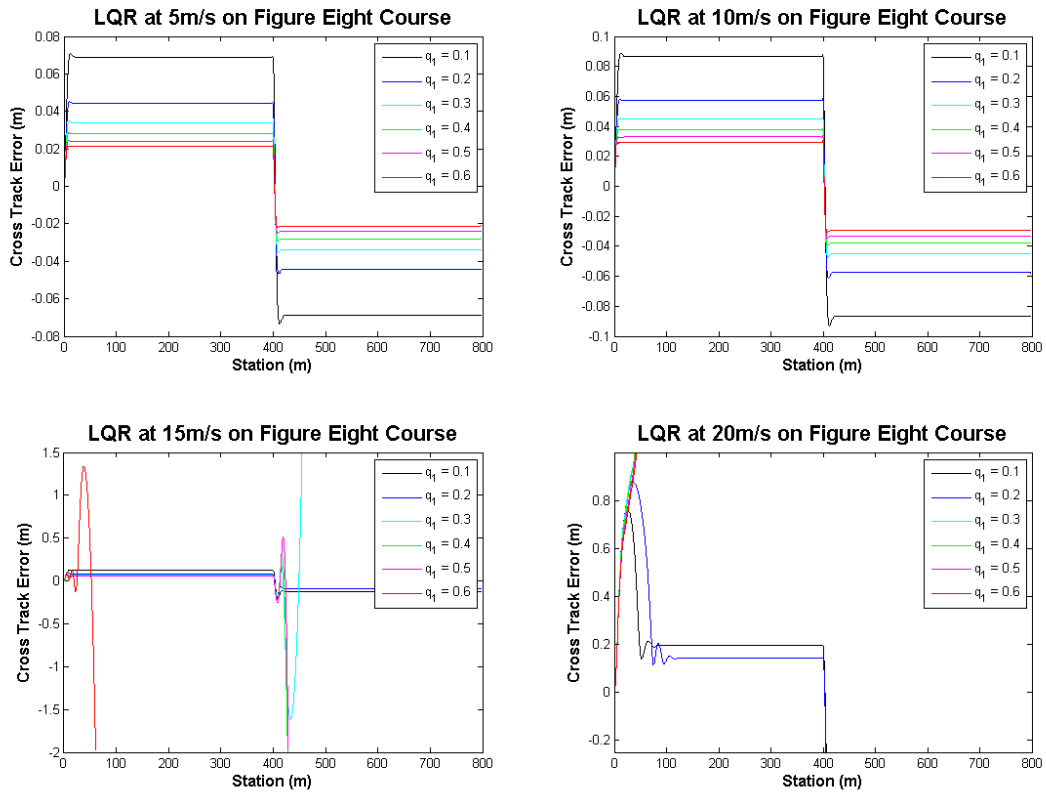
Figure 28: LQR controller at multiple speeds and various gains on the figure eight course

present. An interesting point here is that the steady state error that was previously contributed to neglecting dynamics is still significant. Again, this is because the path dynamics are neglected.

Experiments are conducted on the *Road Course*. Figure 29 illustrates the effects of the tuning parameter on tracking performance during these experiments. Slightly improved performance is achieved in the first two tests, but
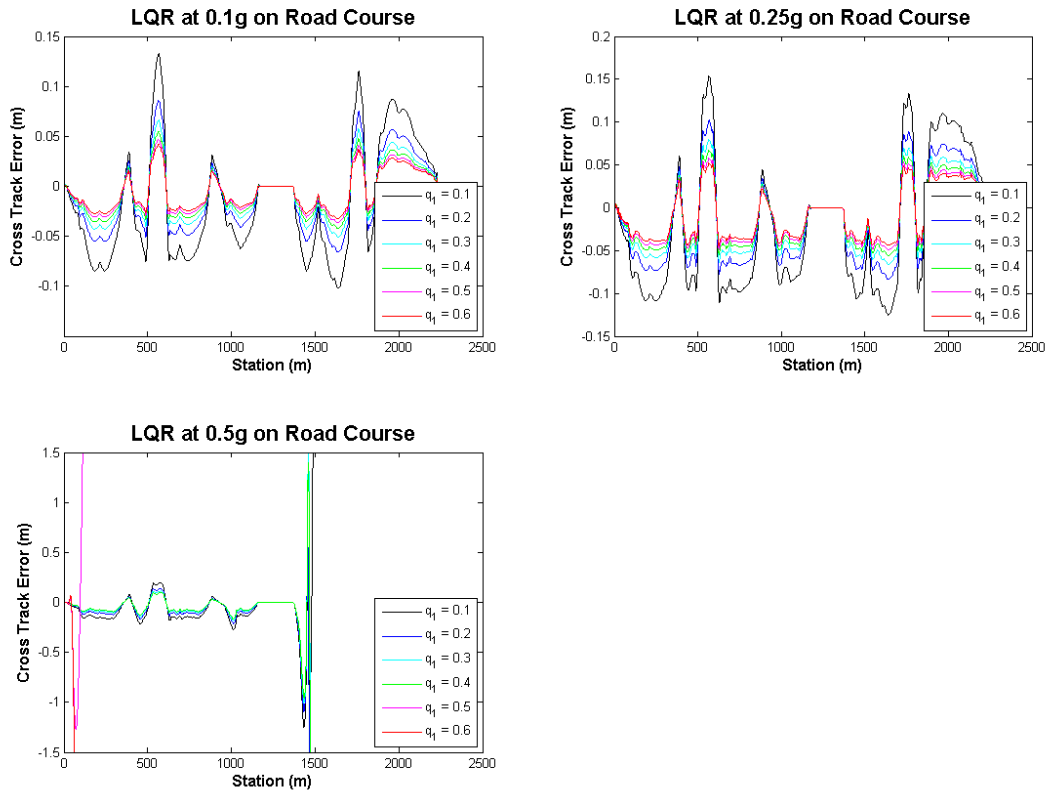
Figure 29: LQR at multiple velocity profiles and various gains on the road course

this method cannot complete the final test. The reason is similar to the kinematic trackers tendency to compensate for unmodeled dynamics with higher gains, only now it is the path that is unmodeled rather than the vehicle dynamics. It can be seen that in some scenarios, the path dynamics are as significant as the vehicle dynamics.

4.3   Optimal Control with Feed Forward Term

Continuing from the previous section, the state space model for the closed-loop system under state feedback is

$$\dot{x} = (A - B_1 K)x + B_2 \dot{r}_{des}.$$

41

Since the $B_2\dot{r}_{des}$ term is present, the states will not all converge to zero while the vehicle is following a constant curvature path, despite the fact that $(A - B_1K)$ is asymptotically stable. In this section, the use of a feed forward term in addition to state feedback to ensure zero steady state error is presented. This is the method used for zeroing steady state error in [11]. Start by assuming the existence of a modified control law of the form

$$\delta = -Kx + \delta_{ff}.$$

Then, the closed-loop system can be written as

$$\dot{x} = (A - B_1K)x + B_1\delta_{ff} + B_2\dot{r}_{des}.$$

Taking Laplace transforms, assuming zero initial conditions, results in

$$X(s) = [sI - (A - B_1K)]^{-1}\{B_1L(\delta_{ff}) + B_2L(\dot{r}_{des})\},$$

where $L(\delta_{ff})$ and $L(\dot{r}_{des})$ are Laplace transforms of δ_{ff} and $\dot{r}_{des}$ respectively. If the vehicle travels at a constant speed v_x on a road with constant radius of curvature R, then

$$\dot{r}_{des} = \text{constant} = \frac{v_x}{R}$$

and its Laplace transform is $\frac{v_x}{Rs}$. Similarly, if the feed forward term is constant, then its Laplace transform is $\frac{\delta_{ff}}{s}$. Using the Final Value Theorem, the steady state tracking error is given by

$$x_{ss} = \lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s) = -(A - B_1K)^{-1}\{B_1\delta_{ss} + B_2\frac{v_x}{R}\}. \tag{33}$$

Evaluation of equation 33 yields the steady state errors

$$x_{ss} = \begin{bmatrix} \frac{\delta_{ff}}{k_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{k_1}\frac{mv_x^2}{R(\ell_f+\ell_r)}\left(\frac{\ell_r}{2c_{\alpha f}} - \frac{\ell_f}{2c_{\alpha r}} + \frac{\ell_f}{2c_{\alpha r}}k_3\right) - \frac{1}{k_1R}\left(\ell_f + \ell_r - \ell_r k_3\right) \\ 0 \\ \frac{1}{2Rc_{\alpha r}(\ell_f+\ell_r)}\left(-2c_{\alpha r}\ell_f\ell_r - 2c_{\alpha r}\ell_r^2 + \ell_f mv_x^2\right) \\ 0 \end{bmatrix}. \tag{34}$$

42

From equation 34, it is clear that the lateral position error e_{cg} can be made zero by appropriate choice of δ_{ff}. However, δ_{ff} cannot influence the steady state yaw error, as seen from equation 34. The yaw angle error has a steady state term that cannot be corrected, no matter how the feed forward steering angle is chosen. The steady state yaw angle error is

$$\theta_{e\_ss} = \frac{1}{2Rc_{\alpha r}(\ell_f + \ell_r)}\left(-2c_{\alpha r}\ell_f\ell_r - 2c_{\alpha r}\ell_r^2 + \ell_f m v_x^2\right)$$

$$\theta_{e\_ss} = -\frac{\ell_r}{R} + \frac{\ell_f}{2c_{\alpha r}(\ell_f + \ell_r)}\frac{m v_x^2}{R}.$$

The steady state lateral position error can be made zero if the feedforward steering angle is chosen as

$$\delta_{ff} = \frac{m v_x^2}{RL}\left(\frac{\ell_r}{2c_{\alpha f}} - \frac{\ell_f}{2c_{\alpha r}} + \frac{\ell_f}{2c_{\alpha r}}k_3\right) + \frac{L}{R} - \frac{\ell_r}{R}k_3$$

which upon closer inspection is seen to be

$$\delta_{ff} = \frac{L}{R} + K_v a_y - k_3\left(\frac{\ell_r}{R} - \frac{\ell_f}{2c_{\alpha r}}\frac{m v_x^2}{RL}\right),$$

where

$$K_v = \frac{\ell_r m}{2c_{\alpha f}(\ell_f + \ell_r)} - \frac{\ell_f m}{2c_{\alpha r}(\ell_f + \ell_r)}$$

is called the understeer gradient and $a_y = \frac{v_x^2}{R}$. Denoting $m_r = m\frac{\ell_f}{L}$ as the portion of the vehicle mass carried on the rear axle and $m_f = m\frac{\ell_r}{L}$ as the portion of the vehicle mass carried on the front axle, $K_v = \frac{m_f}{2c_{\alpha f}} - \frac{m_r}{2c_{\alpha r}}$. Hence

$$\delta_{ff} = \frac{L}{R} + K_v a_y + k_3 e_{2\_ss}.$$

The steady state steering angle for zero lateral position error is given by

$$\delta_{ss} = \delta_{ff} - K x_{ss}$$

$$\delta_{ss} = \delta_{ff} - k_3\theta_{e\_ss}$$

$$\delta_{ss} = \frac{L}{R} + K_v a_y.$$

4.3.1 Tuning the Optimal Controller with Feed Forward Term

Experiments are conducted on the *Lane Change Course*. Figure 30 illustrates the effects of the tuning parameter on tracking performance during these experiments. This method improves performance over the standard LQR method.
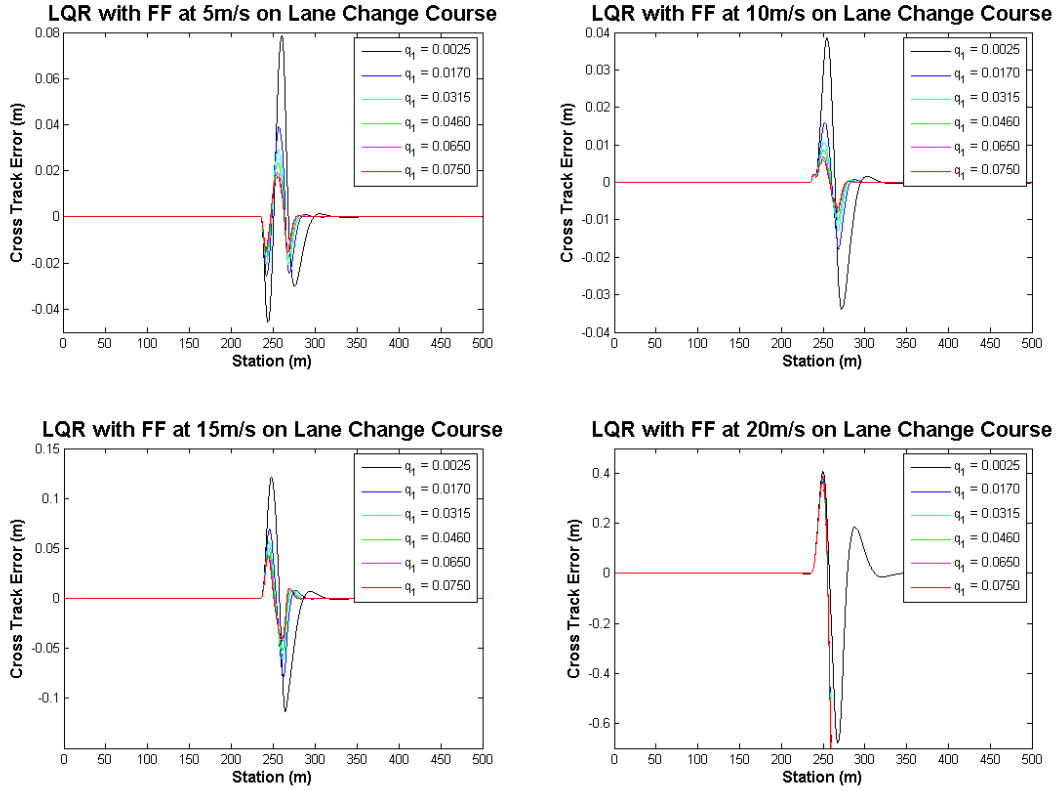

Figure 30: LQR controller with feed forward term at multiple speeds and various gains on the lane change course

Tuning in identical to that of the LQR method with the exception that the gain values are much lower. The higher gain in the LQR method can be attributed to the gain compensation from not having path dynamics.

Experiments are conducted on the *Figure Eight Course*. Figure 31 illustrates the effects of the tuning parameter on tracking performance during these experiments. This method greatly improves the steady state error. However, the

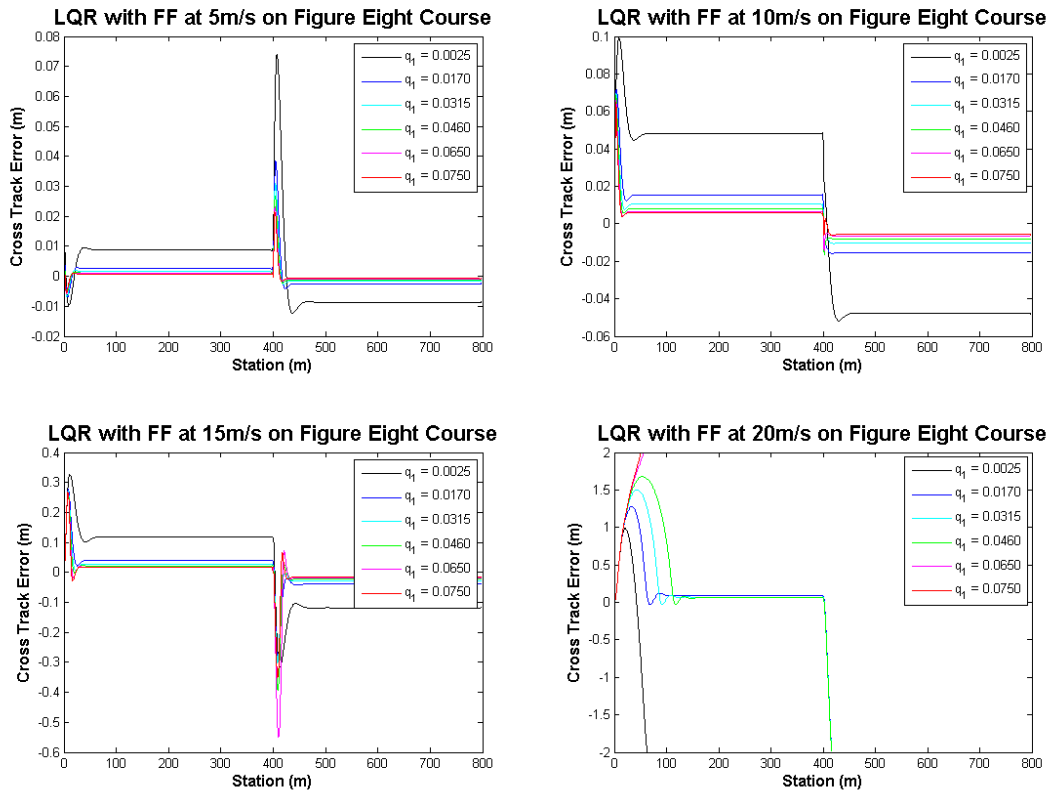
Figure 31: LQR controller with feed forward term at multiple speeds and various gains on the figure eight course

response to the discontinuity is worse. The overshooting of the LQR with a feed forward term can be attributed to the fact that there is no look-ahead down the path in front of the vehicle. The feed forward term can only be reactive. As a result, the final test is complete failure.
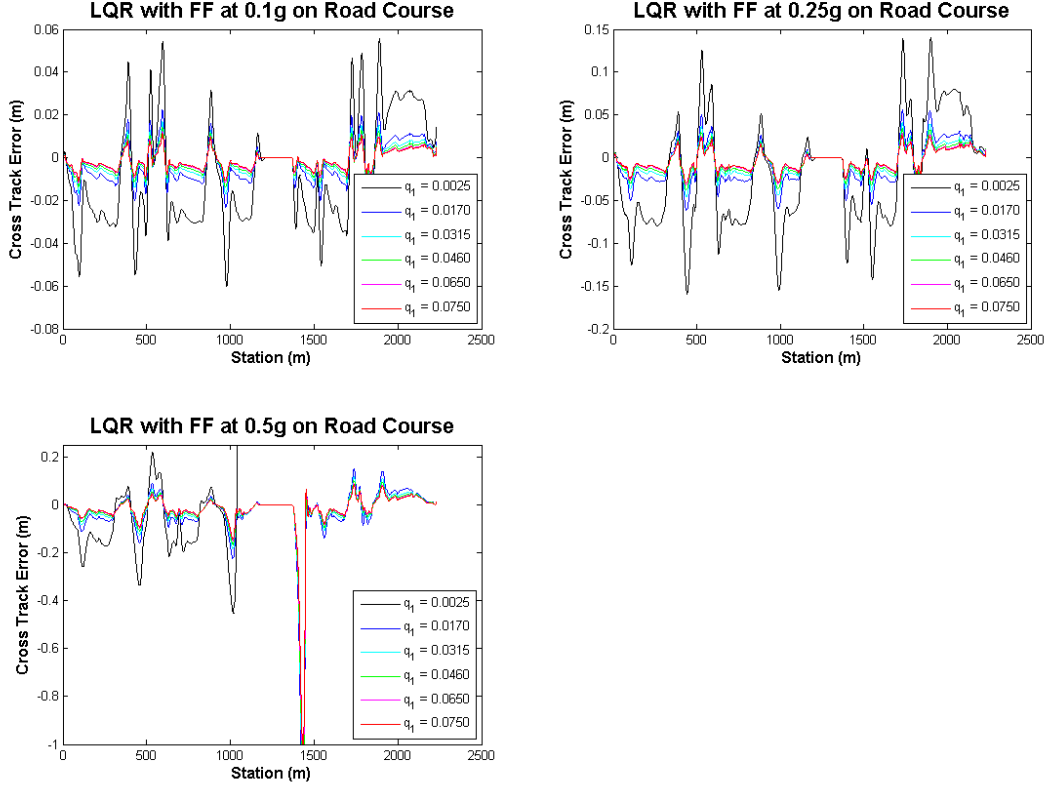
Figure 32: LQR controller with feed forward term at multiple velocity profiles and various gains on the road course

Under varying normal driving scenarios, this method can perform very well when compared to the other methods.

4.4 Optimal Preview Control

To address the overshooting problem from the previous section, a preview of the path ahead of the vehicle is considered. This section presents the application of optimal linear preview control theory to vehicle steering control found in [14]. Another interesting application can be found in [10] as well. As before, the linear dynamic bicycle model is translated to the discrete time-form resulting in the state space form

$$x_v \left(k + 1 \right) = A_v x_v(k) + B_v \delta(k)$$

$$y_v \left(k \right) = C_v x_v(k) + D_v \delta(k)$$

46

and the lateral profile of the path is considered in discrete sample value form, with sample values from past observations of the path ahead being stored as states of the full vehicle/path system. As the system moves forward in time, a new path sample value is read in and the oldest stored value is discarded, corresponding to the vehicle having passed the point on the path to which the oldest value refers. All the other path sample values are shifted through the time step, nearer to the vehicle. The dynamics of this shift register process are represented mathematically by

$$y_r(k+1) = A_r y_r(k) + B_r y_{ri}(k),$$

where A_r is of the form

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

and B_r is of the form

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Combining vehicle and road equations together, the full dynamic system can be written as

$$\begin{bmatrix} x_v(k+1) \\ y_r(k+1) \end{bmatrix} = \begin{bmatrix} A_v & 0 \\ 0 & A_r \end{bmatrix} \begin{bmatrix} x_v(k) \\ y_r(k) \end{bmatrix} + \begin{bmatrix} B_v \\ 0 \end{bmatrix} \delta(k) + \begin{bmatrix} 0 \\ B_r \end{bmatrix} y_{ri}(k).$$

The complete problem is now in standard form,

$$z(k+1) = Az(k) + B\delta(k) + Ey_{ri}(k)$$

$$y(k) = Cz(k) + D\delta(k).$$

47

If y_{ri} is a sample from a white noise random sequence, the time-invariant optimal control minimizing a cost function J, given that the pair (A, B) is stabilizable and the pair (A, C) is detectable, is

$$u^* (k) = -Kz(k)$$

with

$$K = \left(R + B^T PB\right)^{-1} B^T PA,$$

where the objective function to be minimized by the control,

$$J = \sum_{k=0}^{\infty} z^T (k) Qz (k) + \delta (k) R\delta(k)$$

and P satisfies the matrix difference Riccati equation,

$$P = A^T PA - A^T PB\left(R + B^T PB\right)^{-1} B^T PA + Q,$$

where $Q = C^T qC$, containing the diagonal matrix, $diag\,[q_1 q_2 \ldots q_n]$, with terms corresponding to the number of performance aspects contributing to the cost function, and R, corresponding to the control input, steering angle.

Once formulated in this way, the same LQR problem from section 4.3 can be solved to obtain the control law and the same tuning method can be used.

4.4.1 Tuning the Optimal Preview Controller

Experiments are conducted on the *Lane Change Course* with a variety of preview distances. Figures 33-36 illustrate the effects of the tuning parameter on tracking performance during these experiments.

Figure 33: Optimal preview controller with 0.5s preview at multiple speeds and various gains on the lane change course

Figure 34: Optimal preview controller with 1.0s preview at multiple speeds and various gains on the lane change course
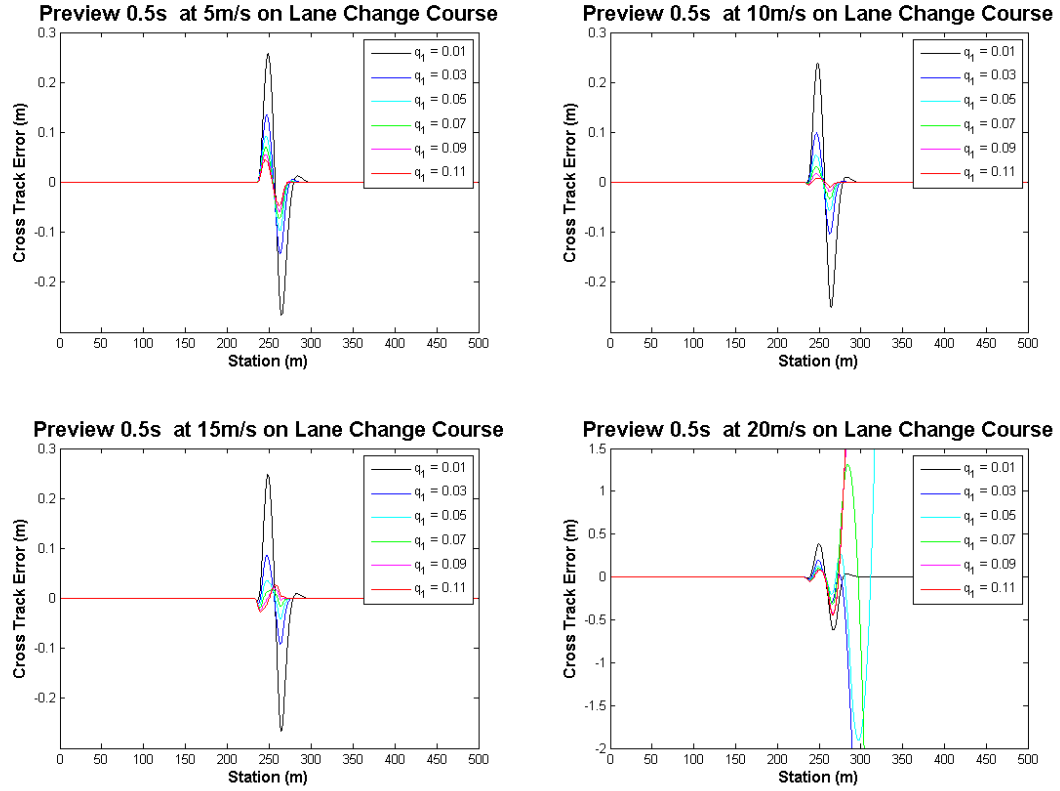
Figure 35: Optimal preview controller with 1.5s preview at multiple speeds and various gains on the lane change course

Figure 36: Optimal preview controller with 2.0s preview at multiple speeds and various gains on the lane change course

Experiments are conducted on the *Figure Eight Course* with a variety of preview distances. Figures 37-40 illustrate the effects of the tuning parameter on tracking performance during these experiments.

Figure 37: Optimal preview controller with 0.5s preview at multiple speeds and various gains on the figure eight course
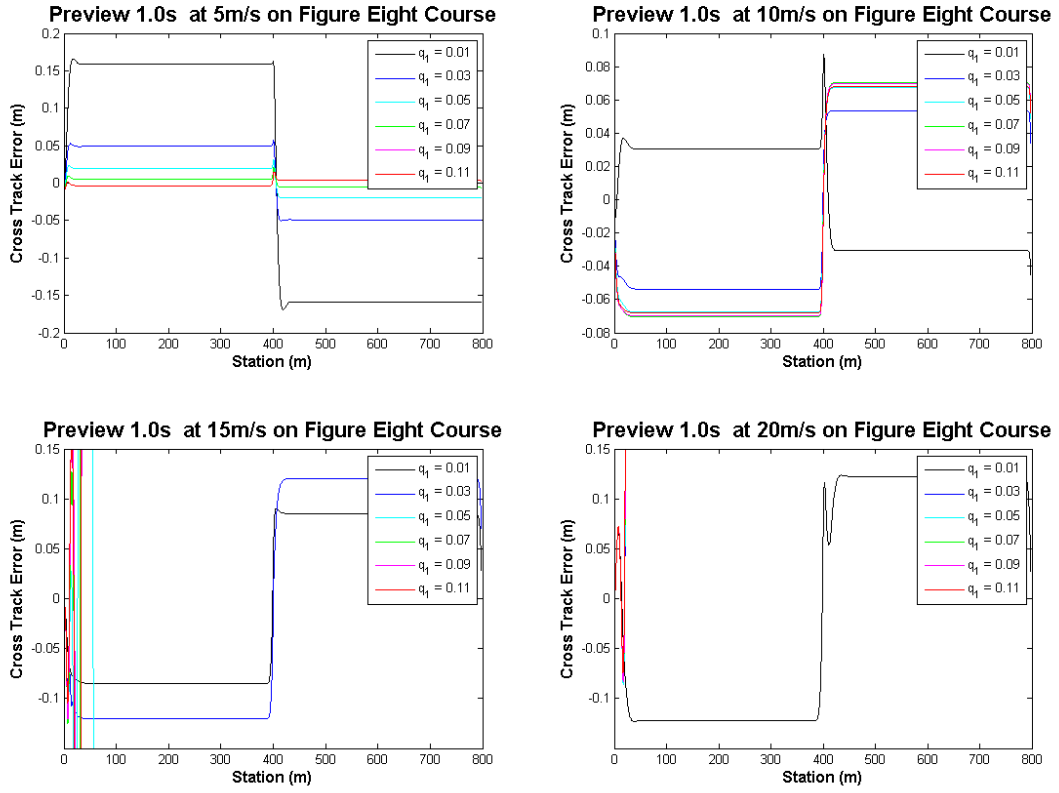
Figure 38: Optimal preview controller with 1.0s preview at multiple speeds and various gains on the figure eight course
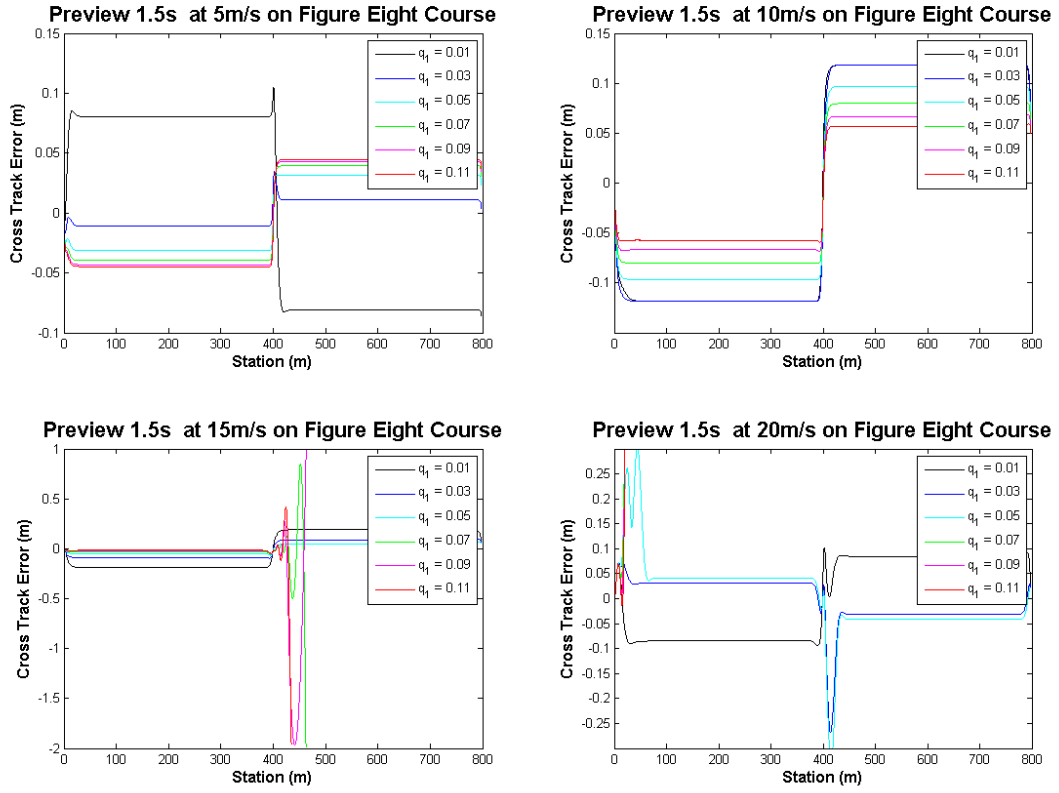
Figure 39: Optimal preview controller with 1.5s preview at multiple speeds and various gains on the figure eight course
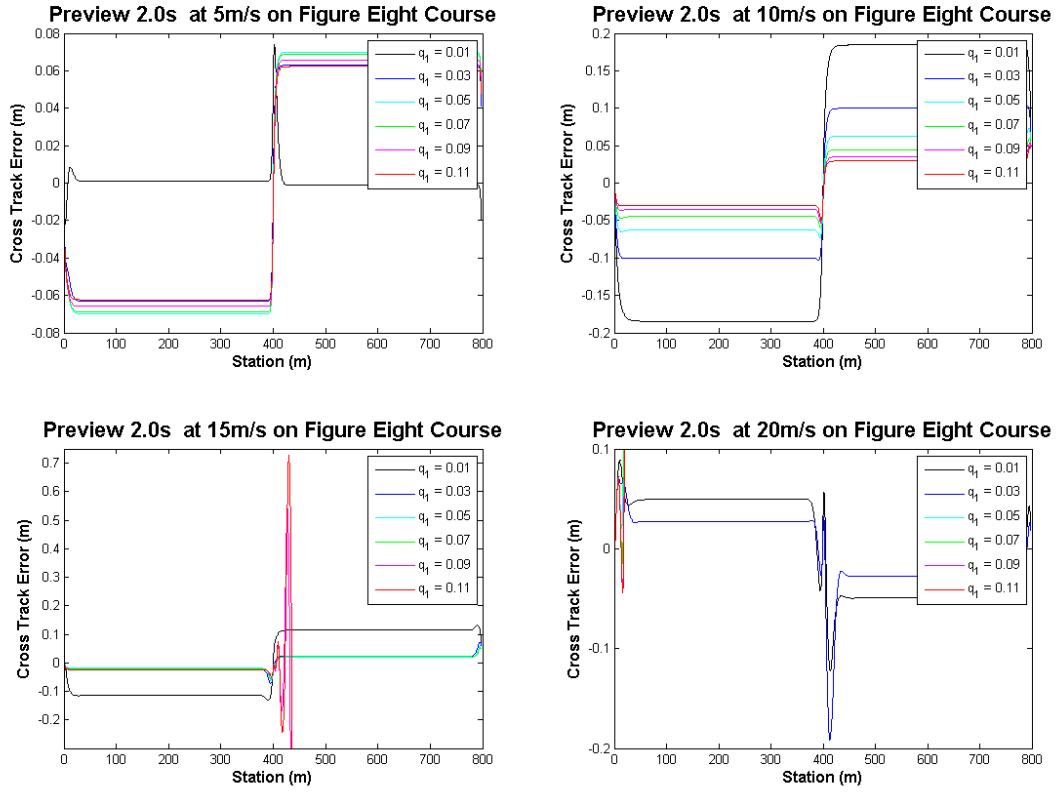
Figure 40: Optimal preview controller with 2.0s preview at multiple speeds and various gains on the figure eight course

Experiments are conducted on the *Road Course* with a variety of preview distances. Figures 41-44 illustrate the effects of the tuning parameter on tracking performance during these experiments.
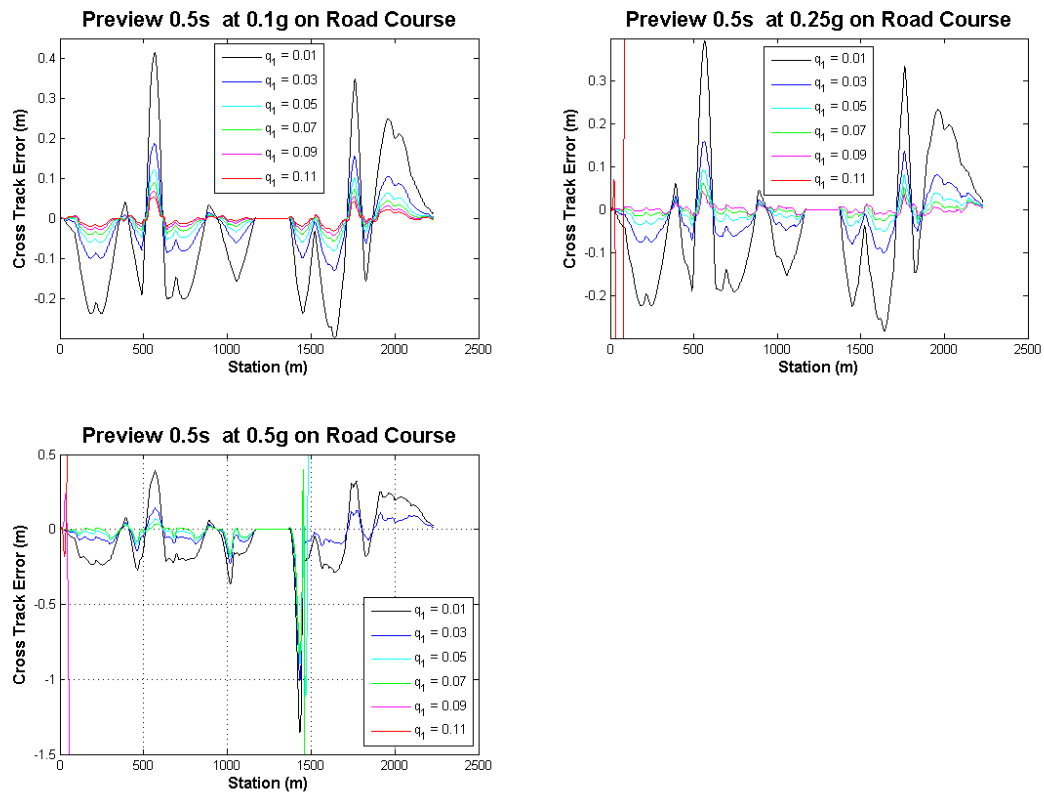
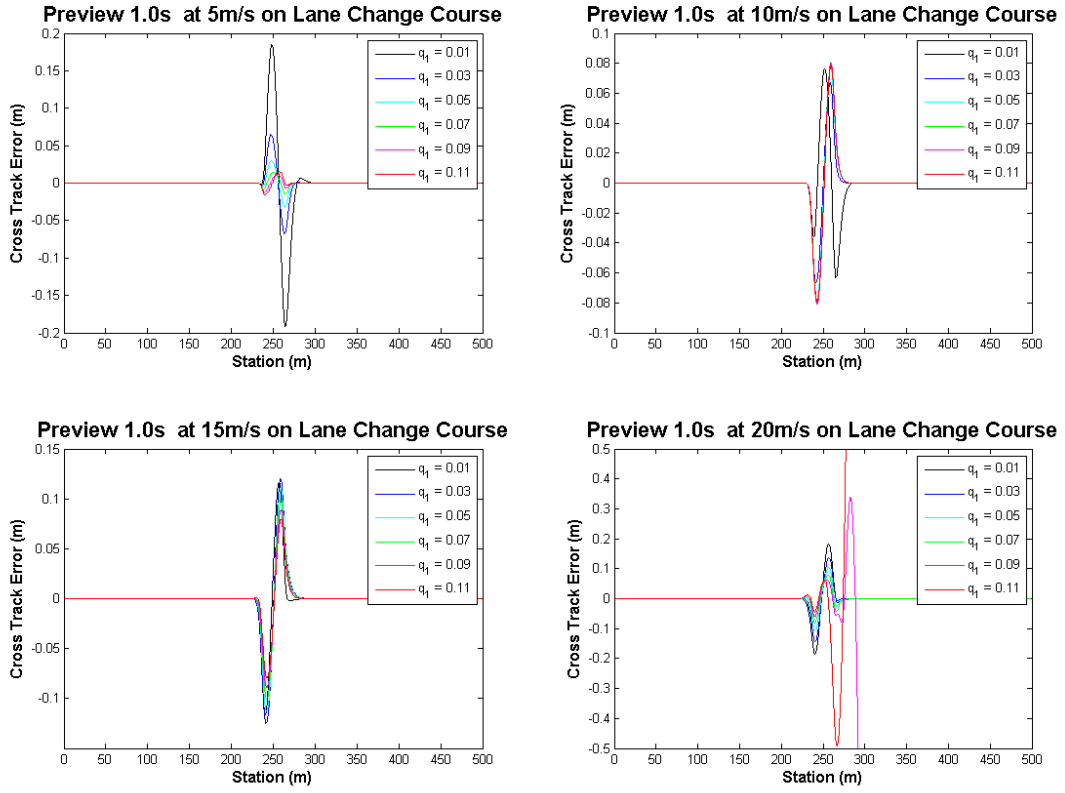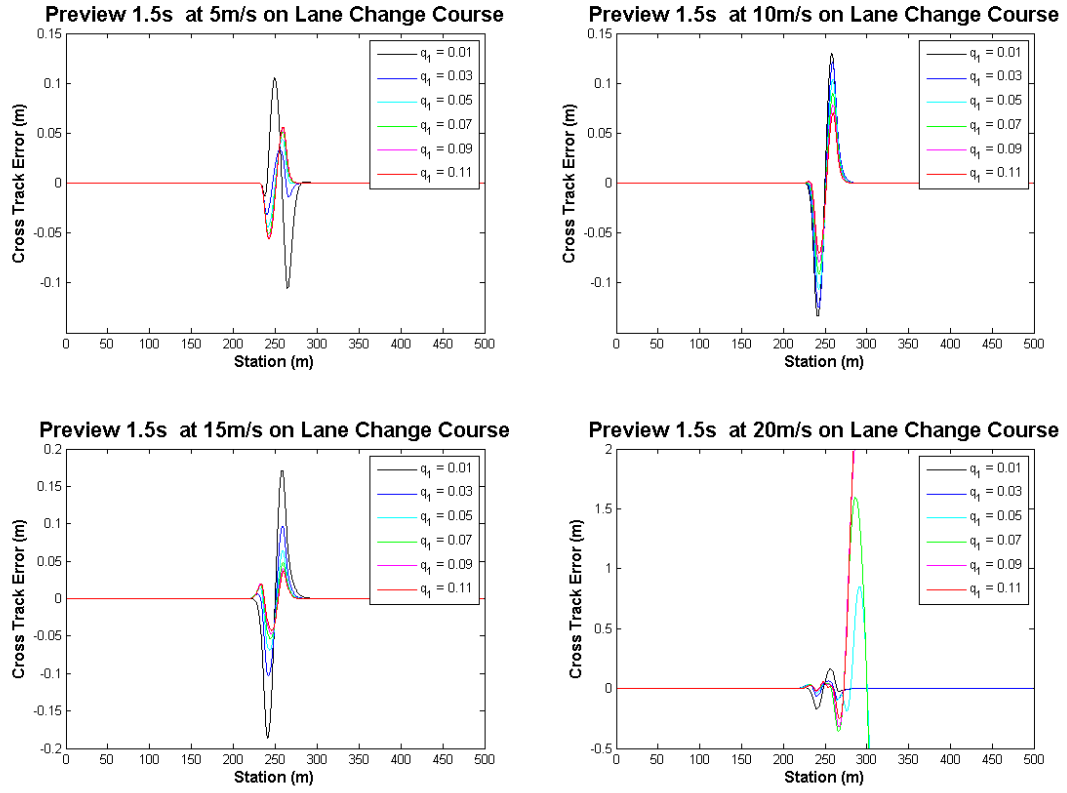Figure 41: Optimal preview controller with 0.5s preview at multiple speeds and various gains on the road course
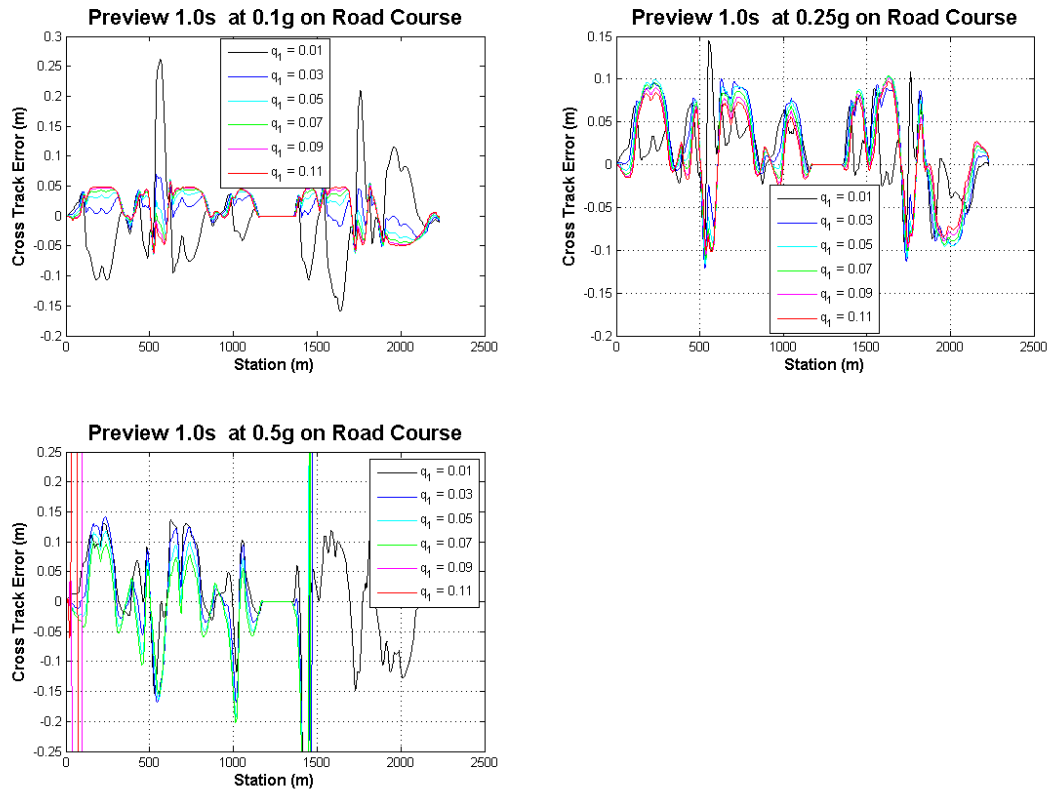
Figure 42: Optimal preview controller with 1.0s preview at multiple speeds and various gains on the road course

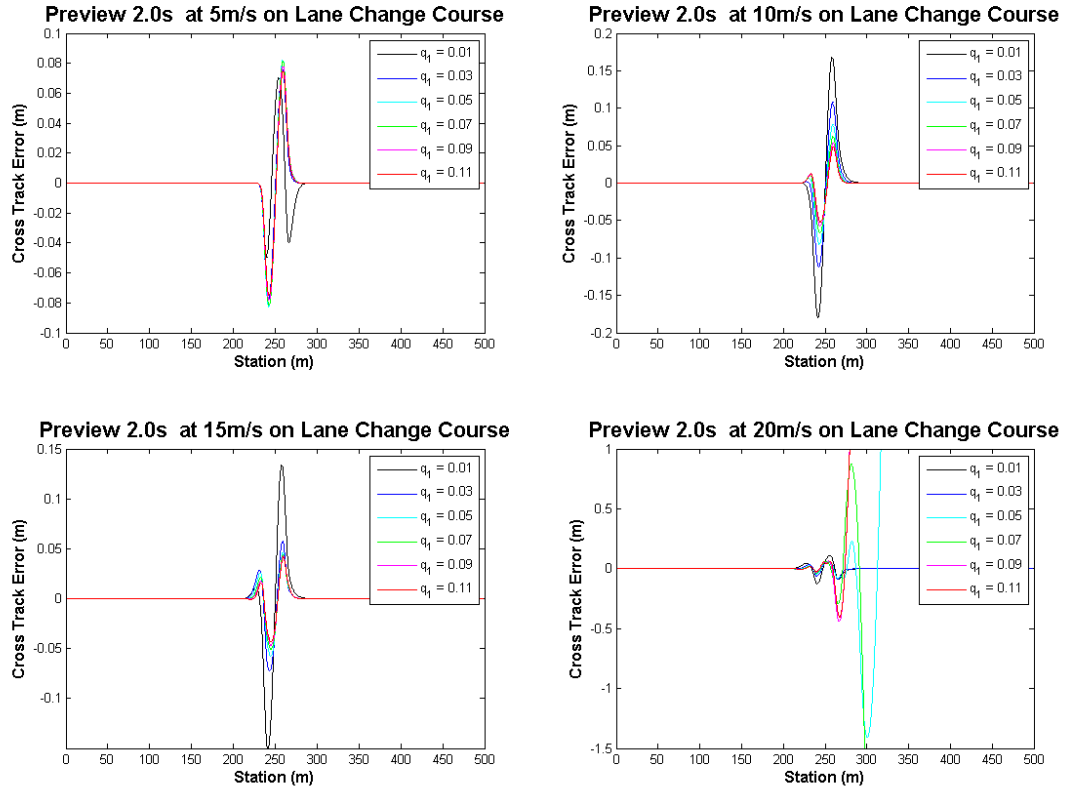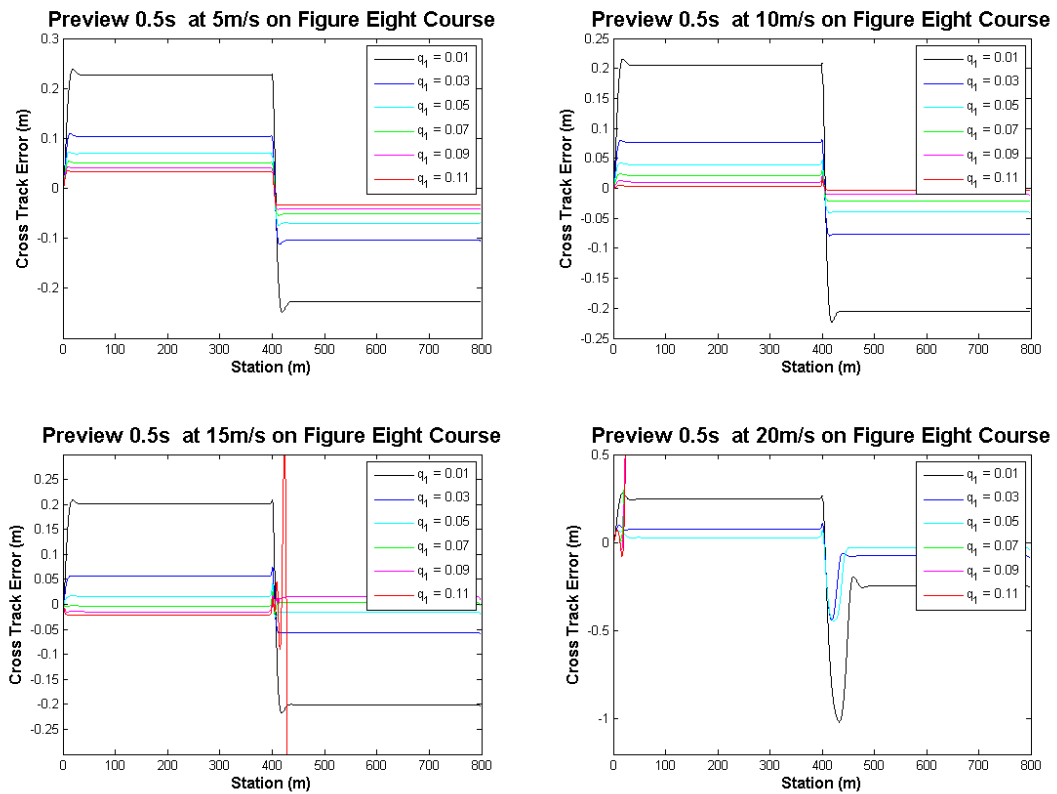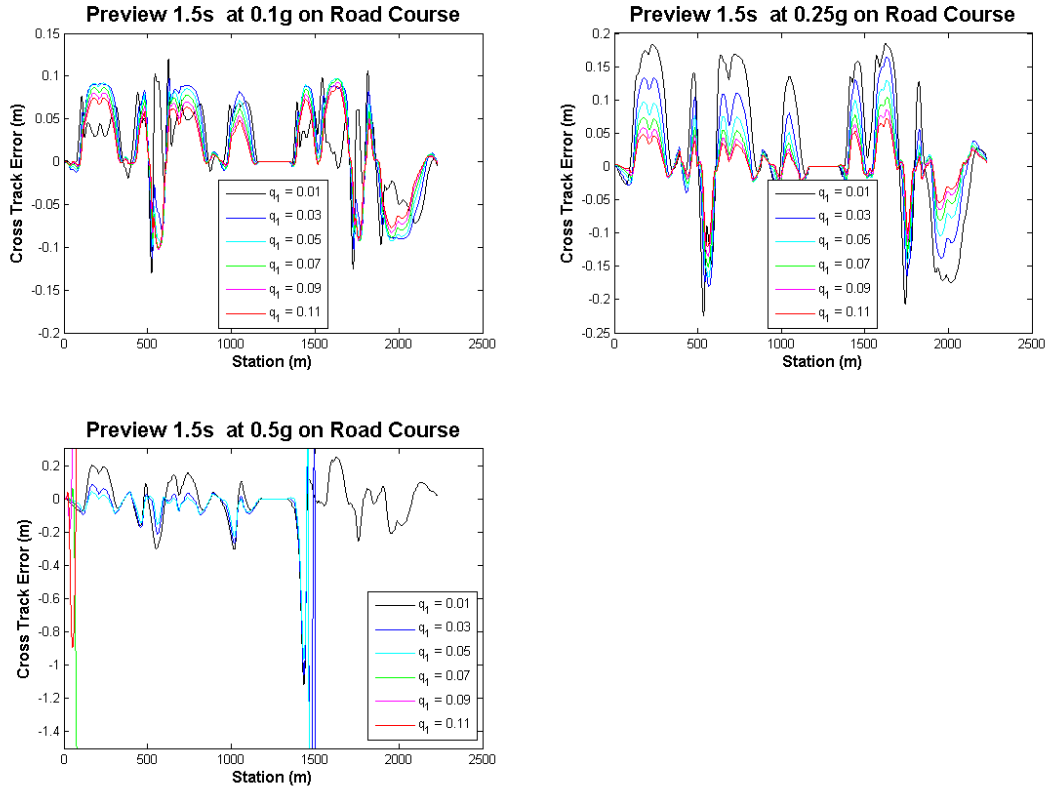Figure 43: Optimal preview controller with 1.5s preview at multiple speeds and various gains on the road course
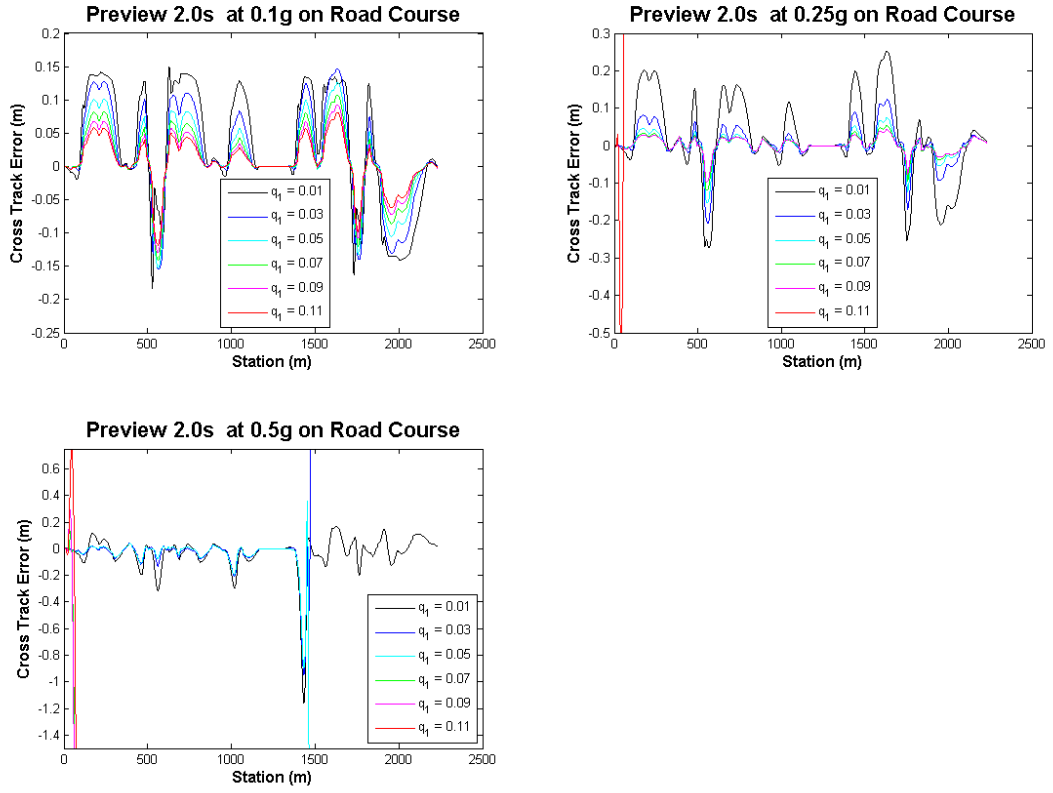
Figure 44: Optimal preview controller with 2.0s preview at multiple speeds and various gains on the road course

5 Performance Comparison

5.1 Tracking Results

Experiments are conducted on the *Lane Change Course*. Figure 45 illustrates the tracking performance of the tuned controllers during these experiments.
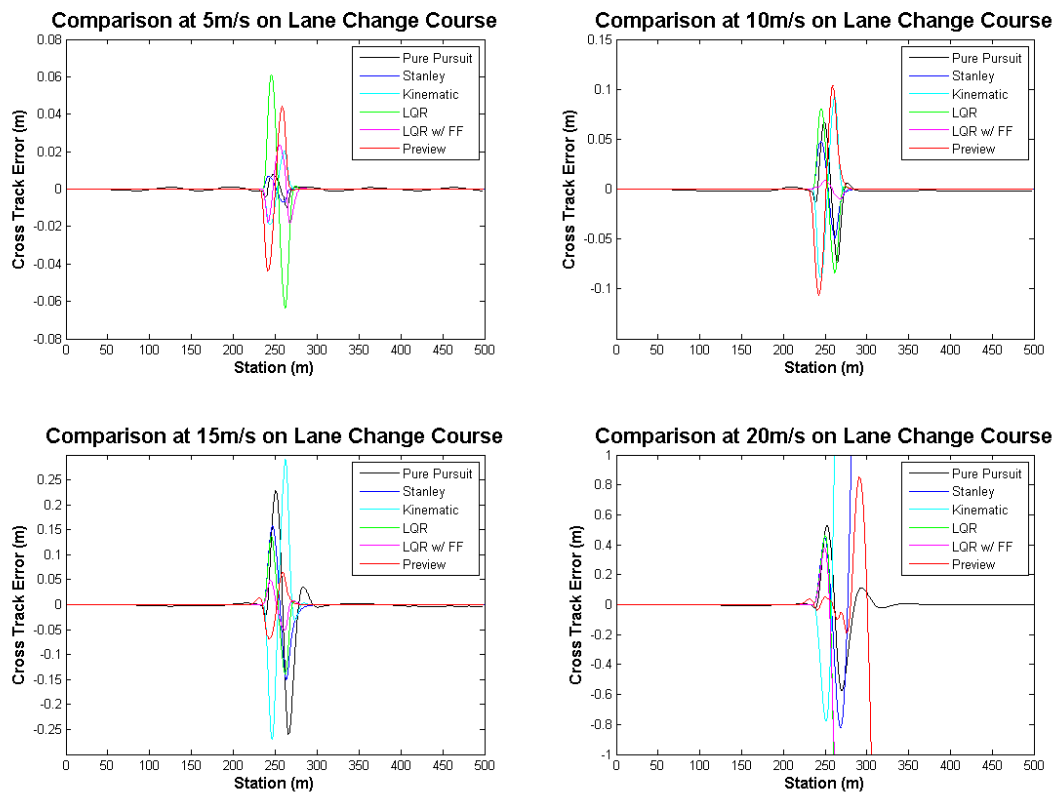
Figure 45: Comparison of the tuned controllers on the lane change course

Experiments are conducted on the *Figure Eight Course*. Figure 46 illustrates the tracking performance of the tuned controllers during these experiments.
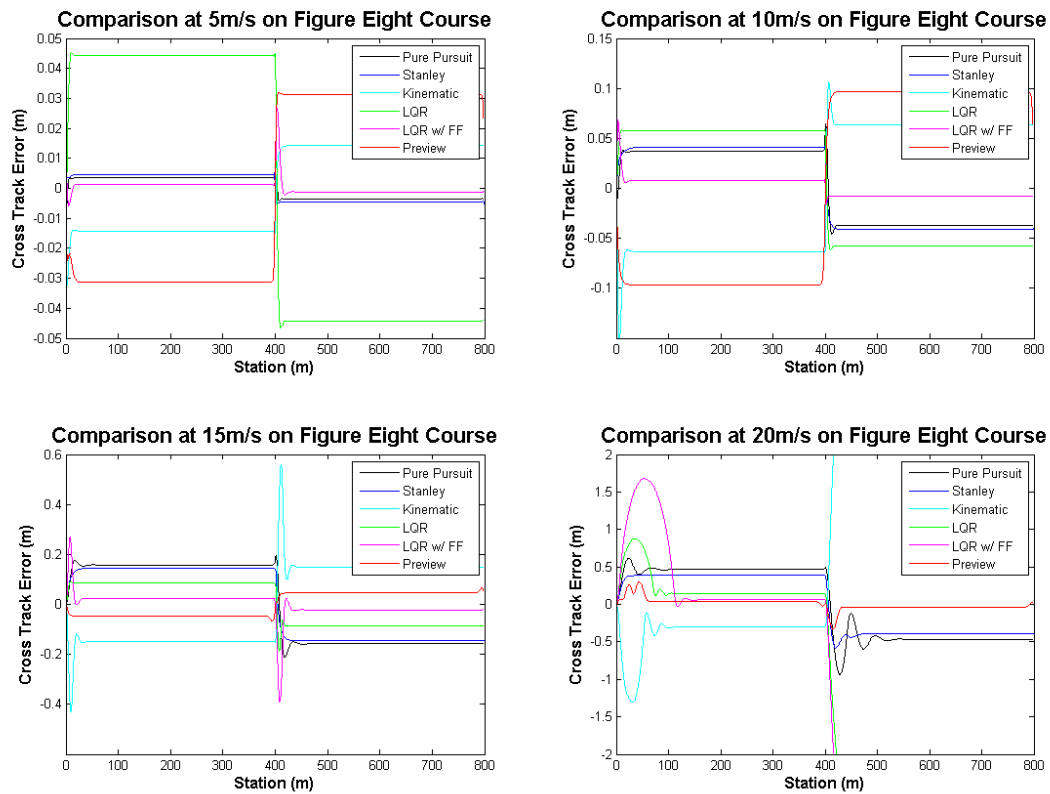
Figure 46: Comparison of the tuned controllers on the figure eight course

Experiments are conducted on the *Road Course*. Figure 47 illustrates the tracking performance of the tuned controllers during these experiments.
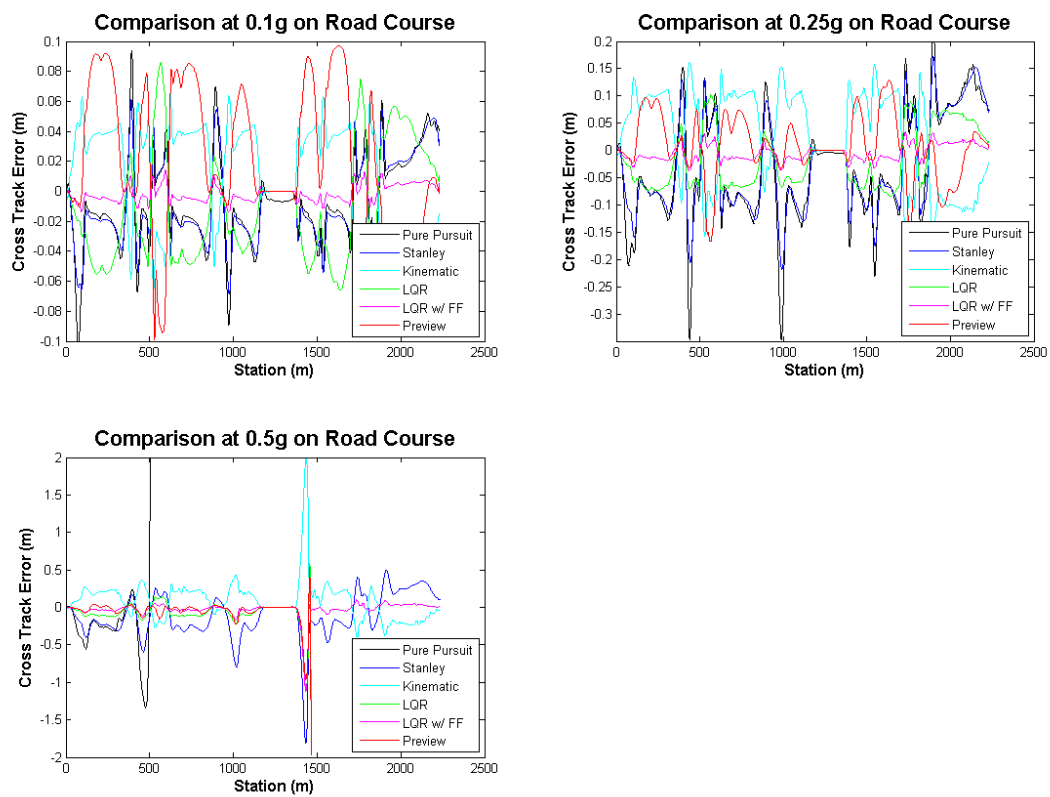
Figure 47: Comparison of the tuned controllers on the road course

An empirical comparison of the path tracking methods can be found in Figure 48. It can be seen that no method is perfect and all of the methods can work well in some applications. Figure 48 lists applications that each method may be well suited for. Chapter 6 discusses this comparison in more detail.

Empirical Comparison of Tracking Results

Tracking Method	Robustness to Disturbances	Path Requirements	Cutting Corners	Overshooting	Steady State Error	Good Applications
Pure Pursuit	Good	None within reason	Significant as speed increases	Moderate as speed increases	Significant as speed increases	Slow driving and/or on discontinuous paths
Stanley	Fair	Continuous curvature	No	Moderate as speed increases	Significant as speed increases	Smooth highway driving and/or parking maneuvers
Kinematic	Poor	Continuous through 2nd derivative of curvature	No	Moderate as speed increases, significant during rapidly changing curvature	Significant as speed increases	Smooth parking maneuvers
LQR with FF	Poor	Continuous curvature	No	Significant during rapidly changing curvature	Minimal until much higher speeds	Smooth high speed driving, urban driving at speed
Preview	Fair	None within reason	Moderate in rapidly changing curvature and/or speed	Moderate in rapidly changing curvature and/or speed	Minimal until much higher speeds	Highway driving at relatively constant speed

Figure 48: Performance comparison table

6 Conclusions and Future Work

This research derives, implements, tunes and compares selected path tracking methods. All of the methods presented are imperfect solutions in some way, but even the simplest methods can perform quite well in some common scenarios. The importance of modeling vehicle and path dynamics is highlighted as vehicle speed is increased and paths become more varying. This paper shows the varying levels of precision, complexity, implementation ease, path requirements and robustness associated with these path tracking methods. The characteristics of these trackers are often complementary and no one tracker is right for every application. Some applications may even benefit from a combination of approaches. Therefore, the understanding to be gained is the level of precision and boundaries of performance to expect from the presented methods as well as the strengths and shortcomings of using the various system models for a variety of applications. This understanding can guide selection of tracking methods for a specific application. The remainder of this chapter summarizes the characteristics of the presented system models and path tracking algorithms. A discussion of future work is also included.

Since a very detailed vehicle model can be difficult to obtain and use, the tracking methods described in this paper make use of system models that approximate vehicle motion. The approximations, simplifications, and linearization of these models vary and one needs to be aware of these shortcuts and the circumstances for which each model provides a reasonable approximation of vehicle motion. A simple kinematic model can suffice when a vehicle closely exhibits kinematic motion, such as driving slowly or performing parking maneuvers. The non-linearity of the steering angle is captured in tight turning scenarios with the kinematic model and the dynamic effects on the vehicle are minimal when driving slowly. A simple dynamic model can suffice in highway driving conditions or when the vehicle is driving at moderate speeds on smoothly varying roads. The dynamics are reasonably approximated for many driving scenarios with this model, but the model starts to break down at very slow speeds and/or tight steering angles such as parking maneuvers. These break downs in the model can be attributed to the velocity being in the denominator in the equations of motion and the linearization about the forward direction with small angle theory applied to the steering angle equation. Additionally, the model starts to break down if vehicle speed is too high on a road that is not straight. This comes from the approximation that the tire dynamics can be represented by a single value describing the ratio between lateral force and slip angle as well as another small angle approximation applied to the side slip equation. This value only represents the tire dynamics in a somewhat linear region of this non-linear function and for a small region of normal force on the tire. Higher speed cornering places the tires in the non-linear dynamics region and causes the suspension to dramatically change the normal force on the tires. The simple dynamic model also requires some parameters of the vehicle to be identified, however the system identification is quite simple using the procedure

described in Section 4.1.3.

Geometric path trackers are simple to understand and implement. The geometric methods presented here achieve the majority of the path tracking performance demonstrated by the other trackers until speeds are significantly increased. These methods include information about the path through a look-ahead distance or considering the path heading, therefore the shape of the path does not influence their performance significantly when operating at low speeds. Pure Pursuit works fairly well and is quite robust to large errors and discontinuous paths. However, it is not clear how to pick the best look-ahead distance. Varying the look-ahead distance with speed is a common approach and is the approach taken in this paper, but it makes sense that the look-ahead distance could be a function of path curvature and maybe even cross track error in addition to longitudinal velocity. Care should be taken to prevent over tuning Pure Pursuit to a specific course since changing the look-ahead distance simply changes the radius of curvature that the vehicle will travel and therefore can compensate for the increased (compared to the kinematic bicycle model prediction) radius that results from the understeer gradient of the vehicle. If tuned to insure stability, performance can be greatly reduced by cutting corners on the path due to a longer look-ahead distance. Steady state error in curves also becomes a problem as speed increases. The Stanley method is the simplest method presented and performs surprisingly well. In general, this method outperforms Pure Pursuit in most scenarios. However, the Stanley method is not as robust to large errors and non-smooth paths. This method is more intuitive to tune when compared to Pure Pursuit, but it suffers from similar pitfalls when tuning. The Stanley tracker can be over-tuned to a specific course in a similar manner because the only way it can overcome dynamic effects is with a high gain that may lead to instability on other paths. In contrast to Pure Pursuit, a well tuned Stanley tracker will not "cut corners" but rather overshoot turns. This effect can be attributed to not having a look-ahead. Similar to the Pure Pursuit method, steady state errors in curves at moderate speeds become significant.

The kinematic controller presented is a little more difficult to understand and implement than the geometric methods. There is a significant, yet manageable, increase in the on-line computational work needed to compute the steering angle velocity. The added complexity of this method increases the chance of making implementation mistakes. An advantage of the kinematic method is that it is easily extended for applications that need to pull a path following trailer. In fact, this method is capable of controlling multiple trailers being pulled by the vehicle. Additionally, this method can be applied to a large class of mobile robots approximated by kinematic models written in chained form. For the car-like robot application, the overall tracking performance and robustness at moderate speed is comparable to the Stanley method. Tuning the three parameters of the resulting control law in the kinematic controller is not as straight forward as the geometric methods, however a relationship between the gains reducing them to a single intuitive parameter was given. This kinematic method requires the curvature of the path and its first two derivatives in

the calculations. Including curvature and its derivatives effectively gives this controller the benefit of a look-ahead, but also requires the path to be smooth. The exclusion of dynamic effects results in the same high gain compensation for unmodeled effects at higher speeds when tuning. Care must be taken to relax the tracking performance to insure stability. A well tuned kinematic tracker will have significant steady state error in curves at moderate speeds.

The Linear Quadratic Regulator (LQR) method uses the dynamic bicycle model to design a path tracker. The model and resulting control law are easy to understand and implement. However, tuning the LQR tracker is more complicated because the solution to the optimal control problem must be solved to obtain the gains. A simplification of the five tuning parameters resulting in a single intuitive parameter was given and many software packages are available to solve the optimal control problem. The additional computation takes place off-line and the on-line controller is very simple. Interestingly, the LQR tracker does not considerably outperform the Stanley or kinematic methods in many scenarios. Since the errors of the previous methods have been largely attributed to unmodeled dynamic effects, one would expect that the LQR method would improve performance. The dynamic bicycle model approximates the lateral dynamics of the vehicle, but in order to enable the use of linear control techniques the path coordinate model is linearized about the forward direction. In other words, the model excludes the non-linear path dynamics and best approximates a vehicle following a straight path. Following a straight path is something all of the methods are pretty good at. However, there are considerable improvements in performance that can be achieved using the LQR approach with the addition of a feed forward term to handle the path disturbance. The choice of this feed forward term was given. The LQR with a feed forward term approach is a great choice for highway driving and many urban driving scenarios. Steady state error approaches zero in curves at moderate speeds. Tuning this controller is identical to the LQR method with the exception that the gain value will be much lower. The lower gain is a result of no longer compensating for the path dynamics with a higher gain. This compensation of unmodeled effects in the standard LQR tracker is similar to those found in tuning the previous trackers. It turns out that for many driving scenarios the dynamics of the vehicle or the path can contribute somewhat equivalent errors if either is unaccounted for. The LQR with a feed forward term method does not solve every problem. The limitations of the dynamic bicycle model still apply to this method. It will not perform as well at very low speeds, during tight maneuvers or during higher speed cornering that forces the tire to operate outside the dynamics region that the simple tire model is valid for. Another important characteristic of this method is that significant overshoot occurs during rapid, even smooth, changes in path curvature. This effect is again attributed to this method not having a look-ahead. This method is not robust like the geometric methods to large errors or path discontinuity.

The Optimal Preview method builds on the LQR method. The LQR with a feed forward term method performs well for many driving scenarios. The shortcomings of the method, for the most part, are during more extreme scenarios

that are not common or scenarios in which another method can perform well. The one exception is the overshooting problem during rapidly changing curvature. The Optimal Preview method provides the LQR method with a look-ahead, or preview, of the upcoming path to address the problem. The idea is that if a rapid change in curvature is known ahead of time the tracker can react earlier to minimize error. The proactive nature of the Preview method often sacrifices some error entering a curve to minimize the overall error through the curve. This approach results in similar performance as Model Predictive Control (MPC) under certain circumstances, but with a computational resources. It is a little more complicated than the LQR method, but it is still easy to understand and implement. A similar method for solving the optimal control solution is used for both the Preview tracker and the LQR tracker. The implementation and tuning are simple extensions to the standard LQR method. The tuning has one additional parameter that represents the preview time used for the look-ahead. This method does not always outperform all other methods. It has the same limitations from using the linear dynamic model as the LQR method. It provides more consistent results over a larger set of scenarios when compared to the LQR with a feed forward term method. The overshooting problem is reduced, however it may not perform as well in the other driving scenarios in which the LQR with a feed forward term excels. Another disadvantage of the Optimal Preview method is the constant velocity assumption. This assumption has been made for every method presented and always has some consequences. The Optimal Preview method is more effected by this assumption because of the preview time. For example, the vehicle could change velocity drastically over the 1-2 second time period the control is being optimized for. The linear time-invariant model results in a an efficient and effective controller, but one needs to be aware of the underlying assumptions.

This paper presents solutions to the path tracking problem for many applications and provided some insight into the choice and implementation of path tracking algorithms. The idea of a hybrid controller that uses two or more of these methods for more general driving has been mentioned. However, as robotic vehicles require better performance and tackle new objectives, the vehicle control methods presented here may be inadequate. This paper has revealed an opportunity for future work and the problems that should be addressed in that work. Future work could include driving scenarios like unexpected emergency maneuvers that pop up in normal passenger car driving or a variety of race car driving scenarios. It is clear that the simple models and linear techniques found here are limiting the performance that robotic cars can achieve. It is also understood that a great deal of work exists in the areas of system modeling and non-linear control. The scope of this paper is to present very efficient control strategies using easily identifiable models. In extending the scope to include new applications, the most obvious next step would be in the area of Nonlinear Model Predictive Control (NMPC) [3]. The use of NMPC with the nonlinear version of the dynamic bicycle model presented here and a planned velocity profile instead of a constant velocity would be a logical next step. Of course this still does not address the simplification of the tire model or the neglecting of suspension effects. This may result

in some small improvements in performance along with a considerable increase in computational resources. NMPC is somewhat understood and methods exist. The physics of car-like vehicles is well understood and high fidelity vehicle models, like the model used for simulation in this paper, exist. NMPC combined with high fidelity modeling requires tremendous amounts of computational resources and most applications need real time control. Therefore, future work may include determining a vehicle model with a subset of variables that captures enough of the dynamic effects to satisfy the requirements of more applications or using machine learning techniques on recorded vehicle data to move the computational burden off-line and produce simpler on-line models. Eliminating the model and the optimization by using machine learning techniques to "learn" the control law from a human driver is agenda for future research.

References

[1] Automotive testing technology international: Awards 2007 special issue, Nov/Dec 2007.

[2] Omead Amidi. Integrated mobile robot control. Technical Report CMU-RI-TR-90-17, Carnegie Mellon University Robotics Institute, 1990.

[3] Eduardo F. Camacho and Carlos Bordons. *Model Predictive Control*. Spinger, 2004.

[4] Stefan F. Campbell. Steering control of an autonomous ground vehicle with application to the DARPA urban challenge. Master's thesis, Massachusetts Institute of Technology, 2007.

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

[6] A. De Luca, G.Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In *Robot Motion Planning and Control*, pages 171–249. 1998.

[7] Mechanical Simulation. *CarSim Data Manual*.

[8] Mechanical Simulation. *The VehicleSim API: Running and Extending VehicleSim Solver Programs*.

[9] R. M. Murray. Control of nonholonomic systems using chained forms. *Fields Institute Communications*, 1:219–245, 1993.

[10] Huei Peng and Masayoshi Tomizuka. Optimal preview control for vehicle lateral guidance. Research Report UCB-ITS-PRR-91-16, California Partners for Advanced Transit and Highways (PATH), 1991.

[11] Rajesh Rajamani. *Vehicle Dynamics and Control*. Spinger, 2006.

[12] Jihan Ryu. *State and Parameter Estimation for Vehicle Dynamics Control using GPS*. PhD thesis, Stanford University, 2004.

[13] C. Samson. Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*, 40(1):64–77, 1995.

[14] R. S. Sharp. Driver steering control and a new perspective on car handling qualities. *Journal of Mechanical Engineering Science*, 219(C):1041–1051, 2005.

[15] Ashish Tewari. *Modern Control Design*. Wiley, 2002.

[16] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[17] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William "Red" Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[18] Chris Urmson, Charlie Ragusa, David Ray, Joshua Anhalt, Daniel Bartz, Tugrul Galatali, Alexander Gutierrez, Josh Johnston, Sam Harbaugh, Hiroki "Yu" Kato, William Messner, Nick Miller, Kevin Peterson, Bryon Smith, Jarrod Snider, Spencer Spiker, Jason Ziglar, William "Red" Whittaker, Michael Clark, Phillip Koon, Aaron Mosher, and Josh Struble. A robust approach to high-speed navigation for unhehearsed desert terrain. *Journal of Field Robotics*, 23(8):467–508, 2006.

[19] Jeffrey S. Wit. *Vector Pursuit Path Tracking for Autonomous Ground Vehicles*. PhD thesis, University of Florida, 2000.

[20] Paul Yih. *Steer-by-Wire: Implications for Vehicle Handling and Safety*. PhD thesis, Stanford University, 2005.