The Pennsylvania State University

The Graduate School

MODEL-BASED INCLUSION OF PREVIEWED INFORMATION FOR LATERAL VEHICLE STATE AND ENVIRONMENT ESTIMATION

A Dissertation in

Mechanical Engineering

by

Alexander Allen Brown

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

August 2013

The dissertation of Alexander Allen Brown was reviewed and approved* by the following:

Sean N. Brennan
Professor of Mechanical Engineering
Dissertation Advisor, Chair of Committee

H.J. Sommer
Professor of Mechanical Engineering

Christopher D. Rahn
Professor of Mechanical Engineering

Jacob W. Langelaan
Professor of Aerospace Engineering

Richard L. Tutwiler
Senior Research Scientist, Penn State Applied Research Laboratory

Karen A. Thole
Professor of Mechanical Engineering
Department Head of Mechanical and Nuclear Engineering

*Signatures are on file in the Graduate School.

Abstract

This dissertation is concerned with the marriage of spatiotemporal preview as it is known in the controls literature with map-based guidance in a simple but effective paradigm, in the interest of making accurate environment and vehicle states accessible with production-grade sensing and computing equipment. It does so without the two-dimensional data association issues of SLAM. Specifically, ideas from linear optimal preview control theory will be employed to develop mathematically tractable and intuitively insightful estimation frameworks that use map, inertial, and forward-looking monocular camera information. The framework developed is intended to provide state and path estimates to an optimal preview controller guiding a vehicle in a static environment at highway speeds. Three variations on preview-based estimation are considered, with experimental results for both human-driven and self-steering vehicles.

Table of Contents

List of Figures

List of Tables

Dedication

This dissertation is dedicated to my wife, Elise Veronica Settanni. She is my North Star. She is all that keeps me from losing my way in the forest forever.

Chapter 1

Introduction

1.1 Background

In a world moving swiftly towards fully autonomous passenger vehicles, their ability to track a particular path using uncertain estimates of both ego-motion and environment remains a key challenge. For feedback control to be successful, a physical system's states must be known, and so must its reference input. This sounds simple enough, but in the context of driver assist and autonomous road vehicle systems, knowing where the vehicle is (states) and where the vehicle is going (reference input) are both difficult, and have been the subject of rigorous study for several decades. For vehicles under robotic steering control, which mimics the "pursuit"-type task shared with aircraft guidance tasks [1], it is not only necessary to know where the vehicle *is*, but also where it *must be* at some lookahead or preview distance in the future.

In the case of active control of a vehicle's steering to traverse a road network, it is generally assumed that maps are available of the road geometry, which offer an autonomous system a pre-defined reference path in a global coordinate frame. Therefore, the vehicle "knows" where it is supposed to go. While this map information is often assumed known for control purposes, it can also be helpful in the estimation of vehicle states. Map information has already shown to be a useful tool in improving vehicle localization accuracy, even with relatively sparse maps [2]. This dissertation develops a novel way of approaching the problem of perceiving a road vehicle's reference path (the road shape) that offers a-priori error statistics estimates given a set amount of visibility from a forward-looking camera. The framework makes use of mapped information, and uses measurements of a lane line to update the vehicle's estimated states, and also updates an estimate of the location of the road in front of the vehicle. This way, optimal preview control strategies can be employed with knowledge of the error expectation of the vehicle's state and environment estimates.

The remainder of this chapter is organized as follows. After a brief introduction to the models of vehicle dynamics that permeate the literature, a background on the history and current state of the art in road vehicle ego-state and environment perception follows suit. Then, a review of common preview-based feedback and feed-forward control strategies that make use of environment information is presented. This discussion is followed by an argument for decomposition of the

requirement for knowledge of what lies *ahead* of the vehicle into two distinct categories: previewed information needed for effective vehicle tracking/state estimation, and previewed information needed for effective vehicle guidance using optimal preview control. An existing optimal preview control framework for vehicles is introduced, and is then used as inspiration in the development of an estimation paradigm. With the addition of a matching optimal preview control design, the resulting stochastic guidance system fuses combinations of map, camera, and inertial measurements with a methodology that has tractable theoretical links to existing methods under reasonable assumptions.

1.2 Vehicle dynamic models

If the reader is unfamiliar with lateral vehicle chassis dynamics, excellent and mature treatments of the subject appear in [3, 4, 5] and others. This section is intended to provide only enough background on vehicle dynamics to assist the reader in digesting the material that follows. It gives some context for the empirical relevance of the bicycle model equations, which are simplistic in the age of fast computers and multibody models, but still adequately descriptive of most driving, and widely used in theory and practice.

First, consider a representative vehicle with assumed "lumped" left and right tires as in Figure 1.2. In this setup, with global coordinates in the familiar East-North-Up (ENU) frame and body-fixed vehicle coordinates in the ISO standard

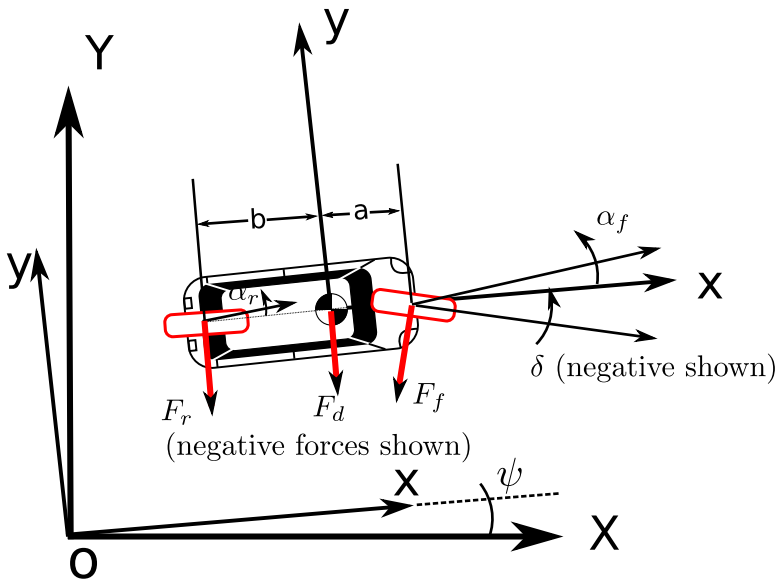

Figure 1.1. Setup of planar vehicle dynamic equations for error-coordinate and local-coordinate representations (negative tire forces shown, all angles positive counterclockwise from axis)

convention. Roll motion from the suspension is ignored, and both lumped tire forces are assumed to act in the vehicle's local y-direction. This is true even though the technically correct representation of $\vec{F}_f$ is $\vec{F}_f = |F_f| \left[\cos(\delta)\,\hat{j} - \sin(\delta)\,\hat{i}\right]$. Longitudinal tire forces are not modeled for any of the work that follows, but the derivations below hold for slight deviations from constant forward speed, as long as braking and acceleration forces are small and do not excite suspension dynamics or push tires into a nonlinear region of behavior.

Both the linearized global equations of motion, deemed as a set the "error-coordinate" model of lateral vehicle dynamics, and the body-fixed coordinate representation of the same can be derived using Figure 1.2.

1.2.1 Tire forces

As mentioned above, tire forces are modeled for the present purposes in the lateral direction only, since these are most important for understanding how the lateral dynamics of a vehicle change under control by a robotic or human driver. In general, tires produce lateral force as a response to angular misalignment between their heading direction (direction in which they point) and their course direction (direction in which they are moving). This angular misalignment, termed "slip angle," is a small signal relative to yaw rate or forward velocity, but it is also too large at appreciable forward velocities to ignore for high-speed lateral guidance. For the purposes of the work to follow, this tire/force relationship is assumed to be linear, is given by:

$$\begin{aligned} F_f &= C_f \alpha_f \\ F_r &= C_r \alpha_r \end{aligned} \tag{1.1}$$

It is worth noting that tire dynamics are quite complex, and hard to capture completely, especially with a linear model. During the construction of Penn State's autonomous testbed, "Big Red," a large amount of effort was expended characterizing effective cornering stiffness values C_f and C_r for the vehicle in its laden and unladen configurations, and for several different sets of tires and tire pressures [6].

1.2.2 Derivation of body-fixed vehicle dynamics

In body-fixed coordinates, the vehicle states are all referenced to the coordinate frame attached to the vehicle's center of mass. Allowing the represented vehicle to have both longitudinal (forward) velocity and lateral (sideslip) velocity at its center of mass, the velocity of the vehicle can be written as in eq. 1.2

$$\vec{v}_{total} = U\hat{\imath} + V\hat{\jmath} \tag{1.2}$$

This obviously ignores any jounce motion of the vehicle. Ignoring also the possible roll and pitch motions of the vehicle, the body-fixed coordinate frame has only one component of angular velocity, given by eq. 1.3.

$$\vec{\omega}_{xyz} = r\hat{k} \tag{1.3}$$

Thus, the slip angles of the tires can be written simply using relative velocity equations:

$$\begin{aligned} \alpha_f &= \frac{V + ar}{U} - \delta \\ \alpha_r &= \frac{V - br}{U} \end{aligned} \tag{1.4}$$

Taking the derivative of the allowed vehicle velocity given by eq. 1.2 allows for the first-principles derivation of the lateral equations of motion for the vehicle. In doing this differentiation, the acceleration of the vehicle is thus given by eq. 1.5.

$$\begin{aligned} \vec{a}_{total} = & \quad \frac{d\vec{V}}{dt} = \frac{dU}{dt}\hat{\imath} + U\frac{d\hat{\imath}}{dt} + \frac{dV}{dt}\hat{\jmath} + U\frac{d\hat{\jmath}}{dt} \\ = & \quad \dot{U}\hat{\imath} + U\vec{\omega_{xyz}} \times \dot{\hat{\imath}} + \dot{V}\hat{\jmath} + V\vec{\omega_{xyz}} \times \hat{\jmath} \\ = & \quad \left(\dot{U} - Vr\right)\hat{\imath} + \left(\dot{V} + Ur\right)\hat{\jmath} \end{aligned} \tag{1.5}$$

Because lateral dynamics of the vehicle are the focus here, this leads to a summation of the forces in the local y-direction as shown in eq. 1.6

$$\sum F_y = \quad ma_y = \quad m(\dot{V} + Ur) = \quad Cf(\frac{V + ar}{U} - \delta) + C_r\frac{V - br}{U} + F_d \tag{1.6}$$

Where the total mass of the vehicle is given by m, and the lateral disturbance force is commonly assumed to be $F_d = -g\frac{e}{100}$, where e is the superelevation, or tilt of the road about the vehicle's x-axis, and g is the gravitational constant, $9.81\frac{m}{s^2}$. This term comes from the definition of superelevation, which is $e = 100\tan\theta$, where θ is the roll angle of the road about the vehicle's x-axis. This is shown in Figure 1.2. Superelevation angles are assumed small in the vehicle dynamic equations, so $\tan\theta \approx \theta$ to maintain linearity.

The moment balance about the yaw (z) axis of the vehicle is then given by eq. 1.7 where the yaw mass moment of inertia of the vehicle is given by I.

$$\sum M_z = \quad I\dot{r} = \quad a(Cf(\frac{V + ar}{U} - \delta)) - b(C_r\frac{V - br}{U}) \tag{1.7}$$

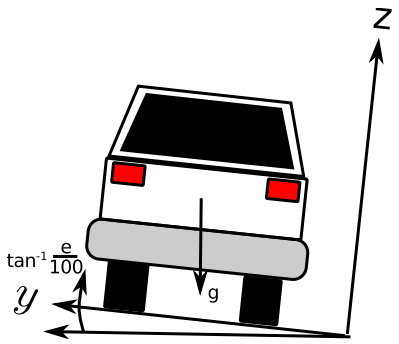

Figure 1.2. Definition of vehicle superelevation

Rearranging these equations results in the familiar state-space representation of body-fixed lateral vehicle dynamics in eq. 1.8

$$\begin{aligned} \dot{V} = &\quad \frac{C_f + C_r}{mU}V + (\frac{aC_f - bC_r}{mU} - U)r - \frac{Cf}{m}\delta - g\frac{e}{100} \\ \dot{r} = &\quad \frac{aC_f - bC_r}{IU}V + \frac{a^2C_f + b^2C_r}{IU} - \frac{aC_f}{I}\delta \end{aligned} \tag{1.8}$$

This is the form in which all local-coordinate preview-filter equations inclusive bicycle model dynamics will appear.

1.2.3 Derivation of lateral vehicle dynamics in error-coordinates

Often, when vehicle dynamics in the context of path-following or lane-keeping are considered, it is convenient to track a vehicle's global lateral position and yaw angle ψ as well as its lateral velocity and yaw rate. In order to do this, consider a situation in which the vehicle stays approximately aligned with the global X-axis as defined in figure 1.2. In general, the transformation from local coordinates to global coordinates for vehicle velocity rotated in-plane by the global yaw angle ψ

is given by eq. 1.9,

$$\begin{aligned} \dot{X} = &\ U\cos(-\psi) - V\sin(-\psi) \\ \dot{Y} = &\ U\sin(-\psi) + V\cos(-\psi) \end{aligned} \tag{1.9}$$

which, when small angle ψ is assumed, leads to eq. 1.10:

$$\begin{aligned} \dot{X} = &\ U + V\psi \\ \dot{Y} = &\ V - U\psi \end{aligned} \tag{1.10}$$

As in eq. 1.5, vector derivatives of velocities are computed, as shown in eq. 1.11, where $\hat{J}$ and $\hat{I}$ represent unit vectors in the global coordinate frame, to differentiate between those used in the body-fixed formulation above. However, although it is not shown, $\hat{k}$ is the same in both formulations, since it is assumed that the vehicle only moves in-plane.

$$\begin{aligned} \frac{d\vec{v}_{total}}{dt} = &\ \left(\dot{U} + Vr + \dot{V}\psi\right)\hat{I} + \left(Ur + V\psi r\right)\hat{J} \\ + &\ \left(\dot{V} - Ur - \dot{U}\psi\right)\hat{J} + \left(Ur\psi - Vr\right)\hat{I} \end{aligned} \tag{1.11}$$

Cancellation of assumed-zero $\dot{U}$ and rearrangement to extract Y components of acceleration only leads to a global Y-acceleration as given by eq. 1.12 when the very small multiplicative nonlinearity Vr is neglected. It is interesting to note that the "coriolis term" Ur in the body-fixed lateral acceleration disappears as a consequence of the small-ψ assumption and coordinate system transformation.

$$a_Y = \dot{V} + Vr\psi \approx \dot{V} \tag{1.12}$$

With the approximations of eq. 1.12, the definition of lateral acceleration does not change, but the definition of slip angles with respect to states does, because of the transformation from V to $\dot{Y}$ in eq. 1.10. Slip angles for error-coordinates are thus given in eq. 1.13.

$$\begin{aligned} \alpha_f &= \frac{V + ar}{U} - \delta \approx \frac{\dot{Y} + ar}{U} - \psi \\ \alpha_r &= \frac{V - br}{U} \approx \frac{\dot{Y} - br}{U} - \psi \end{aligned} \tag{1.13}$$

With this change, the force and moment balances of eqs. 1.6 and 1.7 are trivially repeated. Traditionally, in the literature, the global "error-coordinate" Y is replaced with a lower case y, and local yaw-rate r is replaced with the convenient $\dot{\psi}$, resulting in a state-space equation set as given in eq. 1.14

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{C_f+Cr}{mU} & \frac{-\left(C_f+Cr\right)}{m} & \frac{aC_f-bC_r}{mU} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{aC_f-bC_r}{I_zU} & \frac{bC_f-aC_r}{I} & \frac{a^2C_f+b^2C_r}{IU} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-C_f}{m} \\ 0 \\ \frac{-aC_f}{I_z} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ -\frac{g}{100} \\ 0 \\ 0 \end{bmatrix} e \tag{1.14}$$

This form of the vehicle dynamic equations, while convenient, is not suitable for situations where the small-ψ approximation is violated, such as constant-radius

turns. For simulation and illustration purposes, or for control on a relatively straight path, the error-coordinate form of the bicycle model equations is adequate. But when practical implementation on significantly curved roads is the goal, the body-fixed formulation is often more appropriate.

1.2.4 Discretization of vehicle dynamic models

As the following sections and chapters will demonstrate, the implementation of vehicle dynamic equations is readily realized in discrete time, rather than continuous time. Because of this, each of the two linear vehicle representations above is converted to discrete time using a zero-order hold such that the A and B matrices become discrete-time state and input transition matrices A_d and B_d. For the studies in optimal preview control and preview-based estimation that follow, this discretization is accomplished using the Tustin or bilinear transform approximation of the state transition and input matrices. For a simple linear state-space model, Eq. 1.15 describes its continuous-time dynamics:

$$\begin{aligned} \dot{\vec{x}} &= A\vec{x} + B\vec{u} \\ \vec{y} &= C\vec{x} + D\vec{u} \end{aligned} \tag{1.15}$$

In general, in order to predict a system's states at some time t in the future, it is necessary to solve the matrix form of the first-order continuous-time differential

equation given by Eq. 1.15.

$$\vec{x}(t) = e^{At}\vec{x}(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \tag{1.16}$$

The matrix exponential e^{At} is the analog of the scalar exponential for multivariable first-order differential equations, and is given by:

$$e^{At} = \mathcal{L}\left[s\mathbf{I} - A\right]^{-1} \tag{1.17}$$

Many possible approximations of the matrix exponential exist, but the most convenient and flexible approximation for discretization of linear systems with relatively small sample times is the bilinear or Tustin transform. Using this approximation, one can replace $x(t)$ with a discrete representation of the system's state at a particular sample number, x_k, and replace initial condition $x(0)$ with the sampled value of the system states at the previous timestep, x_{k-1}. In doing so, one can write the discrete-time recursive state space equations for a linear system as in [7]:

$$\begin{aligned} \vec{x}_k = &\quad A_d\vec{x}_{k-1} + \quad B_d\vec{u}_{k-1} \\ \vec{y}_k = &\qquad C\vec{x}_k + \qquad D\vec{u}_k \end{aligned} \tag{1.18}$$

Where the discrete-time matrices A_d and B_d are given by:

$$e\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} A_d & B_d \\ 0 & 0 \end{bmatrix} \approx \left[\mathbf{I} + \frac{dT}{2}\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}\right]\left[\mathbf{I} - \frac{dT}{2}\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}\right]^{-1} \quad (1.19)$$

with dT equal to the time between samples. As derived above, the vehicle dynamic discrete-time equations can be updated at each timestep with the appropriate measured value of the vehicle forward velocity U for simulation, implementation of estimation equations, and control.

1.2.5 Open-loop fidelity of the vehicle dynamic models

While much concern and effort in modern vehicle dynamics research is expended upon high-fidelity multi-body models, which capture minute variations in vehicle states inclusive of tire, suspension, bushing, and aerodynamic phenomena, these effects are overshadowed by the gross variation in lateral vehicle states described by simple, linear dynamics of the bicycle model for most driving circumstances. When effective cornering stiffnesses for a particular vehicle are carefully calculated from steady-state circular driving tests performed with a defense-grade position, velocity and orientation sensor, as in the work by the author and others in [6], and the vehicle's yaw inertia is calculated from steering frequency response data, the bicycle model matches measured vehicle states very well. This is important,

because all of the estimators derived in this document rely on an accurate vehicle model.

To illustrate the degree to which the bicycle model matches actual vehicle behavior, consider a test in which a double lane-change maneuver was performed with Penn State's autonomous testbed "Big Red," where the forward speed and steering input are given by Figure 1.3

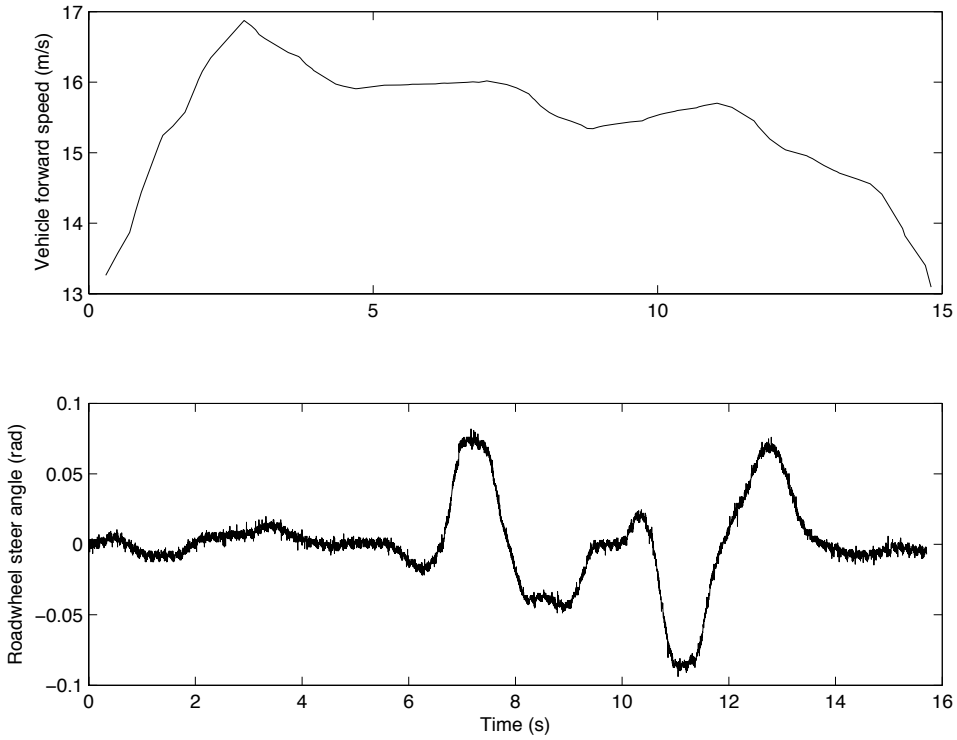

Figure 1.3. Forward velocity and steering angle for a double lane-change maneuver

Data for this maneuver were collected using a defense-grade Inertial Navigation System (INS), which does not rely on a vehicle model for measurement of lateral velocity, yaw rate, or other rotational or linear velocities. The vehicle's lateral acceleration during the maneuver is shown in Figure 1.4. The lateral acceleration

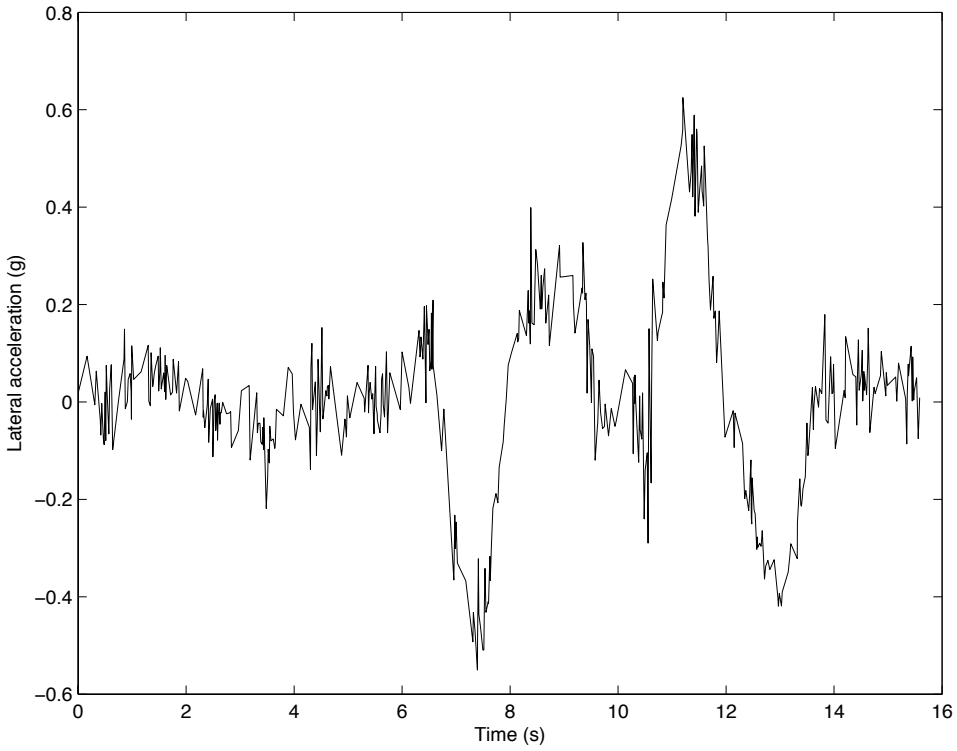

Figure 1.4. Measured lateral acceleration for a double lane-change maneuver

for the maneuver peaks at over $0.5g$, which is fairly aggressive, and definitely forces the tires to approach the nonlinear regions of their lateral force characteristics. However, when a linear time-varying simulation of eq. 1.8 is performed using the forward speed profile to populate the system dynamics and the steering angle as an input, the resulting match between states measured in the field and states predicted by the model is excellent, and shown in Figure 1.5. This level of model fidelity, the result of repeated, redundant parameter identification tests, sets the stage for the inclusion of bicycle model dynamics in the remainder of this document for the purposes of state estimation and lateral vehicle control.

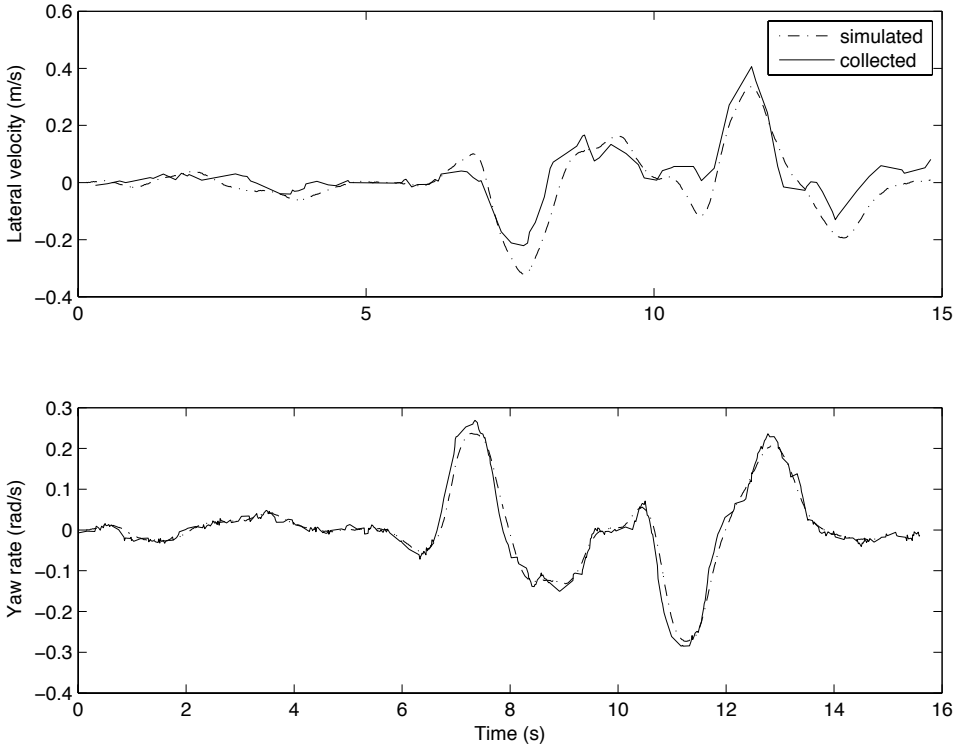

Figure 1.5. Body-fixed vehicle states for a double lane-change maneuver

1.3 Estimation of lateral vehicle dynamic states

While the open-loop match between simulated and measured vehicle states as shown in the preceding section was very good, this is usually only true when vehicle dynamic states are estimated using a research-grade sensor suite, and only under controlled conditions. Estimating vehicle states using production-grade sensing equipment is hard, and forces many production driver assist systems to be quite conservative in anticipation of sensor error [8]. The relatively low signal-to-noise ratio of production sensors makes it challenging to measure vehicle states like sideslip, the angle between the the vehicle orientation and the vehicle's total velocity vector, because sideslip has extremely small magnitudes under normal driving

conditions. For example, a typical passenger vehicle traveling at $U = 20\frac{m}{s}$ (40 mph) forward velocity around a turn with a lateral acceleration of $4\frac{m}{s^2}$ may have a lateral velocity of only $0.5\frac{m}{s}$ (1 mph).

This fact alone makes estimating lateral velocity difficult, and many low-cost sensors suffer from bias instability, quantization effects, and poor resistance to temperature or other environmental variability. As a result, the use of common low-cost inertial sensors in traditional Kinematic Kalman Filters (KKFs) is often out of the question, although success with vehicle sideslip estimation without a model using GPS and yaw gyro measurements was shown in [9]. Some researchers in the vehicle dynamics community have turned towards model-based estimators that make use of known vehicle dynamics to improve estimator accuracy [10, 11]. Some have even found success using model-based estimation techniques to estimate vehicle parameters and/or tire-pavement friction in real time [12, 13, 8]. In an application aimed at predicting lane departure, Mudaliar used a model-based Kalman filter in the design of a lane departure warning system [14].

To provide a brief, basic introduction to the Kalman filter, in the form carried through the remainder of this dissertation, consider the form of the state estimator as given in [15] in eq. 1.20, where Q_k is the covariance matrix of a modeled white noise perturbing system inputs, and R_k is the covariance matrix of the assumed Gaussian white noise perturbing sensor measurements. If the system is linear and observable, and the process and measurement noise are actually white and random

with the modeled covariances, then this is an optimal minimum-variance estimator.

$$\begin{aligned}
\hat{x}_{k|k-1} &= A_{d,k-1}\hat{x}_{k-1|k-1} + B_{d,k-1}u_{k-1} \\
P_{k|k-1} &= A_{d,k-1}P_{k-1|k-1}A_{d,k-1}^T + Q_k \\
\tilde{y}_k &= z_k - H_k\hat{x}_{k|k-1} \\
L_k &= P_{k|k-1}H_k^T\left(H_kP_{k|k-1}H_k^T + R_k\right)^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_k\tilde{y}_k \\
P_{k|k} &= \left(\mathbb{I} - L_kH_k\right)P_{k|k-1}
\end{aligned} \tag{1.20}$$

While these assumptions are never exactly satisfied for a physical system, variations on the Kalman filter are often used to estimate vehicle states by fusing measurements of steer angle δ and other disturbances with measurements from sensors like yaw rate gyroscopes and accelerometers. When the described system dynamics are nonlinear, or system parameters in a linear system are to be estimated as augmented states as in [12, 13], the so-called "extended Kalman filter" can be used, where the system dynamics are linearized at each time step, and the estimation process proceeds as in eq. 1.20. While this is no longer a truly optimal estimation framework, and can introduce biases through linearization, it is experimentally mature. The Kalman filter methodology employed in this dissertation appears as a linear time-varying (LTV) Kalman filter as in eq. 1.20.

Using maps to enhance vehicle localization and state estimation is becoming more and more popular to improve state estimates, and allow a vehicle to register

within a map so that the static pieces of its environment can be perceived without direct measurement. Recent work in[16, 17, 2, 18] makes use of extremely compact maps of roads to localize a vehicle by using a measurement of its pitch angle alone. Additionally, many have turned to the more recent trend of map-based localization using sensors like Light Detection and Ranging (LIDAR) sensors [19] and cameras [20, 21, 22]. Vishisht Gupta was successful in 2008 with global localization and tracking tasks using a GPS, forward-looking camera, and an IMU in [20]. His method used an EKF to track mapped road signs and other features in a video stream, along with horizon shapes (mountains) to aid in localization. Also in 2008, Oliver Pink used a map of road markings generated from an aerial image to match forward-looking camera images using an iterative matching method in [21], achieving good localization and tracking performance. Other uses of aerial imagery and forward-looking cameras for map-based localization include [23], in which the authors use an EKF similar to the one proposed in [20] and using sensors and map sources similar to [21] to achieve localization and tracking. Other similar approaches include the work by Noda in [22]. A review of matching and registration techniques for image-based localization using aerial images is given in [24].

These methods are effective, but make use of either iterative methods or complex mathematical structures to achieve position and velocity state tracking, and treat the map of road features with an eye towards map registration only, rather than tracking them as additional vehicle states with their own statistical confi-

dence levels. This is not a large problem if the environment to be traveled in is fully static, but does present an issue if filter estimates diverge. In this case, a vehicle would have no idea where it is or where it must go, even with attempts at environment measurement. The two filters in this dissertation developed for use with a forward-looking camera address some of these issues, and allow for changing road markings to be updated as previewed states, and readily available to a preview or predictive steering controller. The methodology derived in the chapters that follow, able to function with or without map information, will prove to be a simplification of the famous Simultaneous Localization and Mapping problem, which allows a "robot" to explore its environment, discovering features and localizing both those features and the robot's state relative to them. In this way, the methodology is more flexible than methods that offer map registration only.

1.4 Simultaneous localization and mapping

Simultaneous Localization and Mapping (SLAM) is a problem statement rather than an algorithm. In an unknown environment, a robot wishes to create a map of what lies ahead of and around it, and also wishes to localize itself within that map with the highest degree of accuracy and precision possible [25]. To visualize the meaning of the SLAM problem, consider Figure 1.6, and consider that the vehicle pictured is a generic ground robot, without making assumptions about its dynamic equations.

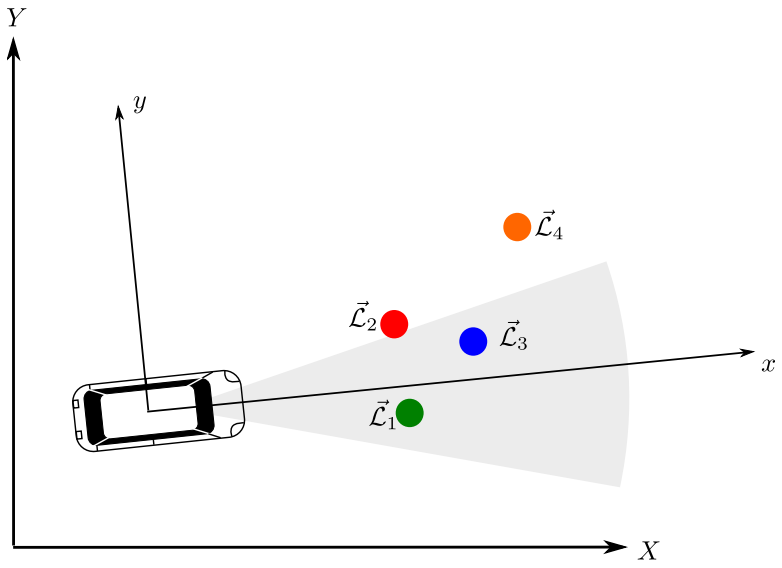

Figure 1.6. Definition of the SLAM problem

Here, note that the vehicle's sensor, which could be a camera, laser, sonar, or something else entirely, is able to measure the two dimensional location of certain features within the scope of its sensors. If no absolute positioning sensor (such as a GPS) is available, the vehicle can only build a local map initialized with a zero position at the start of the procedure, and cannot recover globally referenced position. If, at first, the vehicle can see no features, suppose it maintains estimates of its own states through propagation of the system dynamic equations shown in eq. 1.21.

$$\vec{X}_{r,k} = A_d \vec{X}_{r,k-1} + B_d \vec{u}_{k-1} \tag{1.21}$$

Once the robot is able to observe landmarks in its environment, it tracks those in an estimation framework as well, using measurements of the landmarks relative to the vehicle in combination with vehicle state propagation equations to maintain

estimates of both the environment and the vehicle itself. To do this, Cheeseman, Smith, and Self proposed an Extended Kalman Filter (EKF) solution to the SLAM problem in 1987 [26]. Their solution augments the Kalman-filtered robot state estimate vector with measured landmark positions as they are discovered. While it has been extended and improved vastly since its original inception, this basic methodology is still in wide use today [25]. In this framework, as the robot measures landmarks, and augments its system state vector. This state vector is often set to either grow boundlessly as the vehicle explores its environment, or "forget" map segments whose measurements are corrupted or far away.

$$\tilde{\mathbf{x}} = \begin{bmatrix} \vec{X}_r \\ \vec{\mathcal{M}} \end{bmatrix}_k = \begin{bmatrix} \vec{X}_r \\ \vec{\mathcal{L}}_1 \\ \vec{\mathcal{L}}_2 \\ \vec{\mathcal{L}}_3 \\ \vdots \\ \vec{\mathcal{L}}_n \end{bmatrix}_k \tag{1.22}$$

In Equation 1.22, each state $\vec{\mathcal{L}}_i$ in the map generally consists of a two-dimensional position. Through the propagation of the robot states and repeated measurements of landmarks (including the discovery of new ones), the vehicle uses the measurements of its environment to improve its own state estimates through an estimation framework like the EKF, a particle filter, an Unscented Kalman filter, etc. In

many applications of SLAM, The robot's measurements of its surroundings are accomplished through a laser rangefinder, a camera [27, 28, 29], or other sensor combinations and used to couple the robot's position with that of the stationary landmarks which surround it. SLAM is an incredibly popular field, and comprehensive reviews of many commonly used algorithms are available in review papers such as [30]. Thrun and Montemerlo's seminal 2002 paper [31] propelled the SLAM algorithm past the computational complexity of the EKF by using a Rao-Blackwellized factorization of the posterior probability distribution of robot pose. Their algorithm is also now in wide use in the robotics community, as well as on Stanford University's autonomous road vehicle "Junior" [32].

The fundamental issue with SLAM as a generic solution for autonomous vehicle systems needing knowledge of their surroundings is that the "landmarks" tracked as part of the algorithm can be of ambiguous importance and/or unknown reliability. In other words, in the general formulation of the SLAM problem, it is necessary to define what a "landmark" feature will be. For laser-based methods, this is relatively easy, as data association (the process of determining which landmark as time step $k-1$ is now the same landmark at time k) is easier with reliable two-dimensional position and reflectivity measurements. For lasers, tracked landmark features are also guaranteed to be physical objects, whether stationary or not. For vision-based methods, however, such as [29], features in a camera framework are much more difficult to define. Many times, the features are visual "corners," which

Figure 1.7. Corner feature scarcity during night driving

can provide reasonably dense feature maps even with a monocular camera as in [27]. However, when considering the task of high-speed vehicle guidance, which occurs in a variety of driving environments, lighting conditions, and weather, it is nearly impossible to guarantee that such corners will be found in a vision-based system. Therefore, it is not possible to give realistic confidence bounds on the vehicle's current position relative to the road, or its velocity states, which are necessary for guidance. Consider, for instance, the night time driving scene shown in Figure 1.7, where the only corners in the image are provided by the reflectors far in front of the vehicle.

The absence of traditional features useful in most SLAM frameworks is also apparent with changing weather conditions. Figure 1.8 shows an example of a morning scene on a rural highway, where nearly all visual features are obscured by

Figure 1.8. Corner feature scarcity during adverse weather

fog, other than the lane lines. Lane lines, in the two-dimensional SLAM context, suffer from inadequate feature definitions. They have uniqueness as edges, but suffer as features when data association from time step to time step is required, since the ego-motion of a vehicle along the length of the lane is hard to pick out using vision alone, even for a stereo system that provides three dimensional edges to the SLAM framework. Still, lane lines are excellent and popular candidates for defining a vehicle's reference trajectory in autonomous systems, because they are nearly ubiquitous on public roads, and are designed specifically to be easily seen. Practitioners using lane lines for vehicle guidance have made and continue to make clever assumptions to aid in the perception of the vehicle's relationship to its reference trajectory without the problems they pose for a SLAM framework.

1.5 Vision systems for road scene understanding and vehicle control

Both of the state estimation frameworks described in this dissertation rely on camera measurements of lane position to provide information about the world directly in front of a cruising autonomous car. Cameras are a natural choice for vehicle control primarily because of the immense amount of information that can be extracted from any one image or image sequence, and because lane lines, which define a vehicle's current reference path, are purposefully easy to see in day or night. The problem is that often, images can contain *too much* information, and the information necessary for control is often out of reach or at least difficult to extract from an array of color pixels.

In [33], Joel McCall evaluates the performance of state-of-the-art lane detection systems before presenting his own vision-based driver assist system driven by objectives derived from human factors research. McCall's work in [33] gives some perspective on recent contribution in lane detection research, including attempts to make lane extraction robust to changing weather and lighting conditions, which are incredibly important topics for a commercial implementation of the work described in this dissertation as well. [34] , [35] and [36] also provide excellent reviews of lane-detection and vision-based driver assist technologies.

In general, monocular images pose issues for 3-dimensional analysis of a road

scene, but this limitation is often attacked through the use of either planar road constraints [37] or stereo camera rigs [38]. Nearly all of the vision systems in the literature designed to be coupled with a lateral controller use a geometric model of road edge shape, to which measurements from forward-looking cameras are fitted [37, 33, 36, 34]. A common form of this model as presented in [39] is given below in eq. 1.23, in which C_0, C_1, C_2 define the clothoid fit parameters for the road, ϵ represents the current angular offset between the road tangent and the vehicle's x axis, y_0 represents the vehicle's current offset from the lane edge, Δx is the distance of a measured lane edge from the vehicle, and $\Delta\psi$ is a change in vehicle yaw from time step to time step.

$$\begin{bmatrix} y_0 \\ \tan(\epsilon) \\ C_0 \\ C_1 \end{bmatrix}_k = \begin{bmatrix} 1 & \Delta x & \frac{\Delta x^2}{2} & \frac{\Delta x^3}{6} \\ 0 & 1 & \Delta x & \frac{\Delta x^2}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_0 \\ \tan(\epsilon) \\ C_0 \\ C_1 \end{bmatrix}_{k-1} + \begin{bmatrix} 0 \\ \Delta\psi \\ 0 \\ 0 \end{bmatrix} \tag{1.23}$$

Ernst Dickmanns, in [37], employs a similar spatiotemporal road model to eq. 1.23 extensively in lane-keeping tasks for his team's autonomous vehicles. Dickmanns' group drove autonomously on the Autobahn in Germany in the 1980s using 16MHz microprocessors. Dickmanns' work estimated road curvature as a parameter in his vision-based estimator, and used a simplistic vehicle model in a coupled estimation and control task that fused measurements of the road boundary

at multiple lookahead distances. Here, steering and yaw rate were included in an EKF framework, but it often used a simplified vehicle model neglecting yaw acceleration, and could not deal with jagged or irregular road edges without fitting them to the geometric road model.

Similarly, In [40], C.J. Taylor's work uses vision to estimate the lateral offset of the road at a lookahead distance, as well as the angular misalignment of the road as viewed from the vehicle. Road curvature obtained from a lane recognition algorithm is fed into the combined road-vehicle dynamic equations as an exogenous disturbance. The authors investigate the control-specific effects of increasing lookahead distance, finding that increasing or decreasing preview has specific effects on closed-loop controller-estimator performance and stability. But because their vision-based estimation of lateral offset and angular misalignment are accomplished with a decoupled, simplified filter (neglecting lateral velocity), the full benefits of coupling road geometry measurements with vehicle states in the process of estimating both is not fully achieved. The same is true of similar work in [41], which models vehicle slip angle as a noise-driven integrator, rather than incorporating lateral velocity explicitly in the estimation framework or vehicle model.

Southall and Taylor proposed a method for estimating road shape parameters using a particle filter and a similar model of road geometry to eq. 1.23 in 2001 with [39]. This work employed a particle filter to represent a parametrized version of road geometry similar to that in eq. 1.23. This geometric road model

maintains a pervasive theme in vision-based road scene understanding, marked by the assumption that lane markers are smooth and follow parametric equations. While this is true in principle, the filters developed in subsequent chapters may be extended to make use of jagged or otherwise "unique" lane markings or environmental features to the benefit of the vehicle and road geometry state estimate. Furthermore, erroneous lane detection or limited visibility have unknown effects on this type of algorithm and its accuracy in estimating vehicle states and the vehicle's environment. While [40] investigates the effects of lookahead distance on controller performance for the particular controller employed, the effects of camera measurement noise, lookahead distance, and general lane visibility are not discussed.

To address these issues, a temporal preview horizon framework to estimate previewed road segments assumed to have piecewise constant curvature was proposed in [42]. This framework used a vanishing-point approach to generate a series of estimates of vehicle sideslip β and yaw rate r by assuming that at each of the 9 preview distances in the filter, the road segment had constant curvature. The curvature parameters at each preview distance, ρ_i, were estimated in each frame using a spatial fit, and then recursively corrected at later time steps by enforcing that the road curvature at time k, distance i in front of the vehicle must be the same as the road curvature at time $k+1$, at distance $i-1$. The block diagram representation of this approach is shown in Figure 1.5.

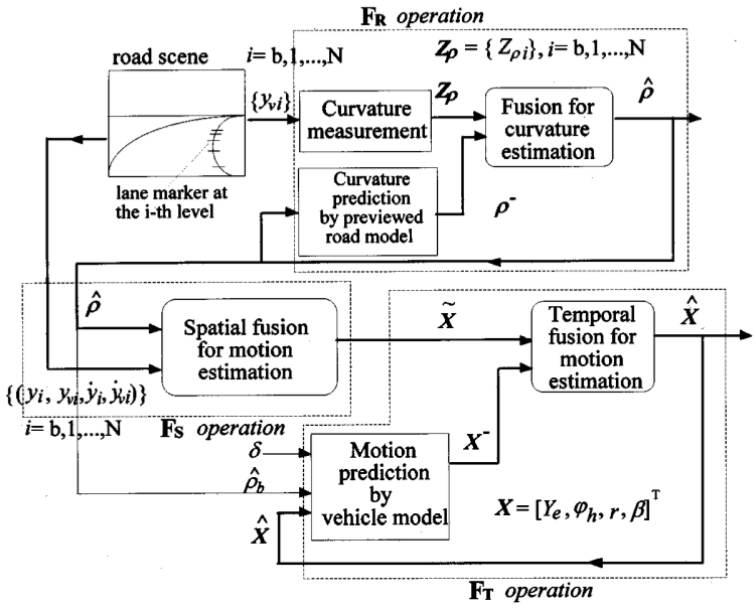

Figure 1.9. A temporal preview-based road curvature and vehicle state estimation approach (Hsu 1997)

The approach in Figure 1.5 is, at first glance, similar to the preview estimation frameworks derived in this dissertation, but the filtered curvature parameters representing road shape were not propagated or inherently corrected by a total vehicle-road motion model, in the fashion of SLAM or as was accomplished in the work of Dickmanns et. al. The road curvatures are maintained in a temporal shift register and propagated using the vehicle's forward speed U, but the total geometry of the road is not. This is an important distinction, because the simple, linear frameworks derived in this dissertation offer a complete fusion of vehicle and previewed road geometry, and without any sort of assumptions about road curvature (or lack thereof) and without the use of a clothoid fit model for road geometry.

Additionally, the filters described above had no provisions for inclusion of map

information, and in brief periods of false or absent measurements, it is not clear from the work how the controllers or estimators would cope, especially on a road with complex geometry. For instance, if a control structure relied on information about the road out to a particular horizon, but the camera could not see far enough, it is unclear what the general implications for a path-following controller would be given this type of estimator, with the exception by Hsu and Tomizuka's work in [43], which does assess controller performance as a function of camera lookahead distance, but only for a simplistic feedforward-feedback control architecture.

The lack of a-priori knowledge about features that haven't been measured yet, a drawback of all vision-based road and vehicle state perception systems without map integration, is also a shortcoming of SLAM. This is true even though the clothoid or curvature fit lane perception models overcome the feature detection problems SLAM would have with lane lines. The methods developed in the following chapters are able to overcome the limitations of SLAM when tracking long lane features, but do not require geometric lane fitting, and are able to use each individual lane measurement to aid in lateral velocity estimation, as one would expect from true SLAM, as well as using all vehicle states to propagate "unseen" or occluded lane features. Additionally, lookahead distance is explicitly linked to expected estimator performance, allowing a lateral controller to be designed a-priori to cope with a given amount of lookahead (preview) distance, in a fashion similar to the methodology in [43].

1.6 Linear models for lateral vehicle control and driver behavior

Vehicle driver modeling has been an important field of study for over 20 years, both as a means to study vehicle dynamics under human control, and to produce robotic systems that replicate human drivers. In fact, some modern, high-fidelity vehicle simulation software packages still make use of driver models that are over 30 years old [14]. By contrast, the DARPA autonomous vehicle challenges have spurred the development of vehicles that map their environments [19, 32, 44, 45, 46, 47, 48, 49, 50], and navigate in them autonomously through the use of cutting edge ego and environment perception techniques and modern control structures.

1.6.1 Fixed-point optimal preview control

In 1980, C.C. MacAdam applied an optimal fixed-point preview controller to vehicle lateral guidance in [51] and showed that the model agreed well with actual human driver behavior. MacAdam's single-point optimal preview controller acts by predicting the vehicle's states over a preview horizon assuming a constant steer angle δ, as in eq. 1.16, and controlling the error between those predicted future states and a previewed reference input with a fixed, optimal gain that minimizes predicted error at the preview distance. This control law can be written quite simply, as in eq. 1.24, where T is the system sampling time, and K is derived from

the objective function, and appears approximately as the inverse of the vehicle dynamic model over the preview horizon length.

$$\delta_{opt} = \frac{y_{road,prev} - y_{predicted}}{KT} \tag{1.24}$$

This controller design provides a very simple way to control a vehicle in error-coordinates, because it can be implemented as an output feedback controller, where future states are predicted in an output equation's C and D-matrix with trivial simplification. The issue with this structure, especially for longer preview horizons, is that the controller will "cut corners," since it has no weight in the optimality function on the error at the current timestep.

1.6.2 Look-ahead feedforward/feedback controllers

A decade later, as a result of the PATH program at the beginning of the 1990s, researchers at U.C. Berkeley [52, 53] developed guidance laws for autonomous vehicle control. These control strategies also used feedforward control acting on previewed road curvature along with feedback at a lookahead distance to achieve vehicle path tracking, but instead of focusing on matching human driver behavior, the aim was to engineer solutions for autonomous vehicles that could be implemented on public highways. Researchers involved in this program, along with others, continued this vein of research through the 1990s [54, 55, 56]. A comprehensive review of vehicle driver modeling for autonomy can be found in [57, 58].

Whether for driver modeling or for vehicle autonomy, nearly all of such research makes use of either previewed or feedforward information in one form or another. In other words, autonomous driving and driver models assume knowledge of what lies *ahead* of the driver. Many of the popular path-following controllers make use of a Proportional-Integral-Derivative controller on projected (geometric, not predicted) error from the path at a given lookahead distance, and augmented with a feedforward term to compensate for road curvature. A classic example of this is shown in Figure 1.10.

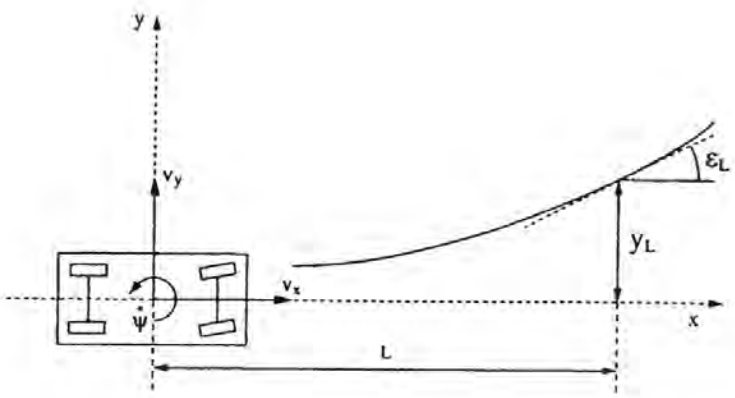

Figure 1.10. Lookahead PID controller setup (Taylor 1999)

In this framework, it is relatively easy to calculate robustness and stability. This is true even given output feedback as in [40]. But although these controllers (and similar approaches, including [37, 52, 53, 54, 55, 56]) can offer insight into the effects of changing preview on the controller, the same concrete relationships are not developed for the road measurement frameworks, except in [43]. These, too, in the case of the vision-based work, were based on geometric fits of road profiles,

rather than a true stochastic picture of the road geometry relative to the vehicle.

Knowing this, vehicle dynamicists in pursuit of vehicle path tracking and semi-autonomous driving often rely on a defense-grade INS system and a low-dimensional map of GPS breadcrumbs to provide the requisite path and preview information, as in [59]. Without explicitly considering implementation beyond the research arena, these algorithms offer excellent performance as long as vehicle states and path (map immediately ahead of the vehicle) are known with a high degree of accuracy.

1.6.3 Model-predictive receding horizon control

While path-tracking using the look-ahead+feedforward approach has shown positive results in the literature, it is often desirable to ensure that the vehicle follow a path in the "best" way possible, using some form of optimal control technique, as in [51]. A modern approach to this problem, employed in studies such as [60, 61, 44], is often referred to as Model-Predictive Control (MPC) or receding horizon control. The general setup of this control framework is shown in Figure 1.11.

The idea is that the grid in front of the vehicle represents a so-called "prediction horizon," where "future" reference states are available either from a map or from SLAM. Each line in the grid represents a future time step, from k to $k + N_p$, where N_p is the size of the control horizon. Model-predictive control is a rich and complex field, but the general concept is to solve the finite horizon dynamic pro-

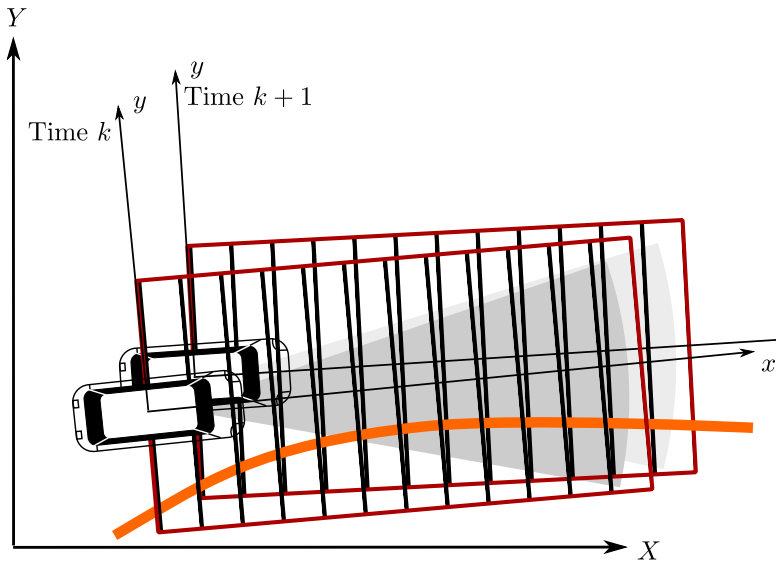

Figure 1.11. Setup of the model-predictive receding horizon control scheme

gramming problem to obtain an optimal input sequence $\vec{u}_{mpc}$, where the sequence represents the open-loop system inputs should optimally take the system through the prediction horizon. For linear systems, permissive of final state weighting and unequal prediction and control horizon lengths N_p and N_m, the problem formulation is laid out in [62], and summarized below. The general objective function for the unconstrained linear optimal control input is given by eq. 1.25.

$$\begin{aligned} J_{N_p,N_m}(x0) = & \inf_{u(\cdot)}[(x(p)-x_r(p))^T P_0(x(p)-x_r(p)) \\ & + \sum_{i=0}^{i=N_p-1} (x(i)-x_r(i))^T Q(x(i)-x_r(i)) \\ & + \sum_{i=0}^{i=N_m-1} u^T(i) R u(i)] \end{aligned} \tag{1.25}$$

where $N_p \geq N_m$. Note that under specialized assumptions, this objective function reduces to the objective function for a standard LQR when $N_m = \infty$. After solving

this objective function for an optimal input sequence $\vec{u}_{mpc}$ at a particular time x_k, only the first input in the sequence corresponding to the optimal control input for time $k+1$ is given to the system, after which the process is repeated recursively, solving for the optimal sequence at each time step and providing the system with only the first input. This methodology is shown qualitatively in Figure 1.12, which illustrates further why the method has earned the moniker of "receding horizon control."

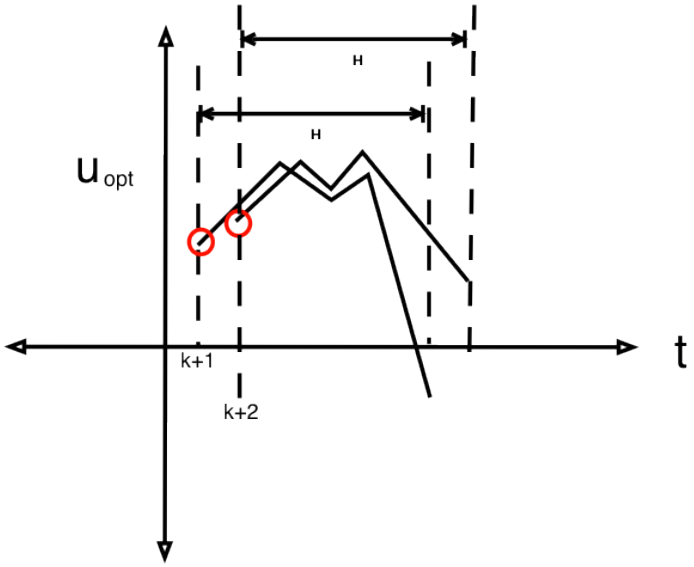

Figure 1.12. Recursive control law of receding-horizon MPC

Although this method appears to be open-loop, and does not necessarily provide the same input sequence at each time step, remember that the optimal control input depends on the most recent system states. Thus, there is implicit feedback in the approach. Stability and performance are difficult to prove for all but the simpler implementations of MPC [62], and it is not immediately clear to a control

system designer whether there is a "best" prediction horizon length over which to optimize. Additionally, for applications in vehicle lateral control, the reference input (previewed road geometry) through the horizon and the vehicle's current states must be known, and recomputed at each time step. Nevertheless, model-predictive control (including nonlinear and constrained variants) and SLAM have found widespread use in modern autonomous vehicles research.

With that said, it is desirable to have bounds on lookahead distance to achieve the best possible controller with the best possible path-following performance. Likewise, it is desirable to know how far ahead of a vehicle a preview sensor must see in order to provide such a controller with the best possible estimates of both vehicle states *and* environment states. In particular, the question of the minimum amount of previewed information necessary for preview control with optimal path tracking, and the minimum amount of previewed information necessary for optimal state and environment estimates is not readily solved by MPC or SLAM. The controller presented in the next section addresses this issue with model-predictive control, while the remainder of this dissertation addresses this issue in an estimation context.

1.7 Review of an optimal preview controller

In contrast to the recursive formulation of MPC, Sharp and Prokop used previewed road geometry in a state feedback form to drive the actions of their optimal pre-

view steering controller in [63] by using a relative motion model of the road moving towards the vehicle. The authors' work along these lines continued through the following decade in [64, 65]. Pick and Cole examined the mathematical relationship between model predictive control theory and Linear Quadratic preview control theory in [58]. This study shows how, under many circumstances, the two approaches can yield similar control laws. The authors also found that there are some instances where similar control laws are not possible. The controller outlined

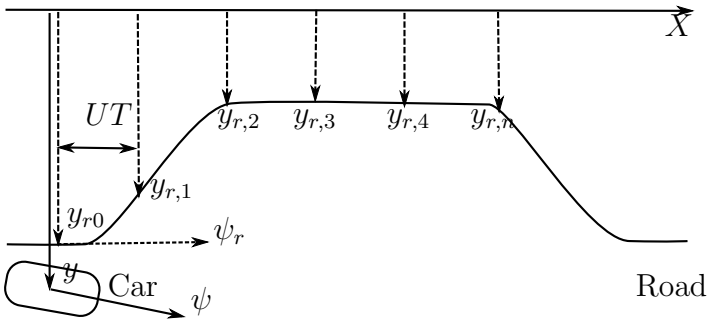

Figure 1.13. Setup of the road-vehicle coordinate systems

below is identical in structure and derivation to the one proposed in [63]. This background is intended to be brief, and the author would like to refer the reader to [63, 64, 66, 58] for a more detailed discussion of its derivation. This model was chosen for its relative ease of implementation as a state feedback controller, and its explicit use of multiple preview points. The constant state-feedback control law obtained using this methodology offers valuable insight into the necessary horizon length for optimal control, given particular weights on output errors. In the controls context, this tackles many of the issues that make MPC's recursive approach intractable for simple stability and performance analysis. As it is used

in the rest of this dissertation, the same approach defeats the limitations of using SLAM on road lane lines, because of the way the filter structure derived requires feature registration in one dimension only.

To describe the road motion relative to axes that slide along the road in error-coorinates with vehicle forward speed U, a shift register representing global road positions ahead of the vehicle is constructed for the vehicle-road system shown in Fig. 1.13. The road position at the preview distance is brought into this system, which lags the previewed measurement backwards through the state space at each time step until it corresponds with the global road position at the current time step k. In this way, one previewed reference geometry point at np time steps in front of the vehicle is coupled to all of the other reference geometry points tracked by the control structure. This is in contrast with MPC, which requires propagation of all vehicle states into the future inclusive of inputs– this motion model only assumes that the vehicle's forward speed U is constant, which is a fair assumption for most driving scenarios.

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 \cdots & 0 \\ 0 & 0 & 1 & 0 \cdots & 0 \\ 0 & 0 & 0 & 1 \cdots & 0 \\ 0 & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} \quad E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{1.26}$$

Upon augmenting the vehicle state vector by adding the road position shift register dynamics into the discrete-time vehicle dynamic equations, the following open-loop system is obtained, where $\vec{x}_k$ is the state vector and $\vec{y}_{r,k}$ is the road's lateral position at time k. Notice that these discrete "lane features" in the shift register are partially dual to the landmarks in a sliding SLAM map as in eq. 1.22. The major difference is that the longitudinal data association is assumed through the shift register matrix.

$$[\vec{z}_k] = \begin{bmatrix} \vec{x}_k \\ \vec{y}_{r,k} \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} \vec{x}_{k-1} \\ \vec{y}_{r,k-1} \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} \delta + \begin{bmatrix} 0 \\ E \end{bmatrix} y_{r,i} \tag{1.27}$$

Notice as well that the augmented state transition matrix is purely diagonal. Thus, there is no coupling here between the road dynamics and the vehicle dynamics, and the two systems essentially act independently of one another. In order to

couple the systems, a discrete-time linear quadratic regulator (LQR) is employed that acts on all of the augmented states. The states are coupled through the quadratic cost function R_1 shown in Eq. 1.28, again exactly as in [63].

$$J = \lim_{n \to +\infty} \sum_{k=0}^{n} \left[\vec{z}^{T}(k) R_1 \vec{z}(k) + \delta(k) R_2 \delta(k) \right] \tag{1.28}$$

with R_1 defined as follows:

$$R_1 = C^T Q C \text{ where } C = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{UT} & \frac{-1}{UT} & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{1.29}$$

and R_2 is chosen as unity. This configuration penalizes vehicle yaw and lateral position error in the LQR design through the diagonal matrix Q.

$$Q = \begin{bmatrix} q_y & 0 \\ 0 & q_\psi \end{bmatrix} \tag{1.30}$$

The key point here is that through the use of the cost function in Eq. 1.28, an optimal preview gain vector can be obtained for the augmented system using MATLAB's DLQR function, which solves the Discrete Algebraic Riccati Equation (DARE) automatically. The augmented system is fed into this function with the previewed road information as the only input. Note that, because there is no way for the controller to influence the road, there is a substantial subspace of this system which is uncontrollable. The coupling between the road geometry and the vehicle

states is through the optimal state feedback control gain K. Once the control loop is closed, its discrete-time dynamics are given by the difference equation, Eq. 1.31.

$$z_k = \left[\begin{bmatrix} A_d & 0 \\ 0 & D \end{bmatrix} - \begin{bmatrix} B_d \\ 0 \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix} \right] z\,(k-1) + \begin{bmatrix} 0 \\ E \end{bmatrix} y_{ri} \qquad (1.31)$$

$$\text{and } K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$$

Consistent with common sense, the controller is unable to influence the road position, as confirmed by the structure of the input matrix through which the optimal controller influences the state vector. Notice, however, that closing the loop with a driver *does* in fact allow the road's absolute geometry to influence vehicle states through the $B_d \begin{bmatrix} K_1 & K_2 \end{bmatrix}$ term. This coupling of environment and physical system in an augmented state-feedback form is the key to using high-fidelity global maps of road geometry and measurements of lane edges at each preview distance to improve vehicle state estimates. This is discussed in the next chapter.

1.8 Summary of research goals

The preceding sections identified key ideas and recent contributions in vehicle autonomy, vehicle state estimation, road scene understanding through computer vision, and the marriage of mapping and localization through SLAM. How are

these four key areas coupled in the automobile driving task? During normal operation, an autonomous vehicle must be aware of its own position relative to a map, since the goal state (usually the lane center) is directly related to the spatial information provided by a map inclusive of lane lines and/or obstacles. In addition to information about the vehicle's current position, however, some information about the road *ahead* of the vehicle is also needed for effective control, as emphasized in Section 1.6. This means that an up-to-date, high-resolution knowledge of the road geometry some finite distance in front of the vehicle, or rather a subset of the total map, is necessary for control as well as estimation. However, based on the preceding review, the following conclusions are evident:

1. SLAM is highly effective, but ungainly in feature-sparse environments, especially using a camera as a sensor.

2. Current coupled road-vehicle state estimators that measure only lanes make too many approximations concerning the relationship between the vehicle and road, precluding an optimal control design with predictable error bounds under noisy output feedback.

3. MPC's recursive nature does not offer the easy relationship between control effort and lookahead distance provided by the optimal preview controller.

4. A need exists for a concrete relationship between perception of previewed states and control using those previewed states, to address the minimum

amount of information needed by a perception system to allow for implementation of optimal preview control for autonomous path-following.

The remainder of this dissertation is organized as follows. Chapters 2 and 3 take the temporal preview horizon concept a step further by fusing map and camera-sensed previewed information to estimate both the goal path and the vehicle's lateral states simultaneously. Through the use of a camera, map, and a yaw rate gyro, an efficient system capable of driving an automobile down a road *within the capabilities of its perception hardware* is the ultimate goal. The two camera-based methods developed in these pages offer perception system designers a way to determine how far in front of a vehicle a robot must see in order to achieve the best possible accuracy. Chapter 4 discusses the practical implementation of the filter developed in Chapter 3, and Chapter 5 shows its merit on public roads and a closed-course test track, for day and nighttime driving. Then, Chapter 6 discusses some of the limitations of using these temporally predictive frameworks for high lateral accelerations and tight, constant-radius turns, while Chapter 7 shows that preview estimation is viable in a control context, discusses design issues with coupled preview control/estimation schemes. Chapter 8 discusses continuing and future studies in this area.

Chapter 2

Camera-Based Preview Estimation in Error Coordinates

2.1 Introduction

The previous chapter introduced the idea of formulating a receding horizon optimal preview control problem as a state feedback system under the assumption of a particular model for relative road motion. Using a sensor capable of measuring the path ahead of the vehicle in local coordinates, like a camera, offers a way to couple previewed map information with vehicle dynamics and inertial measurements in a simple estimation framework that resembles a one-dimensional linear reduction of SLAM.

The idea of using road geometry measured by a camera to aid in vehicle control is not new, as mentioned in Chapter 1. The approaches most closely resembling the

work that follows are those taken by Dickmanns in [37] and by Hsu in [42]. In [37], Dickmanns constructs an Extended Kalman Filter that estimates vehicle states and a parameterized representation of the road geometry ahead of the vehicle. This approach has been refined since its inception in the late 1980s, but the idea of "4-D" (spatiotemporal) perception of the road scene remains the core goal. With this estimation framework, Dickmanns attempts a tightly coupled controller-estimator, and the newest versions even include some sense of 3D geometry, but his approach differs from what follows in a number of important ways:

1. The vehicle model was generally simplified to neglect yaw acceleration.

2. Few road points were measured at each time step to obtain a measure of road curvature.

3. Road geometry was parameterized, and parameters were estimated using a clothoid fit.

4. The controller used for guidance was a simple feedback controller with a feed-forward term accounting for road curvature, but did not address the problem from the standpoint of optimal control, or with intent to form bounds on necessary preview.

5. The filter structure did not employ mapped road geometry, and had no way to deal with complex and potentially occluded road markers.

The preview estimation framework developed below should perform better under a wider range of road conditions, but the method developed by Dickmanns actually has a lot of merit in the sense that his filter only estimates the elements of the road geometry *used by his controller*, and directly couples the estimation processes for road shape and vehicle dynamics. In other words, running a computationally expensive algorithm like a full implementation of SLAM or one of the Monte Carlo map-based localization techniques just to run a path-keeping controller that *uses* only a small subset of the tracked features was not necessary.

Although it is scarce in the literature, the idea of using a temporal preview horizon to aid in vehicle state and environment estimation has also been attempted, most notably by Hsu in [42] in a spatiotemporal framework that included lateral velocity estimates, unlike many of the vision-based methods discussed previously. In this paper, Hsu uses a temporal preview horizon concept similar to those derived in this dissertation to aid in vehicle state and road curvature estimation, but uses a very sparse (9 point) preview horizon, and runs a separate filter framework for spatial estimation of road curvature and vehicle states (within one image) from the one that enforced temporal consistency of those parameters across successive video frames via the preview horizon concept. Thus, there are separate, decoupled estimation frameworks for the vehicle dynamic states and the road curvature values. In addition, the measurement equations are nonlinear, and only derived for a parallel camera. In addition to limiting the applicability of the method, this adds

a significant level of complexity over the methods that follow, and unnecessarily decouples an estimation process that can be completed in one step. The methodology in [42] was tested offline only, and showed estimate errors that increased in certain cases when temporal filtering was used in conjunction with spatial filtering, rather than improving errors. Additionally, there were no provisions in his framework for using map information to temper erroneous lane measurements or to provide road path information beyond the farthest visible lane feature.

In the derivation, discussion, and results that follow, several peculiarities of the road vehicle driving task are exploited to make the estimation of road geometry and vehicle states using a preview-based Kalman filter. Some of these include:

1. Generally, there is a very small misalignment between vehicle and road (small angle approximation applies, except under emergency conditions)

2. Lane lines are stationary. The motion of the road relative to the vehicle is simply the negative of the vehicle's motion relative to the road, so the motion model of road "features" is easy to construct.

3. The path ahead of the vehicle is needed for effective guidance using an optimal preview controller, so the estimator should provide all of the information necessary for autonomous cruising.

It is interesting to note that one of the greatest challenges in vehicle state estimation is the incredibly small signal to noise ratio that exists on states like

lateral velocity. Rather than hindering the preview-estimation framework, this small relative value of lateral velocity versus its longitudinal counterpart is *the entire reason* that representing the road in a shift register is acceptable, because the framework depends heavily on the small-angle approximation.

2.2 Setup of the measurement framework in error coordinates

As in the previous chapter, the following analysis is more tangible when the vehicle is assumed to move in a more or less straight line, so that it and the road ahead may be described by the augmented error-coordinate bicycle model equations of Equation 1.27. This preserves linearity in the global representation of the vehicle dynamic equations, makes map registration easy using odometry, and allows for careful estimator development. The results are applied to a purely local representation of the bicycle model in the next section, for a more flexible implementation.

In the error coordinate form of the vehicle-road system, as mentioned in Chapter 1, there is no inherent coupling between road dynamics (the shift register) and vehicle dynamics. The road dynamics, as described, can be interpreted rather simply: because the path points are globally stationary, their position relative to the vehicle in the pseudo-global x-direction changes as a result the vehicle's forward velocity in the (approximate) x direction by a distance of UT meters with each

loop of the discrete system.

However, even with the error-coordinate representation of the bicycle model, where the vehicle lateral velocity derivative is expressed as if equal to the newtonian acceleration of the vehicle (it is, in general, not), a camera mounted to the vehicle will see the road path points slightly differently than an observer looking down the x-axis. In particular, mismatch between the global y-coordinate of the road point and the "vehicle referenced" measurement of the same will occur as a result of the car's current y-position and yaw angle, as detailed in Figure 2.1.

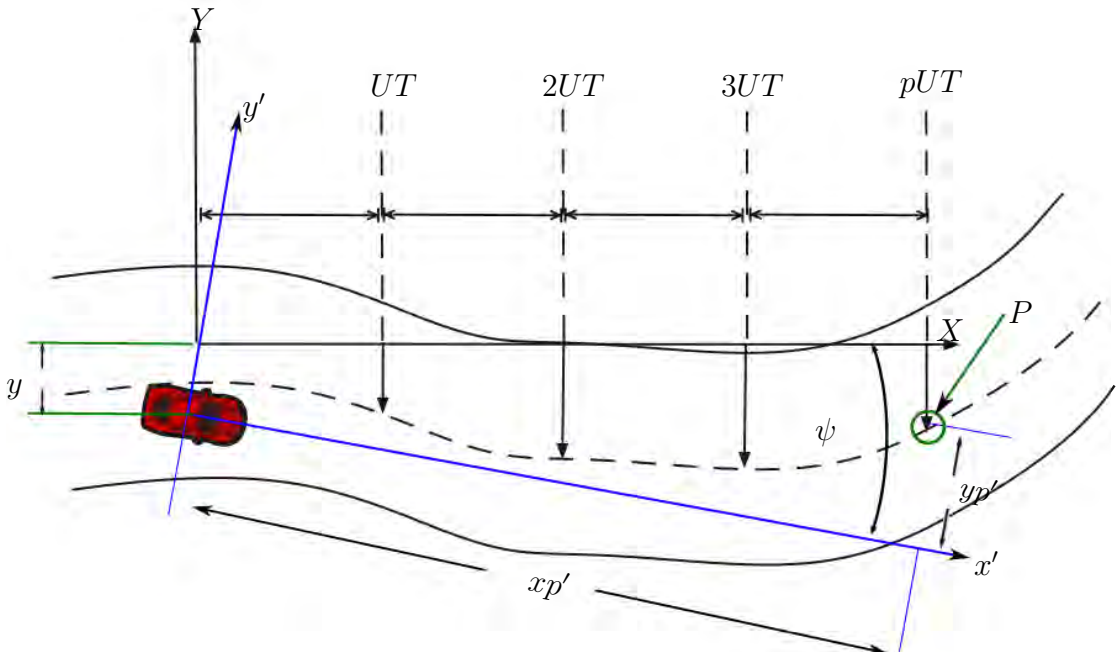

Figure 2.1. Road features on a spatiotemporal horizon for a vehicle moving at a constant speed

As before, the road ahead of the vehicle can be discretized into np points, which range from zero to $UT\,(np-1)$ meters beyond the vehicle CG. As in the previous

chapter, we will set up a motion model in the x-direction for these points that shifts each road point UT meters closer to the vehicle with each time step. As before, the motion model for the road is simply

$$\vec{y_k} = D\vec{y_{k-1}} + E\vec{y_{ri}} \tag{2.1}$$

Where the state transition and input matrices are given by

$$D = \begin{bmatrix} 0 & 1 & 0 & 0\cdots & 0 \\ 0 & 0 & 1 & 0\cdots & 0 \\ 0 & 0 & 0 & 1\cdots & 0 \\ 0 & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{2.2}$$

When the error-coordinate bicycle model is augmented with the road geometry states $\vec{y_k}$, the augmented system is identical to that described in Eq. 1.27. Once again, there is no dynamic coupling between the road and the vehicle without a driver model. However, it is entirely possible to obtain a measurement model that results in a fully observable system when a Kalman filter is designed around the above augmented dynamic equation. The key to this is the realization that, from the vehicle's perspective, observed y-coordinates of road points ahead are *linear combinations of the road states and the vehicle states* as long as the small angle

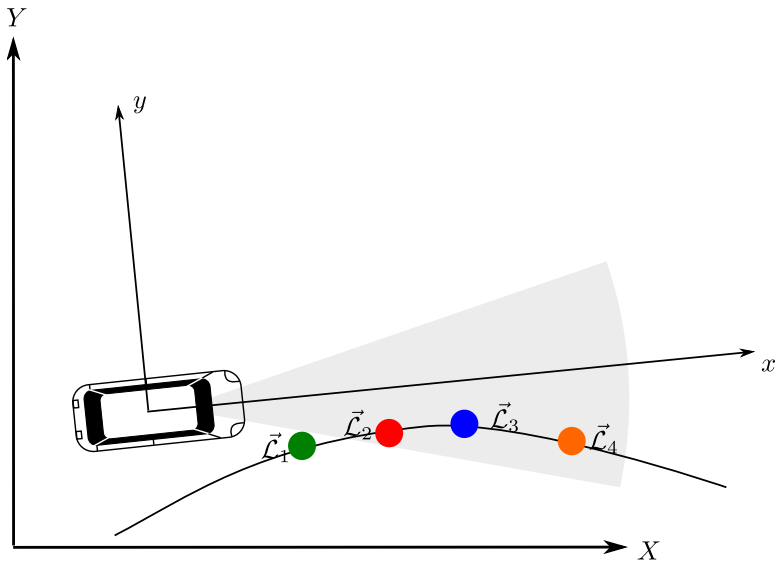

Figure 2.2. Qualitative parallels between preview estimation and SLAM

assumption is not violated. In particular, any point $\vec{y_p}$ ahead of the vehicle can be expressed from a vehicle-fixed frame $(x\prime, y\prime)$ as:

$$\begin{aligned} x'_p = &\quad (x_p - x)\cos\psi + (y_p - y)\sin\psi \\ y'_p = &-(x_p - x)\sin\psi + (y_p - y)\cos\psi \end{aligned} \tag{2.3}$$

Where x and y represent the location of the vehicle. Note that this equation looks a lot like the equation for a landmark measurement in SLAM, and the augmented state vector looks much like the SLAM map of eq. 1.22. Consider a re-mapping of Figure 1.6 to include evenly spaced "landmarks" along a lane marker, as shown in Figure 2.2.

Figure 2.2 shows the landmarks along the edge as evenly spaced dots, but recall that lanes themselves cannot be relied upon to offer good features for two dimensional tracking. Fortunately this is not an issue for the preview estimation

framework. Because of the setup of the road shift register, this estimator is not required to track the x'_p coordinate of the lane edges in the shift register at all, since they are lagged by the road motion model backwards through the state space. This is obviously an approximation from time step to time step, but if the vehicle follows the path closely, as it will under competent closed-loop control, the absolute value of the road points y_p in the shift register is small. Therefore, when the same small ψ assumption central to the error-coordinate derivation of the bicycle model is applied, $x'_p \approx pUT$ and thus the linear measurement equation for the road point's local lateral displacement y'_p becomes:

$$y'_p \approx -pUT\psi + y_p - y \tag{2.4}$$

and the $x'_p \approx pUT$ spatial coordinate, which SLAM might track, is replaced by a temporal coordinate representing how far into the future each feature is. This measurement description can be assembled into an observation equation for use in the design of a linear Kalman filter. In this filter, all road points in the shift register in front of the vehicle starting with y_1 are available for measurement, yields:

$$\vec{z}_k = \underbrace{\begin{bmatrix} -1 & 0 & -UT & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & -2UT & 0 & 0 & 0 & 1 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & 0 & 0 & 0 & 0 & 0 & \ddots & \vdots \\ -1 & 0 & -(np-1)UT & 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \\ y_{r,0} \\ y_{r,1} \\ \vdots \\ y_{r,np-1} \end{bmatrix}}_{\vec{x_k}} \tag{2.5}$$

The idea of the present study is that a Kalman filter, designed with the above measurement equation, and augmented with a yaw rate measurement, can offer improvement in state estimation with increasing preview distances in a way that mimics SLAM, but in a linear framework under the stated assumptions. Mapped road geometry is fed to the system through the road input matrix E, where map registration is achieved in coarse fashion by the GPS sensor, or a fused estimate of odometry achieved through the fusion of GPS position and velocity. Although registration through a "distance traveled" metric is subject to a large amount of error, the road offset at the preview distance is quickly corrected by local measurements at each time step.

It is also worth noting that if no map is available, the estimator, with access

to road measurements out to the preview horizon, will still be able to provide estimates of all states, including lateral velocity. If a map is available, but visibility is low, the mapped input will be propagated back through the motion model until it is corrected by camera measurements– either way, a Kalman filter framework will maintain estimates of the covariance values of vehicle states, along with statistical confidence in the position of each point in the shift register. While the shift register model challenges causality (how does one "know" what will happen in the future?), it will be proven as a good approximation for most driving.

2.3 Local measurement of lane offset using a monocular camera

Researchers have long relied on forward-facing cameras to aid in road scene understanding. In fact, the first autonomous traversal of a public road in the late 1980s was achieved primarily through the use of a forward-facing monochrome camera [37]. Algorithms for lane marker or road edge extraction are various, and range in computational complexity, but for the present study, we will begin with an edge-based extraction of lane lines followed by an inverse perspective transform, where the road in front of the vehicle is treated as a planar, level surface. For the purposes of tracking lane lines in local vehicle coordinates, this might be sufficient, or at least a good place to start.

2.3.1 Lane extraction method

To extract lane geometry, the input video stream is cropped so that only the scan-lines that would map to a planar road surface within the preview horizon are considered. This reduces the likelihood that far-field objects or objects above the horizon line will be encoded as lanes. The image is converted to grayscale, after which OpenCV's Canny Edge filter is applied (Figure 2.3). The resulting image is binary, and consists of thin lines. With proper tuning of the Canny Edge parameters, the lane lines are handily recognized.

After the image is satisfactorily binarized, road edges on the right and left are found by computing the scan-line in the image for each preview distance according to eq. 2.6, and searching for white pixels within a window where lane lines are expected to occur for both straight and curved roads. Lanes are tracked upwards through the image, and their locations are recorded in a vector of pixel-coordinate locations for the transformation to vehicle-fixed coordinates as shown in Figure 2.5. More details are provided in Chapter 4, but a key feature of the preview estimator framework is that it is agnostic of the lane detection algorithm used. The lanes are measured and used for fusion of the map and camera information to estimate vehicle states, but any number of extraction techniques could be used, as long as the modeled noise in the Kalman Filter framework is acceptable.

As shown in Fig. 2.4, each scan line (y_i coordinate) in an image will map directly to a vehicle-fixed x-distance when the road considered is flat. Consistent

Figure 2.3. Canny edge filter applied to grayscale road scene image

with [37], the equations for the inverse perspective transformation are given by Eq. 2.6 below:

$$\begin{aligned} x_p &= h_{cam}\frac{f - y_i \tan\theta}{y_i + f\tan\theta} \\ y_p &= x_i\frac{x_p}{f\cos\theta} \\ z_p &= -h_{cam} \end{aligned} \tag{2.6}$$

where h_{cam} is the height of the camera above the (assumed flat) road surface, f

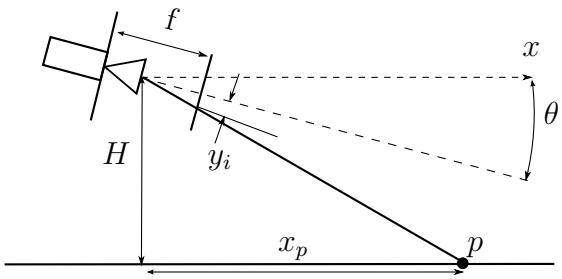

Figure 2.4. Inverse Perspective Transformation for flat roads

is the normalized focal length of the camera in pixels, y_i, x_i represent image plane coordinates, and x_p, y_p, z_p represent vehicle-referenced physical coordinates with the axes oriented as in the bicycle model equations of eq. 3.1.

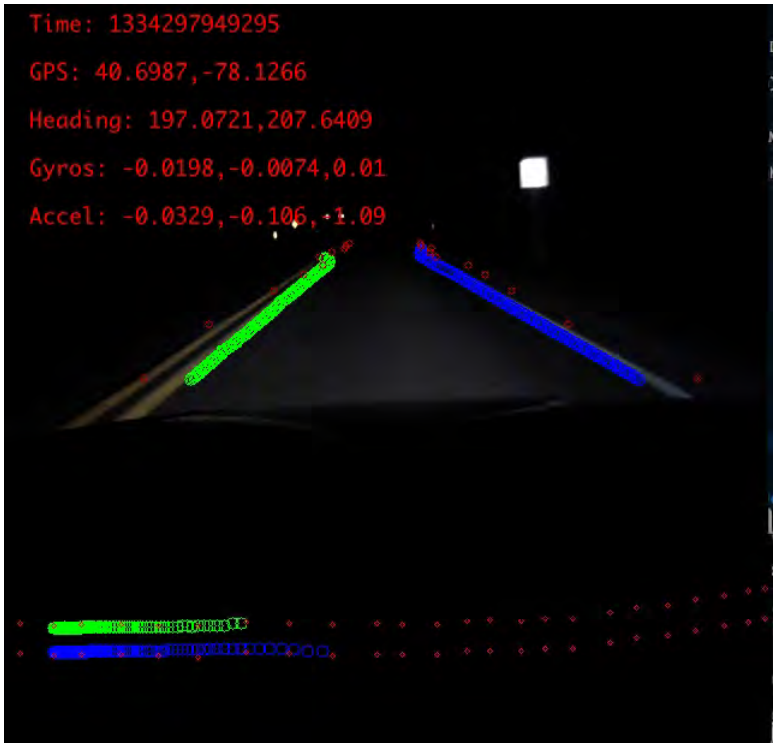

Figure 2.5. edge-based lane extraction during nighttime driving

2.4 Preview measurement noise model

To give appropriate weights to the lane measurements in the Kalman filter structure at each previewed time, especially since the quantization of the lane detection error at a pixel's width is likely to make the noise appear non-Gaussian, an assumed Gaussian noise model reflecting the physical uncertainty of a monocular camera extracting lane lines ahead of the vehicle is now considered. This is a necessary assumption for the use of a Kalman filter, which assumes Gaussian noise in its derivation. Like the canny-based lane extraction method used in this work, most lane-detection methods employ an inverse perspective transformation illustrated in Figure 2.4, with general transformation equations from the camera to vehicle-fixed reference frames given by equation 2.6. These assume a flat road in front of the vehicle, so extra uncertainty should be added to the filter framework to account for dips, hills, and unmodeled vehicle roll and pitch motion.

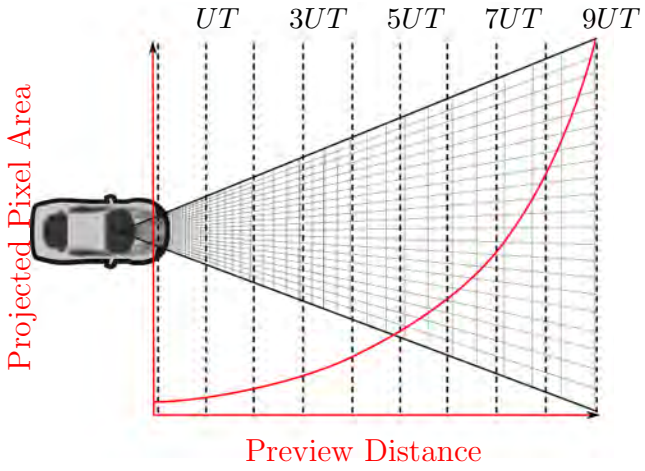

Figure 2.6. Pixel area on road surface

Eq. 2.6 offers yet another source of uncertainty in the lane measurements, because it distorts pixel-level noise based upon the mapping of each pixel onto the road surface, and pixel-width errors in lane edge detection represent larger and larger lateral and longitudinal edge measurement errors as the distance in front of the vehicle increases. To examine a baseline for lane measurement error, a test vehicle was parked in the middle of a typical two-lane road, and a Canny edge-filter-based lane detector was run on video data of the vehicle sitting still in the road. The 1-σ error was not very illustrative in this case, because the noise observed was hardly Gaussian given the quantization effects resulting from non-subpixel edge detection. However, the theoretical (from equation 2.6) and experimentally measured maximum variation in lateral lane position estimates increased linearly with distance, as shown in Figure 2.7.

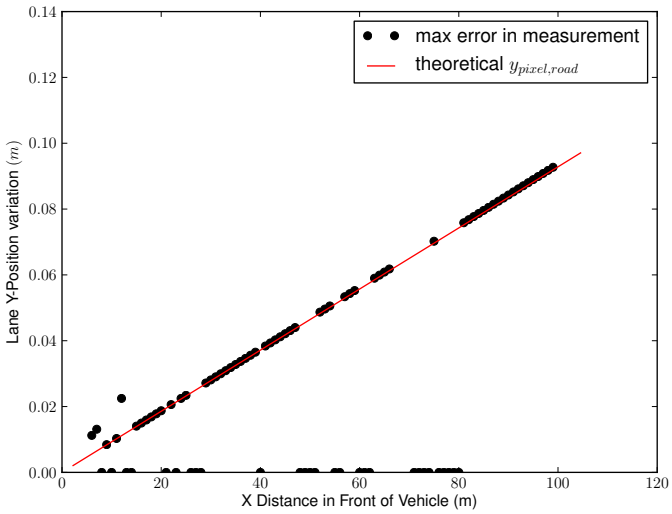

Figure 2.7. Maximum variation in lateral lane position estimate for a static lane detection test

While Figure 2.7 might tempt some to use a linear approximation of lane measurement deviation (a squared approximation of variance) as a function of preview distance, edges detected could also be erroneous in the vehicle's x-direction due to the approximations in the shift register. Pixel-level variation in x will also distort measurements of the y-coordinate of a lane edge, especially for lane features that change lateral position abruptly due to a break, painting error, or other anomaly. Thus, the measurement variance due to pixel-level noise for each preview point is conservatively approximated as each pixel's projected road-coordinate length multiplied by its projected road-coordinate width, which results in a larger variance than simply squaring the pixel width variation. Figure 2.6 illustrates the that while the measurements are taken at particular scan lines corresponding to the evenly spaced preview distances in the shift register D, and the width of each pixel only grows approximately linearly as preview distance increases, the projection of each pixel on the road creates pixel coverage areas that grow nonlinearly. This agrees with intuition where increasingly distant features have decreasing utility for measurement of lateral position.

Unless otherwise noted in the text, the road surface area captured by each pixel was computed as a function of preview distance for a VGA monochrome camera with a downward tilt angle of 5 degrees and a focal length f of 1068 pixels mounted at 1.0m above the road surface. A third-order polynomial fit of total pixel area on the road surface versus preview distance was used as a basis for the diagonal

Table 2.1. Representative Filter Parameters.

Parameter	Value	Units
σ^2_{gyro}	0.00066	$\frac{rad^2}{s^2}$
$\sigma^2_{X_{vehicle}}$	1.5	m^2
$\sigma^2_{cam,x}$	$1.5 \times 10^{-6}x^3$ $+6.5 \times 10^{-6}x^2$ $-5.7 \times 10^{-5}x$	m^2
$\sigma^2_{cam,extra}$	0.1	m^2

measurement noise matrix R in the preview filter. The polynomial representing the assumed lane measurement noise as a function of preview distance x is shown in Table 2.1. Because a yaw rate measurement is also used, Table 2.1 also provides a representative gaussian white noise parameter for an L3G4200D MEMS gyroscope.

Finally, because map registration must be achieved through a measurement of distance traveled in the X direction, representative noise from a simulated longitudinal positioning sensor or algorithm was used to compute the longitudinal noise in $X_{vehicle}$. This variance was not used in a measurement equation, but was used as the process noise affecting $y_{r,i}$ in the filter design since $y_{r,i}$ is computed directly from the vehicle's X-location. The magnitude of this variance, $1.5m^2$, was chosen roughly according to the algorithm performance metrics employed in [16].

2.5 Simulation of state estimates during lane change

The filter described above was used to estimate vehicle and road states using a Kalman filter acting on the vehicle-road system with the measurement matrix H for a lane change maneuver at $20\frac{m}{s}$. The simulation of the lane change maneuver was performed using a linear bicycle model with parameters given in Table 2.2, which represents the vehicle simulated in Sharp's original optimal preview control paper [63] but measurements of vehicle lateral position and yaw rate were corrupted with Gaussian noise as in Table 2.1. For comparison purposes, a standard model-based Kalman filter was also run in parallel using the same measurements (with assumed noise as in the preceding chapter), but without the benefit of simulated camera measurements.The camera measurements fed to the preview estimator were corrupted with Gaussian noise that increased with distance of the road point in front of the current vehicle position as indicated in Table 2.1. Both filters were subjected to steer angle input noise $\sigma_\delta = 0.01rad.$ Simulation results for a lane change maneuver are shown in Figure 2.8.

As Figure 2.8 shows, all four non-previewed vehicle states benefit visibly using the preview estimation framework when compared to an idealized model-based Kalman filter fusing GPS and yaw rate measurements *except* yaw rate, since it is measured directly. Also note that the performance of the GPS sensor in the standard model-based Kalman filter is an idealization, since low-cost GPS systems do not typically have unbiased Gaussian error characteristics (as shown experimen-

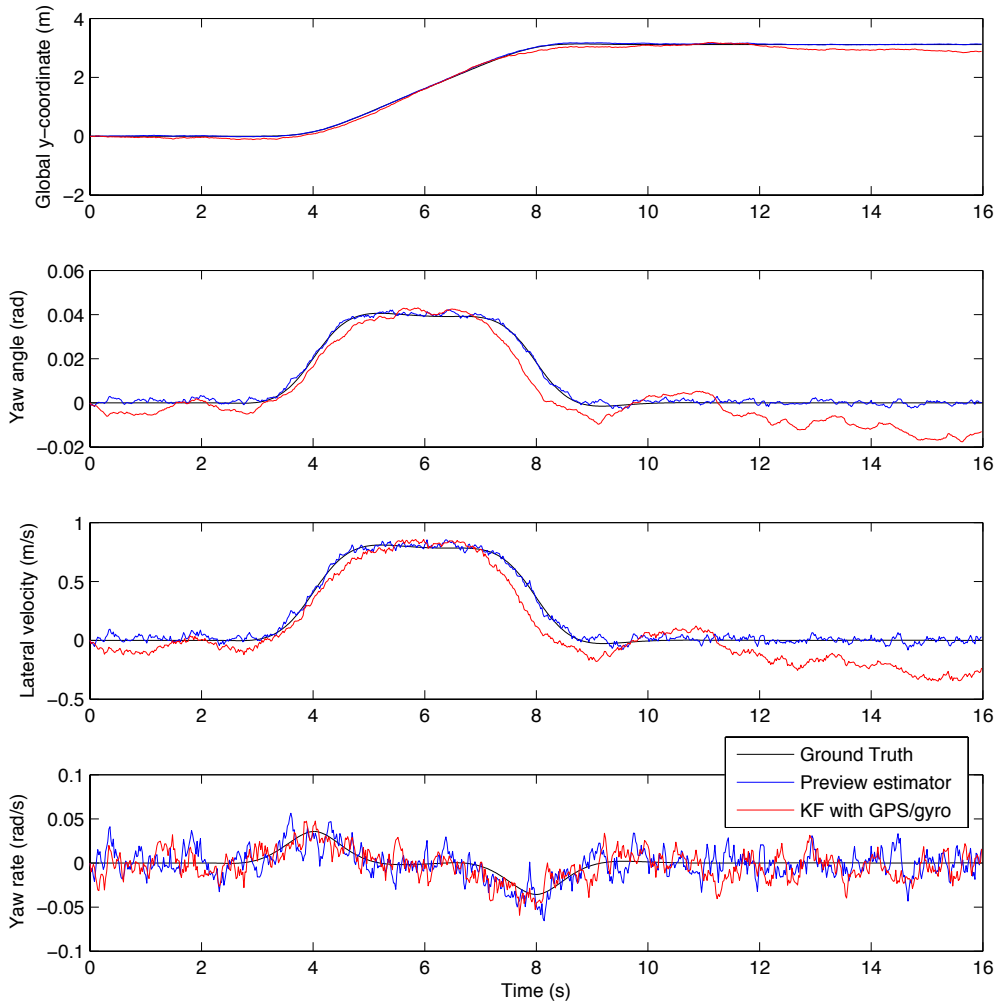

Figure 2.8. True and estimated vehicle states during lane change at $20\frac{m}{s}$ for standard model-based Kalman filter and error-coordinate preview filter

tally in Chapter 5). Because it is somewhat unfair to compare an estimator using a coarse GPS sensor alone with one using a more precise sensor like a camera, consider Figure 2.9, which shows the preview filter's error characteristics for the same simulation as in Figure 2.8, but using preview filters with varying preview

horizon lengths.

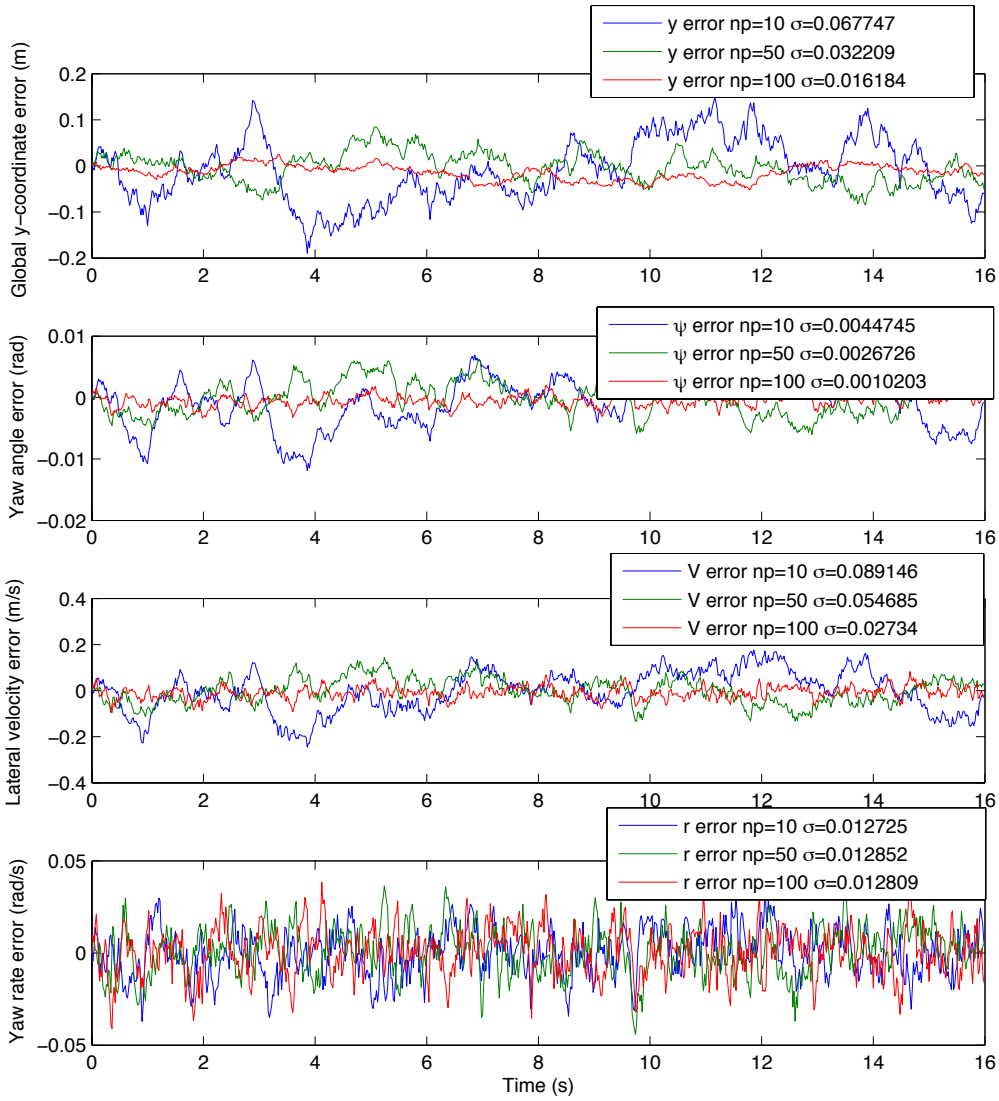

Figure 2.9. Estimate errors during simulated lane change at $20\frac{m}{s}$ for error-coordinate preview filter using various preview distances

This is a primary benefit to the preview estimation framework described– one can determine analytically, based on available road feature visibility, what to expect for non-previewed state estimation accuracy, if the camera noise model is carefully

calibrated. This is true of any Kalman filter, if the number of measurements is known, but the key insight here is that visibility in distance or time maps directly to the number of measurements available to the estimator.

2.6 Calculation of theoretical estimator performance

It may be apparent to the reader at this point that the preview estimator, having full row rank in its observability matrix, is capable of estimating road geometry as well as vehicle states. This has strong parallels to Simultaneous Localization and Mapping (SLAM) algorithms, which pick out road features in a robot's environment, and localize the robot relative to those features. The landmarks in a SLAM framework are stored as states in an augmented state vector, much like the road geometry is in the preview estimator.

While there are definitely parallels between this estimator and SLAM, there are also distinct differences. For one, the shift register representing the road vehicle dynamics in the preview estimator takes care of the propagation of the road geometry towards the vehicle, so the intense data association problems that plague SLAM are not an issue. Because road lane markers are relatively continuous entities, and *because a road vehicle's longitudinal velocity dominates its lateral velocity and changes slowly*, using a shift register to obtain the x-position of any one road

geometry element relative to the vehicle is acceptable. Additionally, the current (error-coordinate) realization of the preview estimator is set up to use map information as an input, so the "mapping" step of SLAM is only partially accomplished. The filter could be thought of as "updating" mapped geometry using measurements from the camera, but a map can not be generated from scratch. Finally, most SLAM algorithms keep track of a nearly ever-increasing number of map features, but the preview estimator only keeps track of features within the time horizon relevant to guidance by optimal preview control.

To isolate potential benefits of increased preview distance/time on the preview estimator's performance under ideal conditions, a series of steady-state Kalman Filters was designed using increasing numbers of preview measurements for various fixed speeds. The resulting filters were used to examine steady-state covariance in each state estimate. These covariance values are illustrative in that they represent the achievable accuracy for a preview filter designed for a given look-ahead condition. These results thus estimate an analytical relationship between lookahead distance and achievable lateral state estimation accuracy.

The measurement equation for each simulated filter included camera measurements of a road lane line in front of the vehicle, starting at around $2m$ from the vehicle's CG . The lane line detection starts $2m$ ahead of the actual CG to account for vehicle's body obscuring the ground directly under the vehicle. The camera noise parameters derived above for a typical monocular camera arrangement were

Table 2.2. Representative Vehicle Parameters.

Parameter	Value	Units
m	1592	kg
I_z	2488	kgm^2
a	1.18	m
b	1.77	m
C_f	-75000	$\frac{N}{rad}$
C_r	-55000	$\frac{N}{rad}$

held constant for each filter. The filters were designed for a vehicle with parameters given in Table 2.2 at constant forward speeds of $U = 20$ to $35m/s$, which represent a range of speeds expected during highway driving. The resulting theoretical steady-state system covariances on each state as a function of preview distance for each test speed are given in Figure 2.10.

It is clear from Figure 2.10 that when using previewed lane line position and yaw rate alone for vehicle state estimation, very good accuracy can be expected, even on difficult to estimate states like lateral velocity. The plot also indicates that an increasing number of temporally previewed features does, in general, benefit the task of vehicle state estimation. However, the figure also indicates that measurements of lane position at distances greater than $40m$ are nearly useless for improving vehicle states across highway driving speeds, at least for this specific camera and gyroscope arrangement. Thus, a preview filter of finite dimension (or a road scene with finite visibility) is sufficient to obtain minimum achievable non-previewed vehicle state estimate covariances.

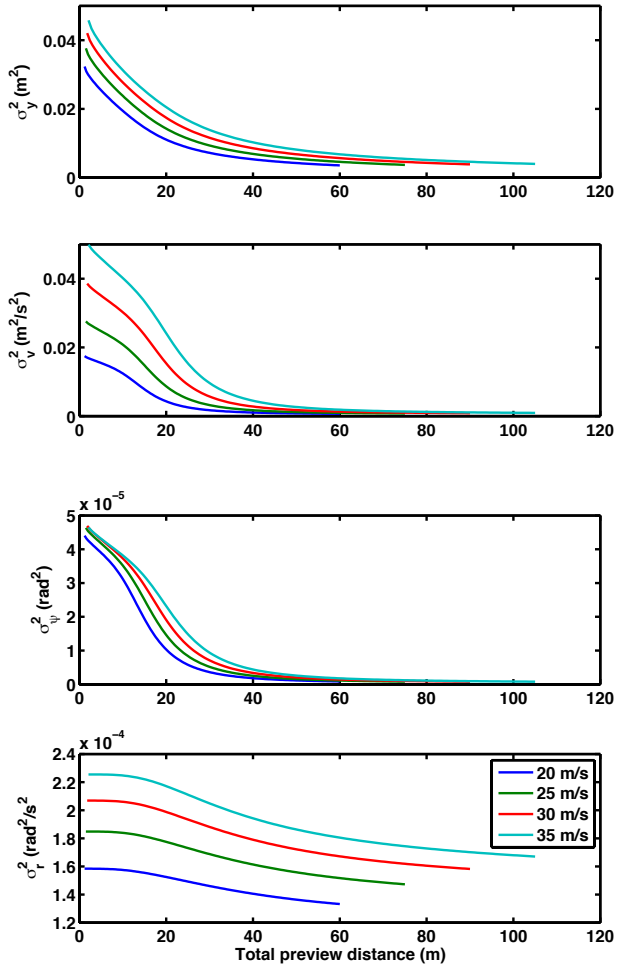

Figure 2.10. Expected variance of vehicle state estimate errors as a function of preview distance

These results are valuable information for selection and orientation of forward-facing monocular cameras and the design of preview-based vehicle guidance algorithms, if camera noise is experimentally determined for a given camera's intrinsic and extrinsic parameters. The reader should tread carefully when interpreting Figure 2.10 literally, however, since in practical implementations of the filter, not all preview states will be available for measurement at all timesteps. Visibility is

a major factor here, and occlusions stemming from other vehicles in the roadway, darkness, fog, snow, or other unforseen occlusive items will change the filter covariance matrix P as time progresses. This is a primary advantage of the preview-filter framework, since visibility is easily correlated with confidence in state estimates. Visibility itself, especially in the context of this preview filter framework, could be developed as a key source of filter and control system diagnostic measures, given the ease with which the accuracy effects of missing features on the preview horizon can be calculated.

Chapter 3

Camera-Based Preview Estimation in Body-Fixed Coordinates

One practical issue with the estimator in the form derived above is that it is based on the error-coordinate representation of the bicycle model. The assumption that the local vehicle reference frame is inertial becomes erroneous for most driving, even when the vehicle is driving on a relatively straight road that turns only gradually. With this in mind, it becomes important to formulate a version of the preview estimator that does not rely directly on such assumptions. Additionally, the error-coordinate formulation above does not address concerns with 3-D road geometry, which often includes a lateral superlevation of turns. This superelevation is readily available from a map with only a slight increase in map complexity, and provides a disturbance input to the vehicle that is often significant for lateral velocity estimation. This will be demonstrated shortly.

3.1 Development of the vehicle-road motion model in body-fixed coordinates

The vehicle-referenced linear bicycle model equations in ISO coordinates, also called body-fixed coordinates, are given below (as derived in Chapter 1, eq. 1.8) in eq. 3.1.

$$\begin{bmatrix} \dot{V} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{C_f+C_r}{mU} & \frac{aC_f-bC_r}{mU} - U \\ \frac{aC_f-bC_r}{I_zU} & \frac{a^2C_f+b^2C_r}{I_z} \end{bmatrix} \begin{bmatrix} V \\ r \end{bmatrix} + \begin{bmatrix} \frac{-C_f}{m} & \frac{-g}{100} \\ \frac{-aC_f}{I_zU} & 0 \end{bmatrix} \begin{bmatrix} \delta_f \\ e \end{bmatrix} \tag{3.1}$$

These equations for local lateral velocity and yaw rate are independent of the road geometry in the xy plane, and the only influence the road has on the states is through the superelevation e. The equation also assumes that the direction of V changes as the vehicle rotates, which is perfectly suited for a motion model with measurements provided by a sensor attached to the vehicle. The road points in the shift register are therefore assumed to move relative to the vehicle based on its local velocity and yaw rate from timestep $k-1$ to timestep k. This idea is illustrated in Figure 3.1. The movement of the vehicle from time k to time $k+1$ is given simply by $\Delta x, \Delta y, \Delta\psi$, with those quantities approximately given by eq.

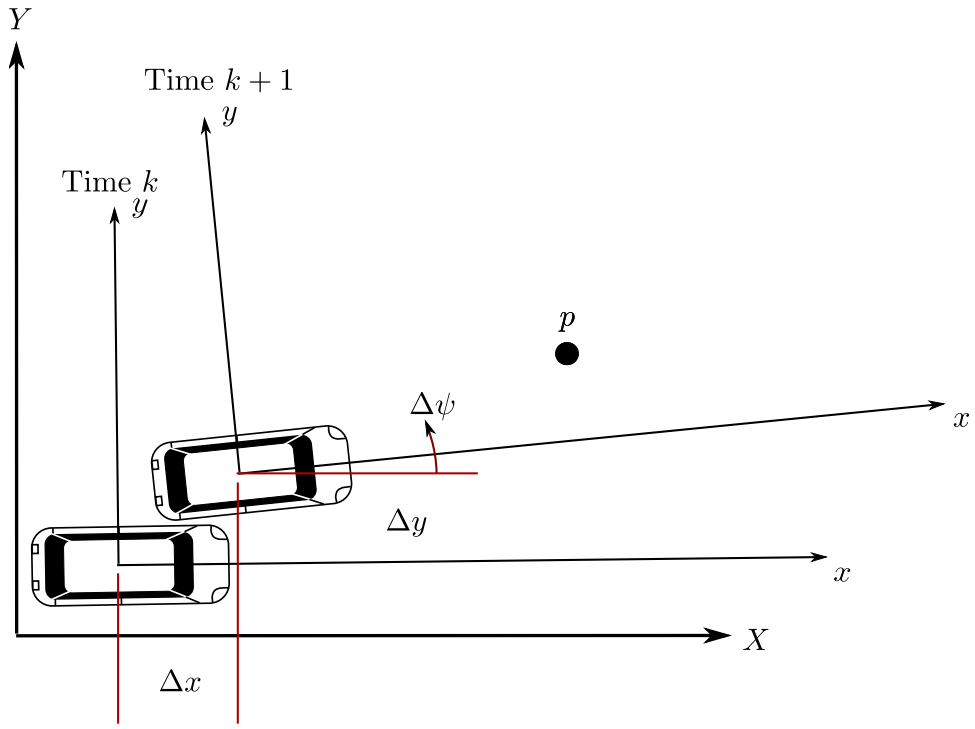

Figure 3.1. Relative road movement from timestep k to timestep $k+1$

3.2 for a small timestep:

$$\begin{aligned} \Delta x &= UT \\ \Delta y &= V_{k-1}T \\ \Delta\psi &= r_{k-1}T \end{aligned} \tag{3.2}$$

When a simple coordinate transformation inclusive of the rotation $\Delta\psi$ is considered, the nonlinear equations expressing the stationary point P from the "new" vehicle-fixed coordinate system at time $k+1$ is given by eq. 3.3.

$$\begin{aligned} x_{p,k} &= (x_{p,k-1} - \Delta x)\cos\Delta\psi + (y_{p,k-1} - \Delta y)\sin\Delta\psi \\ y_{p,k} &= -(x_{p,k-1} - \Delta x)\sin\Delta\psi + (y_{p,k-1} - \Delta y)\cos\Delta\psi \end{aligned} \tag{3.3}$$

Bringing in the small angle assumption for the small movement of each point in the shift register between successive samples, and recognizing that in the shift register framework, $x_{p,k-1} = (p+1)\,UT$, these equations lose their trigonometric terms and eq. 3.4 is obtained.

$$\begin{aligned} x_{p,k} &= ((p+1)\,UT - UT) + (y_{p,k-1} - V_{k-1}T)\,r_{k-1}T \\ y_{p,k} &= -\left((p+1)\,UT - UT\right)r_{k-1}T + (y_{p,k-1} - V_{k-1}T) \end{aligned} \tag{3.4}$$

Then, neglecting the small multiplicative nonlinear terms $-V_{k-1}r_{k-1}T$ and $y_{p+1,k-1}r_{k-1}T$, and recognizing that the road points are still assumed to "move towards the vehicle" with a velocity approximately equal to U from the vehicle's perspective, the relationship between the road offset at time $k-1$ and the offset of that *same* point at time k is given by Eq. 3.5.

$$y_{p,k} = y_{p+1,k-1} - V_{k-1}T - r_{k-1}UT^2(p) \tag{3.5}$$

Here p is the index of the road point under consideration. The small angle assumption, in this case, merely asserts that the change in vehicle yaw angle from time $k-1$ to time k is small so that pUT represents the x distance of the road point at p timesteps ahead of the vehicle. Considering that most vehicles are only capable of $r < 60\,\frac{^\circ}{s}$ in yaw, and that common time steps T for a filter of this nature are on the order of $50HZ$, a vehicle could only reasonably achieve a $\Delta\psi$ of approximately 2° from timestep to timestep, which falls well under the definition of small angles.

Reassembling the road shift register dynamics under this new motion model, the following open-loop, augmented, discrete-time system is obtained in body-fixed coordinates and shown in equation 3.6, where $A11, A12, A21, A22, B11, B12, B21$ come from the discretization of Eq 3.1.

Reassembling the road shift register dynamics under this new motion model, the following open-loop, augmented, discrete-time system is obtained.

$$[\vec{x}_k] = \underbrace{\begin{bmatrix} A11 & A12 & 0 & 0 & 0 & 0 & \cdots & 0 \\ A21 & A22 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -T & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ -T & -UT^2 & 0 & 0 & 1 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -T & -(np-2)\,UT^2 & 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{A_d} \begin{bmatrix} V_{k-1} \\ r_{k-1} \\ y_{0,k-1} \\ y_{1,k-1} \\ \vdots \\ y_{np-2,k-1} \\ y_{np-1,k-1} \end{bmatrix} \tag{3.6}$$

$$+ \underbrace{\begin{bmatrix} B11 & B12 & 0 \\ B21 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{B_d} \begin{bmatrix} \delta \\ e \\ y_{r,i} \end{bmatrix}; \vec{z}_k = \underbrace{\begin{bmatrix} 0 & 1 & 0_{1\times np} \\ 0 & 0 & \mathbb{I}_{np\times np} \end{bmatrix}}_{H} [\vec{x_k}]$$

3.2 Estimation of body-fixed vehicle-road system states

In body-fixed coordinates, the decoupled nature of the open-loop vehicle-road system is no longer an issue. Since each road point's motion is now coupled to the vehicle motion, the measurement equation giving the measurements $\vec{z}_k$ as in Equation 3.6 for this estimator is quite simple. Measurements of states are direct, since state measurements come from a camera, which is always assumed to measure road geometry in the vehicle-fixed reference frame. V and r are both observable if yaw rate or multiple road points are available for measurement. The system is also able to estimate the road geometry ahead of the vehicle, and the observable space of the road geometry is determined by the farthest road point available for measurement by the camera, although a filter without measurements at the preview horizon will still "see" road geometry by way of the model propagation of the road input.

Because a map of road geometry is available, the map information can be used to generate a suitable $y_{r,i}$ value at time $k-1$ by aligning the road marker map with the vehicle coordinate frame. This step provides an input to the road geometry motion model, and allows the road line offset to propagate backwards towards the vehicle as time marches forward. This process is well-defined if the map registration process is known. However, because the filter is no longer keeping track of global states, map registration in the global reference frame must be

accomplished through a parallel process of some kind. In the body-fixed coordinate estimator, vehicle yaw offset from the lane line tangent as defined in Equation 3.7 and lateral position from the lane marker nearest the vehicle CG, given by y_0, are available in the augmented state vector. Unfortunately, global states X and Y are not.

$$\psi_{rel} \approx \frac{\hat{y}_{1,k-1} - \hat{y}_{0,k-1}}{UT} \tag{3.7}$$

To make use of quality GPS-free filtered odometry estimation strategies like the one presented in [17] *or* low-cost GPS sensors, which may provide sufficient longitudinal registration, consider a road map in the configuration shown in Equation 3.8

$$MAP = \begin{bmatrix} \vec{S_r} & \vec{X_r} & \vec{Y_r} & \vec{Z_r} & \vec{\phi_r} & \vec{\theta_r} & \vec{\psi_r} \end{bmatrix} \tag{3.8}$$

where the station S_r, or total distance traveled along the road is available in the map along with cartesian position and orientation of the mapped lane marker at a number of discrete points. The use of the path coordinate S_r allows for a 1:1 lookup of the road marker pose, even for closed-loop paths. At each timestep, a separate estimate of distance traveled by the vehicle (or longitudinal registration through a K-nearest-neighbor (KNN) search in the case of GPS) can be used to find the correct location of the vehicle in the map in path coordinates. Then, using interpolation of the map points and simple coordinate transformation equations, the positions of lane points within the preview horizon can be computed in current

"road frame" coordinates. Finally, the transformation between the road-aligned coordinate frame and the vehicle-fixed coordinate frame is achieved using the lateral offset of the vehicle from the lane line, given by $\hat{y}_{0,k-1}$ and a simple estimate of the vehicle's yaw offset from the road tangent, given by Equation 3.7. Once the transformation of map points in the preview horizon is completed, a suitable value of $y_{r,i}$ can be brought into the filter. This general concept is illustrated by Figure 3.2.

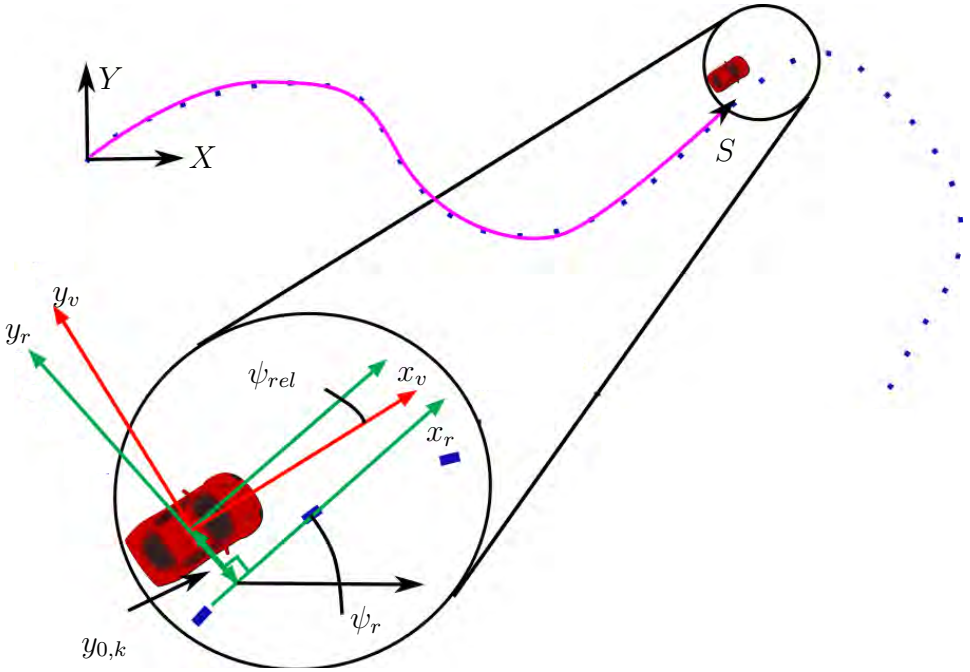

Figure 3.2. Map registration procedure for local-coordinate preview estimator

The filter thus far assumes that the forward speed of the vehicle, U is constant. If there are periods of significant, abrupt acceleration, then the LTI filter structure presented here will likely not function properly. This could be corrected by changing the measurement equations and state transition equations to allow for varying

spatial intervals in the preview horizon that maintain even temporal intervals based on a model for vehicle longitudinal acceleration. This is not demonstrated here as the aim of this work is to introduce the map-based estimator, and thus the focus is to provide lateral vehicle and environment states during cruising.

3.3 Simulations of autonomous driving using the local-coordinate preview estimator

To test the local version of the preview estimator in simulation, an LQR controller was implemented that used the estimator's vehicle and road geometry states. The longitudinal map registration values were given representative noise variance of $1.5m^2$ to simulate a WAAS-enabled GPS system. The vehicle was commanded to follow the simulated right lane edge at Penn State's Larson Transportation Institute (LTI) test track. The map of the test track and the position of the vehicle during the planar simulation are shown in Fig 3.3.

The fictitious camera and yaw rate measurements used in the local preview estimator were generated using Gaussian noise as in Table 2.1. The filter had access to right lane edge measurements as close to the vehicle as the camera geometry would allow ($7.5m$), and at each previewed time step all the way out to the preview horizon.

The lane position estimates quickly (<1 second) converge to their correct val-

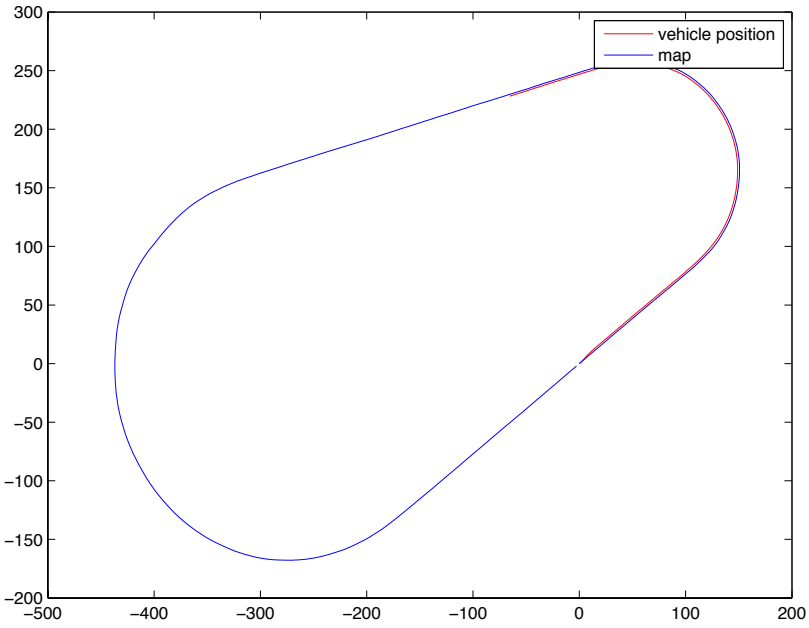

Figure 3.3. Map of fictitious road centerline marking and simulated vehicle position at LTI test track

ues, and the lateral offset of the lane from the vehicle agreed well with the ground truth commanded offset fed to the optimal preview controller of $1.5m$. The agreement between true and estimated non-previewed vehicle and road states during the simulation is shown in Fig. 3.4. These results show promise for this filter's ability to estimate local bicycle model and road geometry states with little more than a camera and a yaw rate gyroscope (if available), and without computationally complex calculations like optical flow or stereo image correspondence, which would be necessary for an implementation of SLAM.

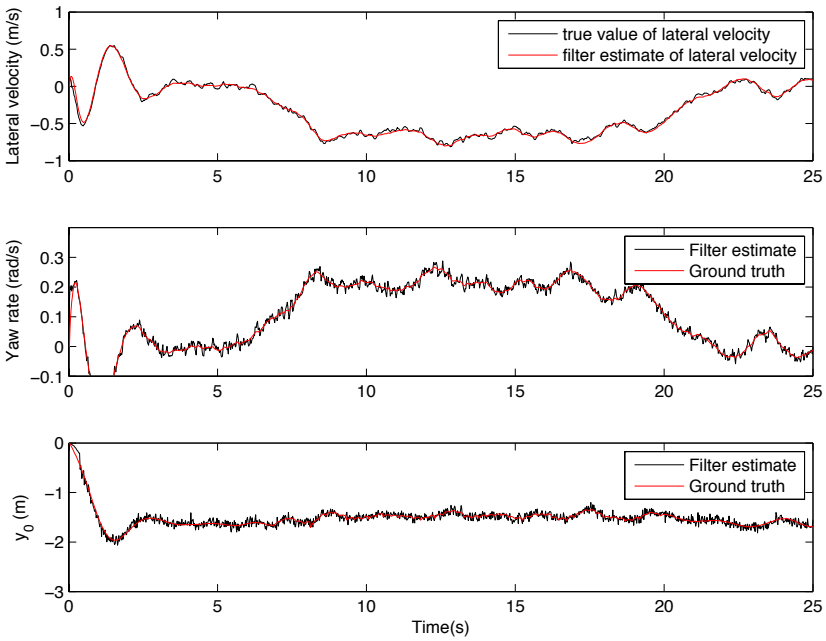

Figure 3.4. Ground truth versus estimated non-previewed states for simulated autonomous driving

3.4 Calculation of theoretical estimator performance

Specifics of the error-coordinate realization of this filter are given in Chapter 2, but recall that the filter shows decreasing improvements to state estimate accuracy as lookahead distance increases when all lane features in the preview horizon (except for those obscured by the vehicle body) are available for measurement. In fact, this is true even without a yaw rate measurement r, and regardless of whether the assumed noise in the camera measurements varies as in Table 2.1 or is a constant value. Theoretical variances on states $\hat{V}$,$\hat{r}$, and $\hat{y_0}$ for steady-state Kalman

estimators for the case in which noise parameters are given by Table 2.1 are shown in Figure 3.5.

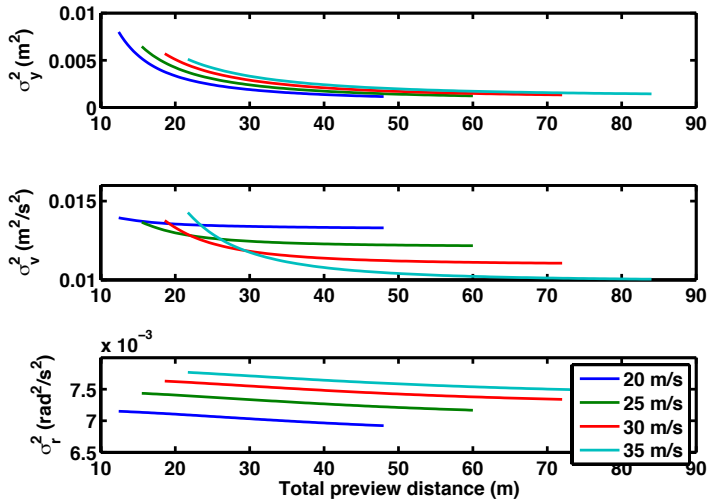

Figure 3.5. Predicted variance of vehicle states as a function of preview horizon length with measurement noise as given in Table 2.1

In Figure 3.5, the expected accuracy of the filter as implemented on Penn State's "Big Red" test vehicle is shown. It is not clear, however, whether the shape of the estimator performance curves has more to do with the yaw rate gyro's accuracy, the preview measurement noise model, or the number and distance of preview points measured. To address the real effects of increased lookahead distance with increasing numbers of features, see Figure 3.6, which differs from Figure 3.5 in that it was designed without the benefit of a yaw rate gyro, and to reflect a preview sensor with a constant $1cm^2$ measurement noise variance.

Figure 3.6 indicates that even for very high speeds, lane features beyond $40m$ do not improve state estimates at all, and suggests that at $20\frac{m}{s}$, a meager $30m$

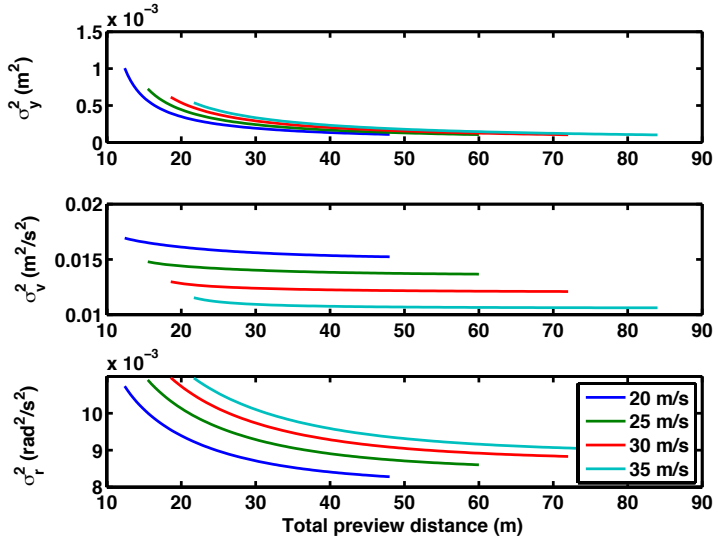

Figure 3.6. Predicted variance of vehicle states as a function of preview horizon length with $\sigma_{camera} = 0.01m^2$, no yaw rate measurement

of lookahead is sufficient to obtain maximum estimator performance for the states close to the vehicle. The fact that there are similar trends in the shape of the variance curves of Figures 3.5 and 3.6 despite the disparate noise models and characteristics indicates that preview horizon length offers decreasing benefits in estimating non-previewed states across measurement noise models. Whether a cubic relationship between x distance and variance is assumed, as in Figure 3.5, or a high-accuracy sensor can provide preview measurements with a small, constant variance as in Figure 3.6, one can compute a preview distance that offers maximum benefit in the estimation of non-previewed states.

Another interesting phenomenon, tacit in Figure 3.5, is that the required lookahead distance has changed with the transformation of the estimator from global to local coordinates, due to the differing use of the road point locations in the

shift register. Whereas in Chapter 2, preview point estimates in the state space represented their global locations, in this framework they are tied to lateral velocity and yaw rate at each time k directly, and represent locations relative to the vehicle's coordinate system. Figure 3.6 was generated using the same camera, yaw rate, and map registration noise values as Figure 2.10. To see the differences in estimator performance between the body-fixed and error-coordinate implementations for identical filter and vehicle designs at one forward speed, see Figure 3.7. This figure shows how the changing coordinate system definition changes predicted filter performance for the same vehicle and sensor configuration.

As Figure 3.7 indicates, the error-coordinate estimator exhibits more sensitivity to preview than the body-fixed version, which is likely due to the differences between coupling road/vehicle states in the measurement matrix H as opposed to coupling them directly in the state transition matrix A_d. Additionally, the error-coordinate estimator measures previewed points as linear combinations of road locations and integrated "global" vehicle states. The fact that y and ψ are not tracked by the body-fixed estimator may contribute to its increased predicted variance on lateral velocity and yaw rate estimates. Conversely, the body-fixed estimator uses lateral velocity and yaw rate to couple relative road movement to previewed geometry estimates directly in the A_d matrix, which could explain the decrease in predicted y_0 variance when moving from the error-coordinate to body-fixed coordinate framework.

In this way, the disparate structure of the two estimators is reflected in their distinct predicted error characteristics. Because the error-coordinate implementation of the preview estimator couples road and vehicle dynamics in measurement rather than in motion model, it is more sensitive to increasing preview distances for short preview horizon lengths, but is able to achieve better lateral velocity and yaw rate estimates with a longer preview horizon, since road motion and vehicle motion are decoupled in the system equations. The direct motion coupling between road and vehicle in the local-coordinate preview estimator implementation confounds estimation of the velocity states V and r, but direct measurement of locally referenced lane positions allows it to achieve better estimates of current lateral offset y_0.

While these differences are interesting, the error-coordiate estimator is far less suited to real-world implementation than the body-fixed version, and the body-fixed preview filter will be carried through the rest of this dissertation in the interest of showing the framework's real-world capabilities.

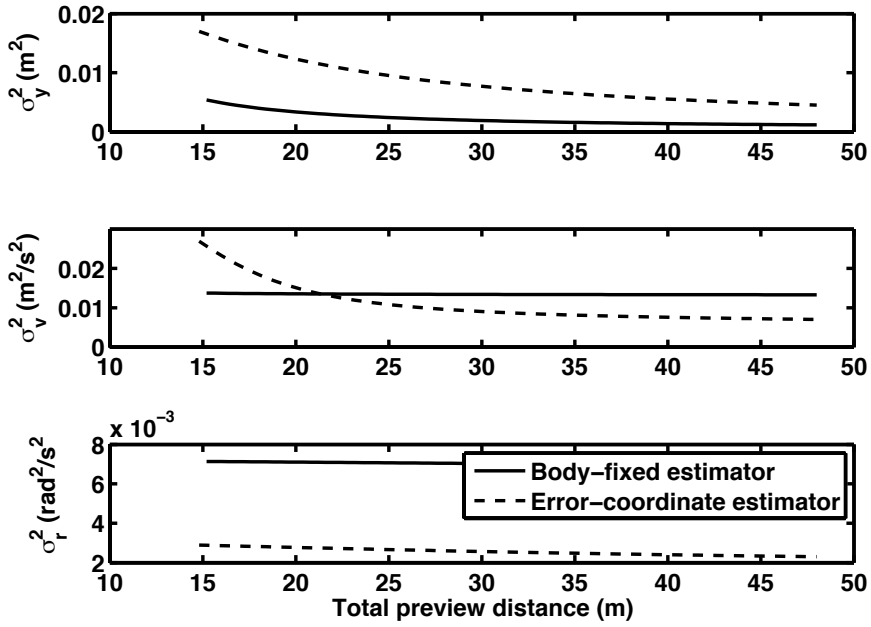

Figure 3.7. Predicted variance of vehicle states at $20\frac{m}{s}$ as a function of preview horizon length for error-coordinate and body-fixed preview estimators

Chapter 4

Implementation and Organization of the Preview Estimator in ROS

The simulations of the preview estimator in local coordinates outlined in the preceding section suggest that the preview estimation framework may be useful for improving state estimates and environment perception using low-cost sensors. However, a significant theoretical and physical divide exists between a pseudo-linear simulation with quality, gaussian simulated measurements and a realization of the preview estimator that can run in real time on low-cost computing equipment. To facilitate real-world tests of the estimator, a transition was made between the MATLAB programming environment and the mixed Python/C++ programming language structures that dominate Willow Garage's Robot Operating System (ROS).

4.1 The Robot Operating System (ROS)

ROS is a collection of command-line and graphical tools for roboticists. ROS's capabilities span hardware abstraction, sensor emulation, multibody simulations, and real-time control tools. The core ROS packages have been installed by the author on hardware ranging from the $35 Raspberry Pi ARM7-powered embedded computer, running at a 700 MHz clock speed, all the way to a 2011 MacBook Pro with a quad-core sandy bridge processor, on operating systems including Mac OS X Lion, Ubuntu 12.04, and Debian Wheezy. ROS is open source, and is developed and distributed under a BSD License [67]. The remainder of this chapter discusses the general organization of the software written in order to implement the preview estimator, and the specifics of the communication, interconnection, and methodology employed in order to make the estimator work with real driving data.

4.2 Realization of preview filter algorithm

While the local-coordinate preview estimation framework is linear in its motion model and filter framework, several sources of nonlinear coupling exist between the filter itself and the inputs and measurements it needs to run. In a sense, the preview estimator trades simplicity and linearity in the estimation process for an increased level of necessary support from auxiliary processes that manage the

feature map, update global vehicle pose estimates, process camera images, and process the feature map to generate suitable inputs for the preview-based motion model. To ensure that some estimate of lane position out to the preview horizon is maintained, even without a fully populated measurement matrix H, global map registration is necessary to provide a suitable and representative value of y_{in} to the filter motion model. Global map registration as explained in Chapter 3, is undoubtedly a nonlinear process, but is necessary to obtain y_{in}.

All tests of the preview filter were performed using the Robot Operating System (ROS) where the tasks necessary to run the filter were broken down into "nodes," with each performing a specific tasks, and "topics" that represent data streams flowing between the nodes as needed. The general setup of the system is shown in Figure 4.2. ROS handles many of the timing, coordinate transformation, and communication details automatically, so changes to the filter framework are rapidly and readily made.

Figure 4.1 shows a sample of the match between an actual forward-looking camera image and the mapped lane feature (right lane line) obtained using a preview filter during a straight-road 55-mph test, along with generated state estimates for a period of manual sinusoidal steering. The visual agreement between camera image and mapped lane geometry indicates a good estimate of local (and thus global) vehicle yaw angle and lateral position. The filter implemented used $np = 50$ and $T = .02s$ to put the preview horizon at $1s$ and $25m$, offering the full accuracy

benefits of preview for the camera used as discussed in Chapter 3.

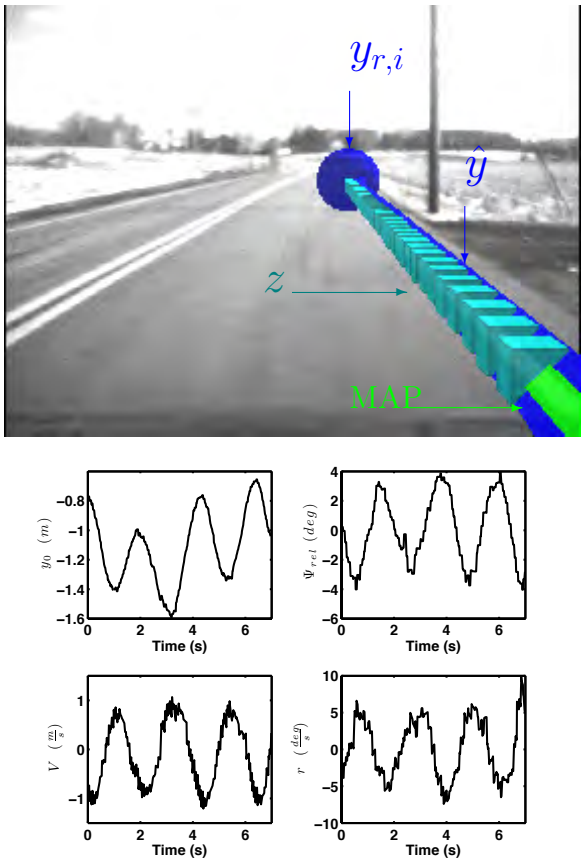

Figure 4.1. Filter results for sinusoidal steering at a speed of 55mph

Figure 4.2 shows the interconnection between all of the "nodes" that comprise a ROS-enabled system. Each has its own specific functions, as detailed below, except the "map registration" node, whose function is described in Chapter 3.

4.2.1 Map server and map generation

The procedures for creating maps of the right lane edge for use in the preview filter fell into two distinct categories: maps generated using public GIS data, and maps generated in-house using high-fidelity sensors. For early tests of the preview filter,

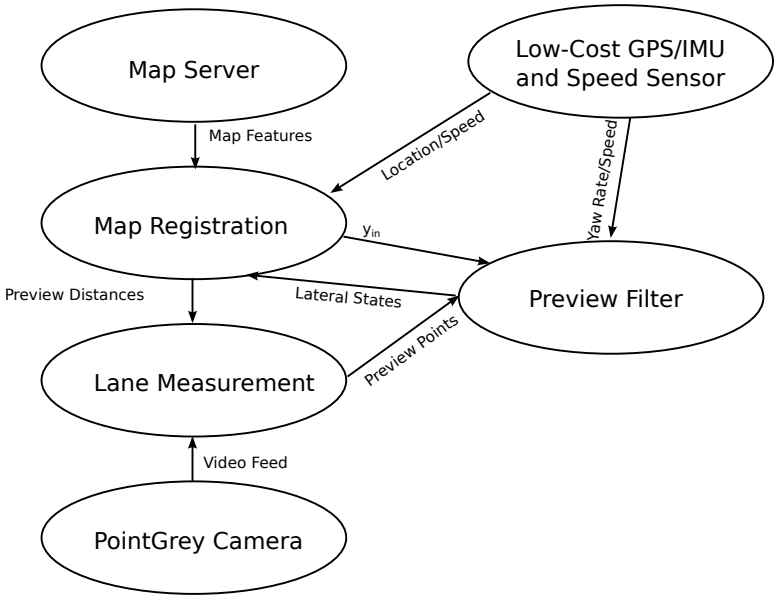

Figure 4.2. Functional Layout of Algorithm in ROS

maps were generated using online aerial imagery/map software by creating a list of points representing the right lane edge. This approach to generating mapped lane features is not new at all– in fact, this is the general method used in many map-based vision techniques, like those in [21, 22, 24]. While some efforts in using lane feature maps generated from satellite imagery like [22] use automatic feature extraction techniques, for this dissertation it was determined for the sake of brevity and to eliminate erroneous lane marker detection that tests of the preview filter using imagery-generated maps would be performed on maps encoded by hand. Google Earth works very well for this purpose, because it offers users the option to export a path as a Keyhole Markup Language (kml) file after clicking on a series of points in the Google Earth user interface. Figure 4.3 shows the process of selecting road edge points, which can be done with a great deal of precision at high zoom

levels.

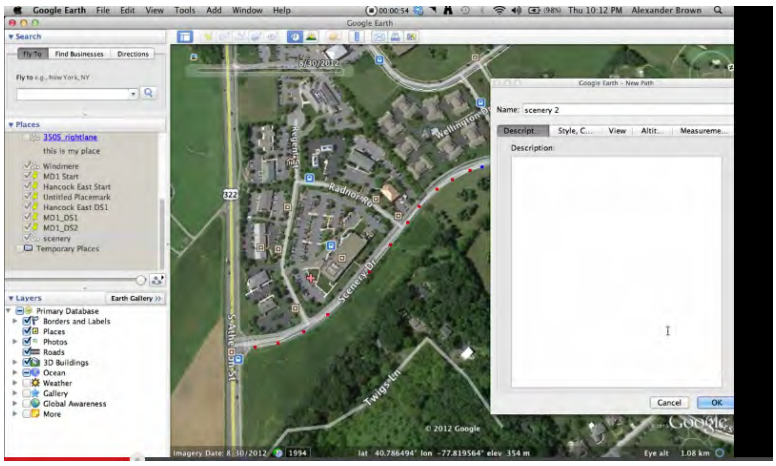

Figure 4.3. Building a road feature map in Google Earth

While this method of selecting points is ad-hoc and subject to human error, the level of precision offered in Google Earth is substantial (global accuracy concerns will be discussed in Chapter 5). The only obvious issue with this method, the lack of elevation data present in the exported kml file, was remedied handily with a small Python script that separately accessed Google's elevation data to generate road height profiles. After exporting the kml file representing the right road marker, a python script parsed the file, and queried Google Maps' web Application Programming Interface (API) for elevation data corresponding to each point. The web API gives rather precise JavaScript Object Notation (JSON)-encoded elevation information, which was used to complete the map description for use in the preview filter.

Maps generated with this method, when tested for agreement in lateral position estimates (lane offset y_0) by comparison with a RTK GPS/INS system, were

found to have insufficient global accuracy for a fair evaluation. This will be shown explicitly in Chapter 5, but in summary, the RTK GPS/INS system showed lateral offsets that did not match intuition or video data, and showed the vehicle exhibiting frequent departures from the lane when the GIS-generated maps were used. It should be noted, however, that this global error appeared not to affect lateral position or yaw rate estimate accuracy.

With this in mind, maps were generated using the RTK GPS/INS system and a downward-facing Sick LMS500 LIDAR sensor as shown in Figure 4.4 where mapping with this equipment was possible. Lane edges were extracted from the LIDAR data using a simple threshold-based approach on infrared reflectivity. This was accomplished by scanning outwards from the center of the LIDAR scan until a reflectivity value higher than an experimentally-determined threshold was found, and storing this point as the edge of the lane, after a coordinate transform to a globally-referenced set of East North Up (ENU) axes.

At Penn State's Larson Transportation Institute Test Track, multiple laps were performed to ensure repeatability in the mapping process and to confirm the fidelity of the Novatel Span DL4 GPS/INS system. A sample overlay of mapped LIDAR points onto the forward-looking camera image during the high-fidelity mapping is shown in Figure 4.5.

Lane markers at the track were exceptionally easy to extract from the LIDAR data, because of the reflective glass beads used in lane marker paint. After

transforming the LIDAR points into a map frame using ROS's coordinate transformation API, and extracting lane features, the points were saved for use in the map server. Instead of saving all mapped intensities and positions, as is the standard for many of the map-based localization and tracking approaches using LIDAR data, this study needs only a very compact version of the map, representing just the right lane marker's position.

With the map saved as a comma-separated-value (CSV) file, the ROS map server node broadcasts map data as a "latched topic," which ensures that excess bandwidth is not eaten up by the repeated transfer of static data. The map format upon transfer between ROS nodes is similar to the format given in eq. 3.8, but in a packed data form common to ROS.

Figure 4.4. Lidar/GPS mapping rig on test vehicle

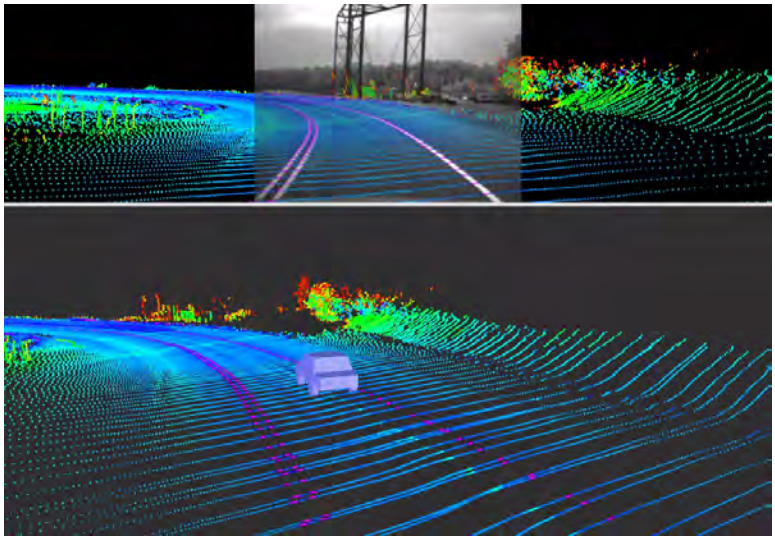

Figure 4.5. Mapping of Penn State Test Track

4.2.2 Low-cost GPS/IMU and speed sensor

In Figure 4.2, the node labeled "Low-Cost GPS/IMU and Speed Sensor" exists physically as a ROS-enabled BeagleBone embedded single-board computer interfacing with GPS and IMU modules through an Inter-Integrated Circuit (I2C) interface. This unit measures forward speed by communicating with a vehicle's CAN bus and/or wheel encoders. ROS allows for a high level of abstraction, and the map registration and inertial measurements could come from any number of sources. The embedded BeagleBone board with IMU and GPS sensors attached is shown in Figure 4.6.

The BeagleBone itself runs Ubuntu 12.04 Linux, and runs a bare-bones variant of ROS, so that once it has connectivity to the main data acquisition/filtering computer (in most cases during tests in this dissertation, a 2011 MacBook Pro), there is full connectivity between all ROS nodes and topics on the network. This

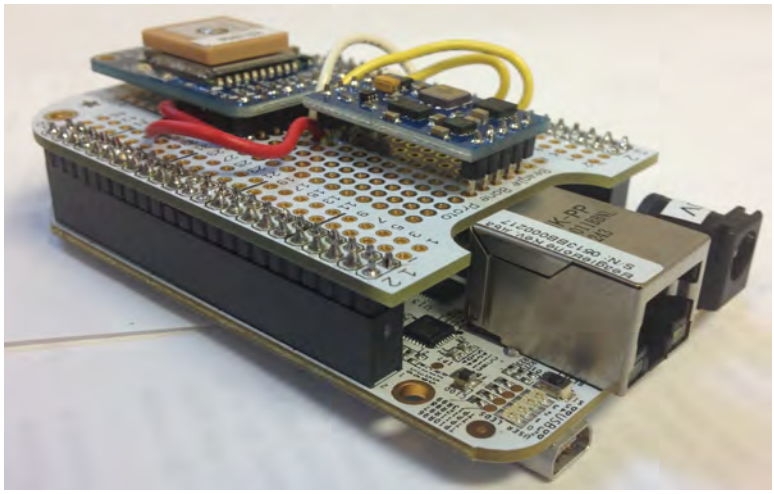

Figure 4.6. BeagleBone with custom sensor suite

allows for high-level programming of the hardware/software interfaces for data collection, and seamless integration with high-level control and filtering processes.

With that said, there proved to be very little qualitative difference between running the camera interface node and the node that parsed data from the USB-powered OBDII/CAN bus speed sensor module (not shown) on the BeagleBone versus running those hardware-level interface nodes on the main filtering computer. In most cases, it was easier to leave all hardware connected to the BeagleBone, and leave the high-level filtering and visualization tasks to the MacBook.

4.2.3 Lane measurement

For all tests in this document, a PointGrey Research Firefly MV USB 2.0 monochrome camera with VGA resolution comprised the hardware portion of the lane measurement system. This camera, chosen for its price ($35) and low pixel-level noise,

Figure 4.7. PointGrey Firefly camera

produces a quality video stream, although many modern production-grade cameras at least approach, if not surpass its performance. The camera is shown in Figure 4.7

Time stamped camera images brought into the lane measurement node were cropped to reflect the road vanishing point calculated from eq. 2.6 according to downward tilt. Then, a canny-filtered version of the image was searched for white pixels according to the expected locations of a straight lane, with a search area wide enough to encompass curved roads, but small enough to attempt to avoid extraneous markings outside of the lane.

Note that because of ROS's easily expandable framework, any number of lane detection techniques could have been used in the preview filter. The canny-based constrained search was chosen for simplicity, and to decrease likelihood that edges

in the middle of the lane would throw off a true "dumb" search working outward from the center of the image at each time step.

4.2.4 Preview Filter

Figure 4.8 shows how local estimates of non-previewed vehicle states are fed back into the map registration procedure. This is accomplished by combining them with coarse longitudinal registration from GPS or an alternative odometric algorithm to obtain $y_{r,i}$ through a coordinate transformation. For the error-coordinate realization of the preview filter, this feedback of estimator states is not needed. Note, too, that Figure 4.8 shows a steady-state oberver gain L, while in practice, L is updated according to the standard current-measurement Kalman filter equations as derived in [15], and shown briefly with time-varying system and measurement matrices in Equation 4.1.

$$\begin{aligned}
\hat{x}_{k|k-1} &= A_{d,k-1}\hat{x}_{k-1|k-1} + B_{d,k-1}u_{k-1} \\
P_{k|k-1} &= A_{d,k-1}P_{k-1|k-1}A_{d,k-1}^T + Q_k \\
\tilde{y}_k &= z_k - H_k\hat{x}_{k|k-1} \\
L_k &= P_{k|k-1}H_k^T\left(H_kP_{k|k-1}H_k^T + R_k\right)^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_k\tilde{y}_k \\
P_{k|k} &= \left(\mathbb{I} - L_kH_k\right)P_{k|k-1}
\end{aligned} \tag{4.1}$$

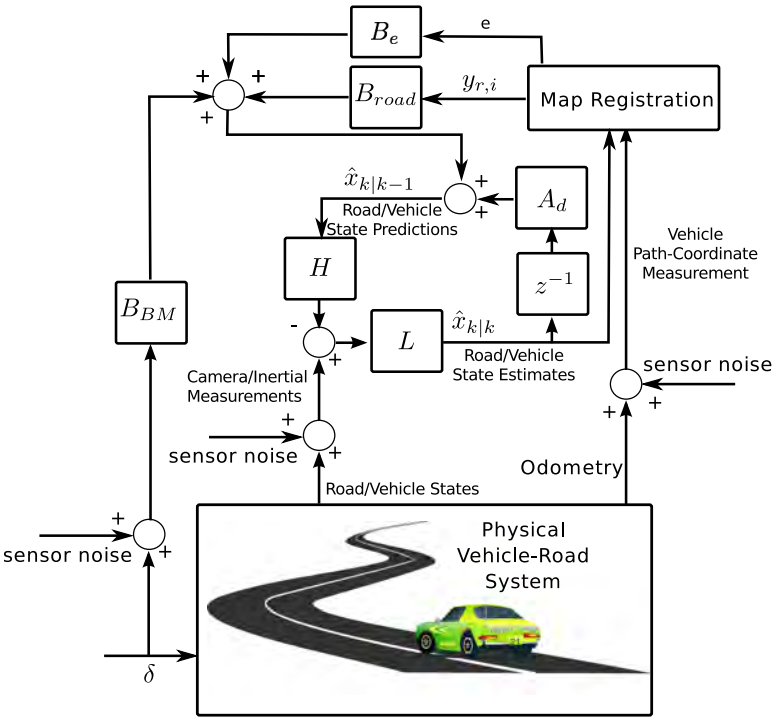

Figure 4.8. Block diagram layout of preview filter

Chapter 5

Experimental Validation of Preview Estimation Framework

To assess whether the theoretical advantages of the preview filter framework translate to real life, and if they do, to what degree, two separate batteries of testing were performed. The first, on an exceptionally tortuous road with a speed limit of $55mph$, was performed without the aid of steering sensor or superelevation inputs to the filter, and was performed with a map generated from GIS data. The second was performed at Penn State's Larson Transportation Institute test track, and included a filter setup that had access to a high-resolution low-cost steering sensor and a LIDAR-generated map of road superelevation at every point. For both tests, the filter itself was run separately on collected data, but ROS is set up to emulate real-time performances, with data playing back at exactly the rate they did in the actual event. Actual real-time tests of the preview filter are not

shown until Chapter 7, where the preview filter is run in real-time along with an optimal preview controller.

5.0.5 Sensors and data acquisition

In both of the testing scenarios, non-previewed vehicle states were compared with states provided by a Novatel SPAN RTK GPS system augmented with a Honeywell HG1700 ring-laser gyroscope. The $2cm$ position accuracy provided by the Novatel system was deemed sufficiently accurate to act as a fair ground-truth measurement of lane position, as long as the global accuracy of the map is equal to or better than the accuracy of the Novatel system. The lateral velocity and yaw rate estimates from the system have proven extremely accurate and reliable in controlled tests at Penn State's Larson Transportation Institute test track. The Novatel Span system's INSPVA (Inertial Navigation System Position Velocity Acceleration) data were collected at $50HZ$, and raw IMU measurements from the Honeywell HG1700 were collected at $100HZ$. The preview filter ran with a 50-timestep preview horizon, with a sample time of $0.02s$. All messages and state estimates from the preview filter were time-stamped at the time of creation according to the computer's internal clock, and ground truth data from the GPS/INS system were timestamped using GPS time for comparison.

Low-cost sensor inputs to the preview filter for both tests were provided with the same hardware setup as described in Chapter 4, with the exception of the

Table 5.1. 1999 Hyundai Elantra Vehicle Parameters

Symbol	Value	Unit
m	1436	kg
b	1.489	m
a	1.069	m
I_{zz}	2500	kg·m^2
$C_{\alpha f}$	-60000	N/rad
$C_{\alpha r}$	-80000	N/rad

speed measurement, which was provided by a Controller Area Network (CAN)-bus sensor for the testing with the 1999 Hyundai, and differentiated quadrature encoder measurements on the 1989 GMC "Big Red," since this vehicle was manufactured without a user-serviceable interface to diagnostic data.

5.1 Testing on Pennsylvania Route 350

In order to test the preview filter algorithm on as simple a hardware set as possible and in as real a scenario as possible, a test was performed using a 1999 Hyundai Elantra on Pennsylvania state route 350. Parameters for the Hyundai were compiled using a combination of physical measurements and consultation of both CarSim equivalent vehicles (tire parameters) and NHTSA vehicle parameter databases [68] and are given in Table 5.1.

The test, run at night, spanned the global position trace shown in Figure 5.1. This road segment is deceptively curvy, with blind turns made by rocky abutments obscuring far lane measurements through most of the sweeping turns. These turns

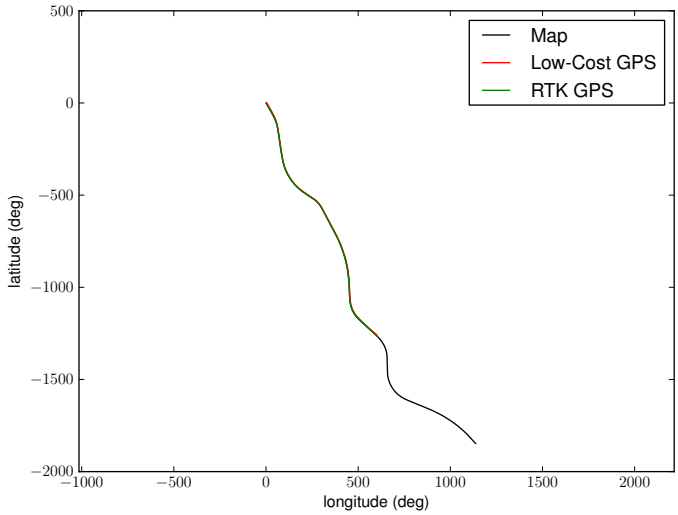

Figure 5.1. Global position trace during Route 350 test

are relatively flat (little superelevation), which makes them good candidates for testing the preview filter without superelevation influences. An approximately constant speed of $55MPH$, the road's speed limit, was maintained for the entire traversal.

As mentioned in the preceding section, the preview filter itself only had access to sensor information from the low-cost BeagleBone-based sensor block. Longitudinal map registration as described in Chapter 4 was accomplished using the KNN nearest-neighbor search method. In other words, the odometric path coordinate S used in the preview filter's map registration procedure was computed *relative to the map*. The resulting error in longitudinal path coordinate S when compared to the S-coordinate computed using the position from the Novatel system is shown in Figure 5.2.

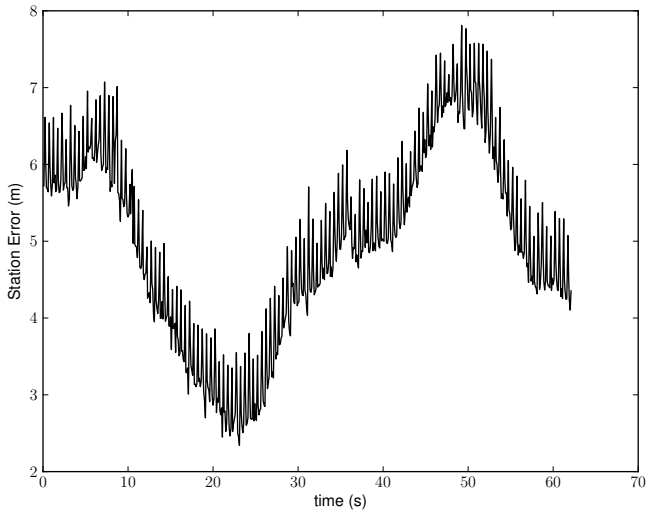

Figure 5.2. Error in path-coordinate registration

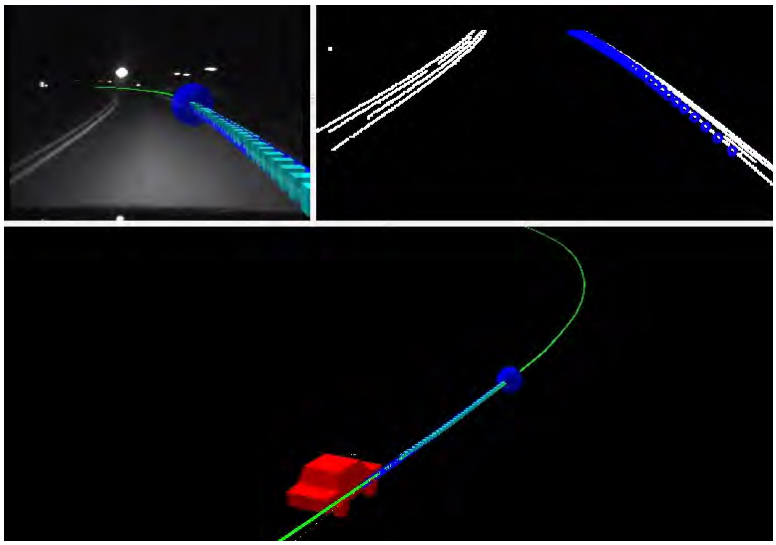

Figure 5.3. Raw lane measurements, global position estimates, and image overlay during Route 350 traversal

The results shown in Figure 5.2 are typical for the low-cost GPS system used, and have proven to be accurate enough for use with the preview filter, as evidenced by the augmented-reality-type overlay of map, image, and preview filter estimate given by ROS's visualization software RVIZ, shown in Figure 5.3.

This figure shows the upcoming turn in the correct location, even though it has not entered the preview filter framework yet, which indicates that the small errors in longitudinal map registration are overshadowed by the generally gradual curvature of a public road. Further, any lateral errors introduced by longitudinal error are quickly corrected by camera measurements once they enter the preview filter.

5.1.1 Results

The non-previewed states tracked by the preview filter, V, r, and y_0 are all primary candidates for validation. The bicycle-model state estimate match between RTK GPS/INS is shown in Figure 5.4.

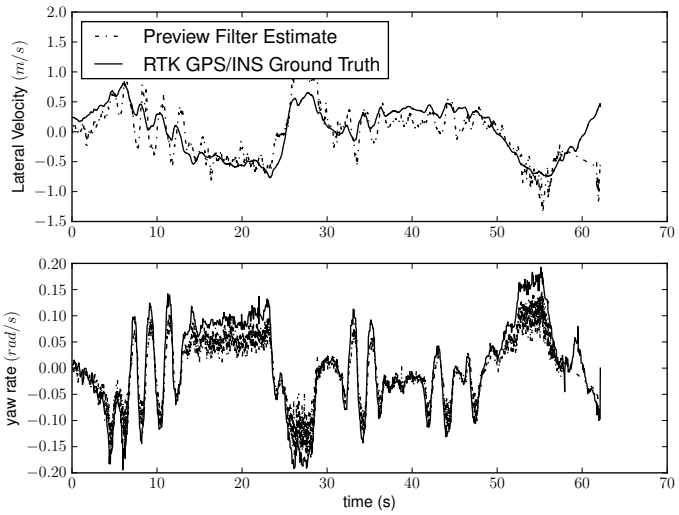

Figure 5.4. Vehicle State Comparison

The filter's ability to estimate lateral velocity is rather good, especially with

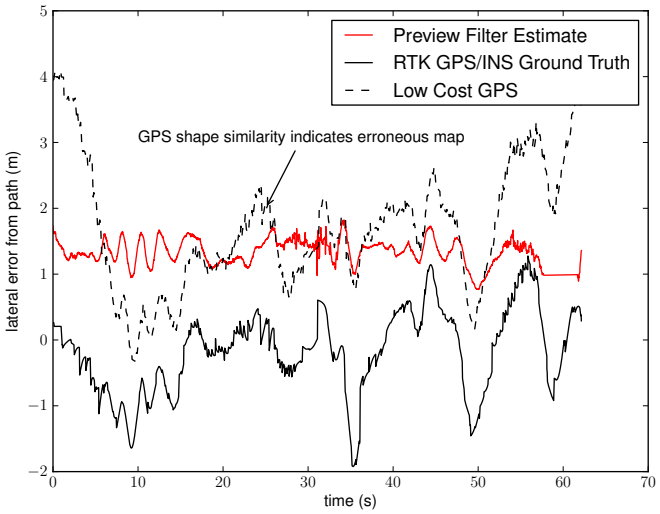

Figure 5.5. Lateral offset from the mapped lane marker

the lack of access the filter has to steering angle and the poor accuracy of the map. Yaw rate is also quite good, aside from some minor scale factor issues, which may have to do with the small assumed variance in steer angle δ, which can make a Kalman filter "pull" estimates towards the model prediction, which in this case, again, had no access to steer angle. Still, this match suggests that the preview filter could only improve with the inclusion of model inputs other than $y_{r,i}$, as demonstrated in the second test at Penn State's test track.

The situation is a little more complex when the lateral offset y_0 is considered. The match between this state and the "ground truth" is shown in Figure 5.5. While at first glance it appears that the preview filter was unable to correctly measure y_0, it is important to look at the scale and shape of the "ground truth" and raw low-cost GPS measurements of y_0. The vehicle, according to the Novatel system,

is crossing the lane boundary, and swinging wildly around in the lane throughout the traversal. This indicates not that the position measurements from the Novatel system are erroneous (the low-cost GPS measurement shows a similar shape), and not that the preview filter estimate is corrupted, but rather that the *map* is at fault. Remember, this is a measure of *globally referenced* lateral position offset, compared to the local one that the preview filter maintains. This means that the preview filter, even in the face of a map with global errors, is able to correctly assert the trends in vehicle lateral position seen in the video data. While these results do not speak to its absolute accuracy, this is addressed by the following experiment, in which a higher-fidelity map is used.

5.2 Testing at the Larson Transportation Institute test track

To mitigate potential effects of an erroneous map built from GIS data, the map used in the testing at the Larson Transportation Institute test track was constructed using a combination of the Novatel INS system and a downward-facing LIDAR scanner, as explained in Chapter 4. This allowed for an absolute globally-referenced map of the right lane line to be used. The lane lines at the Larson Transportation Institute track are of standard width and composition. They were painted as part of a cooperative agreement with the Pennsylvania Department of Transportation

(PENNDOT) in May of 2013, as shown in Figure 5.6. The lines were scanned with the LIDAR rig immediately after painting to ensure a high-quality map.

Figure 5.6. Lane lines being painted at the Larson Transportation Institute test track

The paint, like the paint on all public roads in the area, is impregnated with glass beads so that it luminesces at night. This also aids in the mapping step, because the glass beads boost the infrared reflectivity measured by the LIDAR, aiding in classification of the lane edges.

5.2.1 Vehicle Setup

For tests at the LTI test track, given the abundance of redundant sensors on Penn State's autonomous testbed "Big Red," a 1989 GMC 2500, this vehicle was used for all tests at the closed course testing facility. This vehicle's inertial and tire parameters are given in Table 5.2, and it is shown in Figure 5.7.

Parameters for the GMC 2500 were taken from previous physical tests conducted by the author Dr. Sittikorn Lapapong as part of his PhD work in [69]. Tire cornering stiffness values were obtained through exhaustive testing of the ve-

Figure 5.7. 1989 GMC 2500 "Big Red"

Table 5.2. 1989 GMC 2500 Vehicle Parameters

Symbol	Value	Unit
m	2579	kg
b	1.964	m
a	1.39	m
I_{zz}	5411	kg·m^2
$C_{\alpha f}$	-75700	N/rad
$C_{\alpha r}$	-83700	N/rad

hicle's tires, as documented for various tire pressure values in [6]. The vehicle speed during this test was roughly 20 $\frac{m}{s}$ throughout the traversal. The test included sinusoidal steering on several of the straight sections of the road to excite the vehicle dynamics for the purposes of testing the preview filter.

Figure 5.8 shows the ground truth position trace and map for the counter-clockwise validation run around the test track, along with the forward speed of the vehicle during the test. Low-cost sensor data were augmented with "ground truth" data from the INS system for comparison and validation of the filter. As Figure 5.8 indicates, the speed during the test was not entirely constant, but the

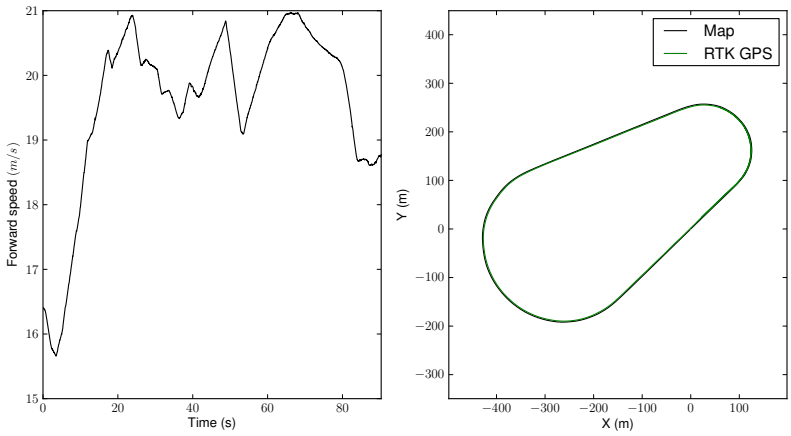

Figure 5.8. Test trajectory and speed profile

driver attempted to maintain a fairly constant speed throughout the entire loop. Steering angle was measured by a string potentiometer at the steering rack, with effective roadwheel angle computed from the average angle of the left and right wheels. These were calibrated as in [69] using skid-plates on both front wheels.

Once again, longitudinal map registration was achieved using the low-cost GPS aboard the BeagleBoard embedded computer using a K-Nearest-Neighbor (KNN) search algorithm. Longitudinal map registration accuracy obtained from the low-cost GPS was within $3m$ for the majority of the tests, which was confirmed by comparison to the Novatel DGPS unit. Therefore, curves were anticipated correctly by the preview filter through the road input y_{in}, even when the lane detection step did not see lane edges out to the preview distance. The preview filter maintained 50 preview points with a timestep of $0.02s$, for a total of $1s$ of preview. Camera measurements arrived from the PointGrey camera at $15Hz$. Note that because the camera measurement update rate is slower than the system propagation frequency,

the statistics of the filter's estimates are slightly different than they would be with a $50Hz$ measurement frequency. These statistics are still governed by eq. 4.1, with the equation for $P_{k|k-1}$ inflating the system covariance matrix in propagation appropriately between measurement updates.

5.2.2 Results

Figure 5.9 shows the lateral velocity and yaw rate match between the ground truth INS system and the preview filter for the test run. Once a relatively steady-state speed is achieved at approximately 10s, the states match very well. Scale-factor errors causing overprediction of $\hat{V}$ as seen in Figure 5.9 at 15 to 25 seconds and again from 65 to 75 seconds cause some input-correlated noise characteristics. This can be seen more clearly in 5.11. These scale-factor errors in transient maneuvers are often caused by unmodeled tire or suspension dynamics that manifest as "notch filters" at certain frequencies [70].

Astute readers will note that between approximately 10s and 25s, the vehicle is in an ostensibly aggressive steady-state turn with a yaw rate of 0.2 $\frac{rad}{s}$, which, according to Table 5.2 and a steady-state analysis of bicycle model dynamics, should result in a steady-state lateral velocity of nearly 0.7 $\frac{m}{s}$. However, because of the superelevation of the test track through both turns, which is brought in via the map data server to the propagation step of the filter equations, the steady-state lateral velocity does not reach the value it would on a flat road. This observation is

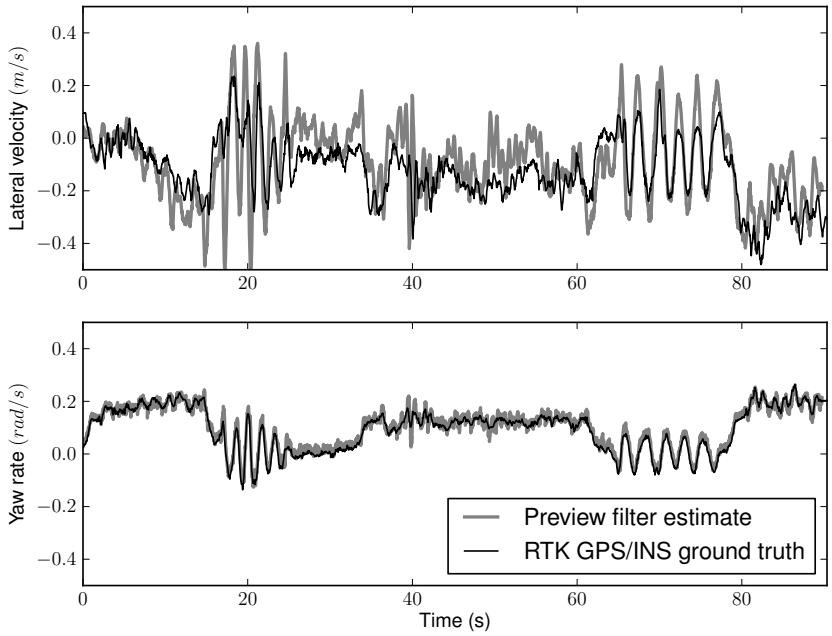

Figure 5.9. Planar velocity state estimates versus ground truth for test track data

important to note for vehicle control purposes since most prior research on lateral vehicle control and state estimation neglects this system input. This research shows that the influence of road superelevation can be quite significant in correctly estimating vehicle states–particularly lateral velocity. Both the ground-truth value of V obtained from the INS system and the filter estimate $\hat{V}$ correctly reflect the effects of the known superelevation through the turns.

$\hat{V}$ and $\hat{r}$, however, are only two of the 52 states maintained by the preview filter in the test. While it would be impractical to show the global error of each preview point y in the filter shift register, the point y_0 represents the vehicle's current lateral offset from the lane, which is another extremely important state for vehicle

control. The methodology in [63] indicates that when using Optimal Preview Control for guidance, preview points far ahead of the vehicle are of eventually decreasing utility for control, which is a similar conclusion to the one drawn in Chapter 3, but points close to the vehicle, especially the current lateral offset, are very important. The match between ground truth and the preview filter's estimate of lateral offset is shown in Figure 5.10. Ground truth, in this case, refers to the lateral offset from the map (which was generated using LIDAR during a separate traversal), as measured by the RTK differential GPS system.

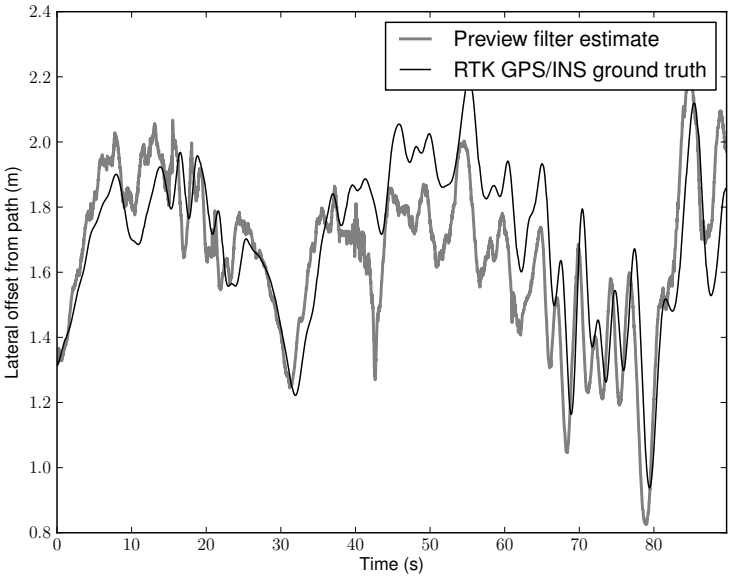

Figure 5.10. Lateral position offset from mapped path estimate versus ground truth of test track data

The lateral offset, too, matches very well with the INS ground truth data for the run, although the estimator performance bounds suggested in Chapter 3 are not quite acheived. This is to be expected, given the fact that the linear

approximations inherent in the shift register and vehicle dynamics model are, in fact, approximations. The relative accuracy of the filter given its simplicity is notable. Additionally, not all preview points in the shift register are available for measurement at all times. This very fact makes the preview filter capable of self-diagnosis via a lookup table of expected system covariance as a function of preview points available, for comparison with filter-maintained state confidence values. In the validation run examined, anywhere between 5 and 40 of the 50 preview samples were measured for each update step, even though the filter was designed to track 50 preview points – not all were available for measurement at any given time. Once again, if no lane was "found" at a particular preview distance for measurement, that state's statistics for the propagation step were governed by the Kalman filter equations of eq. 4.1. This allows the filter to maintain estimates of unseen preview points with statistical confidence bounds, even when those points are not found during a camera update.

To examine the performance of the filter, and address any issues with modeling error, it makes sense to examine the error characteristics of these three states individually. These are shown in Figure 5.11. While the error in $\hat{r}$ is arguably white, which follows naturally given its direct measurement and low sensor variance, the errors in $\hat{V}$ and $\hat{y_0}$ are not as well-structured. While the errors are generally agreeably small in maximum and in standard deviation, there appear to be systematic components in each, which may have causes in measurement model error (e.g. ap-

proaching non-flat portions of the track, roll motion of the vehicle) or in timing. The sinusoidal error in $\hat{V}$ occurs on flat portions of the test track, and appears to be due to a very slight time lag in $\hat{V}$ when compared to the value of V provided by the ground truth INS system. This is a bit anomalous, as physical systems are always more complex and higher order than their simplified models. Therefore one generally expects the system responses measured on a vehicle to more often lag behind model-based state estimates, rather than the opposite situation observed here. Because the real measurements lead the estimator, it suggests that there are some signal timing or modeling errors that are effectively adding a small amount of phase lag to the estimated lateral velocity. Further investigation of this phenomenon is ongoing, but the overall small error in $\hat{V}$ and its steady-state agreement with the defense-grade INS system show the capabilities of the preview filter's simple, straightforward structure.

In regards to $\hat{y_0}$, recall that the forward-looking camera can not directly measure this state, so it is propagated back using state estimates from the closest available preview measurement. Thus, small errors in U, longitudinal acceleration or deceleration, and other unmodeled error sources may contribute to its final value. Additionally, differences between the way a canny filter picks out a lane edge in image data and the way the threshold algorithm finds the lane edge in the LIDAR data could also contribute to high-frequency error. The simplistic planar lane assumption, and even the linearization of the relative road motion model

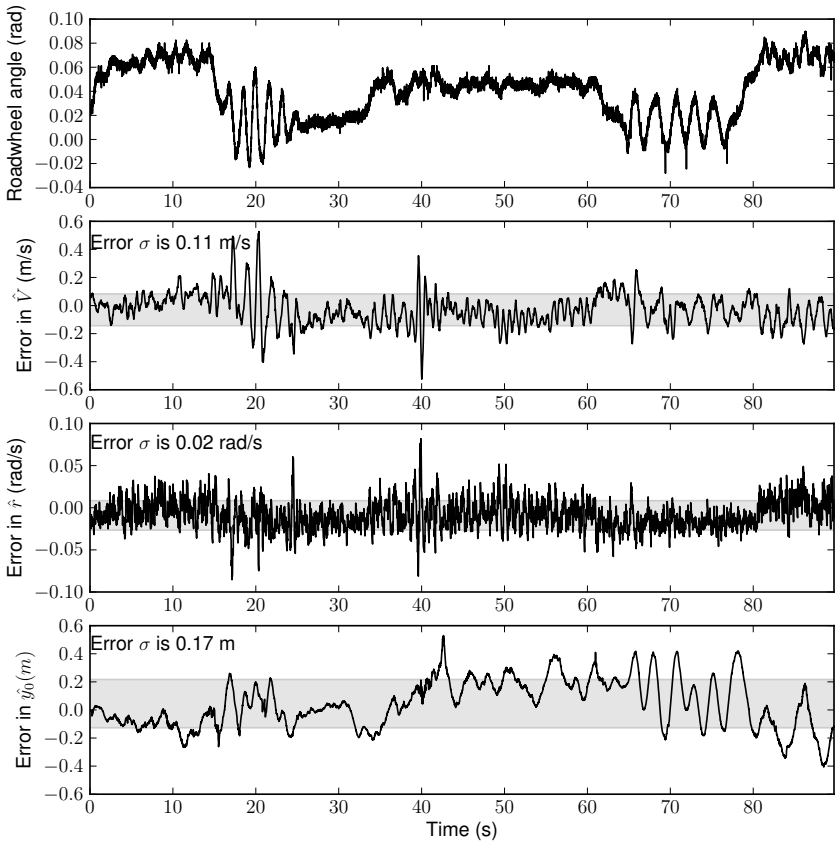

Figure 5.11. Error in planar vehicle states for test track data

could contribute to low-frequency error while the vehicle traverses a sharp turn. Any of these error sources might explain some of the low-frequency bias error seen between 45 and 75 seconds, which occurs on a portion of the test-track with a changing bank angle.

Finally, to address the way the filter functions in the measurement update step, the innovations for all camera preview measurements for all preview points and all timesteps are shown in Figure 5.12. Figure 5.12 shows, from a top-down view,

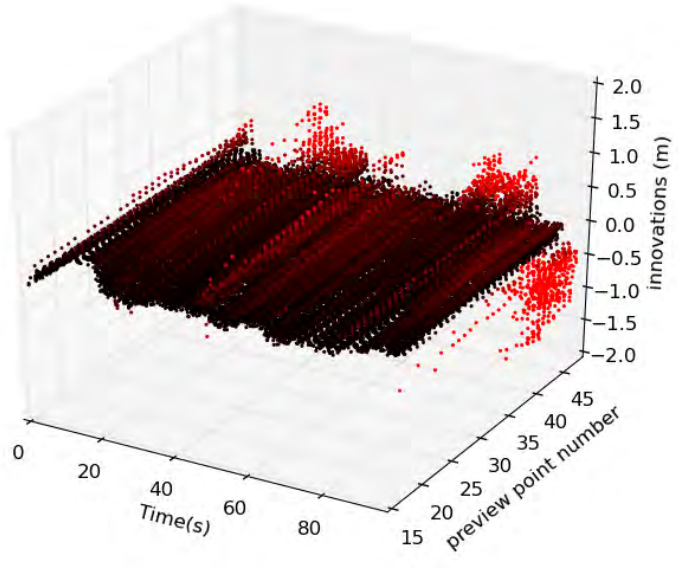

Figure 5.12. Preview measurement innovations for all preview points

the visibility space as a function of time for the data collection run–a pronounced dip in visible features close to the vehicle occurs at 10s into the run, for example. This figure also shows that the innovations at the far end of the preview horizon are markedly large, which is an expected result when an error in longitudinal map registration occurs, since the y_{in} value generated from the map is not corrected until the camera can see it. Kalman Filters, when functioning perfectly, have zero-mean innovations with white noise characteristics. To examine the mean and standard deviation of the innovations at each preview point for the whole run, consider Figure 5.13.

Figure 5.13 shows that innovations out to preview point 40 (.8s or 16m in front

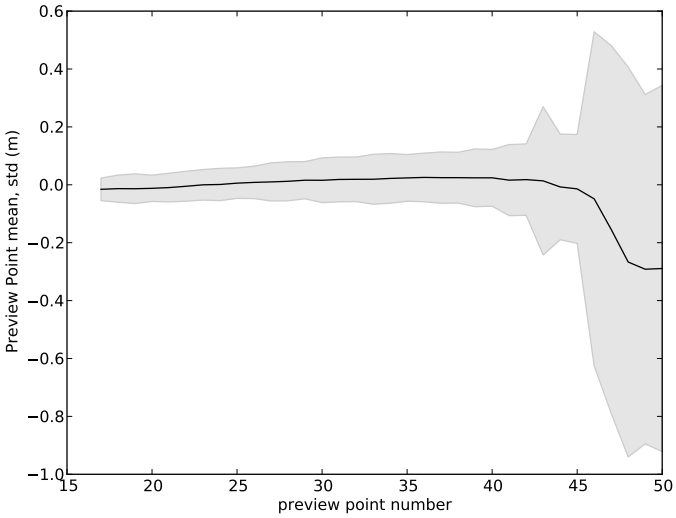

Figure 5.13. Preview measurement innovations for all preview points (mean of innovations shown in black, covariance of innovations shown in grey)

of the vehicle) are nearly zero mean, but that beyond this point, innovations are biased, which is an artifact of a mean bias in the GPS longitudinal map registration, similar to the mean bias shown in Figure 5.2. While this is an unfortunate side-effect of this registration method, it does not have a detrimental effect on the majority of the road geometry estimates in the preview horizon. Longitudinal registration methods with zero-mean error characteristics, like the GPS-free pitch-based localization approach in [2], may eliminate this bias in innovations near the preview horizon. Overall, however, the innovations statistics on the preview horizon indicate that the shift register motion model for the road geometry is not introducing significant biases into the filtered estimate of roadway geometry, at least for the conditions tested. The potential for modeling error-induced bias does exist, however, and is discussed in the next chapter.

5.3 Performace of local-coordinate estimator under ideal conditions

The preceding section showed that while the local-coordinate preview filter provides good vehicle state estimates during a challenging traversal of a test track with banked turns, and under aggressive sinusoidal steering, it also suggests that the filter does not perform quite as well as the results of Chapter 3 would suggest. The predicted variance values reported in that chapter for the non-previewed state estimates suggest that the filter should be more accurate than the tests at the Larson Institute Test Track showed. Taking the square root of the variances predicted by steady-state Kalman filter designs for varying preview distances and speeds should give theoretical 1-σ variation for state estimate errors in the filter, with the assumed process and measurement noise values. These are shown for Big Red at varying speeds in Figure 5.14. For the $20m$ preview distance and $20\frac{m}{s}$ forward speed of the the tests at the test track, the algorithm predicts 1-σ bounds for $\hat{y}_0$,$\hat{V}$, and $\hat{r}$ of $0.058m$, $0.12\frac{m}{s}$, and $0.08\frac{rad}{s}$, respectively. By comparison, the physical results featured 1-σ error bounds on the same states of $0.17m$, $0.11\frac{m}{s}$, and $0.02\frac{rad}{s}$. This is a fair agreement considering the assumptions in the filter's derivation.

As these numbers show, the predicted bound on yaw rate r is worse than the error observed in the tests. This is likely due to the inflated noise assumed for the camera and yaw rate measurements to deal with unmodeled error sources.

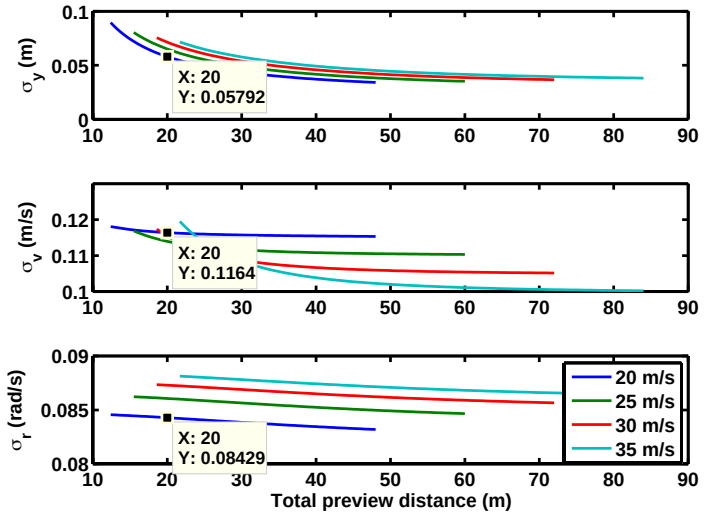

Figure 5.14. Theoretical standard deviation of non-previewed state estimates for Big Red

The lateral velocity bound generally agrees with Figure 5.11's statistics, but the predicted bound on lateral position, y_0, is not achieved by the filter in that test. The next chapter deals with some possible sources of this disagreement.

Because the test presented above was performed with aggressive steering, it is possible that the error in the state estimates resulted from this excitation. Most driving using the filter, even under autonomous control, will not feature such pronounced sinusoidal steering. Therefore, an experiment was conducted to examine whether a more realistic traversal enables the filter to achieve the lateral position accuracy predicted by the steady-state Kalman filter analysis. The vehicle was driven on a straight section of the track at $20\frac{m}{s}$ with the same filter design as before, but the test did not feature sinusoidal steering to excite vehicle dynamics. Figure 5.15 shows the state estimate errors for this case, along with the standard

deviation of the error in each non-previewed state. Figure 5.15 shows that for

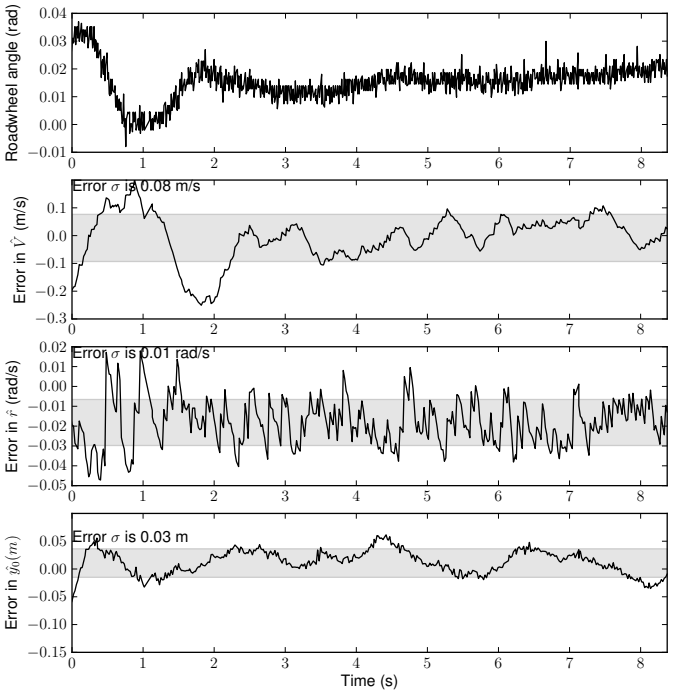

Figure 5.15. State errors for $20\frac{m}{s}$ straight-running traversal of test track's north straightaway

a short, straight-run traversal with little steering excitation, the filter surpasses predicted error characteristics for all non-previewed states, with the $3cm$ 1-σ error exceptionally close to the accuracy of the ground truth Novatel SPAN system (about $2cm$) and the accuracy of the map. Indeed, the estimator error is so close to the expected error in the "truth" measurement from the DGPS/INS that the estimator's error is not likely to be statistically significant. Additionally, it can be considered to be on the order of the predicted accuracy of the filter according to Figure 5.14.

Interestingly enough, although the lateral dynamics of the vehicle were not significantly excited during this traversal, and the driver maintained "lane-center" through the straight segment of road, there was a surprising amount of lateral deviation from the road center's theoretical offset of $1.625m$ from the right edge for the test track lane. To illustrate this further, the lateral position estimate versus ground truth, in an analogous figure to Figure 5.10, is shown in Figure 5.16. The most interesting thing about Figure 5.16 is the pronounced variation in offset

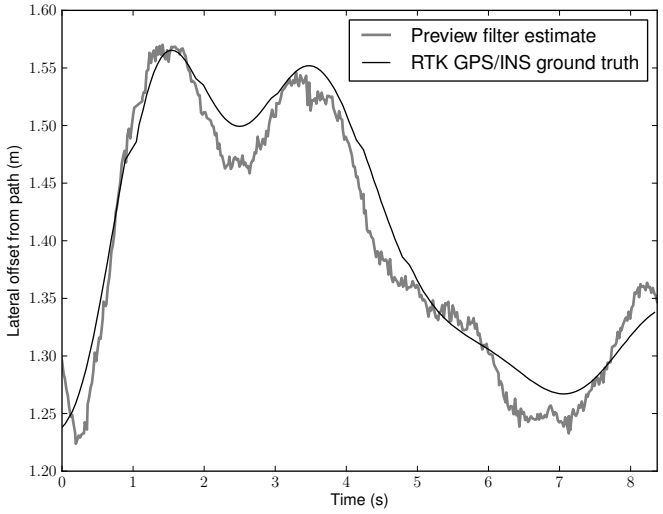

Figure 5.16. Lateral position offset from mapped path estimate versus ground truth for $20\frac{m}{s}$ straight-running traversal of test track's north straightaway

from the right lane edge. The waviness in the lane line itself is likely "at fault" for a large portion of this variation. This supports the general assertion of the preview filter framework that there is significant information content in the lateral deviation of lane lines from their "ideal" geometry. This is an effect not captured in lane tracking methods that assume parametric equation fits for road geometry.

Additionally, it is important to note that both the ground truth data and the filter's estimate of lateral position capture this effect. The map of the lane line reflected the lane's actual position, the DGPS accurately measured the vehicle's offset from the lane, and the preview filter's estimates of lateral offset agreed. This agreement would not be possible with an erroneous map, an incorrect algorithm implementation, or significant unmodeled error sources.

Finally, during this test, the driver was instructed to drive as straight as possible, and maintain a position in the center of the lane. The observed variation of almost $30cm$ or 1 foot over the 9-second straightaway indicates that the driver appeared to be tolerant of a significant amount of lateral position error. This tolerance suggests that a bound of $30cm$ on tracking error in a combined perception/guidance system would achieve a performance similar to that of the human driver in the test shown, but that a controller with an error on the order of $3cm$ might be unnecessary. The difference in lateral positioning performance might not even be perceptible to vehicle occupants.

Even with these results, which showcase the preview filter's ability to estimate vehicle states in ideal conditions, it is apparent that during challenging maneuvers, there are error sources not accounted for in the preview filter's derivation. One of these is discussed in the next chapter, using a concrete simplification to examine error bounds over a wide range of expected driving conditions.

Chapter 6

Causality and Steady Curves

Errors in the preview estimator framework can arise from a number of possible sources. These include the following:

1. Errors arising from the assumption that sensor noise is Gaussian with zero-mean and the modeled covariance. This includes noise on the steering angle δ, the errors in the GPS-based longitudinal map registration, and measurements from the yaw rate gyroscope, which were inflated to deal with unmodeled error sources.

2. Errors arising from erroneous velocity. The forward speed U is considered a parameter, but is actually a measured quantity at every time step.

3. Errors arising from unmodeled tire dynamics.

4. Errors associated with the discretization of the vehicle model, and the Euler-integral approximation of the motion of the road relative to the vehicle.

5. Errors arising from the "future predictions" of the shift register, which could be significant for particular geometry. These will cause disagreement between measurements of previewed geometry and predictions, because the measurements of road points will repeatedly correct causal errors in the shift register road motion model.

While almost all of these errors plague every Kalman filter-based estimator in one form or another, the one item unique to the preview estimator specifically in this dissertation is Item 5. Because of this, we will address this source of error in detail.

One of the key assumptions in the whole preview estimation and control frameworks is that the x-distance traveled relative to the vehicle of each road feature is constant and equal to UT meters with each timestep. This is usually a decent assumption, but when the road is turning significantly, it is not the x-distance that is UT meters with each timestep, it is the S-distance. These two are the same for a straight road, but can be quite different for a curved one. It would be an exercise in futility to consider all possible modeling errors arising from all possible road geometries that may violate the straight preview horizon assumption. However, since most highways are comprised of straight and curved sections, these are the two that will receive a thorough treatment in this chapter. This chapter investigates the modeling error that arises from the assumption of a straight preview horizon when in fact the road is curved, and the vehicle is in a steady-state turn.

In Chapter 3, it was shown that state matches in simulation showed that the local coordinate preview filter performed as expected. However, for the sections of the simulation where the vehicle traversed a constant turn, the simulated estimate error plots tell an interesting and different story, as in Figure 6.1. While the error in

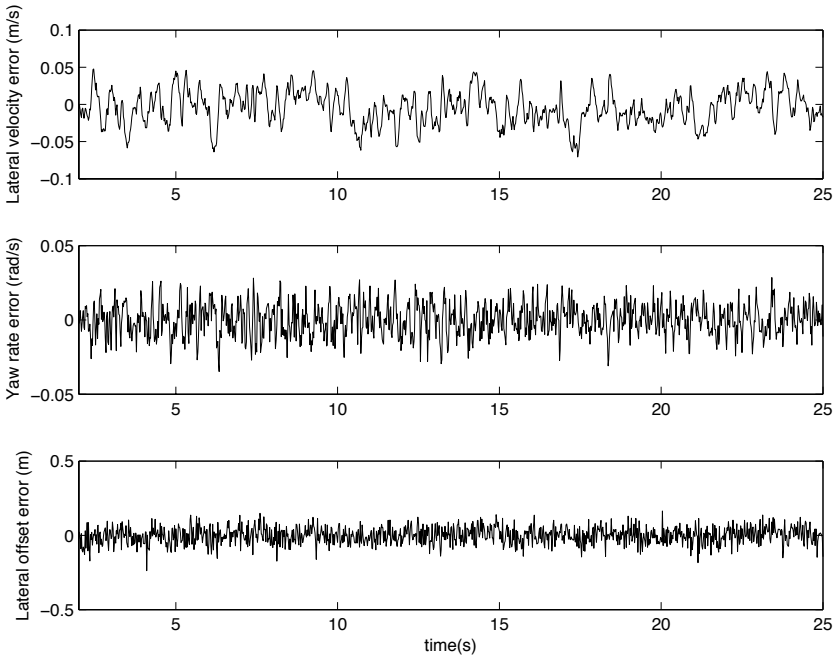

Figure 6.1. Error in measurement for constant-radius turn for planar simulation of preview estimator

the non-previewed states appears at first glance to be zero-mean, close inspection reveals that the mean of the estimate errors might not remain identically zero, which they should indeed do, since the simulation was purely planar with a linear vehicle model, and all measurements were corrupted with zero-mean Gaussian noise. To investigate the possibility of systematic errors corrupting the mean of the estimator errors, see Figure 6.2. Here, the exact same simulation of control

and estimation using the preview filter/preview controller was performed, but the "measurements" of lane geometry, steer angle, and yaw rate had no noise added to them. In this way, the estimator was tested as a weighted sum of inputs and measurements that should, in theory, result in a zero-error estimate.

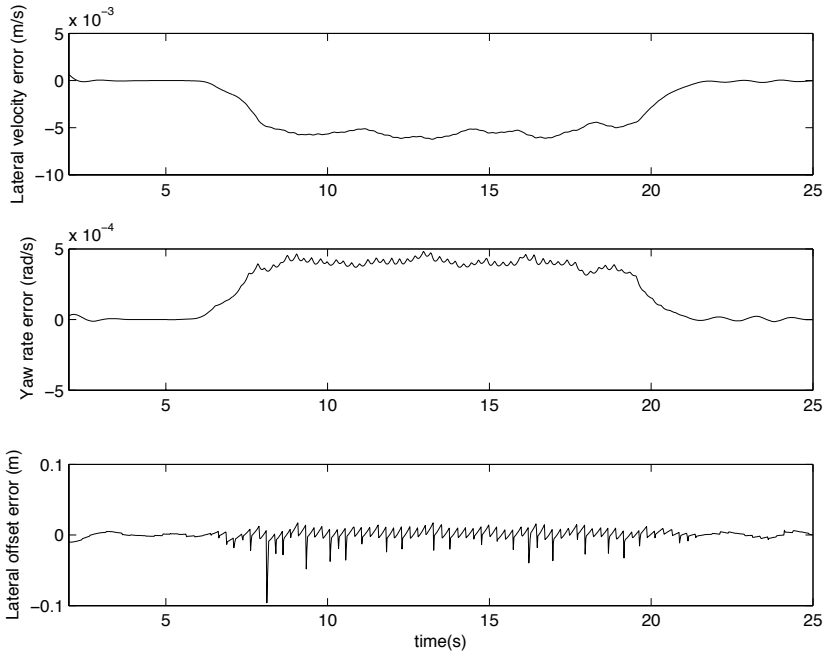

Figure 6.2. Error in measurement for constant-radius turn for planar simulation of preview estimator with no measurement noise

While the noise in Figure 6.1 does appear white with near zero mean, the resulting errors are not identically zero mean as expected. It is thus evident that there are modeling issues that result in small preview filter estimate biases when the vehicle traverses a constant radius turn. Figure 6.2 does, however, suggest these errors will be small relative to the Gaussian modeled noise sources, especially violation of the temporal preview assumption caused by a "wandering" vehicle that

does not follow a curve perfectly. But without knowing their underlying causes, and bounds on their magnitudes, they cannot be ignored. This chapter investigates these error sources by looking at an alternative "ideal preview scenario" from the one through which the local-coordinate preview filter was derived.

6.1 Road curvature preview horizon bias

Because the causality of the temporal preview horizon is not exact unless the vehicle follows a straight path perfectly, some bias errors will appear in the estimates of V and r, and also in the preview points y_p, when the vehicle is violating this assumption. Remember that the predictive nature of the preview horizon asserts that a point pUT meters in front of the vehicle is also pT timesteps in front of the vehicle. If the vehicle's path is anything other than a straight line in the local x direction for the entire horizon length, this is bound to be somewhat incorrect. Errors arising from this phenomenon will have two separate types:

1. Mismatch between the assumed straight preview horizon and the actual preview horizon shape (along the path) will cause structured error in the system dynamic model, i.e. the vehicle-road A_d matrix.

2. The preview filter will attempt to measure lane features along the assumed straight preview horizon, rather than at the correct preview distances in front of the vehicle. This will create direct error in the measurements.

To determine the source and expected magnitude of bias errors in preview-filter state estimates, consider Figure 6.3, which gives a qualitative picture of the measurement error that can result by assuming that a point P at p timesteps in front of the vehicle lies along the vehicle's x-axis rather than along the path traveled. One of the key underlying assumptions in the preview filter is that we know, in a temporal sense, *when* the reference path points on the preview horizon will pass underneath the vehicle. In reality, this is a causally flawed premise, because any path deviation or curvature causes an increase in path length when compared to a perfectly straight line. This effectively causes the vehicle to drive farther on curves than on straights given the same, straight preview horizon assumption. Therefore, the timing of a vehicle's path is in disagreement between straight and curved road segments. The magnitude of the resulting error is discussed in the sections that follow.

To mathematically analyze this error, note that when a vehicle is following a path of arbitrary shape, and under competent control, the temporal preview horizon lies approximately along the path, not along the vehicle's x-axis. The x-axis and the temporal horizon line up exactly only when the vehicle is traveling along a straight road. In the straight road case, even with slight angular misalignment between vehicle and path, the small angle assumption justifies the fact that measurements of y_p, or the y_{road} measurement at p timesteps in front of the vehicle, is acquired at pUT meters in front of the vehicle, and in local coordinates. The

degree to which the geometry introduces error depends on the radius of curvature of the road R, the vehicle's forward speed U, the timestep T, and the preview point p in question. Additionally, consider that the shift register dynamics in the vehicle motion model, which depend on preview points that are *temporally even* in spacing, will also be erroneous for highly curved roads. This effect is shown in Figure 6.3, which shows a vehicle in the midst of a constant-radius turn traversal.

To address each of these issues, recall that the equation of a circle in cartesian coordinates is given by:

$$\begin{aligned} y &= R\,(\sin\theta - 1) \\ x &= R\,(\cos\theta) \end{aligned} \tag{6.1}$$

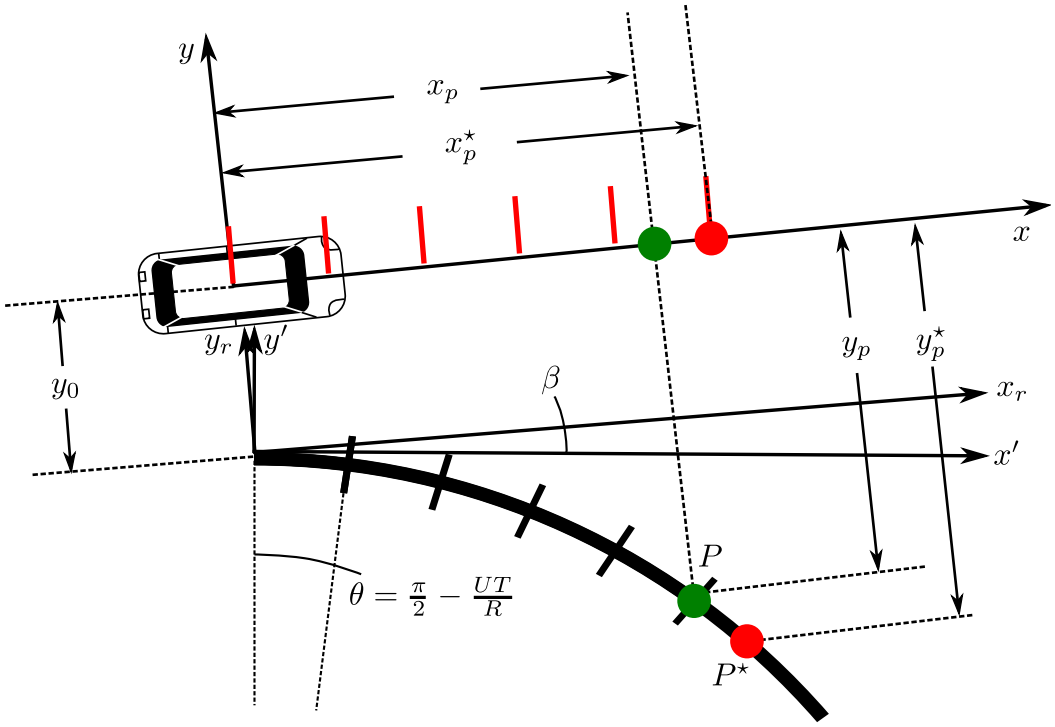

Figure 6.3. Error in measurement for constant-radius turn

where θ is the subtended angle in the circle's circumferential arc. Now, imagining that the vehicle's controller enables it to traverse the arc at a constant lateral offset, in the same way the estimator's predictive shift register has assumed the preview horizon to extend straight in front of the vehicle, it is clear that the preview horizon points must lie along the path itself rather than on an imaginary arm extending straight in front of the vehicle. These "true" preview points are shown as black tick marks in Figure 6.3. This path, for the simple circular arc representing the right lane edge of a right-hand turn, has a length of:

$$S = R\theta \tag{6.2}$$

which means that for a given preview point at p time steps in front of the vehicle, the angle subtending the vehicle's circular trajectory of length $S = pUT$ meters for a right-handed turn as shown in Figure 6.3, is:

$$\theta = \frac{\pi}{2} - \frac{pUT}{R} \tag{6.3}$$

This implies, in turn, that the true x', y' coordinates of the preview point P in a constant-radius, right-hand turn is actually equal to:

$$\begin{aligned} y'_P &= R\sin\left(\frac{\pi}{2} - \frac{pUT}{R}\right) - R \\ x'_P &= R\cos\left(\frac{\pi}{2} - \frac{pUT}{R}\right) \end{aligned} \tag{6.4}$$

When the coordinate transforms from the x', y' coordinates to the x, y coordinates are considered, which consist of a rotation through the steady-state sideslip angle β followed by a translation in the y-direction, the true x, y coordinates of the preview point at p time steps in front of the vehicle are given by:

$$\begin{aligned} y &= R\cos\theta\sin\beta + (R(\sin\theta - 1))\cos\beta + y_0 \\ x &= R\cos\theta\cos\beta - (R(\sin\theta - 1))\sin\beta \end{aligned} \tag{6.5}$$

However, with a straight preview horizon, the x coordinate in the A_d-matrix of the local-coordinate preview estimator equations, eq. 3.6 is *assumed* to be pUT, so that for the constant radius turn case, the error in the state transition matrix resulting from the curved path is given by:

$$\Delta A_d = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & \epsilon_{x,1}T & \vdots & 0 & \\ 0 & \epsilon_{x,2}T & \vdots & 0 & \\ 0 & \vdots & \vdots & 0 & \\ 0 & \epsilon_{x,np}T & \vdots & 0 & \\ 0 & 0 & 0 & 0 & \end{bmatrix} \tag{6.6}$$

where T is the time spacing between preview points, and $\epsilon_{x,i}$ are the errors in the preview point's x location in front of the vehicle. One way to think of each of these

errors is as an error in the "lever arm" that maps yaw rate motion to the lateral movement of each preview point from time step to time step. These individual errors $\epsilon_{x,i}$ are given by the difference between the assumed and actual lever arm lengths:

$$x_p - x_p^\star = \epsilon_{x,p} \tag{6.7}$$

These errors are easily calculated, since the true value of θ for the constant radius turn is known. However, while this analysis accounts for propagation error in the filter equations, consider that the filter, under the assumption of a straight preview horizon, "looks for the preview points" during the measurement step in the wrong spatial x locations. In other words, the measurements of the assumed preview point $P^\star$ as shown in Figure 6.3 will not produce the same y-values as a measurement at the actual preview point P. The magnitude of this error can be calculated for a given vehicle design, speed, and turn radius by solving for the subtended angle $\theta^\star$ of the assumed preview point, and then calculating, based on eq. 6.1 and eq. 6.5 the expected true value of the preview point's lateral offset y_p. This analysis leads to a vector of expected measurement errors for the constant-radius turn, given by eq. 6.8.

$$y_p - y_p^\star = \epsilon_{y,p} \tag{6.8}$$

To analyze the magnitude and shape of these error sources in detail, and see how they conspire to corrupt the expected value of the preview filter's state estimates,

the steady-state vehicle attitude angle β is necessary, along with a steady-state lane offset y_0. The sideslip angle β Comes from the vehicle dynamics, and depends on the vehicle's lateral acceleration, a function of turn radius and forward speed for a given vehicle design. Also note that these equations change slightly depending on whether the filter tracks the right or left lane edge, and whether it traverses a left or right hand turn. For a filter that tracks the right lane edge in a left-hand turn, as was the case for the simulation shown in Figure 6.1, the analysis is repeated by adjusting eq. 6.5 and eq. 6.3 accordingly to reflect the changes in the geometry, so that $\theta = \frac{pUT}{R} - \frac{\pi}{2}$ and $y'_p = R\,(\sin\theta + 1)$. The resulting differences in errors between the two curve directions will be shown in the following sections.

6.2 Steady state estimation error

As discussed in the preceding section, there is, in general, a difference between prediction and reality in the preview estimation algorithm. This suggests that while the Kalman filter structure of the estimator is generally effective in estimating vehicle states, some of the fundamental assumptions of the Kalman filter are violated for its derivation in the preview filter context. Among these, which include contributions from non-Gaussian camera noise and system linearization, the modeling error associated with the causal violation of the temporal horizon is an intriguing, unexpected guest. Two distinct geometric error sources in the propagation and measurement step combine in tandem to corrupt the normal zero-mean expecta-

tion for Kalman filter estimates. To begin to understand these errors, assume first that the Kalman gains have reached a steady-state value, and ignore (supposedly zero-mean) sensor noise for simplicity. Then, the estimator equations reduce to those of a simple Linear Time-Invariant (LTI) discrete observer. This is shown in eq. 6.9, with assumed system matrices and signals denoted with a $\star$ superscript.

$$\begin{aligned}\hat{x}^{\star}_{k|k} &= A^{\star}_{d}\hat{x}^{\star}_{k-1|k-1} + B^{\star}_{d}u_{k-1} + L\left(y^{\star}_{k} - H\hat{x}^{\star}_{k|k-1}\right) \\ &= A^{\star}_{d}\hat{x}^{\star}_{k-1|k-1} + B^{\star}_{d}u_{k-1} + L\left(y^{\star}_{k} - H\left(A^{\star}_{d}\hat{x}^{\star}_{k-1|k-1} + B^{\star}_{d}u_{k-1}\right)\right) \\ &= \left(\mathbb{I} - LH\right)A^{\star}_{d}\hat{x}^{\star}_{k-1|k-1} + \left(\mathbb{I} - LH\right)B^{\star}_{d}u_{k-1} + Ly^{\star}_{k}\end{aligned} \tag{6.9}$$

Recall that in the special case where the assumed system matrices $A^{\star}_{d}, B^{\star}_{d}$ are identical to the true system matrices, A_d, B_d, the error dynamics of the estimator are given by eq. 6.10:

$$e_k = x_k - \hat{x}_k = \left(A_d - LH\right)e_{k-1} \tag{6.10}$$

which is stable (e_k approaches zero) as time progresses. For this to occur, the eigenvalues of $(A_d - LH)$ must lie within the unit disk on the complex plane.

However, when the assumed and actual system dynamics differ, i.e. A_d does not equal $A^{\star}_{d}$, the discrete equation for error is more complex. If the assumed estimator is stable, estimates of system states will reach a steady value. In other

words, $\hat{x}_k = \hat{x}_{k-1} = \hat{x}_{ss}$. When eq. 6.9 is rearranged with this substitution, the following equation is obtained:

$$\hat{x}_{ss} = \left[\mathbb{I} - A_d^\star + LHA_d^\star\right]^{-1} \left[\left(\mathbb{I} - LH\right) B_d^\star u_{ss} + L y_{ss}^\star\right] \tag{6.11}$$

where the steady-state input u_{ss} is obtained by solving the vehicle dynamic equations, as in eq. 3.1 for a steady-state value of δ according to the vehicle parameters and the desired steady-state lateral acceleration $U \cdot r_{ss}$. In a similar fashion, a steady-state value of sideslip angle β and a vector of assumed and true lane measurements can be obtained from the preceding section's analysis when a vehicle is assumed to be traversing a particular geometry at a particular speed. In compact form, the true values of the steady-state vehicle and road states can also be obtained from any given combination of $\delta, y_{r,i}, e$ for the preview filter by performing a simple algebraic reduction of the system equations at steady state:

$$x_{ss} = \left[\mathbb{I} - A_d\right]^{-1} B_d u_{ss} \tag{6.12}$$

In this way the steady-state estimator error for the zero-noise case can be obtained for a steady-constant-radius curve as:

$$e_{ss} = x_{ss} - \hat{x}_{ss} \tag{6.13}$$

For the simulation results shown in Figure 6.2, where a steady-state yaw rate of $0.2\frac{rad}{s}$ and a forward velocity U of $20\frac{m}{s}$ was achieved during the left-hand curve, the measurement error $\epsilon_{y,p}$ from eq. 6.8 and the lever-arm error $\epsilon_{x,p}$ from eq. 6.7 are shown in Figure 6.4.

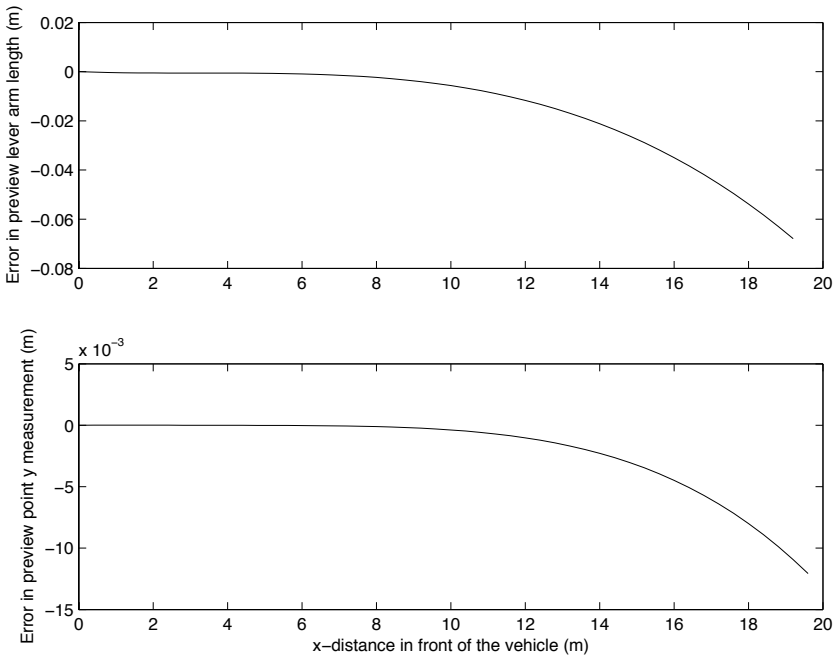

Figure 6.4. Error sources for a straight-horizon local preview filter when a vehicle traverses a left curve of $100m$ radius at $20\frac{m}{s}$ while tracking the right road edge at a $1.625m$ offset

Figure 6.4 shows these error sources for a particular scenario. It is important to remember that these errors will change with curve direction, lateral offset, and whether the preview filter tracks the left or right lane edge. This is true even for a particular vehicle, speed, and turn radius. Continuing with the simulation of Figure 6.2, the total expected value of the error in y_p for each preview point

in the preview horizon due to each of these errors is shown in Figure 6.5. These

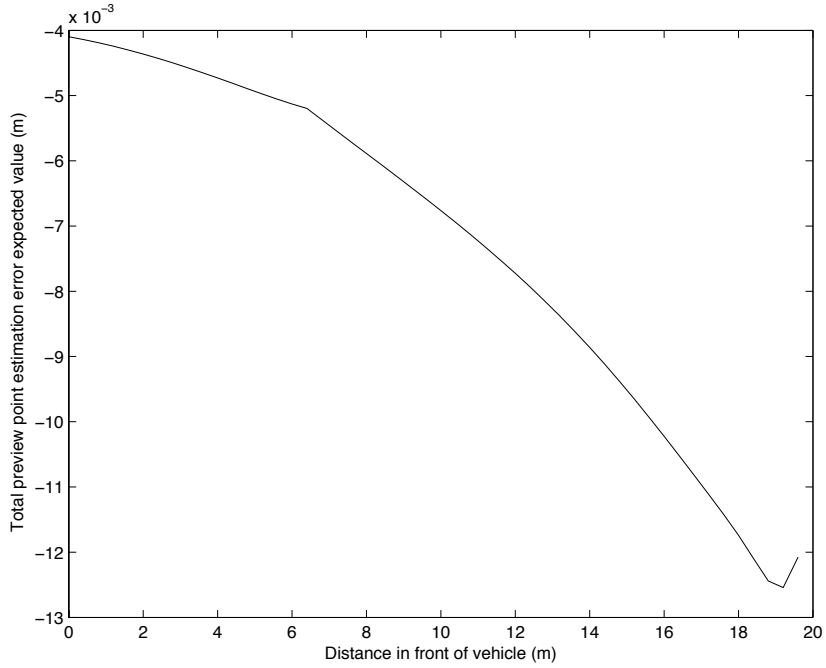

Figure 6.5. Expected value of total error in each preview point location for a straight-horizon local preview filter when a vehicle traverses a left curve of $100m$ radius at $20\frac{m}{s}$ while tracking the right road edge at a $1.625m$ offset

expected preview point errors are very small, in most cases less than 10% of the modeled camera noise, and generally less than $1cm$ for this preview horizon length (50 points, or 1s). The expected values of the errors in V and r for this scenario are 3.2×10^{-3} and 1.4×10^{-3} respectively. While these do not match the biases seen in figure 6.2 exactly, recall that the local-coordinate preview horizon is built upon the small angle assumption, an Euler-integration of preview point motion, and the assumption that y_p is small throughout the preview horizon. As preview horizons get longer, curves get sharper, and time steps increase, all of these assumptions

will contribute to the total bias. This suggests that while it may be tempting to recast the preview horizon motion model to account for path shape, and use a map of road geometry to determine which point is *actually* p time steps ahead of the vehicle for each measurement, these efforts will likely be thwarted or at least overshadowed by measurement noise, especially in longitudinal map registration.

6.3 Contributions of temporal horizon errors for a range of realistic driving conditions

Even though steady-state bias errors caused by the temporal preview horizon assumption were small in the preceding section, it makes sense to analyze their relative contributions for a range of realistic conditions, since they may change drastically across speeds and lateral acceleration values. To do this, note that simulations have shown that the bias errors are greater in cases where the road feature being tracked is on the *inside* of the turn traversed. In other words, right turns result in larger preview horizon and vehicle state estimate errors for a filter that tracks the right lane edge. The errors for a right-hand turn with a filter tracking the right lane edge can be seen for the same speed, radius, and vehicle as in the preceding section in Figure 6.6 below.

The difference in error magnitude is significant, so it makes sense to consider right-hand turns for a worst-case analysis of the steady-state biases caused by the

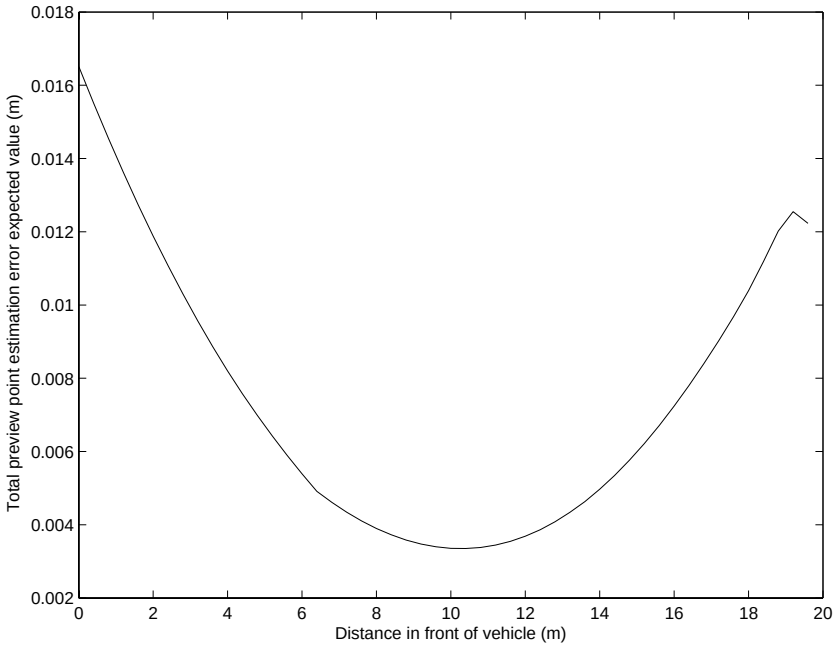

Figure 6.6. Expected value of total error in each preview point location for a straight-horizon local preview filter when a vehicle traverses a right curve of $100m$ radius at $20\frac{m}{s}$ while tracking the right road edge at a $1.625m$ offset

acausal nature of the preview filter's predictive horizon. To perform this analysis over a range of vehicle speeds, consider that highways in the United States are designed according to the American Association of State Highway And Transportation (AASHTO) Green Book [71], which states that vehicles should be subjected to an experimentally determined lateral acceleration that decreases with increasing forward speed. This design "side friction factor," or normalized lateral acceleration of the vehicle's center of mass, is used in eq. 6.14 to determine a minimum curve radius for a particular superelevation e and forward speed U.

Table 6.1. AASHTO Design Side Friction Factors

Speed (mph)	25	30	35	40	45	50	55	60	65	70	75	80	85
f_{design}	.23	.20	.18	.16	.15	.14	.13	.12	.11	.10	.09	.08	.07

$$R_{min} = \frac{U^2}{g\left(\frac{e}{100} + f_{design}\right)} \tag{6.14}$$

where $f_{design} = \frac{a_{y,CG}}{g}$ and g is the gravitational constant. The design side friction factors for a range of applicable highway design speeds are shown in Table 6.1 and taken from [71].

Using these equations, it is possible to obtain an idea of what slip angle β a vehicle can expect to experience while cruising at a road's design speed by specifying speed alone. Additionally, consider that on most limited-access highways, lanes are 12 feet or $3.65m$ wide, so a vehicle could be expected to maintain a steady-state position in the middle of the lane, or at a $1.83m$ offset from the right lane edge. With these two pieces of information, it is possible to examine the preview-point location and vehicle state errors for a range of speeds and preview horizon lengths. For a filter designed as in Table 2.1 with $1s$ of preview and a time step of $0.02s$, the biases as a function of preview point and speed are given in Figure 6.7, and the biases in the non-previewed states are given in Figure 6.8.

It is interesting to see the large effect higher lateral acceleration has at low design speeds when compounded with geometrically tighter radii. Figure 6.7 shows

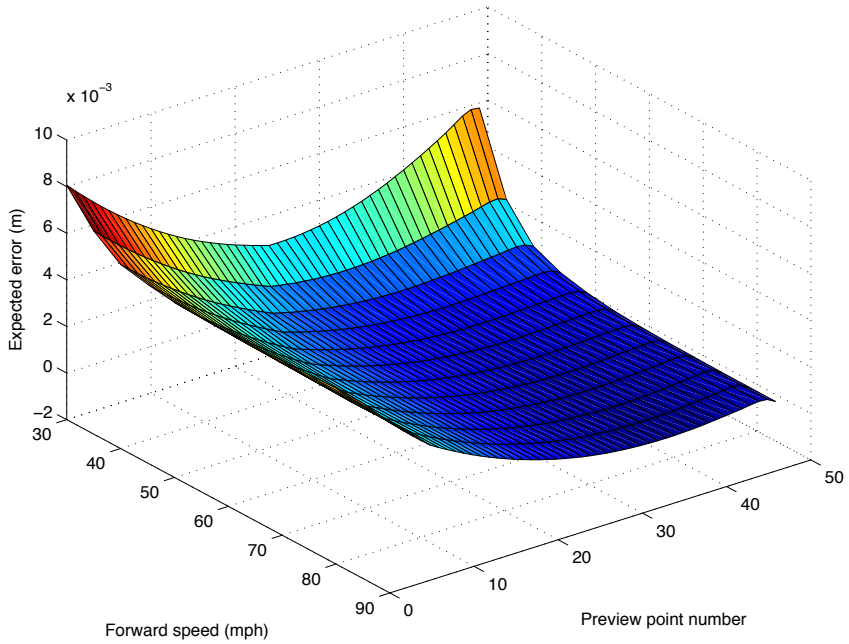

Figure 6.7. Expected value of total error in each preview point location for a straight-horizon local preview filter with $1s$ of preview when a vehicle traverses a right curve of AASHTO design while tracking the right road edge at a $1.83m$ offset

that at the end of a $1s$ preview horizon, the expected bias error in the path can be almost $1cm$, but that at higher design speeds, this bias essentially disappears. The biases in the non-previewed states, too, are worse at lower speeds than at high speeds due to the roadway designs described by eq. 6.14.

It stands to reason that because longer preview horizons require measurement and propagation of more of a turn's curvature, increasing the horizon length could, in fact, worsen bias errors. This is confirmed by Figure 6.9, which shows that errors in previewed road states are quite significant at the end of the preview horizon for low speeds. For $30mph$, the steady-state error at the end of the preview horizon

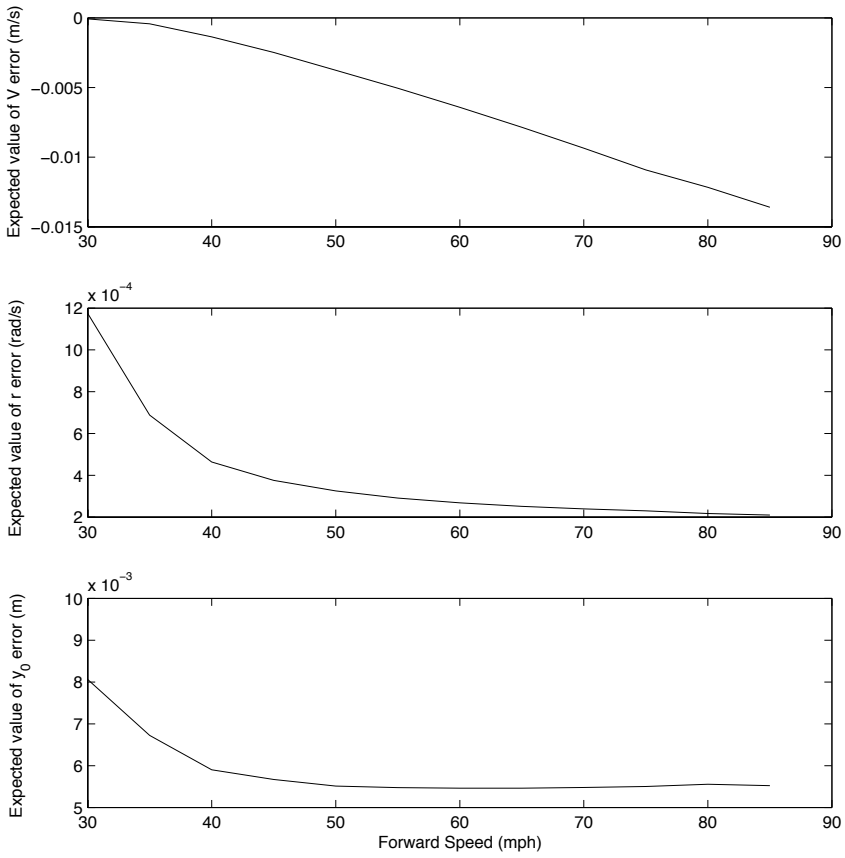

Figure 6.8. Expected value in non-previewed states for a straight-horizon local preview filter with $1s$ of preview when a vehicle traverses a right curve of AASHTO design while tracking the right road edge at a $1.83m$ offset

is over $0.12m$, even though the current lateral offset estimate as shown by Figure 6.10 is only biased by $0.2cm$. This is important information for the design of a preview filter driving an optimal preview controller in environments where such high design lateral accelerations are present, e.g. highway off-ramps.

In conclusion, while for most highway driving, the biases on steady curves caused by temporal horizon assumptions are small contributors to the overall es-

timate bias, they are exacerbated for roads with high design lateral acceleration.

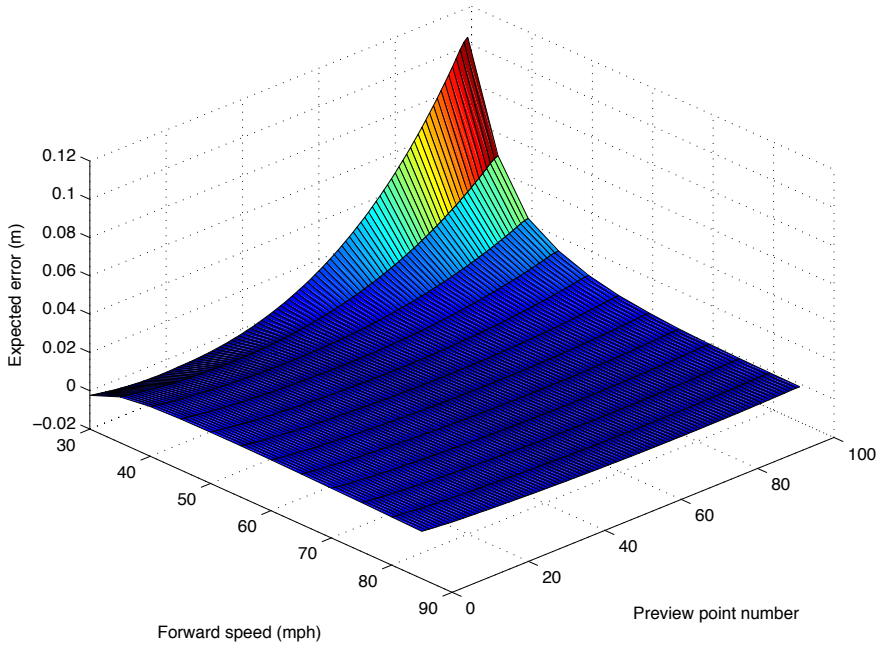

Figure 6.9. Expected value of total error in each preview point location for a straight-horizon local preview filter with $2s$ preview when a vehicle traverses a right curve of AASHTO design while tracking the right road edge at a $1.83m$ offset

6.4 Using path coordinates to mitigate temporal horizon assumption biases

In situations where these biases are expected to become significant, they can be somewhat mitigated by constructing the preview measurement vector and A_d matrix by using the mapped road geometry. In doing this, the path-coordinate S is used to look up true lever arms and preview point locations based on the actual,

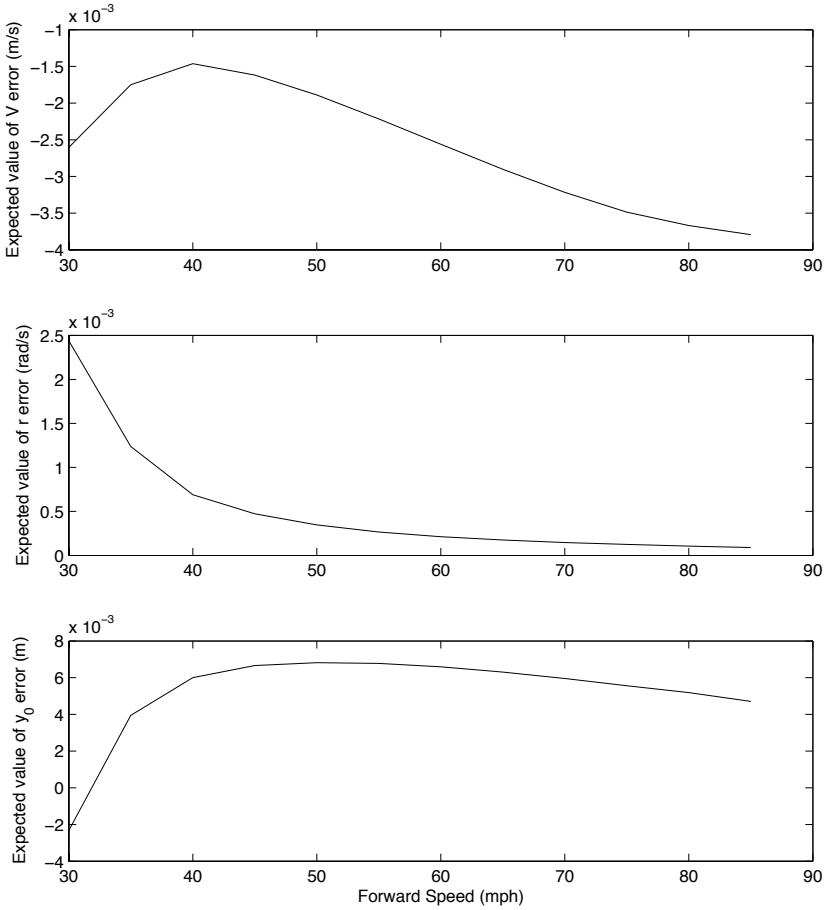

Figure 6.10. Expected value in non-previewed states for a straight-horizon local preview filter with $2s$ preview when a vehicle traverses a right curve of AASHTO design while tracking the right road edge at a $1.83m$ offset

arbitrary shape of the road marker, so that turns, straights, and transitions are accurately accounted for. This requires very good longitudinal map registration, and thus may not be able to provide significant improvement in state estimates for the physical experiments in this dissertation, where coarse longitudinal registration and other error sources dwarf those contributed by causal concerns. To

see if exact longitudinal registration combined with correct station registration is indeed helpful, consider a simualtion of a vehicle and preview estimator traversing a $50m$ radius right-hand turn at $10\frac{m}{s}$, approximating the AASHTO condition for a $25mph$ curve. The simulation was performed without corrupting measurements or inputs with noise, and the resulting errors are shown in Figure 6.11. Even though no measurement or input noise was added, some bias and periodic variation in the error are still present due to mismatch between the shift register's model and the global motion of the vehicle, due to the approximations in the shift register equations, discussed below.

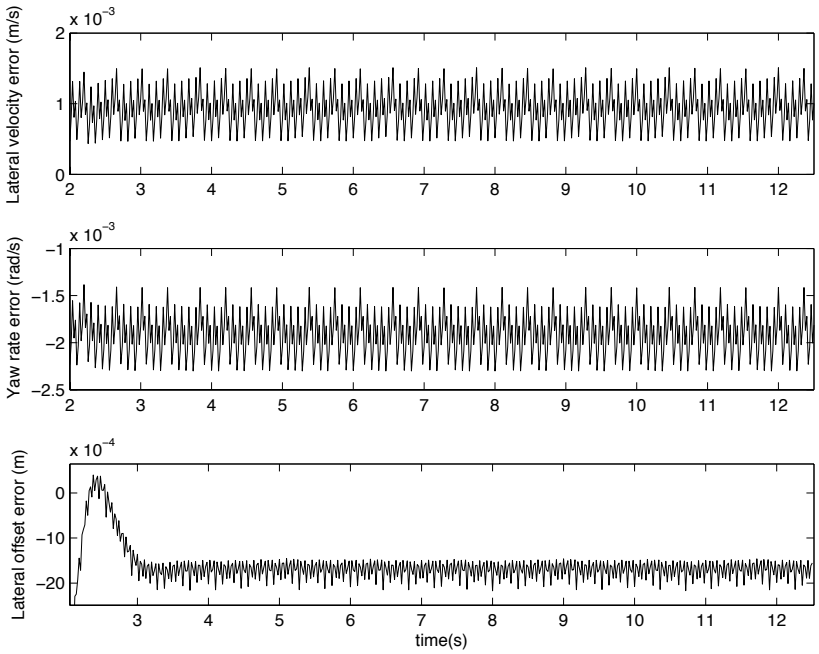

Figure 6.11. Errors in non-previewed states for a noiseless planar simulation of the path-coordinate horizon preview filter for a $50m$-radius right turn at $20\frac{m}{s}$ with perfect longitudinal map registration

This simulation was performed with a simulated LQG controller following the path at the specified offset, so error in path tracking along with the geometric approximations in the preview horizon motion model corrupt the estimates and keep them from approaching zero, or from dipping to the level predicted by Figure 6.10. This does indicate that for non-previewed states, using a path-based horizon is likely superfluous. These errors combine to produce the "noise" apparent in figure 6.10, as a result of the prediction and correction steps in the Kalman filter. But for a combined controller-estimator that needs accurate preview point estimates as well as vehicle states, it is important as well to examine this method's effect on the preview point biases, which were quite high for this scenario as shown in Figure 6.10. These are shown for the "ideal" simulation with no measurement or input noise added in Figure 6.12. Note that the "noise" seen in this plot is due to the disagreement between the shift register model's Euler approximation of road motion movement, which also prevents a complete disappearance of bias error.

While the non-previewed states do not enjoy reduction in bias error with the path-based preview horizon modification, the preview points themselves, because they are directly measured, are much more accurately portrayed. Figure 6.12 suggests that even at the preview horizon, the error in preview point position resulting from the estimator's combination of inputs and measurements is corrupted by less than $1cm$. In filter/controller designs where very accurate odometry estimates are available for longitudinal registration, this method could provide more accurate

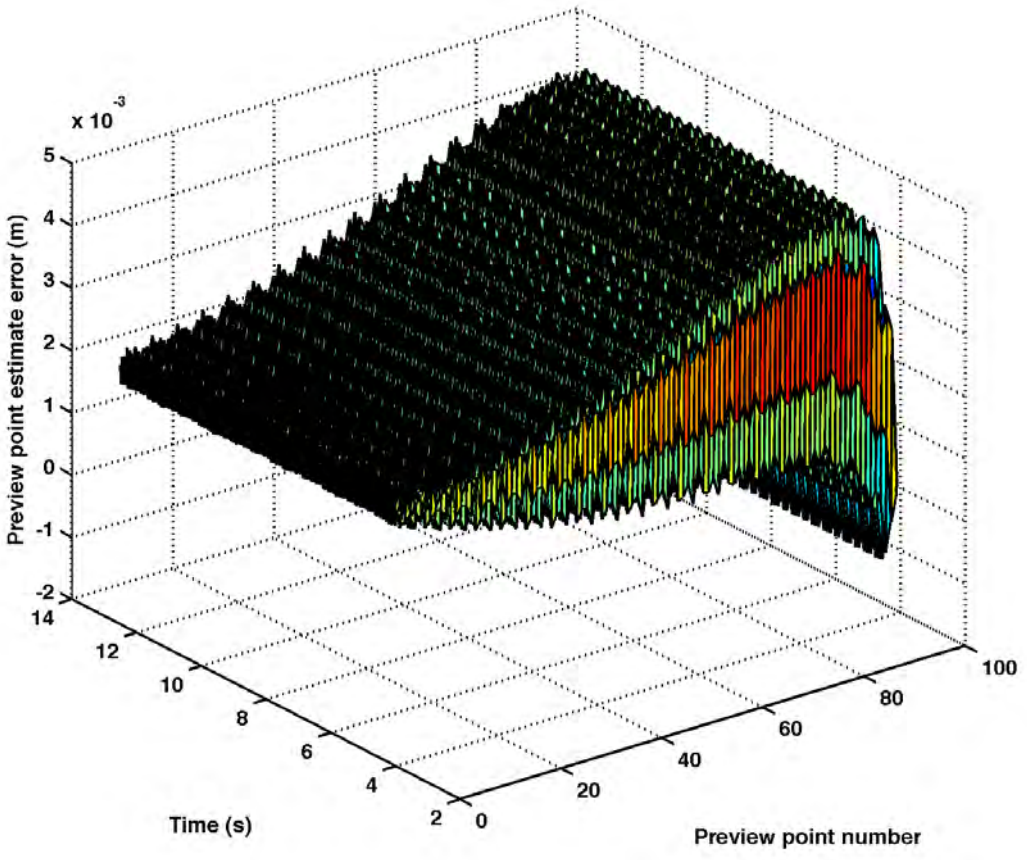

Figure 6.12. Errors in preview point states for a noiseless planar simulation of the path-coordinate horizon preview filter for a $50m$-radius right turn at $20\frac{m}{s}$

estimates than the straight preview horizon assumption. However, for the low-cost GPS system used for longitudinal map registration in this dissertation, this modification to the filter framework was deemed unnecessary.

Chapter 7

Validation of the Preview Estimation Framework for Closed-Loop Lane-Keeping

Chapter 1 reviewed a linear optimal preview controller as in [63]. More recent efforts have offered extensions to this model of human drivers to include concepts neuromuscular (actuator) dynamics [72] or control near the limit via linearization [73], but applications of this control framework to physical vehicles in a semi-autonomous capacity is scarce in the literature, with many autonomous vehicle practitioners finding success with modified single-point preview PID [59] or model-predictive control. Researchers interested in implementation of fully autonomous driving tend to build dense maps, localize and plan paths using computationally intensive methods as in [19, 32, 44, 45, 46, 47, 48, 49, 50], and attack the con-

trol task without explicitly considering the limitations of the estimation scheme on system performance. Even so, the performance of these guidance algorithms can still be quite good because state and ego-motion estimates from sensors like LIDAR scanners and differential GPS are very good. This is an excellent and adequate approach given the vehicles and sensors used, as the sensor fusion methods developed are high-quality.

However, problems arise when considering the jump from current driver-assist technologies to full autonomy. Sensors available to production vehicles are often less accurate than those employed by research-grade autonomous testbeds. Therefore, a need exists to marry the ideas of high-speed vehicle guidance using simple, linear preview-based control, like the method in [63], to an estimation framework that allows for analysis of stability margins and minimum visibility for a closed-loop perception and lateral tracking system, as well as the use of production-quality sensors. This chapter investigates the feasibility of using the local-coordinate preview estimator as a perception framework for a lane-keeping controller, including issues with the transformation of the control framework from the global to the local coordinate system, and results from four physical tests.

7.1 Design and simulation of the optimal preview controller-estimator

In [63], Robin Sharp showed that when designing the optimal preview controller (reviewed in Chapter 1) in error-coordinates, the resulting steering angle under closed-loop control, δ, is invariant under the transformation of the vehicle-road system from error (global) to body-fixed (local) coordinates. This occurs through the disappearance of the feedback gain on global yaw angle ψ and the vehicle's global y-position, since these states have no meaning in a vehicle-fixed framework. In this way, Sharp suggests that the optimal preview control law described by eq. 1.31 be modified to retain only the feedback gains on lateral velocity V and yaw rate r, followed by the preview gains on previewed road geometry as given in [63].

This idea was tested at Penn State's Test Track, where there is a long straight followed by the smaller-radius turn on the track's East end. Semi-autonomous tests are reasonably easy to perform in this area, with the only obstacles to safety being the unavoidable steep dropoff on the outside of the turn, and changing pavement quality on the transition from straight to curved road section. This lies in stark contrast with the North West end of the test track, which features a narrow bridge and constant bus traffic.

With safety in light of the steep bank on the outside of the turn in mind, simulations and tests for validation were performed at a modest $25mph$ target speed.

At this speed, the preview filter was run with $1.5s$ of preview, an experimentally determined balance between achieving non-previewed state accuracy and avoiding lane measurement corruption due to causal errors as discussed in Chapter 6, along with camera measurement errors due to elevation changes on corner entry. The theoretical variances on the non-previewed states as a function of preview horizon length are shown in Figure 7.1.

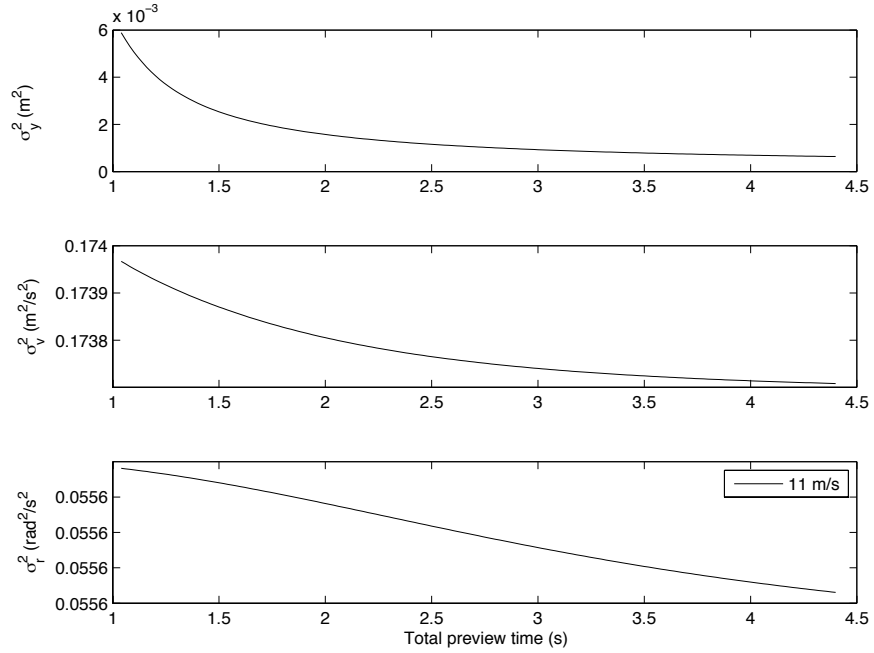

Figure 7.1. Local-coordinate preview estimator variance on non-previewed states as a function of preview time for $U = 11\frac{m}{s}$ and $dT = 0.02s$

Note that at $1.5s$ of preview, or $np = 75$ in this test, most of the achievable accuracy in non-previewed states is achieved, but not all. Also note that because of the lower speed, more preview points are obscured by the car's body, so the measurement matrix H in the preview filter is more sparse, and the occluded

preview points near the vehicle must be propagated backwards through the shift register without correction. This is an interesting conundrum, with the preview filter framework showing increasing utility at higher, rather than lower speeds. As discussed previously, this is due to a combination of the differences in spacing in the temporal horizon and the higher signal-to-noise ratio of small states like V at higher speeds.

With the preview horizon in the preview filter framework constructed with $1.5s$ of preview, the Sharp optimal preview control gains for Big Red are shown in Figure 7.2. This figure shows that for this control weighting scheme, the optimal preview gains level out to zero at the preview horizon, so that "full preview" has been achieved by the controller, and it needs no further road information for guidance as long as previewed road geometry estimates are bounded. The weight on lateral position error, q_y, was tuned specifically to achieve this result, so that the preview filter's "visible space" was not exceeded. If it was, it is possible that mapped lane features could be used for preview points needed by the controller beyond the filter's horizon.

To see qualitatively how the gains acting on the position of each preview point estimate in a controller-estimator framework relate to those points' expected statistics, consider Figure 7.3. This figure shows the Sharp optimal preview gains in black, bounded in grey by the theoretical variance of each preview point's position estimate from a steady-state Kalman filter design of the preview filter. It shows

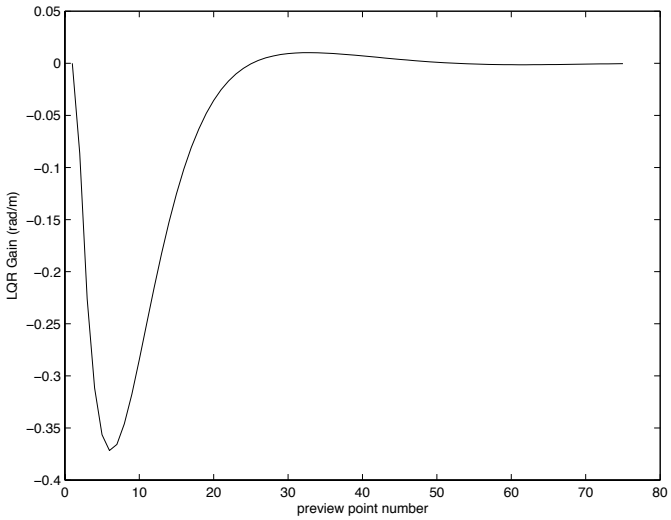

Figure 7.2. Sharp optimal preview feedback gains on each preview point for $U = 11\frac{m}{s}$,$R2 = 0.05$,$q_y = 1$,$q_\psi = 0$, and $dT = 0.02s$

very clearly how growing uncertainty in preview point measurements relates to their "importance" to the optimal preview controller. Subjectively, it makes sense that if the optimal preview controller is designed so that the preview point variances grow significantly only after the control gains on the preview points have died out, controller performance should be good. Remember, however, that errors in the previewed state estimates must be bounded to ensure a bounded steering angle via the Sharp optimal preview control law. Outliers, even at preview distances where the control gains are small, could still cause problematic closed-loop behavior. In the tests performed, this was not an issue, but in cases where large outliers are a concern, the preview filter framework could be used to check for unexpectedly large innovations in the measurement update step to determine whether outliers may be present.

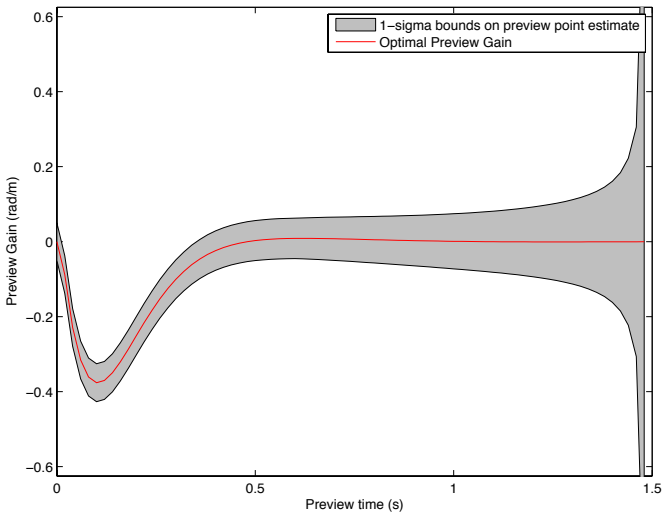

Figure 7.3. Sharp optimal preview feedback gains on each preview point with steady-state $1-\sigma$ bounds on preview point estimate error for $U = 11\frac{m}{s}$,$R2 = 0.05$,$q_y = 1$,$q_\psi = 0$, and $dT = 0.02s$

To see how the controller performs under ideal circumstances, a closed-loop simulation of Big Red following the South straight and East turn of the test track was performed. In contrast to the simulation results shown in Figure 3.4, this simulation was performed inclusive of the superelevation value e of the road, which provides an exogenous disturbance that the preview filter "knows about" through an input model, but the controller has no special provision for rejecting. The road bank angle versus the station coordinate S is shown in Figure 7.4, and was obtained from the high-fidelity LIDAR/GPS mapping procedure as outlined in Chapter 4.

With the superelevation as described by the map feeding a disturbance into the vehicle states (and estimation framework, which has access to superelevation via map data), the optimal preview controller/estimator outlined above was able

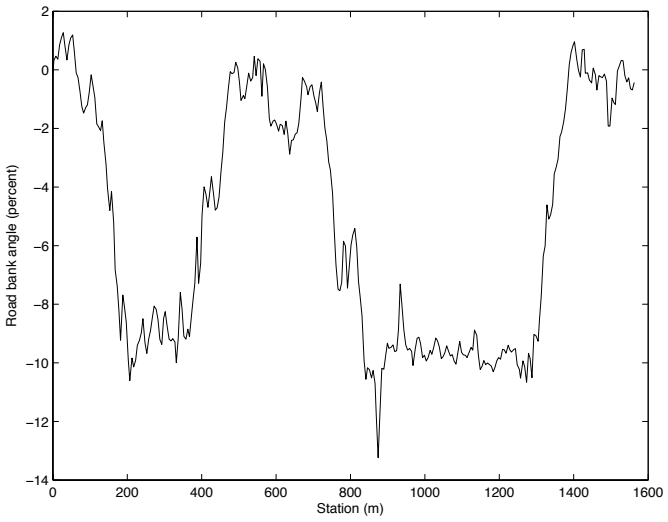

Figure 7.4. Larson Transportation Institute road bank angle (deg) as a function of path-coordinate S

to maintain extremely good path tracking in simulation. Non-previewed state estimates for the simulation are shown in Figure 7.5, where the controller was instructed to maintain a $1.625m$ lateral offset from the right lane line. The controller described exhibits the expected anticipatory behavior, and maintains this offset very well, even in the face of the superelevation disturbance.

One interesting thing to note about the simulation is that the the lateral velocity V and the yaw rate r have the same sign, even though the simulated Big Red is traveling above its "zero-crossing" speed, which, in vehicle dynamics terminology, basically indicates that the sideslip angle β should be negative, resulting in a negative V. The lateral velocity V is not negative, however, because of the road superelevation. This is important information, and indicates that in this case, superelevation is important to the estimator. To show this effect on estimator per-

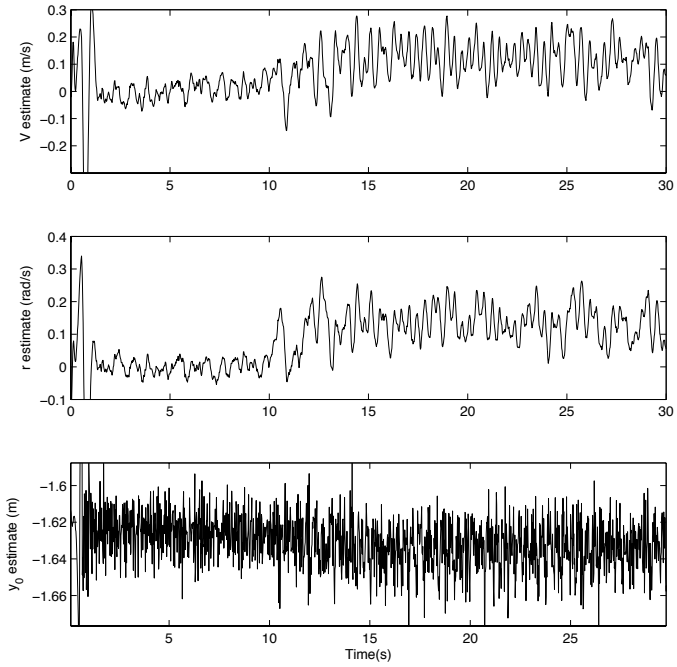

Figure 7.5. Simulation of $11\frac{m}{s}$ traversal of test track East turn under preview control/estimation, where controller attempted to maintain a $1.625m$ offset from the right lane edge

formance, consider Figure 7.6. This figure shows filter state estimates versus true states for a test case wherein the preview filter had no access to the superelevation disturbance. During that simulation, the preview controller was able to reject the superelevation disturbance with only a $10cm$ bias in tracking performance, but lateral velocity estimates were exceptionally noisy, and biased from truth.

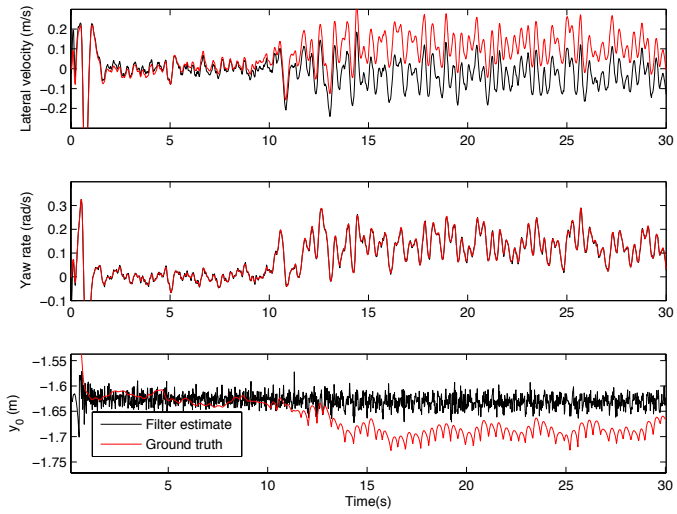

Figure 7.6. Simulation of $11\frac{m}{s}$ traversal of test track East turn under preview control/estimation, with assumed zero superelevation in the preview filter.

7.2 Physical testing of the optimal preview controller-estimator

As mentioned in the preceding chapters, the Larson Transportation Institute Test Track is home to "Big Red," a retrofitted 1989 GMC2500 4x4 often used for teleoperation during vehicle dynamics tests where human operation would be far too dangerous. As such, it is outfitted with an Aerotech rotary stage that acts as a servomotor for controlling the steering, and linear motors to control throttle and brake. All sensors are ROS-enabled, so their control via preview control and estimation was straightforward.

For the four runs shown in this section, forward speed was the responsibility of the "backup driver," who monitored the test for safety, and possessed a "dead

man's switch" to disable the motor and halt the test as soon as his/her hands left a two-handed grip on a repurposed gaming controller. The backup driver was responsible for maintaining a $25mph$ speed throughout the test while the optimal preview estimator/controller steered the vehicle around the same turn simulated in the preceding section. A view from inside of the cockpit during one of the test runs is shown in Figure 7.7.

Figure 7.7. First-person view of $11\frac{m}{s}$ traversal of test track East turn under preview control/estimation

The tests were each run with the same control law as shown in Figure 7.2, and as before, the coarse longitudinal map registration for finding $y_{r,i}$ was accomplished with a low-cost $10Hz$ GPS module on the ROS-enabled Beaglebone, and measurements of yaw rate came from the IMU on the same. Aside from the expensive ($100k) sensor suite needed for map generation, the entire cost of the sensing equipment needed for the tests, including the PointGrey camera, was roughly $230. Preview filter estimated states for four separate traversals of this same area on the track are shown in Figure 7.8.

The temporal variation in the $\hat{y_0}$ position may seem troubling when observed as

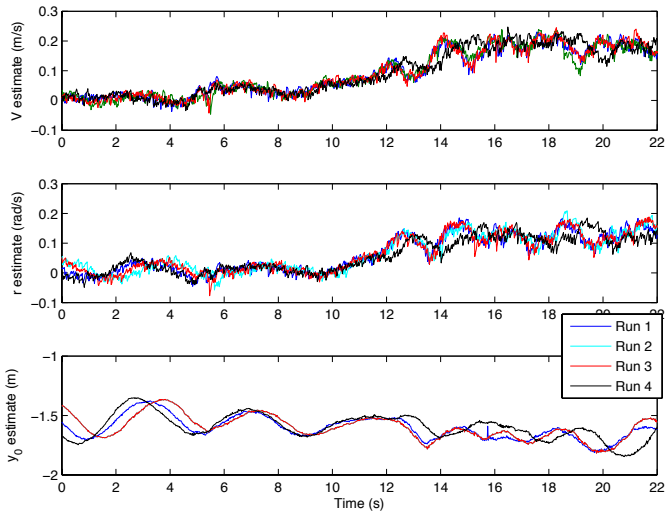

Figure 7.8. Experimental states during $11\frac{m}{s}$ traversal of test track East turn under preview control/estimation versus time

a time series, since it suggests that the controller's performance was highly initial-condition dependent. It is indeed true that the preview controller-estimator is essentially a Linear Quadratic Gaussian (LQG) controller, and these are notorious for having uncertain robustness. However, several possible sources of this bias were present:

1. The backup driver was responsible for longitudinal speed control, certainly not maintaining a perfect $11\frac{m}{s}$.

2. There are errors as discussed in Chapter 6 with the known approximations in the vehicle dynamics and shift register.

3. No actuator dynamics were considered.

4. The test featured nearly constant violation of the planar road assumption

due to undulations in the test track surface and the banked turn.

Because of this, and despite the fact that the vehicle was able to maintain a satisfactory position in the lane, the oscillations in the vehicle position should be the subject of further study. In fact, work is ongoing to quantify robustness of the optimal preview LQG under certain assumptions, but is beyond the scope of this dissertation.

Because the test vehicle was driven with slightly different initial conditions in each traversal, it is wise to consider the variation in states during the traversals as a function of the S-coordinate rather than time. This will expose, at least to a degree, whether road-specific disturbances are contributing to the oscillations observed. This is shown in Figure 7.9. When the runs are compared with each other

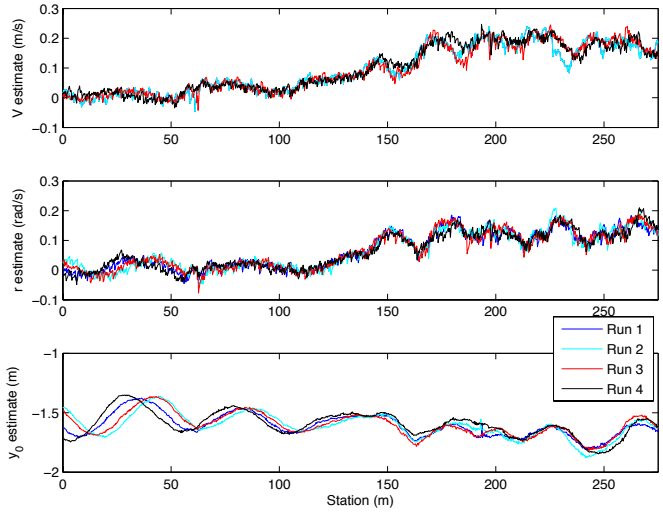

Figure 7.9. Experimental states during $11\frac{m}{s}$ traversal of test track East turn under preview control/estimation versus station

this way, it becomes clear that, apart from some initial errors as each estimator

converges, the oscillations in vehicle lateral offset from the desired position of $1.625m$ correlate strongly with position on the track, suggesting that they are the result of map errors. These errors could manifest in the filtered estimate of road position due to violation of the lane measurement node's flat road assumption, or possibly the result of the controller attempting to respond to small lateral lane position variations unsuccessfully due to deadband or rod-end compliance in the steering rack, which is quite significant for Big Red at 24 years old.

Even with the increase in variation about the reference offset of $1.625m$ from simulation to experiment, however, it is clear that the preview estimator offers a reliable source of vehicle and road states to the optimal preview controller, and allows the vehicle to navigate an appreciably sharp turn with heavy banking. This paves the way for detailed investigations into the true nature of the relationship between previewed state perception and path tracking for semi-autonomous lane-keeping systems.

Chapter 8

Conclusions and Future Work

This dissertation looked at a novel paradigm to use temporally previewed information to obtain sub-lane-level lateral positioning accuracy, using a combination of a map and other sensors to allow the estimation framework to track its environment in a control-relevant, receding horizon. First, two realizations of the map-based preview estimation scheme were derived, each with its own particular benefits. Each provided a-priori predictions of estimate errors for a given lookahead distance. These estimation frameworks' theoretical simplicity and flexibility improved upon the point-based limitations of a SLAM approach for lane finding and vehicle motion recovery, the rigid map registration requirements of traditional map-based localization, and the parametric fitting assumptions of older lane-based vehicle state estimators. Treating the mapped road location at the edge of the preview horizon as a model input allowed these estimators to correct lane goemetry where measurements were available. Lane measurements simultaneously improved

lateral vehicle state estimates, and using current vehicle states to propagate "unseen" but mapped lane features in occluded regions of the temporal shift register.

After deriving each of these estimation frameworks, the body-fixed coordinate version of the estimator was tested in two locations and three separate test events using maps generated from either high-accuracy LIDAR data or coarse Google Earth breadcrumbs. Google Earth data were determined not to provide sufficient global accuracy in lateral vehicle position, but did not harm local offset estimates significantly. LIDAR-generated lane line maps, in conjunction with the preview filter, offered between $3cm$ (straight-line driving) and $17cm$ $1-\sigma$ error bounds with very small bias errors in lateral position. The accuracy of the estimator clearly depended on the degree to which vehicle dynamics were excited and the degree to which the path at hand was followed in accordance with the temporal preview horizon assumption.

Then, to address the error sources unique to the temporal preview estimation paradigm, the special case of constant-radius turning as a source of bias error in the preview filter was considered. Bias values resulting from the causal violation of the assumed temporal preview horizon for a range of expected driving conditions suggested that these effects are larger on roads with higher design lateral accelerations (lower design speeds), but that biases on non-previewed states are small. For a preview filter designed as in Chapter 5, with $20m$ of preview and a $20\frac{m}{s}$ forward speed, and a road with a $45mph$ design speed like the Larson Institute Test

Table 8.1. Preview Estimation Error Budget

State	σ_{exp}	σ_{pred}	$\bar{e}_{causal}$
$\hat{y}_0(m)$	0.17	0.058	-0.0025
$\hat{V}(\frac{m}{s})$	0.11	.12	0.00025
$\hat{r}(\frac{rad}{s})$	0.02	0.08	0.0057

Track, the predicted accuracy, constant-curve bias, and actual accuracy during the aggressive one-lap traversal are shown in Table 8.1.

In Table 8.1, $\bar{e}_{causal}$ is the expected causal bias error in each state for the track's turn radii as examined in Chapter 6. From this table, it is clear that for non-previewed states V and r, the preview filter framework as designed performs similarly to predictions using a steady-state Kalman filter analysis like the one that generated Figure 5.14. For the test track's constant-radius curves, biases expected from violation of causality in the temporal preview horizon assumption are very small compared to other error sources. Unmodeled measurement error sources in the camera, such as roll and pitch motion of the vehicle (which will add input-correlated bias), non-planarity of the road on turn entry/exit (which will add location-dependent measurement bias), and errors in the preview horizon resulting from the euler model of relative road motion and the non-constant forward speed U are likely at fault for the disparity in predicted versus observed error in y_0. These error sources are not discussed in detail in this dissertation, but are targets of ongoing work. Recall, too, that the preview filter outperformed estimated error bounds on y_0 for a straight-line traversal of the test track's North straightaway,

where it achieved a 1-σ error of $3cm$ when compared to the ground truth DGPS system. This test also suggested, via Figure 5.16, that even with the preview filter's experimental accuracy under significant lateral vehicle dynamics excitation, $17cm$ accuracy is on the same order as a human's lane-keeping error when driving a straight portion of road.

Finally, the preceding chapter showed, through simulation and physical experiment, that it is possible to design a perception and lateral state control system with explicit knowledge of how the length of the spatiotemporal horizon affects both the perception and control tasks individually, so that they may be designed to work in concert. Additionally, the experimental results have shown that it is indeed possible to obtain lateral states sufficient for lateral control with very low-cost sensors.

8.1 Further developments of the preview filter framework

However, by no means is the preview filter concept developed here complete as delivered. This dissertation's primary goal was to develop the preview estimation framework, but many extensions, improvements, and further studies are possible. Any number of additional sensors, measuring any of the states or linear combinations thereof, could be added with appropriate noise models to improve state

estimates even further. The measurement of multiple lane features is also a possibility, especially if the map specifies what each of these features should look like, and the lateral offset from the lane center of each one. In this way, the mathematical dimension of the filter could be maintained with one shift register, and the only change would be the addition of rows to the measurement matrix H. In this way, the measurements of each lane feature would appear as repeated measurements of the lane center via map-specified offsets. These repeated measurements would likely reduce the variance in the state estimates in a very predictable manner given by the Central Limit Theorem.

The use of map information to relax the planar road assumption used in this dissertation for forward-facing monocular cameras is also possible. Using mapped elevation profiles for a road in combination with the coarse longitudinal map registration procedure could provide the lane measurement system with a 1:1 lookup of scan-lines for every preview distance, even for roads with non-planar geometry. This could aid in the estimation process while avoiding bootstrapping issues, since the longitudinal map registration procedure is decoupled from the lateral states corrected by the preview filter. It would also allow the framework to avoid looking for road features over the crest of a hill, for instance.

In an application where speeds may vary slowly, such as an Adaptive Cruise Control (ACC) and autonomous lane-keeping system for a vehicle on the highway, a compromise in preview filter shift register size and accuracy concerns would be

necessary in maintaining a constant number of previewed features, np. Statistical analysis of visibility on roads is possible, and is a candidate for inclusion in the map. Additionally, it is possible for a perception/control system designer to use multiple cameras for preview point measurement (such as a side-view camera) to make lane measurements possible at both close and far ranges. In [37], Dickmanns recommends the use of multiple cameras with differing zoom levels to aid in perception on different scales. This is also a wonderful idea for the preview filter framework, as multiple lane features could be tracked at different lookahead scales, and is planned for future work.

Finally, because the physically implemented preview filter algorithm as presented in this document was programmed in Python for simplicity and rapidity of development, it may not be especially suited for migration in whole to a low-cost single-board computer like the Beaglebone. With that said, the algorithm itself is simple, and requires only a finite dimension in the system matrices, so it may be possible to implement it entirely on a very low-cost ARM-powered computer with translation to C++. The low-cost computer already handles all of the requisite sensor data acquisition, so migration would be of the filter pieces only. This would allow for a self-contained perception system with a very low price tag, and exceptional accuracy.

8.2 Relating visibility and driveability

As the preceding chapter hinted, the use of the preview-based LQR in practice, with output feedback, requires much more study before it could be implemented on a production vehicle. Because sensor selection and preview horizon length are tunable parameters, as are the LQR control weights in the preview controller, it is likely possible to develop concrete guidelines for the deployment of coupled perception and lateral control systems using the preview filter in conjunction with an optimal preview controller, in a manner similar to the analysis presented in [43] for a different class of vision-based controller/estimators.

A thorough robustness analysis could uncover issues that bound the types of sensors that should be used in a closed-loop lane-keeping system. Additionally, this same sort of analysis could be very useful for an implemented system with a set amount of visibility, where a driver-assist or semi-autonomous system would have to make decisions about whether automated steering of a vehicle is feasible for a given driving situation. The preview filter framework, with its ability to link preview sensor noise models to bounds on state estimate accuracy, offers the system a way to determine this. Work in this area is ongoing.

Bibliography

[1] Sheridan, T. B. (1966) "Three models of preview control," *Human Factors in Electronics, IEEE Transactions on*, (2), pp. 91–102.

[2] Dean, A. and S. Brennan (2009) "Terrain-based road vehicle localization on multi-lane highways," in *American Control Conference, 2009. ACC'09.*, IEEE, pp. 707–712.

[3] Pacejka, H. (2005) *Tyre and vehicle dynamics*, Butterworth-Heinemann.

[4] Milliken, W. F., D. L. Milliken, and L. D. Metz (1995) *Race car vehicle dynamics*, vol. 146, Sae International Warrendale, PA.

[5] Karnopp, D. (2004) *Vehicle stability*, vol. 171, CRC Press.

[6] Brown, A., S. Lapapong, K. Swanson, and S. Brennan (2011) "Modeling of Planar Vehicle Dynamics During Gradual Tire Deflation," in *22nd International Symposium on Dynamics of Vehicles on Roads and Tracks, Manchester, UK*.

[7] DeCarlo, R. A. (1989) *Linear systems: A state variable approach with numerical implementation*, Prentice-Hall, Inc.

[8] Bevly, D., J. Ryu, and J. Gerdes (2006) "Integrating INS sensors with GPS measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness," *Intelligent Transportation Systems, IEEE Transactions on*, **7**(4), pp. 483–493.

[9] Daily, R. and D. Bevly (2004) "The use of GPS for vehicle stability control systems," *Industrial Electronics, IEEE Transactions on*, **51**(2), pp. 270–277.

[10] Kiencke, U. and A. Daifl (1997) "Observation of lateral vehicle dynamics," *Control Engineering Practice*, **5**(8), pp. 1145–1150.

[11] Sebsadji, Y., S. Glaser, S. Mammar, and J. Dakhlallah (2008) "Road slope and vehicle dynamics estimation," in *American Control Conference, 2008*, IEEE, pp. 4603–4608.

[12] Ray, L. (1995) "Nonlinear state and tire force estimation for advanced vehicle control," *Control Systems Technology, IEEE Transactions on*, **3**(1), pp. 117–124.

[13] Ray, L. R. (1997) "Nonlinear tire force estimation and road friction identification: simulation and experiments," *Automatica*, **33**(10), pp. 1819–1833.

[14] Mudaliar, N., D. LeBlanc, and H. Peng (2004) "Linear estimator for road departure warning systems," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 3, IEEE, pp. 2104–2109.

[15] Simon, D. (2006) *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, Wiley-Interscience.

[16] Jerath, K. and S. Brennan (2011) "GPS-Free Terrain-Based Vehicle Tracking Performance as a Function of Inertial Sensor Characteristics," in *Proceedings of the 2011 Dynamic Systems and Control Conference, Arllington, VA, USA*.

[17] Dean, A., P. Vemulapalli, and S. Brennan (2008) "Highway evaluation of terrain-aided localization using particle filters," in *Proceedings. 2008 Dynamic Systems Control Conference*, pp. 20–22.

[18] Dean, A., R. Martini, and S. Brennan (2011) "Terrain-based road vehicle localisation using particle filters," *Vehicle System Dynamics*, **49**(8), pp. 1209–1223.

[19] Levinson, J., M. Montemerlo, and S. Thrun (2007) "Map-based precision vehicle localization in urban environments," in *Proceedings of the Robotics: Science and Systems Conference*, Citeseer.

[20] Gupta, V. (2010) *Vehicle localization using low-accuracy GPS, IMU and map-aided vision*, Ph.D. thesis, The Pennsylvania State University.

[21] Pink, O. (2008) "Visual map matching and localization using a global feature map," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, IEEE, pp. 1–7.

[22] Noda, M., T. Takahashi, D. Deguchi, I. Ide, H. Murase, Y. Kojima, and T. Naito (2011) "Vehicle ego-localization by matching in-vehicle camera images to an aerial image," in *Computer Vision–ACCV 2010 Workshops*, Springer, pp. 163–173.

[23] Lee, Y., W. Yu, J. Cho, et al. (2011) "Adaptive localization for mobile robots in urban environments using low-cost sensors and enhanced topological map," in *Advanced Robotics (ICAR), 2011 15th International Conference on*, IEEE, pp. 569–575.

[24] Dey, N., R. Mohanty, and K. Chugh (2012) "On Road Navigation System Using Spatial and Motion Image Processing for Automatic Navigation System," *International Journal of Innovation, Management and Technology*, **3**(1), pp. 80–83.

[25] Thrun, S., W. Burgard, D. Fox, et al. (2005) *Probabilistic robotics*, vol. 1, MIT press Cambridge.

[26] Cheeseman, R., R. Smith, and M. Self (1987) "A stochastic map for uncertain spatial relationships," in *Proceedings of 4th International Symposium on Robotics Research*.

[27] Smith, P., I. Reid, and A. Davison (2006) "Real-time monocular SLAM with straight lines," in *British Machine Vision Conference*, vol. 1, pp. 17–26.

[28] Milford, M. J. and G. F. Wyeth (2008) "Single camera vision-only SLAM on a suburban road network," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, IEEE, pp. 3684–3689.

[29] Clemente, L. A., A. J. Davison, I. Reid, J. Neira, and J. D. Tardós (2007) "Mapping large loops with a single hand-held camera," in *Robotics: Science and Systems*.

[30] Durrant-Whyte, H. and T. Bailey (2006) "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *Robotics and Automation Magazine*, **13**(99), p. 80.

[31] Montemerlo, M., S. Thrun, D. Koller, and B. Wegbreit (2002) "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the National conference on Artificial Intelligence*, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, pp. 593–598.

[32] Montemerlo, M., J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, et al. (2008) "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, **25**(9), pp. 569–597.

[33] McCall, J. and M. Trivedi (2006) "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *Intelligent Transportation Systems, IEEE Transactions on*, **7**(1), pp. 20–37.

[34] Bertozzi, M., A. Broggi, and A. Fascioli (2000) "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous systems*, **32**(1), pp. 1–16.

[35] Cheng, H. (2011) "Road Detection and Tracking," *Autonomous Intelligent Vehicles*, pp. 33–59.

[36] Kastrinaki, V., M. Zervakis, and K. Kalaitzakis (2003) "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, **21**(4), pp. 359–381.

[37] Dickmanns, E. and H. Wünsche (2007) *Dynamic vision for perception and control of motion*, Springer.

[38] Bertozzi, M. and A. Broggi (1998) "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, **7**(1), pp. 62–81.

[39] Southall, B. and C. Taylor (2001) "Stochastic road shape estimation," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, IEEE, pp. 205–212.

[40] Taylor, C., J. Košecká, R. Blasi, and J. Malik (1999) "A comparative study of vision-based lateral control strategies for autonomous highway driving," *The International Journal of Robotics Research*, **18**(5), pp. 442–453.

[41] Redmill, K., S. Upadhya, A. Krishnamurthy, and U. Ozguner (2001) "A lane tracking system for intelligent vehicle applications," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, IEEE, pp. 273–279.

[42] Hsu, J.-C., W.-L. Chen, R.-H. Lin, and E. C. Yeh (1997) "Estimations of previewed road curvatures and vehicular motion by a vision-based data fusion scheme," *Machine vision and applications*, **9**(4), pp. 179–192.

[43] Hsu, J.-C. and M. Tomizuka (1998) "Analyses of vision-based lateral control for automated highway system," *Vehicle system dynamics*, **30**(5), pp. 345–373.

[44] Thrun, S., M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. (2006) "Stanley: The robot that won the DARPA Grand Challenge," *Journal of field Robotics*, **23**(9), pp. 661–692.

[45] Chen, Q., U. Ozguner, and K. Redmill (2004) "Ohio state university at the 2004 darpa grand challenge: developing a completely autonomous vehicle," *Intelligent Systems, IEEE*, **19**(5), pp. 8–11.

[46] Ozguner, U., C. Stiller, and K. Redmill (2007) "Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience," *Proceedings of the IEEE*, **95**(2), pp. 397–412.

[47] Ozguner, U., K. A. Redmill, and A. Broggi (2004) "Team TerraMax and the DARPA grand challenge: a general overview," in *Intelligent Vehicles Symposium, 2004 IEEE*, IEEE, pp. 232–237.

[48] Behringer, R., S. Sundareswaran, B. Gregory, R. Elsley, B. Addison, W. Guthmiller, R. Daily, and D. Bevly (2004) "The DARPA grand challenge-development of an autonomous vehicle," in *Intelligent Vehicles Symposium, 2004 IEEE*, IEEE, pp. 226–231.

[49] Kammel, S., J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. v. Hundelshausen, et al. (2008) "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, **25**(9), pp. 615–639.

[50] Patz, B. J., Y. Papelis, R. Pillat, G. Stein, and D. Harper (2008) "A practical approach to robotic design for the darpa urban challenge," *Journal of Field Robotics*, **25**(8), pp. 528–566.

[51] MacAdam, C. C. (1981) "Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving," *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-11**(6), pp. 393–399.

[52] Shladover, S., C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown (1991) "Automated vehicle control developments in the PATH program," *Vehicular Technology, IEEE Transactions on*, **40**(1), pp. 114–130.

[53] Peng, H. and M. Tomizuka (1991) "Preview control for vehicle lateral guidance in highway automation," in *American Control Conference, 1991*, IEEE, pp. 3090–3095.

[54] Tan, H., J. Guldner, S. Patwardhan, C. Chen, and B. Bougler (1999) "Development of an automated steering vehicle based on roadway magnets-a case study of mechatronic system design," *Mechatronics, IEEE/ASME Transactions on*, **4**(3), pp. 258–272.

[55] Guldner, J., H. Tan, S. Patwardhan, C. D. of Transportation, B. I. o. T. S. University of California, P. for Advanced Transit, and H. (Calif.) (1997) *On fundamental issues of vehicle steering control for highway automation*, California PATH Program, Institute of Transportation Studies, University of California, Berkeley.

[56] Guldner, J., H. Tan, and S. Patwardhan (1997) "Study of design directions for lateral vehicle control," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 5, IEEE, pp. 4732–4737.

[57] Macadam, C. (2003) "Understanding and modeling the human driver," *Vehicle System Dynamics*, **40**(1-3), pp. 101–134.

[58] Cole, D., A. Pick, and A. Odhams (2006) "Predictive and linear quadratic methods for potential application to modelling driver steering control," *Vehicle System Dynamics*, **44**(3), pp. 259–284.

[59] Kang, J., R. Y. Hindiyeh, S. Moon, J. C. Gerdes, and K. Yi (2008) "Design and testing of a controller for autonomous vehicle path tracking using GPS/INS sensors," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, pp. 6–11.

[60] Falcone, P., F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat (2007) "Predictive active steering control for autonomous vehicle systems," *Control Systems Technology, IEEE Transactions on*, **15**(3), pp. 566–580.

[61] Keviczky, T., P. Falcone, F. Borrelli, J. Asgari, and D. Hrovat (2006) "Predictive control approach to autonomous vehicle steering," in *American Control Conference, 2006*, IEEE, pp. 6–pp.

[62] Nevistić, V. and J. A. Primbs (1997) "Finite receding horizon linear quadratic control: a unifying theory for stability and performance analysis," URL http://authors.library.caltech.edu/28065/.

[63] Sharp, R. and V. Valtetsiotis (2001) "Optimal preview car steering control," *Vehicle System Dynamics*, **35**, pp. 101–117.

[64] Sharp, R. S. (2007) "Motorcycle Steering Control by Road Preview," *Jurnal of Dynamic Systems, Measurement, and Control*, **129**, pp. 373–381.

[65] Sharp, R. S. and H. Peng (2011) "Vehicle Dynamics Applications of Optimal Control Theory," *Vehicle System Dynamics*, **49**(7), pp. 1073–1111.

[66] Pick, A. and D. Cole (2004) "Neuromuscular dynamics and the vehicle steering task," *Vehicle System Dynamics*, **41**, pp. 182–191.

[67] (2013), "ROS.org," .
URL `http://www.ros.org/`

[68] (2013), "NHTSA Vehicle Parameter Database," .
URL `http://www-nrd.nhtsa.dot.gov/database/aspx/vehdb/queryvehicle.aspx`

[69] Lapapong, S. (2010) *Vehicle rollover prediction for banked surfaces*, Ph.D. thesis, The Pennsylvania State University.

[70] Lapapong, S., A. A. Brown, and S. N. Brennan (2011) "Frequency characteristics and explanation of notches seen in frequency responses of vehicles," in *Proceedings of the TSME International Conference on Mechanical Engineering. October, Krabi, Thailand*.

[71] Aashto, A. (2001) "Policy on Geometric Design of Highways and Streets," *American Association of State Highway and Transportation Officials, Washington, DC*, **1**, p. 990.

[72] Pick, A. J. and D. J. Cole (2008) "A Mathematical Model of Driver Steering Control Including Neuromuscular Dynamics," *Journal of Dynamic Systems, Measurement, and Control*, **130**.

[73] Thommyppillai, M., S. Evangelou, and R. Sharp (2009) "Car driving at the limit by adaptive linear optimal preview control," *Vehicle System Dynamics*, **47**(12), pp. 1535–1550.

Vita

Alexander Allen Brown

Education:

Ph.D. in Mechanical Engineering, *2013*
The Pennsylvania State University, University Park, PA, USA

M.S. in Mechanical Engineering, *2012*
The Pennsylvania State University, University Park, PA, USA

B.S. in Mechanical Engineering, *2008*
The Pennsylvania State University, University Park, PA, USA

Professional Experience:

Graduate Teaching Assistant
Department of Mechanical and Nuclear Engineering
The Pennsylvania State University, University Park, PA
January 2012–May 2013

Graduate Research Assistant
Department of Mechanical and Nuclear Engineering
The Pennsylvania State University, University Park, PA
June 2008–May 2013

Summer Engineering Intern
MTD Products, Inc., Valley City, OH
June 2006–August 2006, June 2007–August 2007

Undergraduate Teaching Assistant
Undergraduate Machine Shop
The Pennsylvania State University, University Park, PA
August 2006–May 2008