Title
Cooperative Multiple-Sensor Fusion for Automated Vehicle Control

Authors
Hedrick, J. K.
Jang, J.
Potier, A.

Publication Date
2004-04-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Cooperative Multiple-Sensor Fusion for Automated Vehicle Control

J. K. Hedrick, J. Jang, A. Potier
University of California, Berkeley

California PATH Research Report
UCB-ITS-PRR-2004-10

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

Final Report for Task Order 4243

April 2004

ISSN 1055-1425

Cooperative Multiple-Sensor Fusion for Automated Vehicle Control

Final Report

PATH Task Order 4243

J. K. Hedrick

J. Jang

A. Potier

Mechanical Engineering Department

University of California at Berkeley

Berkeley, CA 94720

Abstract

The number and quality of sensors available for both on-board vehicle and infrastructure-based sensing is increasing while the cost of these sensors is rapidly decreasing. On-board vehicle sensors can be utilized for both individual vehicle safety as well as automated vehicle control. It is becoming imperative that "fusion" techniques be developed, i.e., methods to combine the wide variety of sensors available so that reliable and accurate information can be obtained even though individual sensors may fail or become extremely noisy. The process of sensor fusion combines multiple sensor measurements in order to provide an accurate and coherent view of the vehicle and its environment.

Sensor fusion has recently been studied in the Advanced Transportation Management and Information System (ATMIS) for the California Department of Transportation. The objective of this project with the Partners for Automated Transit and Highways (PATH) group was to investigate sensor fusion methods for automated vehicle control. The focus of the first year was to complete the development and implementation of a modular sensor fusion architecture for longitudinal vehicle control in platoon formation. The sensor models, closed-loop vehicle models, and mathematical framework were then programmed and simulated with the use of MATLAB/Simulink in order to analyze the performance of the proposed system.

Longitudinal control of vehicles in leader-follower mode is simple to implement and requires small computational ressources. Lateral control, however, is more difficult to achieve due to the non-linear equations of motion goverened by the vehicle dynamics. The PDA method can then be applied to the expanded, three-dimensional vehicle model to obtain fused estimates for the lateral and angular states. Additional sensors, such as magnetic road markers embedded in certain freeways, DGPS, and accelerameters, can also be used to achieve lateral control. Sensor fault diagnostic systems are discussed for future investigation.

To increase accuracy of target tracking, IMM algorithm was applied to three cases, i.e. longitudinally moving target tracking under no measurement noise, longitudinally moving target tracking under measurement noise, and longitudinal / lateral moving target tracking under measurement noise. The IMM algorithm reduces the RMS position error by more than 65 % as well as indicates the current mode of a target with probability. The final estimate which is fused using the measurements from three sensors, two radars and one lidar, reduces the RMS position error by 80%. Even though the degree of error reduction depends on the choice of the process noise matrix that is determined by the maximum acceleration of the vehicle, the IMM algorithm and sensor fusion technique would improve the target tracking performance.

Contents

List of Figures

Chapter 1

Introduction

The number and quality of sensors available for both on-board vehicle and infrastructure-based sensing is increasing while the cost of these sensors is rapidly decreasing. On-board vehicle sensors can be utilized for both individual vehicle safety as well as automated vehicle control. It is becoming imperative that "fusion" techniques be developed, i.e., methods to combine the wide variety of sensors available so that reliable and accurate information can be obtained even though individual sensors may fail or become extremely noisy. The process of sensor fusion combines multiple sensor measurements in order to provide an accurate and coherent view of the vehicle and its environment.

Sensor fusion has been widely used in applications such as robotics, geological surveying, traffic management, and more recently in the Advanced Transportation Management and Information System (ATMIS). The initial proposal that was submitted to the Partners for Automated Transit and Highways (PATH) group in 2002 anticipated a project duration of two years. The focus of the first year was to complete the development and implementation of a modular sensor fusion architecture for longitudinal vehicle control in platoon formation. The sensor models, closed-loop vehicle models, and mathematical framework were then programmed and simulated with the use of MATLAB/Simulink in order to analyze the performance of the proposed system.

The process of sensor fusion can be functionally represented by three tasks: data alignment, data association, and data fusion. Inherent in all aspects of the fusion process is the use of models describing the sensors as well as the vehicles and the environment in which they operate. The first task, data alignment, entails the incorporation of all available sensor data and

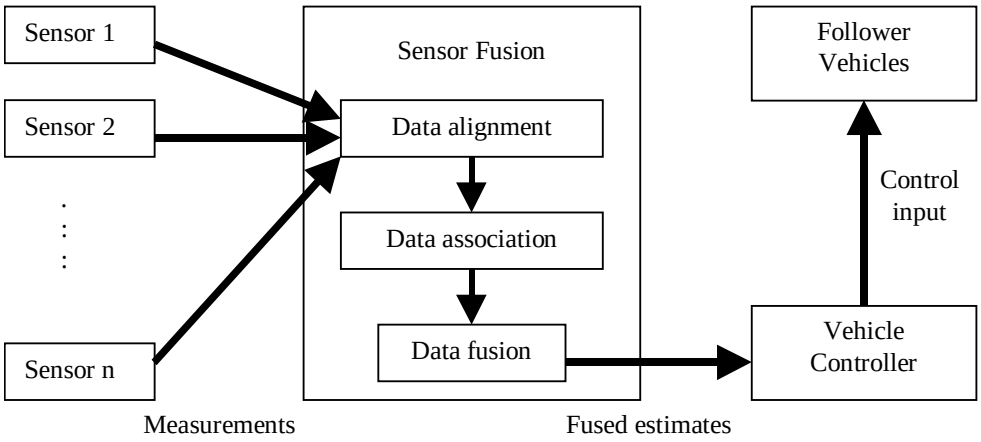

Figure 1.1 Sensor fusion process

their transformation into a common spatial and temporal reference frame. The second task, data alignment, involves the validation or rejection of the sensor data with the use of gating which is performed by verifying that the measurement lies within a region predicted by the model. Finally the third task, data fusion, combines the validated measurements using a weighted average or least-squares estimation method to provide an estimate of the signal with a lower variance. A summary of the sensor fusion process is given in Figure 1.1.

Chapter 2

Longitudinal Response and Control Using the Kalman Filter

2.1 Introduction

The Kalman filter is a set of mathematical equation that provides an efficient computational (recursive) solution of the least-squares method. This algorithm, in particular, is suitable in estimating the state of a system from measurements containing random process noise. In the following two chapters, two vehicles, one leader and one follower, are considered in order to determine a control scheme that allows one to maintain a specified relative position and velocity between two consecutive vehicles. As the first step in a longitudinal vehicle control, the Kalman filter was implemented into a vehicle model and simple simulations were performed.

Three different sensors which are currently available on PATH experimental vehicles were modeled based on the sensor specifications and simulations using different sensors were compared. Table 2.1 lists the sensor specifications, approximate costs for the hardware, and characteristics.

2.2 Kalman Filter Algorithm

In stochastic state estimation problems, the dynamic system is governed by

sensor	specifications	Approximate cost	characteristics
FMCW Radar	Range 2-100m at 0.3m res., Range rate at 0.3m/s res., Azimuth at 0.002 rad res., 10deg FOV	$3,000	Insensitive to harsh condition, possibility to track multiple targets
Lidar	Range 0.1-60m at 0.1m res., 20-80deg FOV	$1,000-10,000	Sensitive to visibility and dirt
DGPS	Position +/-0.01m, Velocity +/-0.01m/s	$5,000+	Degraded performance when less than 5 satellites in LOS

Table 2.1 Sensors which are currently used on PATH experimental vehicles

$$x(k+1) = F(k)x(k) + G(k)u(k) + w(k) \tag{2.1}$$

$$z(k) = H(k)x(k) + v(k) \tag{2.2}$$

The random variable $w(k)$ and $v(k)$ represent the process and measurement noise which are white and gaussian with a covariance matrix $Q(k)$ and $R(k)$, respectively. They are also uncorrelated with each other. The Kalman filter is the estimation algorithm which satisfies two statistical conditions such as the expected value of our estimate is equal to the expected value of the state and an estimation algorithm should minimize the expected value of the square of the estimation error [2.1].

The Kalman filter estimates a state at some time and then obtains feedback in the form of noisy measurement. These two steps are called prediction and correction.

1. Prediction

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k) \tag{2.3}$$

$$P(k+1|k) = F(k)P(k|k)F(k)' + Q(k) \tag{2.4}$$

where $P(k+1|k) = E[(x(k+1) - \hat{x}(k+1|k))(x(k+1) - \hat{x}(k+1|k))']$

The prediction step is responsible for predicting the current state and covariance matrix,

used in time t+1 to predict the previous state

2. Correction step

$$K(k+1) = P(k+1|k)H'(k+1)[H(k+1)P(k+1|k)H'(k+1) + R(k+1)]^{-1} \quad (2.5)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)[z(k+1) - H(k+1)\hat{x}(k+1|k)] \quad (2.6)$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1)H(k+1)P(k+1|k) \quad (2.7)$$

The first task during the correction step is to compute the Kalman filter gain $K(k+1)$. The next step is to actually measure the process to obtain $z(k+1)$,and then to generate an a posteriori state estimate by incorporating the measurement as in (2.6). The final step is to obtain an a posteriori error covariance estimate via (2.7)[2.2, 2.3].

2.3 Kalman Filter Simulation Results

The purpose of this simulation is to show how the Kalman filter can be implemented into a vehicle control model. In order to simplify the real vehicle model, two assumptions were made that a vehicle does not have a length, which means vehicle is considered a point, and the mass is 1kg. The leader and follower vehicles were traveling at a constant velocity of 10m/s and the initial distance between two vehicles is 30m. The follower vehicle tried to keep the same velocity but decrease the distance from the leader vehicle up to 10 m. After 25 seconds, the follower vehicle was increasing the distance again.

Sensor 1, sensor 2, and sensor 3 are FMCW radar, lidar, and DGPS, respectively. Three sensors have different sampling times and measure noise covariances.

- Sensor 1 (FMCW radar)

  sampling time : 0.02 sec.

  measurement noise covariance : $R = \begin{bmatrix} 0.6667 & 0 \\ 0 & 0.1333 \end{bmatrix}$

- Sensor 2 (lidar)

  sampling time : 0.04 sec.

measurement noise covariance : $R = \begin{bmatrix} 0.1333 & 0 \\ 0 & 0.6667 \end{bmatrix}$

- Sensor 3 (DGPS)

sampling time : 0.5 sec.

measurement noise covariance : $R = \begin{bmatrix} 0.2667 & 0 \\ 0 & 0.0333 \end{bmatrix}$

The simulation results are shown in the following figures. Different measurement noise covariances affect the result of filtered estimates as well as unfiltered measurements. Figure 2.1 and figure 2.2 show that the position error after filtering decreased more than 50 % even while the velocity error did not change a lot. On the other hand, the velocity error clearly decreased up to more than 60% when sensor 2 was used. By comparing these two simulation results, it can be concluded that the Kalman filter improves the error reduction when the measurement noise covariance of a sensor is big. Even though the result of DGPS simulation shows that the maximum error of filtered estimates is bigger than that of unfiltered measurements, entire error was decreased.

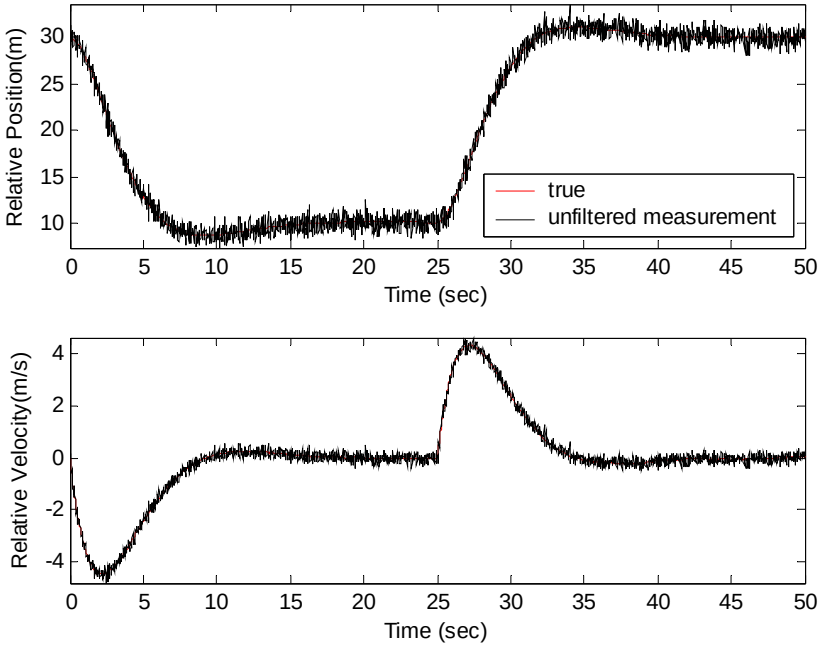

Figure 2.1 Unfiltered measurement using sensor 1

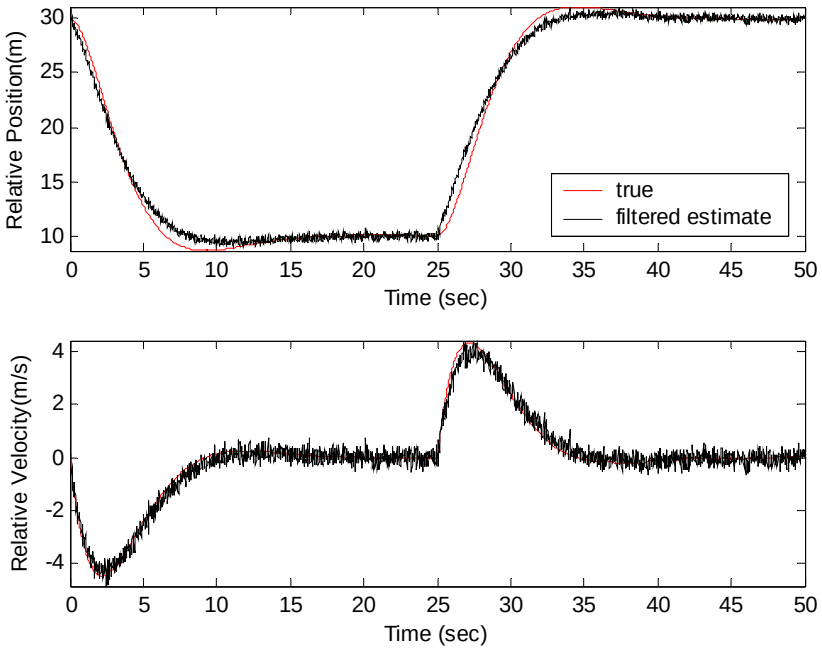

Figure 2.2 Filtered estimate using sensor 1

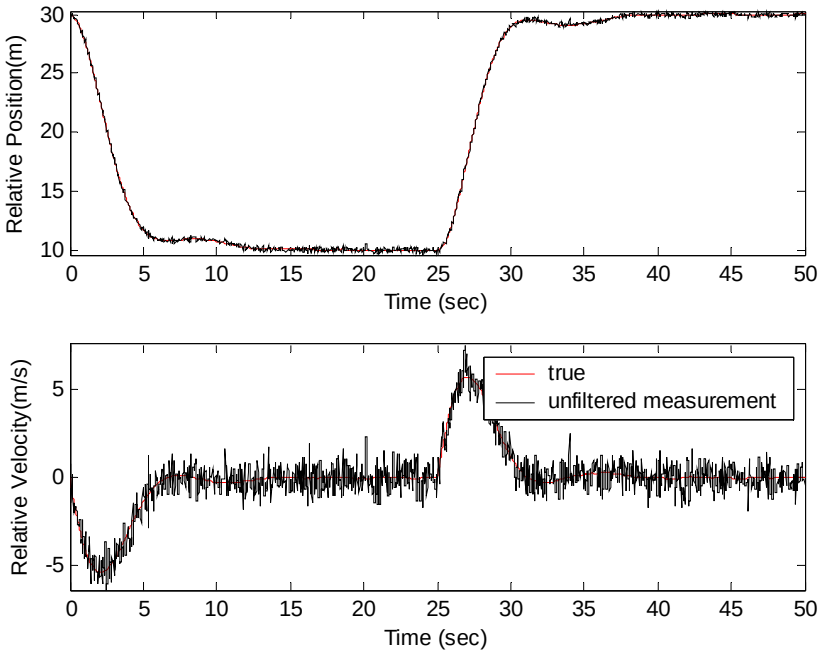

Figure 2.3 Unfiltered measurement using sensor 2

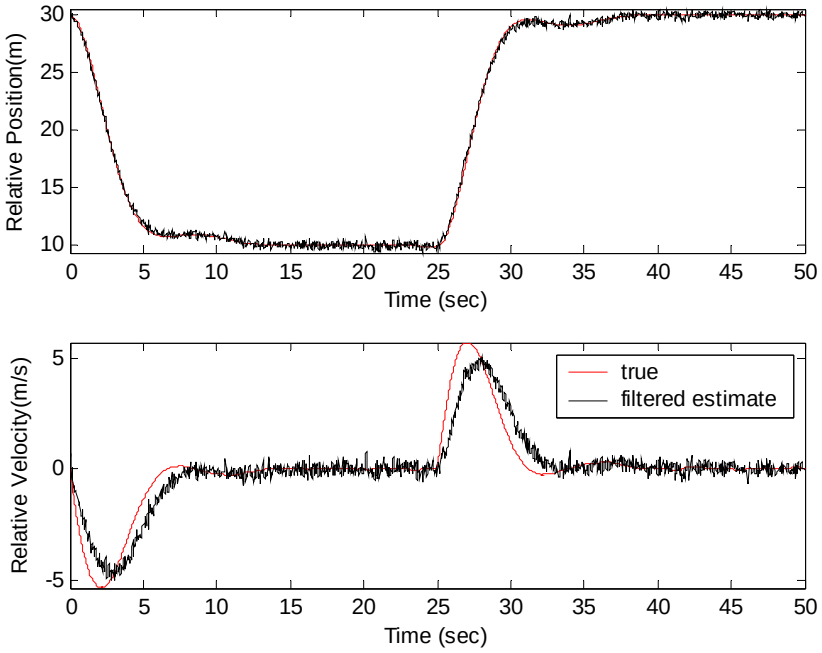

Figure 2.4 Filtered estimate using sensor 2

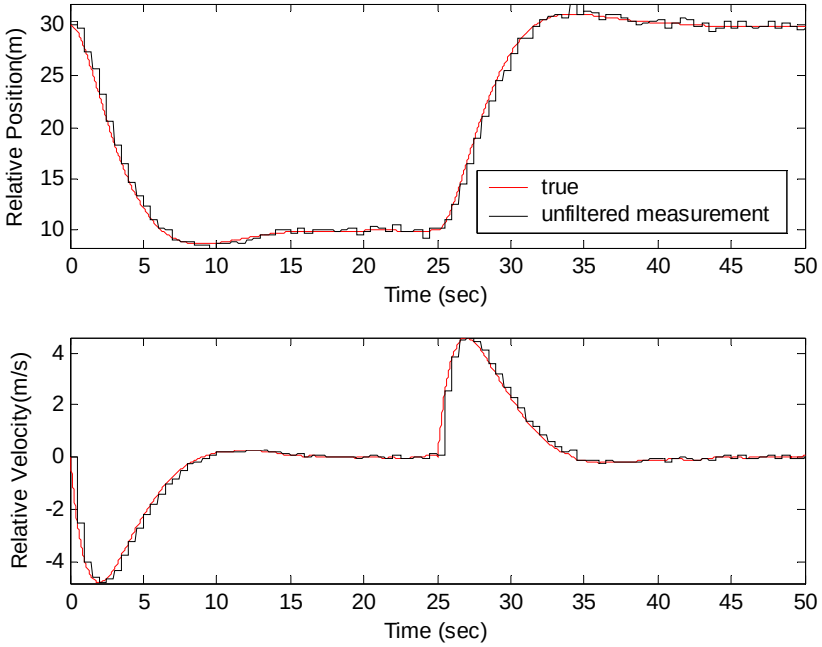

Figure 2.5 Unfiltered measurement using sensor 3

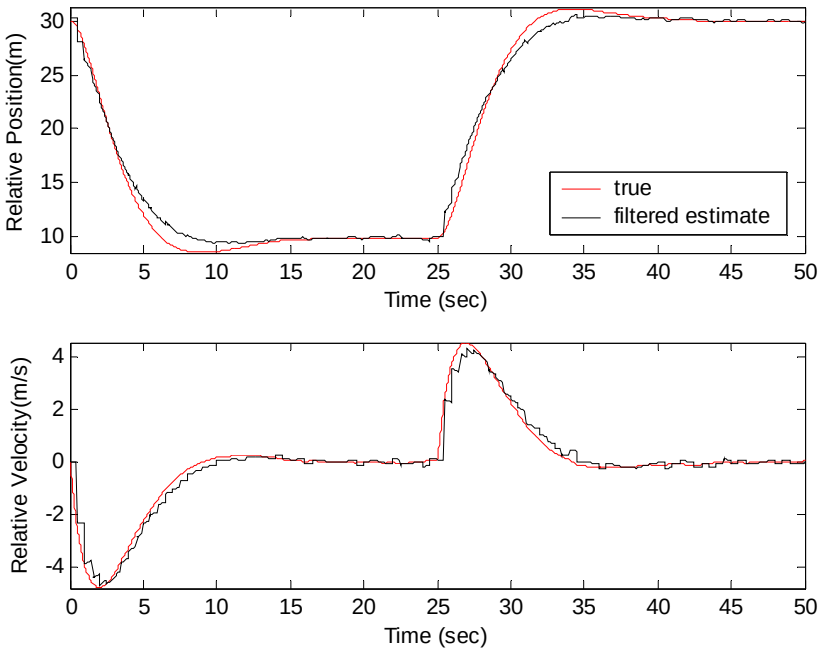

Figure 2.6 Filtered estimate using sensor 3

Chapter 3

Probabilistic Data Association Filter

3.1 Introduction

The Probabilistic Data Association Filter (PDAF) is a Bayesian approach to multiple target measurement association that uses conditional probabilities rather than likelihood measures. Introduced by Bar-Shalom and Tse in 1975, the PDAF has been used extensively in sensor fusion applications to combine multiple measurements from a single sensor such that the output estimate has minimum estimation error covariance. [3.1] It assumes that each target is isolated from all other targets and considers all other measurements external to a given validation region as false clutter points. [3.2]

In the case of multiple sensors, the PDAF can be structured as a sequential Kalman Filter that weights the correction for each sensor based on the validity of each sensor measurement. Furthermore, studies have shown a certain number of advantages with sequential filtering versus parallel filtering of all sensor measurements such as greater estimation accuracy and less computation. [3.3]

3.2 PDAF Algorithm

The set of 'validated' measurements contains $n[k]$ elements and is obtained from a sensor at time k. This set can be expressed as

$$Z[k] = \{ z_i[k] \}_{i=1}^{n[k]} \tag{3.1}$$

and the set of all measurements up to and including time k is denoted as

$$Z^k = \{Z[l]\}_{l=1}^{k} \tag{3.2}$$

The i-th measurement at time k, $z_i[k]$, is considered validated if it falls within a particular 'validation' region. The residual, or innovation, corresponding to measurement $z_i[k]$ and a particular model is given by

$$v_i[k] = z_i[k] - \hat{z}[k \mid k-1] \tag{3.3}$$

where $\hat{z}[k \mid k-1]$ is the predicted measurement estimate and is given by

$$\hat{z}[k \mid k-1] = H\hat{x}[k \mid k-1] \tag{3.4}$$

The measurement matrix H is equal to the C matrix defined in the state-space output of the model and $\hat{x}[k \mid k-1]$ is the predicted state estimate. Combining Equations 3.3 and 3.4 yields the following expression for the measurement residual:

$$v_i[k] = z_i[k] - H\hat{x}[k \mid k-1] \tag{3.5}$$

With the innovation covariance, $S[k]$ as given before, the elliptical validation region is defined as

$$\{z : v_i[k]^T \bar{S}[k]^{-1} v_i[k] \le g^2\} \tag{3.6}$$

where g is a threshold determined from the χ-squared distribution tables with n_z degrees of freedom corresponding to the probability α_1 of rejecting the correct data return. The number of degrees of freedom is equal to the dimension of the measurement vector. In the two-dimensional case the area of this region is denoted

$$V = g^2 \pi \det S^{1/2} \tag{3.7}$$

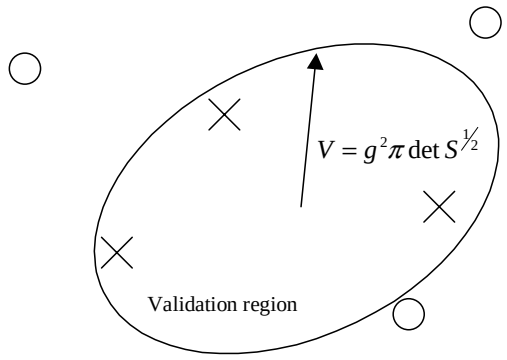

Figure 3.1 Validation gating using the χ-squared distribution

Therefore, an increase in g or $S[k]$ will increase the size of the validation gate depicted in Figure 3.1.

The PDAF technique states that the best estimate of the target state is the conditional mean based upon all observations that with some non-zero probability originated from the target, i.e.,

$$\hat{x}[k \mid k] = E\left[x[k] \mid Z^k \right] \tag{3.8}$$

The expected value of the state estimate is given by

$$E\left[x[k] \mid Z^k \right] = \sum_{i=0}^{n[k]} \beta_{k,i} E\left[x[k] \mid \chi_i[k], Z^k \right] \tag{3.9}$$

where $\chi_i[k]$ denotes the event that the i-th validated measurement, $z_i[k]$, is correct ($i = 1, 2, ..., n[k]$), $\chi_0[k]$ is the event that no validated measurement is correct.

$$E[x[k] \mid \chi_i[k], Z^k] = \hat{x}_i[k \mid k] \tag{3.10}$$

Combining Equations 3.8-3.10 yields the following expression for the state estimate conditioned upon all prior measurements:

$$\hat{x}[k \mid k] = \sum_{i=0}^{n[k]} \beta_i[k] x_i[k \mid k] \tag{3.11}$$

The probabilistic data association (PDA) is given by

$$\beta_i[k] = P\{\chi_i[k] \mid Z^k\}, i = 0,1,...,n[k] \tag{3.12}$$

$$\text{and} \quad \sum_{i=0}^{n[k]} \beta_i[k] = 1 \tag{3.13}$$

A primary assumption made in the derivation of the PDA method is that the Probability Density Function (PDF) of the state conditioned upon past observations, the prior-PDF, is assumed normal with mean $\hat{x}[k \mid k-1] = Fx[k-1 \mid k-1]$ and covariance $P[k \mid k-1]$, i.e.,

$$p\left(x_k \mid Z^{k-1}\right) = N\left(x[k]; \hat{x}[k \mid k-1], P[k \mid k-1]\right) \tag{3.14}$$

This assumption leads to the following equations for the PDA method: [3.4]

$$\beta_i[k] = f_k\left(x_i[k]\right)\left[b[k] + \sum_{i=1}^{n[k]} f_k\left(x_i[k]\right) \right]^{-1} \tag{3.15}$$

$$\beta_0[k] = b[k]\left[b[k] + \sum_{i=1}^{n[k]} f_k\left(x_i[k]\right) \right]^{-1} \tag{3.16}$$

where

$$\begin{aligned} f_k(x_i[k]) &= (1-\alpha_1)^{-1} N(x_i[k]; \hat{x}[k \mid k-1], S[k]) \\ &= (1-\alpha)^{-1} N(\nu_i[k]; 0, S[k]) \end{aligned} \tag{3.17}$$

is a truncated normal density that is zero outside the validation region and

$$b[k] = n[k] V[k]^{-1} \frac{(\alpha_1 + \alpha_2 - \alpha_1\alpha_2)}{(1-\alpha_1)(1-\alpha_2)} \tag{3.18}$$

where α_2 is the probability that the correct return will not be detected.

The state estimate conditioned on measurement i being correct is

$$\hat{x}_i[k \mid k] = x[k \mid k-1] + W[k]v_i[k] \tag{3.19}$$

with the filter gain $W[k]$ given by

$$W[k] = P[k \mid k-1]H^T S[k]^{-1} \tag{3.20}$$

Thus the state estimate for one model in the PDAF is

$$\hat{x}[k \mid k] = \sum_{i=0}^{n[k]} \beta_i[k] x_i[k \mid k] = x[k \mid k-1] + W[k]v[k] \tag{3.21}$$

where the combined innovation is

$$v[k] = \sum_{i=1}^{n[k]} \beta_i[k] v_i[k] \tag{3.22}$$

The state estimate given by Equation 3.21 has the associated covariance [3.5]

$$\begin{aligned} P[k \mid k] &= \beta_0[k]P[k \mid k-1] + \left[1 - \beta_0[k]\right] P^o[k \mid k] \\ &+ W[k]\left[\sum_{i=0}^{n[k]} \beta_i[k] v_i[k] v_i[k]^T - v[k]v[k]^T\right] W[k]^T \end{aligned} \tag{3.23}$$

where $P^o[k \mid k]$ is the covariance associated with the condition of knowing the correct measurement and is obtained from the following expression

$$P^o[k \mid k] = P[k \mid k-1] - W[k]S[k]W[k]^T \tag{3.24}$$

The first term in the covariance given by Equation 3.24 reflects the possibility that there are no validated measurements returned from the target. The second term ensures that only a validated measurement might be a correct measurement, while the third term is a positive definite matrix that increases $P[k \mid k]$ as a result of measurement origin uncertainty.

3.3 PDAF Implementation

The PDAF can be broken down into two basic computational stages at each discrete time step k:

1) Prediction of fused state estimate and estimation error covariance based on dynamic system model

$$x(k+1) = Ax(k) + w(k)$$
$$y_1(k) = C_1 x(k) + v_1(k)$$
$$\vdots$$
$$y_4(k) = C_4 x(k) + v_4(k)$$

$$\hat{x}(k \mid k-1) = A(k-1 \mid k-1)$$
$$P(k \mid k-1) = AP(k-1)A^T + Q$$

$$z_1(k) = C_1 \hat{x}(k \mid k-1)$$
$$S_1(k) = C_1 P(k \mid k-1) C_1^T + R$$
$$v_1(k) = y_1(k) - z_1(k)$$
$$V(k) = v_1^T(k) S_1^{-1}(k) \quad v_1(k)$$

2) Validation of all measurements from each sensor using g-sigma gating based on predicted measurement and measurement covariance. For each sensor,

$$\hat{x}(k \mid k-1) = x_{i-1}(k \mid k)$$
$$P(k \mid k-1) = P_{i-1}(k)$$

a) If the sensor measurement is valid, use in Kalman filter to correct fused state estimate and estimation error covariance

$$if \quad V(k) < g^2,$$
$$\hat{x}_1(k \mid k) = \beta_0 \hat{x}_1(k \mid k-1) + \beta_1 \left(\hat{x}_1(k \mid k-1) + W(k) v_1(k) \right)$$
$$W(k) = P(k \mid k-1) C_1^T S^{-1}(k)$$
$$P_1(k) = \beta_0 P(k \mid k-1) + (1-\beta_0)(I - W(k)C) P(k \mid k-1) + \beta_1 (1-\beta_1) W(k) v_1(k) v_1^T(k) W^T(k)$$

b) If the sensor measurement is not valid, do not use in Kalman filter

$$if \quad V(k) \geq g^2,$$
$$\hat{x}_1(k \mid k) = \hat{x}_1(k \mid k-1)$$
$$P_1(k) = P(k \mid k-1)$$

The vehicles and sensors are modeled as signal-flow block diagrams and then simulated with the MATLAB/Simulink software package, as shown in Figures 3.2 and 3.3. The PDA method can then be applied with the signals illustrated in Figures 3.4a and 3.4b and the PDAF code described above is implemented with m-file scripts. It should be noted that the input and sensor measurement signals are corrupted by noise using a random variable generator that adds the random number to the given signal.

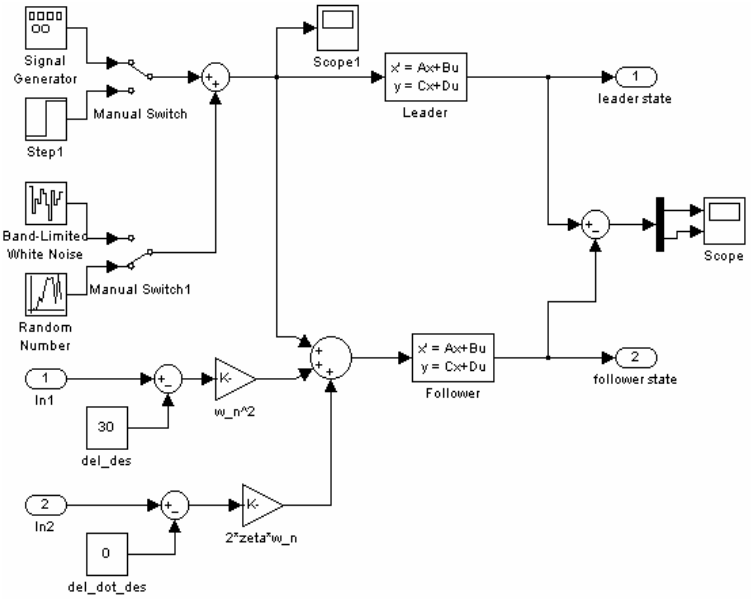

Figure 3.2 Leader and follower vehicle model

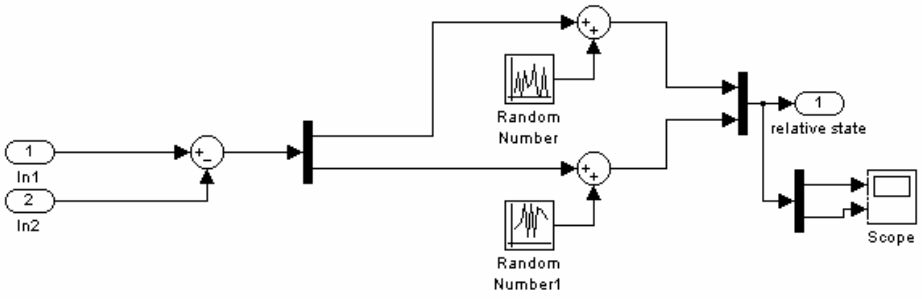

Figure 3.3 Sensor model

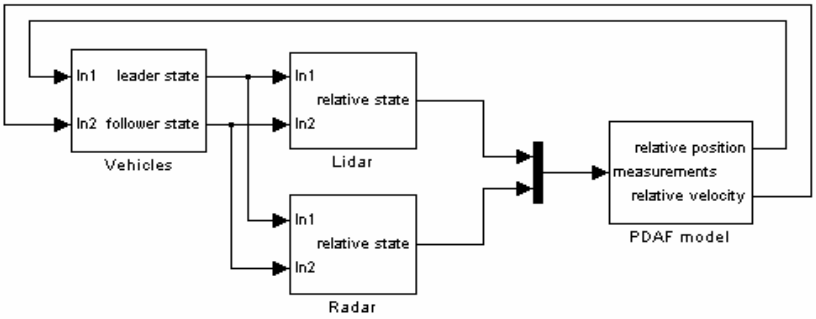

Figure 3.4a PDAF model

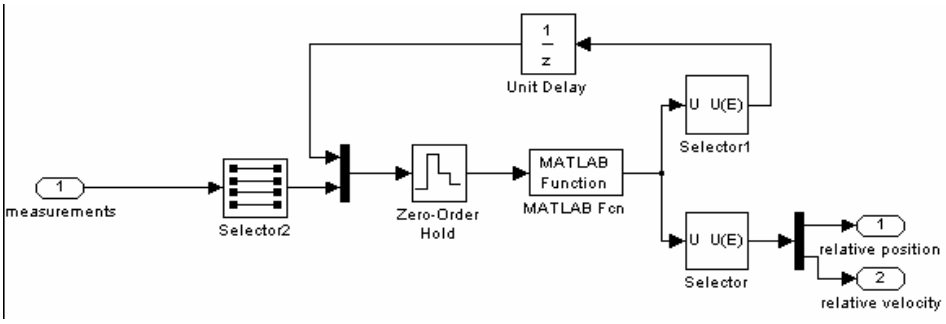

Figure 3.4b PDAF block

3.4 PDAF Simulation Results

For the following 50 s simulations, the initial position and velocity of the leader vehicle are 30 m and 10 m/s, respectively, while the follower vehicle has an initial position of 0 m and a velocity of 10 m/s. The requirements are to maintain a constant relative velocity between the two vehicles and to maintain a constant distance between them, initially 10 m, then 30 m after 25 s. In addition, at time t = 30 s, sensor 3 drops out and no longer returns measurements.

As shown in Figures 3.5 and 3.6, the desired relative position and velocity between the two vehicles are obtained within 7 s of each change maneuver by the leader. The fused state estimates resulting from the PDA method are much cleaner than the noisy measured signals. Furthermore, the filtering process is relatively unaffected by the loss of measurements from sensor 3 when it drops out 30 s into the simulation.

Additional sensors can be added as needed; results for a four-sensor simulation are given in Figures 3.7 and 3.8 where the model now makes use of sensors that respond to counts of magnetic road markers embedded in the lanes of certain freeways within the PATH AHS architecture such as the I-15 High Occupancy Vehicle (HOV) lanes.

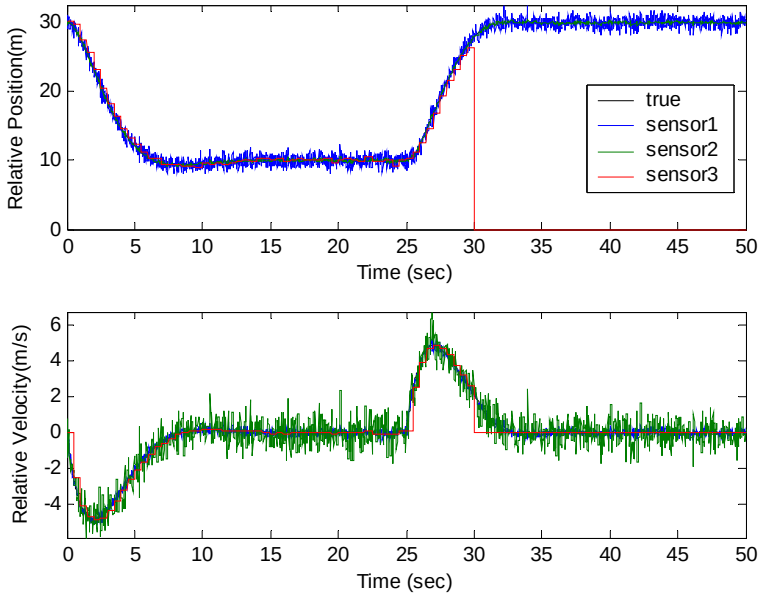

Figure 3.5 Sensor measurements

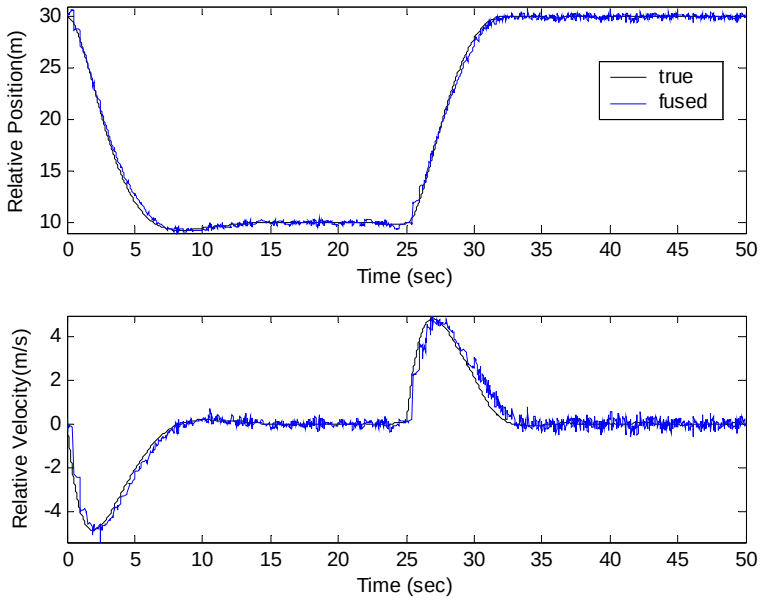

Figure 3.6 Fused estimates

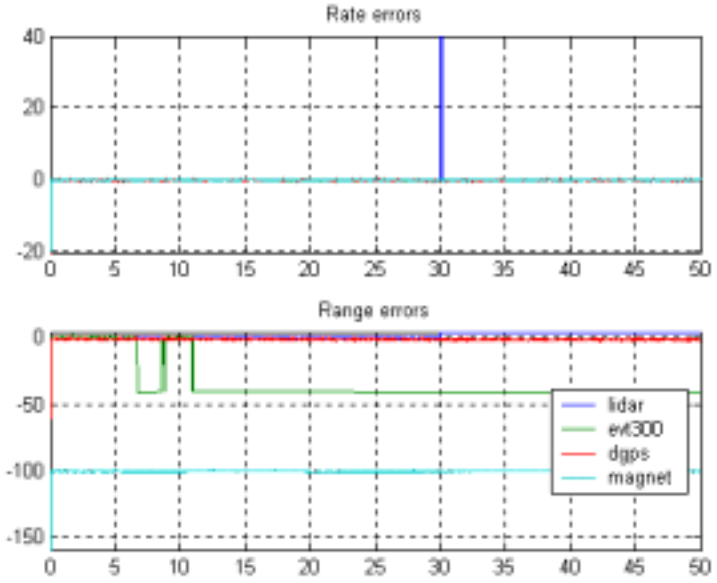

Figure 3.7 Rate and range sensor errors

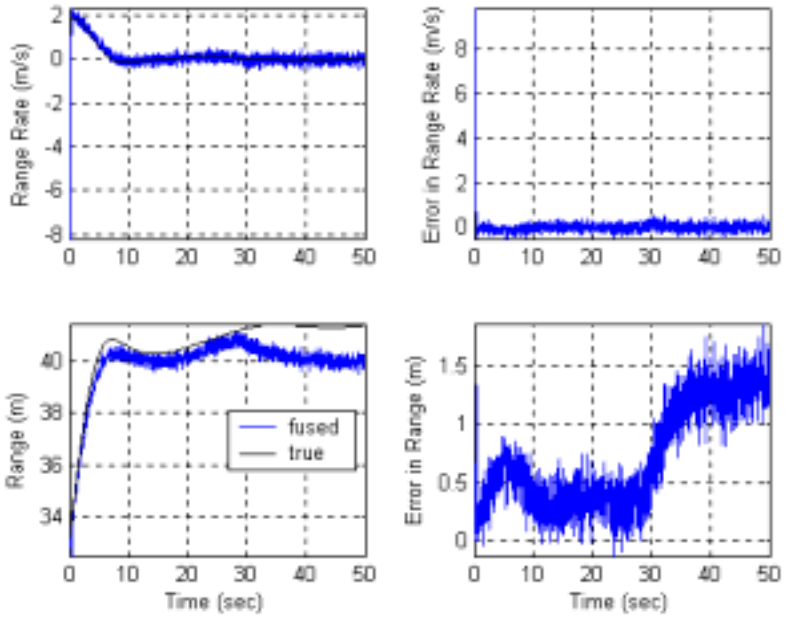

Figure 3.8 Fused range, rate, and associated errors

Chapter 4

Lateral Response and Control

4.1 Introduction

The vehicle control model previously developed, although suitable and effective for longitudinal control, does not account for rotation and lateral motion of the vehicle due to steering inputs. In this section an analysis of vehicle steering design and the cornering forces developed in the tires during a turning maneuver are first presented, followed by a formulation of the three-dimensional equations of motion. From these equations are obtained the longitudinal, lateral, and rotational states of the vehicle. Finally, simulation results of vehicle transient response to steering and acceleration inputs are then presented and discussed for feasibility of implementing real-time lateral control.

4.2 Ackerman Steering Geometry

The two most common steering mechanisms employed by road vehicles are the 'rack-and-pinion' and the 'recirculating-ball' designs. The 'rack-and-pinion' type is the choice of most manufacturers and the 'recirculating-ball' type is a past favorite because the balls act as a rolling thread bearing, thus causing less friction and wear.

The steering systems used on motor vehicles vary widely in design but are functionally quite similar. The steering wheel connects by shafts, universal joints, and vibration isolators to the steering gearbox whose purpose is to convert the rotary motion of the steering wheel to a translational motion appropriate for steering the wheels. The lateral translation produced by the

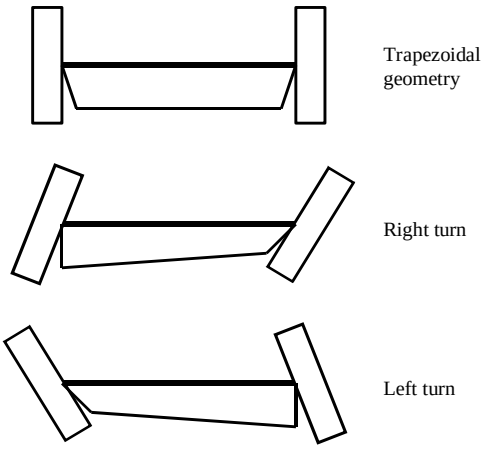

Figure 4.1 Steering configurations

gearbox is relayed through the linkages to steering arms on the left and right front wheels, as shown in Figure 4.1.

The kinematic geometry of the relay linkages and steering arms is generally not a parallelogram, which would produce equal left and right steering angles, but rather a trapezoid to more closely approximate "Ackerman" geometry that steers the inside wheel to a greater angle than the outside wheel. Indeed, the prime consideration in the design of the steering system geometry is minimum tire scrub during cornering. This requires that during the turn all tires should be in pure rolling without lateral sliding. [4.1] To satisfy this requirement, the wheels should follow curved paths with radii originating from a common center, as shown in Figure 4.2.

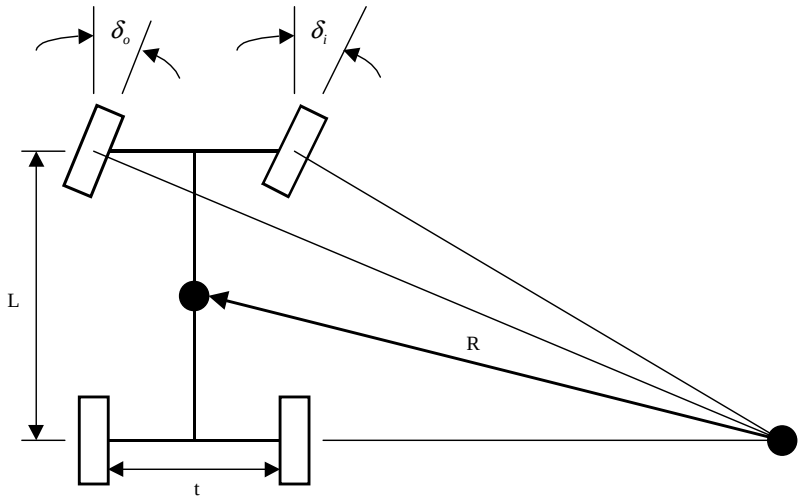

Figure 4.2 Ackerman geometry

Ackerman steering geometry requires that

$$\delta_o = \tan^{-1}\frac{L}{R + t/2} \tag{4.1}$$

$$\delta_i = \tan^{-1}\frac{L}{R - t/2} \tag{4.2}$$

For small angles, as are typical of most turning maneuvers, the arctangent of the angle is very nearly equal to the angle itself, in radians. At high speeds the radius of turn is much larger than the wheel-base of the vehicle. Small angles can then be assumed and the difference between steer angles on the outside and inside front wheels is negligible. Thus, for convenience, the two front wheels can be represented by one steer angle with a cornering force equivalent to both wheels and is approximated as

$$\delta_f = \frac{L}{R} \tag{4.3}$$

4.3 Tire Cornering Forces

Steady-state handling performance is concerned with directional behavior of a vehicle during a turn under non-time-varying conditions. An example of a steady-state turn is a vehicle negotiating a curve with constant radius at constant forward speed. In the analysis of steady-state handling behavior, the inertia properties of the vehicle are not involved.

When a vehicle is negotiating a turn at moderate or higher speeds, however, the effect of the centrifugal force acting at the center of gravity can no longer be neglected. To balance the centrifugal forces, the tires must develop appropriate cornering forces, i.e. a side forces acting on each of the tires which produces a slip angle. [4.2] Thus, when a vehicle is negotiating a turn at moderate or higher speeds, the four tires will develop appropriate slip angles. To simplify the analysis, the pair of tires on an axle are represented by a single tire with double the cornering stiffness, as shown in Figure 4.3. The handling characteristics of the vehicle depend, to a great extent, on the relationship between the slip angles of the front and rear tires, α_f and α_r, respectively.

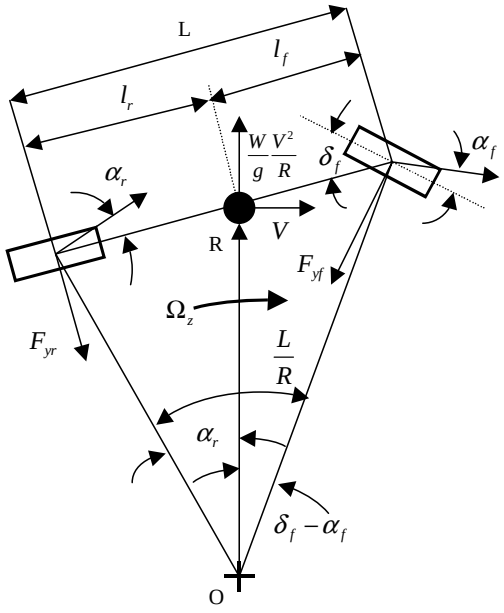

Figure 4.3 Simplified vehicle model

The steady-state handling response of a vehicle at moderate and higher speeds will have a more complex relation with the steering input than that at low speeds. From the geometry shown in Figure 4.3, the relationship between the steer angle of the front tire δ_f, turning radius R, wheel base L, and the slip angles of the front and rear tires α_f and α_r, respectively, is given by

$$\delta_f - \alpha_f + \alpha_r = \frac{L}{R} \tag{4.4}$$

Equation 4.4 indicates that the steer angle δ_f required to negotiate a given curve is a function of not only the turning radius R but also the front and rear slip angles α_f and α_r, respectively. [4.3] These slip angles are dependent on the side forces acting on the tires and their cornering stiffness. The cornering forces on the front and rear tires F_{yf} and F_{yr} can be

determined from the equilibrium of the vehicle in the lateral direction. For small steer angles, the cornering forces acting at the front and rear tires are approximately given by

$$F_{yf} = \frac{W}{g}\frac{V^2}{R}\frac{l_r}{L} \tag{4.5}$$

$$F_{yr} = \frac{W}{g}\frac{V^2}{R}\frac{l_f}{L} \tag{4.6}$$

where W is the total weight of the vehicle, g is the acceleration due to gravity, V is the vehicle forward speed. [4.4]

The normal load on each of the front wheels W_f and that on each of the rear wheels W_r under static conditions are expressed by

$$W_f = \frac{Wl_r}{2L} \tag{4.7}$$

$$W_r = \frac{Wl_f}{2L} \tag{4.8}$$

Equations 4.7 and 4.8 can be rewritten as

$$F_{yf} = 2W_f\frac{V^2}{gR} \tag{4.9}$$

$$F_{yr} = 2W_r\frac{V^2}{gR} \tag{4.10}$$

The slip angles α_f and α_r are therefore given by

$$\alpha_f = \frac{F_{yf}}{2C_{\alpha f}} = \frac{W_f}{C_{\alpha f}}\frac{V^2}{gR} \tag{4.11}$$

$$\alpha_r = \frac{F_{yr}}{2C_{\alpha r}} = \frac{W_r}{C_{\alpha r}}\frac{V^2}{gR} \tag{4.12}$$

where $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness of each of the front and rear tires, respectively. [4.5] The cornering stiffness of a tire varies with a number of operational parameters including inflation pressure, normal load, tractive or braking effort, and lateral force. It may be regarded as a constant only within a limited range of operating conditions. [4.6]

4.4 Transient Response

A vehicle may be regarded as a control system upon which various inputs are imposed. During a turning maneuver the steer angle induced by the driver can be considered as an input to the system and the motion variables of the vehicle, the state variables, may be regarded as outputs that can be measured by sensors for feedback. The ratio of the yaw velocity, lateral acceleration, or curvature, to the steering input can then be used for comparing the response characteristics of different vehicles and their response characteristics.

Between the application of steering input and the attainment of steady-state motion, the vehicle is in transient state. The behavior of the vehicle in this period is usually referred to as transient response. The overall handling qualities of a vehicle depend, to a great extent, on its transient behavior. The optimum transient response of a vehicle is that which has the fastest response with a minimum of oscillation in the process of approaching the steady-state motion.

In analyzing the transient response, the inertia properties of the vehicle must be taken into consideration. During a turning maneuver, the vehicle is in translation as well as in rotation. To describe its motion, it is convenient to use a set of axes fixed to and moving with the vehicle body because with respect to these axes the mass moments of inertia of the vehicle are constant, whereas with respect to axes fixed in space the mass moments of inertia vary as the vehicle changes its orientation.

To formulate the equations of transient motion for a vehicle during a turning maneuver, it is necessary to express the absolute acceleration of the center of gravity of the vehicle (i.e., the acceleration with respect to axes fixed in space) using the reference frame attached to the vehicle body. [4.7]

Let ox and oy be the longitudinal and lateral axes, respectively, fixed to the vehicle body with origin at the center of gravity, and let V_x and V_y be the components of the velocity V of the center of gravity along the axes ox and oy, respectively, at time t as shown in Figure 4.4.

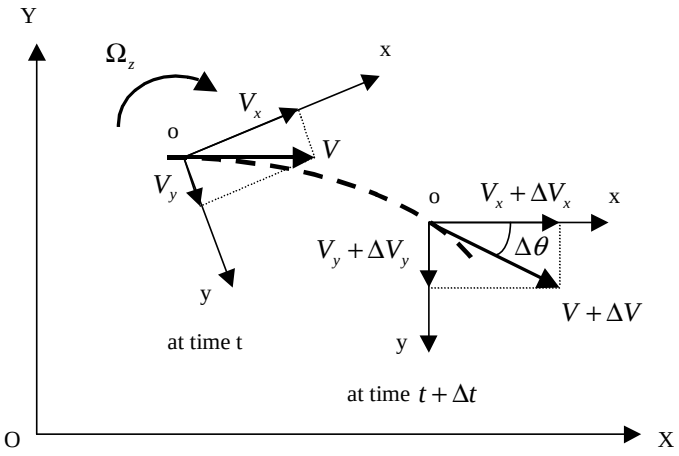

Figure 4.4 Coordinates fixed to vehicle body

As a vehicle is in both translation and rotation during a turn, at time $t + \Delta t$, the direction and magnitude of the velocity of the center of gravity as well as the orientation of the longitudinal and lateral axes of the vehicle change. The change of the velocity component parallel to the ox-axis is given by

$$\begin{aligned}&(V_x + \Delta V_x)\cos\Delta\theta - V_x - (V_y + \Delta V_y)\sin\Delta\theta \\ &= V_x\cos\Delta\theta - \Delta V_x\cos\Delta\theta - V_x - V_y\sin\Delta\theta - \Delta V_y\sin\Delta\theta\end{aligned} \tag{4.13}$$

Consider that $\Delta\theta$ is small and neglect second-order terms, the above expression becomes

$$\Delta V_x - V_y\Delta\theta \tag{4.14}$$

The component along the longitudinal axis of the absolute acceleration of the center of gravity of the vehicle can be obtained by dividing the above expression by Δt. In the limit, this gives

$$a_x = \frac{dV_x}{dt} - V_y\frac{d\theta}{dt} = \dot{V}_x - V_y\Omega_z \tag{4.15}$$

The component dV_x/dt (or $\dot{V}_x$) is due to the changing magnitude of the velocity component V_x and is directed along the ox axis, and the component $V_y\, d\theta/dt$ (or $V_y\Omega_z$) is due to the rotation of the velocity component V_y. Following a similar approach, the component of the absolute acceleration of the center of gravity of the vehicle along the lateral axis is

$$a_y = \frac{dV_y}{dt} + V_x \frac{d\theta}{dt} = \dot{V}_y + V_x\Omega_z \tag{4.16}$$

Referring to Figure 4.5, for a vehicle having plane motion, the equations of motion using the axes fixed to the vehicle body are given by

$$m(\dot{V}_x - V_y\Omega_z) = F_{xf}\cos\delta_f + F_{xr} - F_{yf}\sin\delta_f \tag{4.17}$$

$$m(\dot{V}_y + V_x\Omega_z) = F_{yr} + F_{yf}\cos\delta_f + F_{xf}\sin\delta_f \tag{4.18}$$

$$I_z\dot{\Omega}_z = l_f F_{yf}\cos\delta_f - l_r F_{yr} + l_f F_{xf}\sin\delta_f \tag{4.19}$$

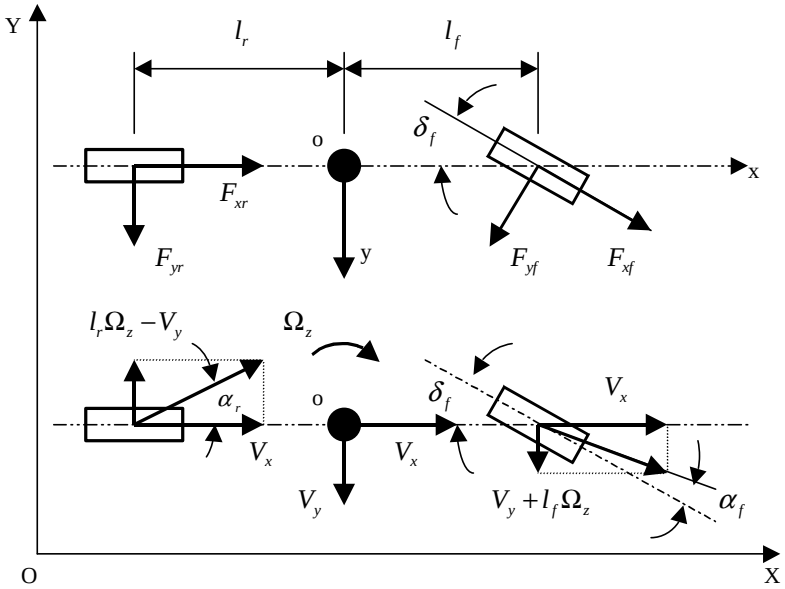

Figure 4.5 Vehicle angles and dynamic components

where m is the mass of the vehicle and I_z is the mass moment of inertia of the vehicle about the z axis.

In deriving the above equations, it is assumed that the vehicle body is symmetric about the longitudinal plane (i.e., the xoz plane) and that roll motion of the vehicle body is neglected. If the vehicle is neither accelerating nor decelerating in the longitudinal direction, the first equation of motion given by Equation 4.17 may be omitted and the lateral motions of the vehicle are governed by Equations 4.18 and 4.19.

The slip angles α_f and α_r can be defined in terms of vehicle motion variables Ω_z and V_y. Referring to Figure 4.5 and using small angle approximations,

$$\alpha_f = \delta_f - \frac{l_f \Omega_z + V_y}{V_x} \tag{4.20}$$

$$\alpha_r = \frac{l_r \Omega_z - V_y}{V_x} \tag{4.21}$$

The lateral forces acting on the front and rear tires are a function of the corresponding slip angle and cornering stiffness and are expressed by

$$F_{yf} = 2C_{\alpha f}\alpha_f \tag{4.22}$$

$$F_{yr} = 2C_{\alpha r}\alpha_r \tag{4.23}$$

Combining Equations 4.17 through 4.23 and assuming that F_{xr} is zero, the equations of motion of the vehicle become

$$\dot{V}_x = u\cos\delta_f + V_y\Omega_z - \frac{2C_{\alpha f}}{m}\left(\delta_f - \frac{l_f\Omega_z + V_y}{V_x}\right)\sin\delta_f \tag{4.24}$$

$$\dot{V}_y = -V_x\Omega_z + \frac{2C_{\alpha r}}{m}\left(\frac{l_r\Omega_z - V_y}{V_x}\right) + \frac{2C_{\alpha f}}{m}\left(\delta_f - \frac{l_f\Omega_z + V_y}{V_x}\right)\cos\delta_f + u\sin\delta_f \tag{4.25}$$

$$\dot{\Omega}_z = \frac{2l_fC_{\alpha f}}{I_z}\left(\delta_f - \frac{l_f\Omega_z + V_y}{V_x}\right)\cos\delta_f - \frac{2l_rC_{\alpha r}}{I_z}\left(\frac{l_r\Omega_z - V_y}{V_x}\right) + \frac{l_f mu}{I_z}\sin\delta_f \tag{4.26}$$

In the above equations, δ_f represents the steer angle of the front wheel as a function of time. If, in addition to the steer angle and longitudinal acceleration, external forces or moments, such as aerodynamic forces and moments, are acting on the vehicle, they should be added to Equations 4.24 through 4.26 as additional input variables. When the input variables, such as steer angle and external forces, and the initial conditions are known, the response of the vehicle, expressed in terms of the state variables as functions of time, can be determined by solving the differential equations.

4.5 Simulation Results

For the following simulations, the selected test vehicle is a Honda Accord. Parameters for the Honda Accord used in the experiments were $m = 1590kg$, $I_x = 2920kg.m^2$, $l_f = 1.22m$, $l_r = 1.62m$, and $c_f = c_r = 2 \times 60000N/rad$. The cornering stiffness is increased by a factor of two since both of the tires are lumped together at the front and at the back of the vehicle. [4.8]

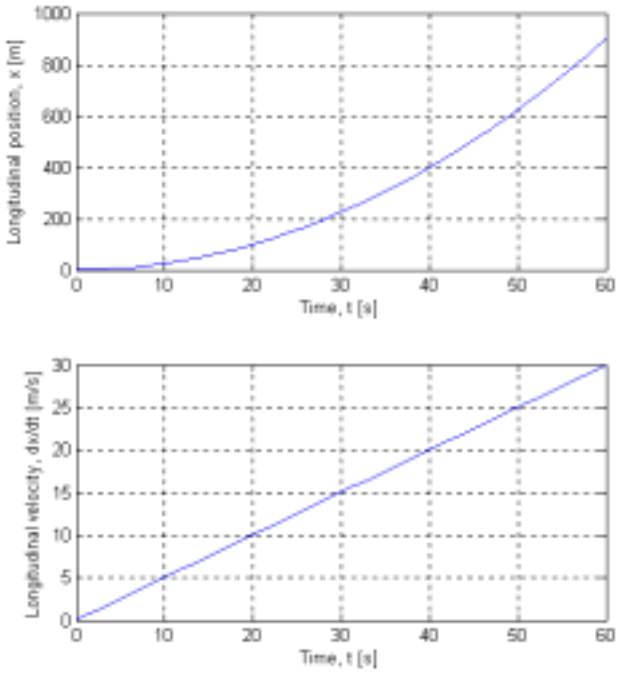

Figure 4.6 Constant longitudinal acceleration

4.5.1 Constant longitudinal acceleration

The first simulation set is a validation of the three-dimensional model for the longitudinal response of the vehicle. A constant acceleration of $u = 0.5m/s^2$ is applied to a vehicle initially at rest, i.e., with zero initial conditions, and no steering angle is applied, i.e., $\delta_f = 0$, such that vehicle is vehicle is accelerating in the longitudinal direction only.

As shown in the plots given in Figure 4.6, the longitudinal position response is parabolic over time and the longitudinal velocity response is linear. At $t = 60s$, the position $x = 900m$ and the velocity $v_x = 30m/s$ which match the expected results from a second-order, linear, longitudinal model under constant acceleration.

4.5.2 Constant steering angle with initial velocity

In the second simulation set, a constant steering angle $\delta_f = 0.01rad$ is applied to a vehicle travelling with an initial velocity $v_0 = 25m/s$. For each simulation, the vehicle state responses are obtained and compared for various longitudinal acceleration inputs. Results for all six states and the vehicle trajectory are given in Figures 4.7 through 4.10.

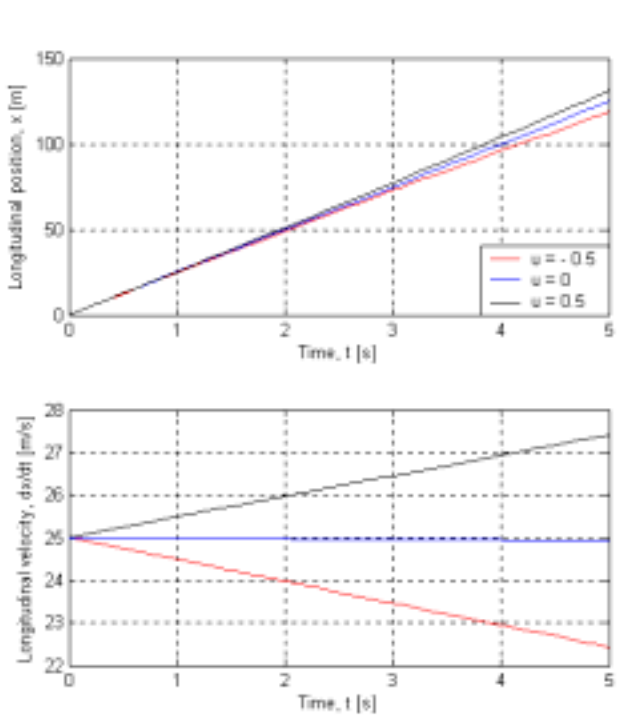

Figure 4.7 Longitudinal response for constant steering angle with initial velocity

From the plots given in Figure 4.7, the constant steering angle input results in a decrease in the longitudinal velocity. Indeed, for zero acceleration, the longitudinal velocity drops from its initial value of $v_0 = 25m/s$, although this drop is rather negligible over the time span of 5 seconds.

As shown in Figure 4.8, the vehicle responds quickly to a small steering input since the initial velocity is large. The vehicle begins by moving in the opposite direction of that desired as it first swerves in the postive y-direction for the initial 0.25 seconds. The vehicle then moves in the negative y-direction, as desired. This phenomenon is due the initial cornering force on the tire, and for zero acceleration, the lateral velocity reaches a steady-state value of –0.12 m/s after 1 second.

From the plots given in Figure 4.9, the yaw angle is roughly linear versus time for various longitudinal acceleration inputs. The first peak in the angular velocity response is consistent with the initial cornering force upon a steering input which causes a momentary rotation in the opposite direction. The lateral velocity reaches a steady-state value of 0.62 rad/s after 1 second.

As shown in Figure 4.10, after 5 seconds, the vehicle has traveled a longitudinal distance of roughly 120 m. The difference in lateral position between positive and negative acceleration is small, less than 0.3 m after 5 seconds, but increasing.

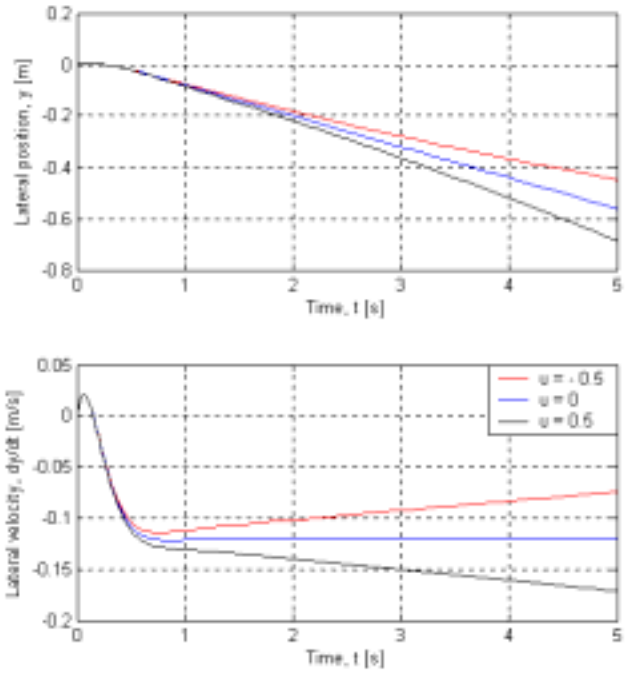

Figure 4.8 Lateral repsonse for constant steering angle with initial velocity

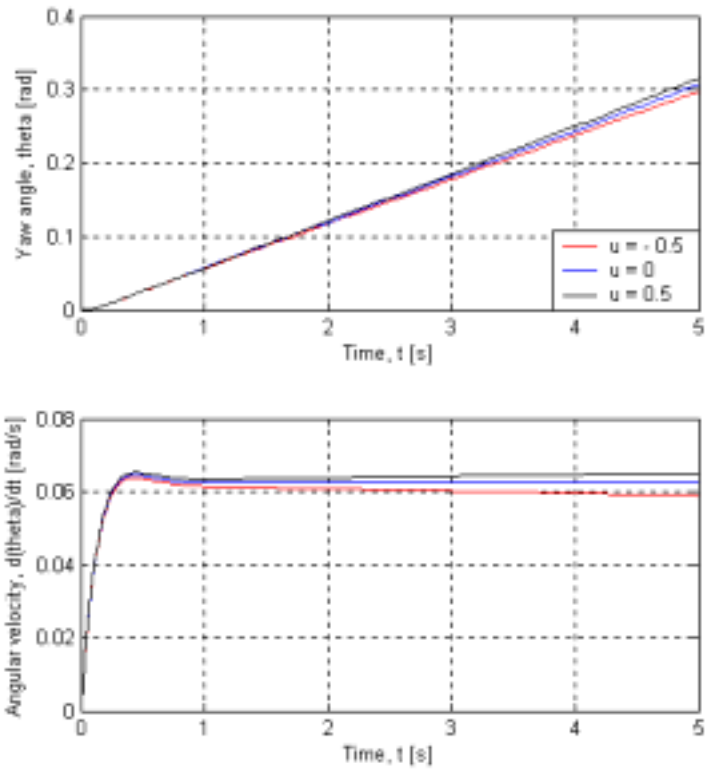

Figure 4.9 Angular response for constant steering angle with initial velocity

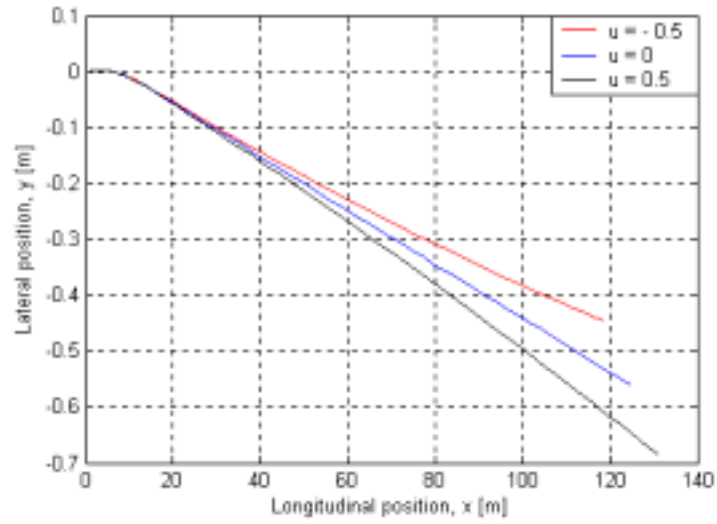

Figure 4.10 Trajectory for constant steering angle with initial velocity

4.5.3 Zero acceleration with initial velocity

In the third simulation set, zero longitudinal acceleration is applied to a vehicle travelling with an initial velocity $v_0 = 25m/s$. For each simulation, the vehicle state responses are obtained and compared for various steering angle inputs. Results for all six states and the vehicle trajectory are given in Figures 4.11 through 4.14.

From the plots given in Figure 4.11, the longitudinal velocity decreases for increasing steering angle due to the tire cornering forces that tend to oppose forward motion as the vehicle maneuvers a turn. For a steering angle of 0.1 radians, roughly 6 degrees, the longitudinal velocity drops to 19.5 m/s after 5 seconds. At higher velocity, smaller steering angles should be applied in order to minimize the longitudinal velocity gradient.

As shown in Figure 4.12, the vehicle responds quickly to the steering input since the initial velocity is large. The vehicle again begins by moving in the opposite direction of that desired as it first swerves in the postive y-direction for the initial 0.3 seconds. The vehicle then moves in the negative y-direction, as desired. For a steering angle of 0.01 radians, roughly 0.6 degrees, the lateral velocity reaches a steady-state value of –0.15 m/s after 1 second.

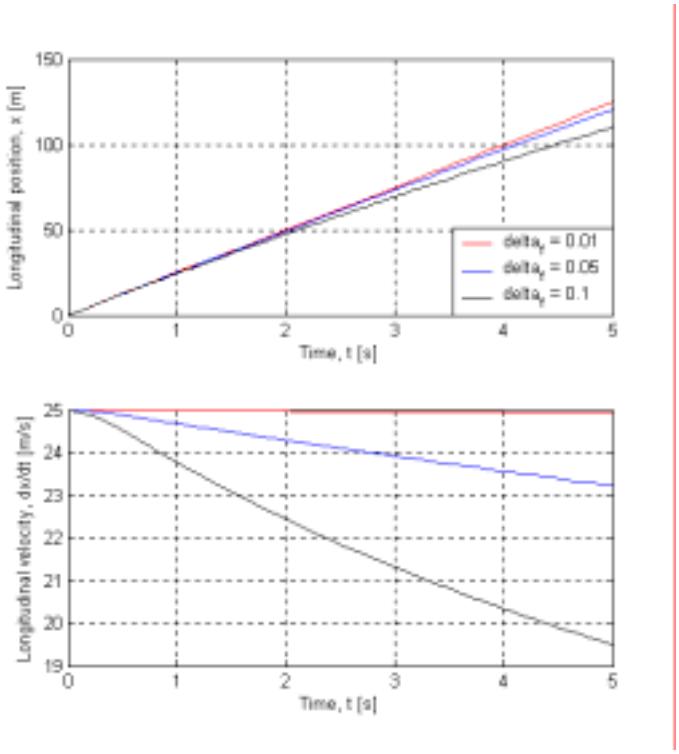

Figure 4.11 Longitudinal response for zero acceleration and initial velocity

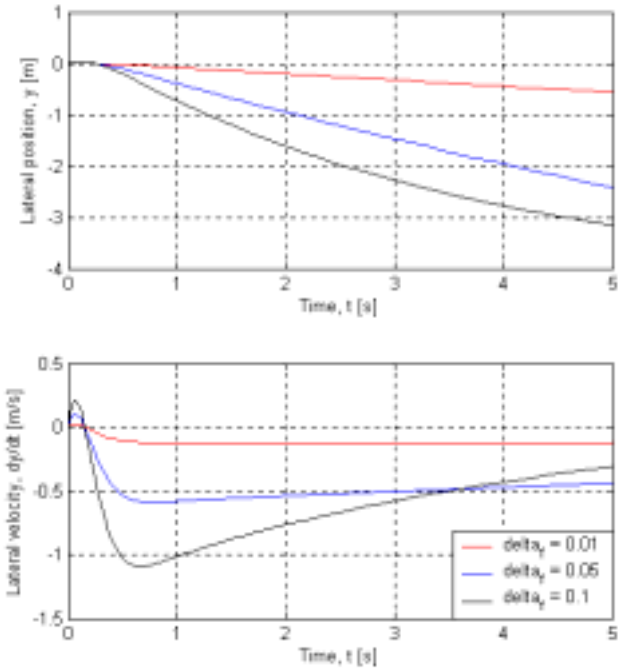

Figure 4.12 Lateral response for zero acceleration and initial velocity

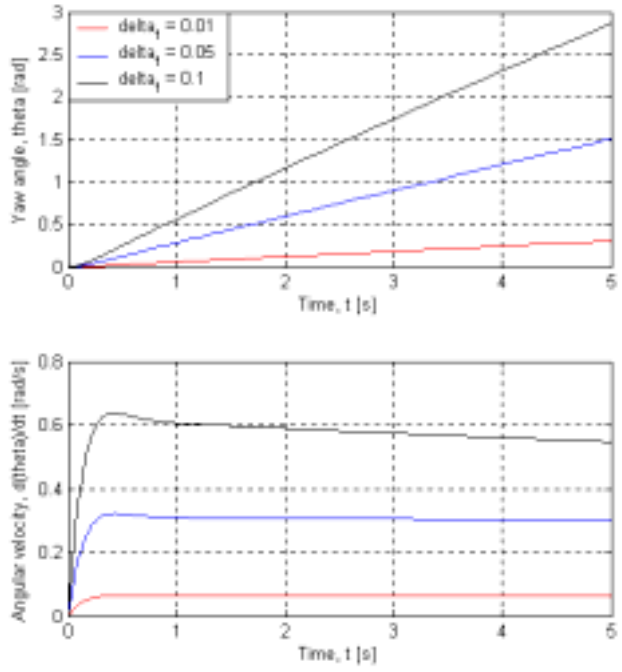

Figure 4.13 Angular response for zero acceleration and initial velocity

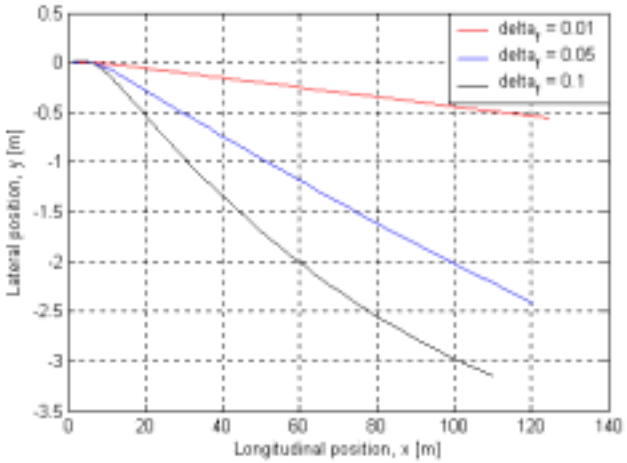

Figure 4.14 Trajectory for zero acceleration and initial velocity

From the plots given in Figure 4.13, the yaw angle is roughly linear over time for all three steering angles. For the smaller steering angle of 0.01 radians, the angular velocity response reaches a steady state value of 0.06 rad/s after 0.25 seconds while the angular velocity response exhibits an initial peak at about 0.3 seconds for larger steering angles before approaching its final value.

As shown in Figure 4.14, after 5 seconds, the vehicle has traveled a longitudinal distance of roughly 120 m. The difference in lateral position between the small and larger steering angles is more significant, almost 3 m after 5 seconds. Steering angles of less than 0.1 radians are more suitable for turning and lane-changing maneuvers.

Chapter 5

Sensor Fusion Using the Interacting Multiple Model (IMM) Algorithm

5.1 Introduction

In Automated Highway System (AHS), the safety of drivers is the most critical issue. In the same sense, tracking performance can be evaluated by how accurately and quickly we can estimate target position and velocity. In order to improve the quality of estimation, accurate, which implies expensive, sensors can be used if finances are not a concern. The alternative is to increase the number of sensors. The question to be addressed is then how to improve the accuracy of target estimation by using the limited number of sensors.

Many of the adaptive algorithms based on Kalman filter have been proposed to accomplish this task, and one of the effective methods is the Interacting Multiple Model (IMM) algorithm. It has multiple models, each of which matches to a particular mode such as uniform motion with constant velocity and maneuvering motion with acceleration. Sensor fusion using the IMM algorithm combines the estimates which are produced by equivalent numbers of filters and modes.

In this chapter, we consider a single target in planar motion utilizing three different sensors. The estimate with a single sensor is first considered to evaluate the IMM algorithm, and then sensor modeling and longitudinal/lateral motion estimate will be discussed in detail.

5.2 The IMM Algorithm

The main idea of the IMM algorithm is to weight the estimates from the filters matched to the different modes. Different modes have different state space models. The weights are based on the time variant mode probabilities that imply how close the estimate from each filter is to the corresponding model. Since the IMM algorithm mixes the estimates from different models instead of choosing which mode is true in each time step, it is called a soft switching algorithm, which does not include hard decision [5.1]. Figure 5.1 is the structure of the IMM algorithm when two models are used.

In each time step, the IMM algorithm performs four steps to yield the overall state estimates.

1. Interacting or mixing of the estimates: from the estimate, $\hat{x}_i(k-1/k-1)$ and mode probability, $\mu_i(k-1)$ of each filter in the previous step, obtain the mixing estimate, $\hat{x}_i^o(k-1/k-1)$ and covariance, $P_i{}^o(k-1/k-1)$ under the assumption that a particular mode is in effect at the present time. The mixing estimate is used as an initial state in current step.

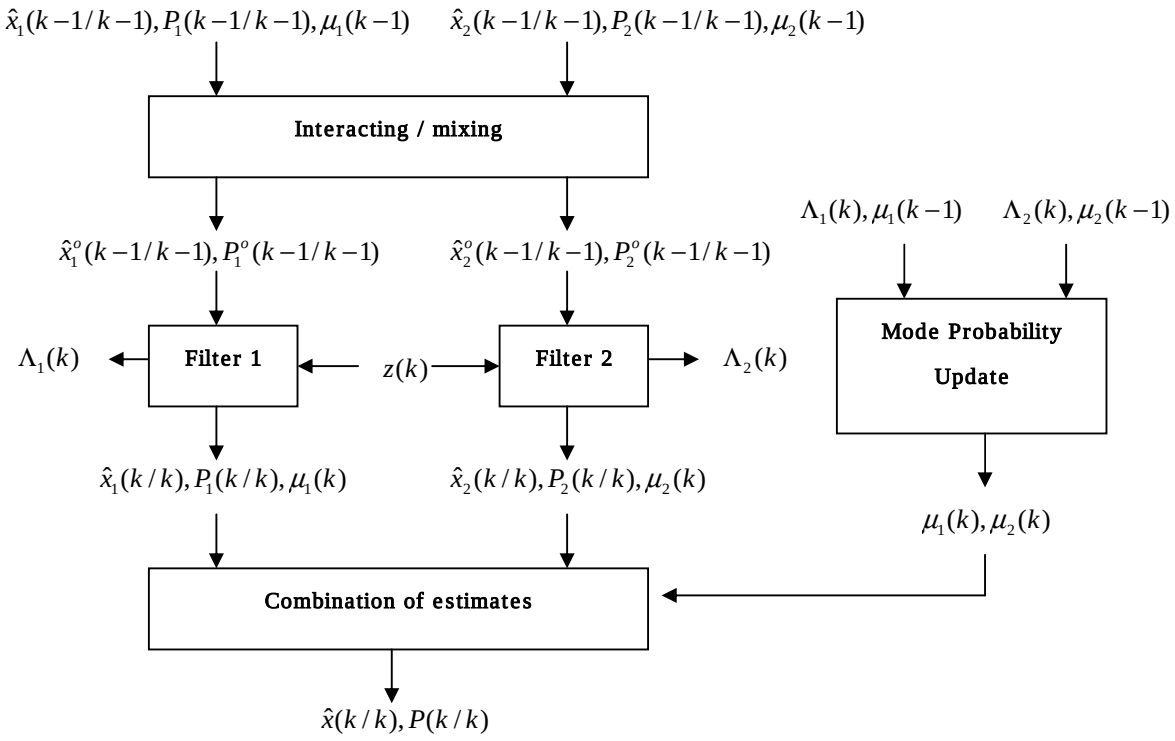

Figure 5.1 Structure of the IMM Algorithm

2. Model-conditional filtering: using the Kalman filter, update state and covariance, $\hat{x}_i(k/k), P_i(k/k)$.
3. Mode probability update: using the likelihood function, update the mode probability, $\mu_i(k)$.
4. Combination of estimates: based on the mode probability, weigh the estimate from each filter and combine them, $\hat{x}(k/k), P(k/k)$.

5.2.1 Target modeling

Since a moving target has time variant dynamics, a number of models can be established to describe a target's motion. Even though more models can give better overall estimates, it is less efficient since more time will be required to yield the estimate. In the sense of acceleration of a target, the constant velocity and acceleration modes are most commonly considered to build models. In this project, these two models are used. Linear accelerations are normally quite small and thus can be reasonably covered by a process noise in a nearly constant velocity model, i.e. the uniform motion (constant velocity model) plus a zero-mean noise with an appropriate covariance representing the small acceleration [5.2]. Alternatively, this mode can be described as a constant velocity model with no process noise. On the other hand, the acceleration mode has the acceleration increment during the sampling time and this should be included in state space model. When the state space equation is given by

$$x(k) = Fx(k-1) + Gw(k-1), \tag{5.1}$$

x is the state vector of a target defined as

$$x = [\xi \quad \dot{\xi} \quad \ddot{\xi} \quad \eta \quad \dot{\eta} \quad \ddot{\eta}]' \tag{5.2}$$

where ξ and η denote longitudinal and lateral position respectively. . In equation (5.1) w signifies process noise, which is zero-mean, white, and Gaussian with covariance $R(k)$.

The state transition matrices and the noise gain matrices are

$$F_1 = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.3}$$

and
$$F_2 = \begin{bmatrix} 1 & T & 0.5T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0.5T^2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 1 & 0 \\ 0 & 0.5T^2 \\ 0 & T \\ 0 & 1 \end{bmatrix} \tag{5.4}$$

where subscripts 1 and 2 in equations (5.3) and (5.4) denote mode 1 (constant velocity motion) and mode 2 (acceleration motion) and T is the sampling time.
The measurement model is

$$z(k) = Hx(k) + v(k) \tag{5.5}$$

$$H_1 = H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{5.6}$$

Measurement matrix implies that only position of the target is measured from each sensor. The mode sequence is assumed to be a first order Markov chain with transition probabilities

$$\pi_{ij} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{5.7}$$

5.2.2 Sensor modeling

For simulation, three sensors are modeled under the assumption that they are radars, which is most commonly used in target tracking. Since a radar measures the range and azimuth, they should be converted to the Cartesian coordinates. In simulation, predetermined target trajectory is converted to the polar coordinates and then converted to the Cartesian coordinates again after mixing with measurement noise. The most crucial value of the sensor is the standard deviation of the range measurement and the azimuth measurement, σ_r and σ_θ respectively. The approximation of the covariance of these converted measurement errors can be calculated as follows [5.2].

$$R = \mathrm{cov}[w] = \frac{\sigma_r^2 - r^2\sigma_\theta^2}{2} \begin{bmatrix} b + \cos 2\theta & \sin 2\theta \\ \sin 2\theta & b - \cos 2\theta \end{bmatrix} \tag{5.8}$$

where

$$b = \frac{\sigma_r^2 + r^2\sigma_\theta^2}{\sigma_r^2 - r^2\sigma_\theta^2} \tag{5.9}$$

Three sensors are used, each with different standard deviations of the range and azimuth measurement.

- Sensor 1 and sensor2 : $\sigma_r = \max\{0.5m, 0.05r\}$, $\sigma_\theta = 0.01rad$
- Sensor 3 : $\sigma_r = \max\{0.3m, 0.03r\}$, $\sigma_\theta = 0.005rad$

5.2.3 Data fusion

In order to fuse the estimates, the sequential process is used. After the first step of the IMM algorithm is completed using $\hat{x}_i(k-1/k-1)$ and $P_i(k-1/k-1)$, the measurement in the second step is performed with the measurement from sensor 1. Sensor 2 and sensor 3 measurements are used to update the predicted states and covariances, $\hat{x}_i(k/k)$ and $P_i(k/k)$. That is, the second step is repeated three times.

5.3 The IMM Algorithm Verification

Before applying the IMM algorithm to a planar motion target tracking, the algorithm was verified by applying to the longitudinally moving target. In this case, the measurement noise can be assumed to be determined exclusively by σ_r. The state transition matrix, noise gain matrix, and measurement matrix can be described by the following matrices:

$$F_1 = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} T \\ 1 \\ 0 \end{bmatrix}, \quad H_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{5.10}$$

$$F_2 = \begin{bmatrix} 1 & T & 0.5T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad G_1 = \begin{bmatrix} 0.5T^2 \\ T \\ 1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{5.11}$$

$$Q_2 = \mathrm{cov}(v) = 0.055^2 \tag{5.12}$$

where Q_2 is the process noise matrix for the second mode.

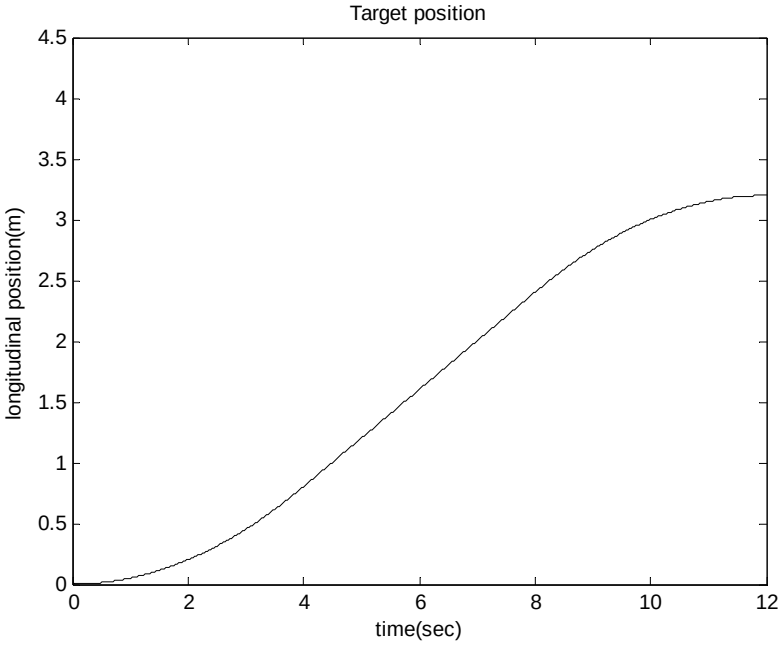

Figure 5.2 Longitudinally Moving Target Trajectory

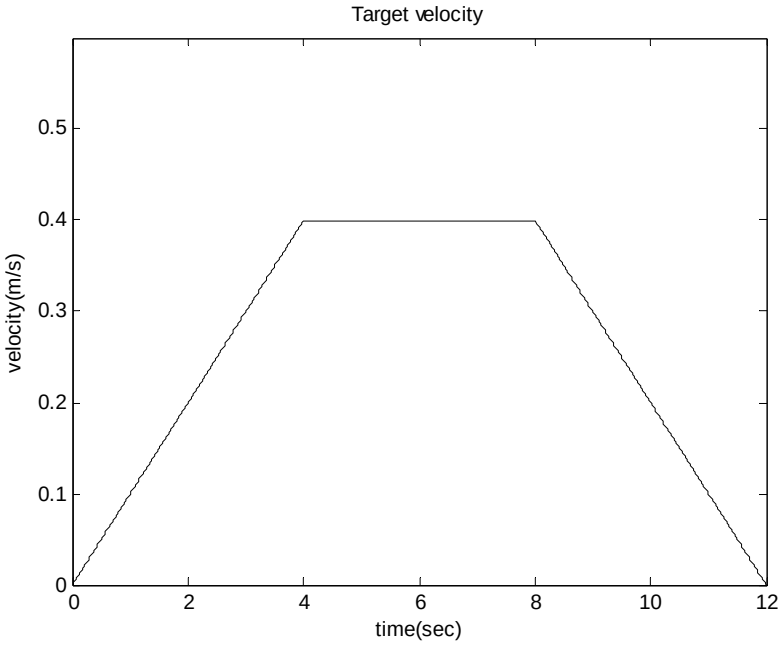

Figure 5.3 Longitudinally Moving Target Velocity

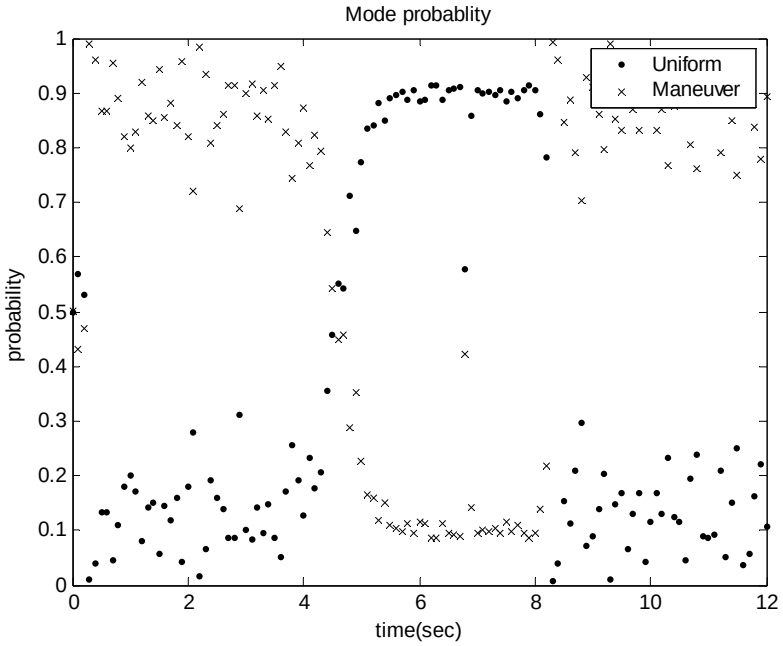

Figure 5.4 Mode Probability in Longitudinally Moving Target Tracking (without measurement noise)

5.3.1 Simulation with no measurement noise

The first simulation is performed without measurement noise. Since there is no measurement noise, perfect tracking is achieved. The most interesting part is shown in Figure 5.4. According to the target velocity plot (Figure 5.3), two mode changes occur at 4 and 8 seconds. Based on the IMM algorithm, each filter should catch the mode change and it should change mode probabilities, i.e. mode probability for uniform motion is expected to increase after 4 seconds and decrease again after 8 seconds. In Figure 5.4, the expected result is shown. Before 4 seconds, mode probability for uniform motion stays under 0.3 and mode probability for maneuvering motion stays over 0.7. Right after 4 and 8 seconds, two mode probabilities start changing and they are completely switched in 0.7 seconds. Between 5 and 8 seconds, the mode probability for uniform motion is almost 0.9.

5.3.2 Influence of transition probabilities

Initially, transition probabilities are set as $\pi_{11} = 0.95, \pi_{12} = 0.05, \pi_{21} = 0.05, \pi_{22} = 0.95$ and the simulation result is shown in Figure 5.4. Since it is assumed that the probability that a target keeps its current mode is high, mode changes occur in short period and clearly presented in Figure 5.4. On the other hand, when different values are chosen as transition probabilities,

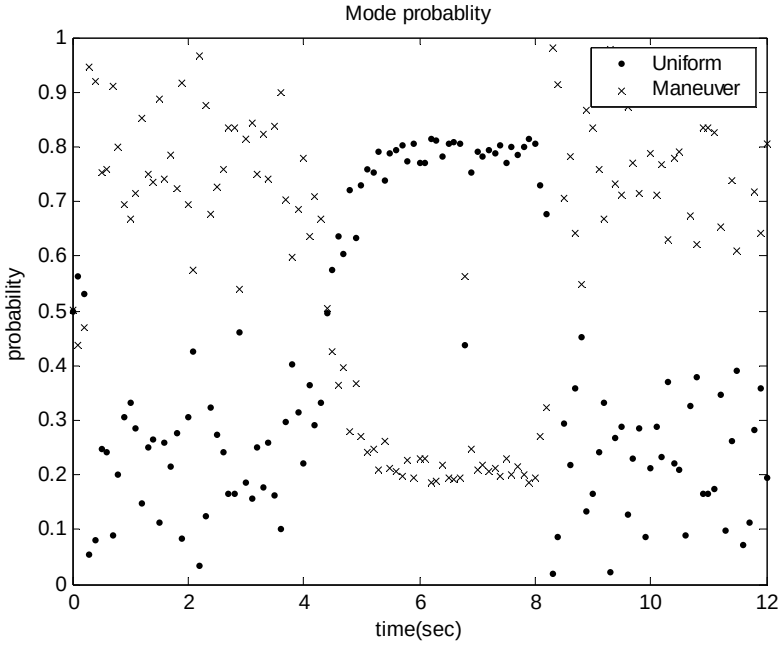

Figure 5.5 Mode Probability : $\pi_{11} = 0.9, \pi_{12} = 0.1, \pi_{21} = 0.1, \pi_{22} = 0.9$

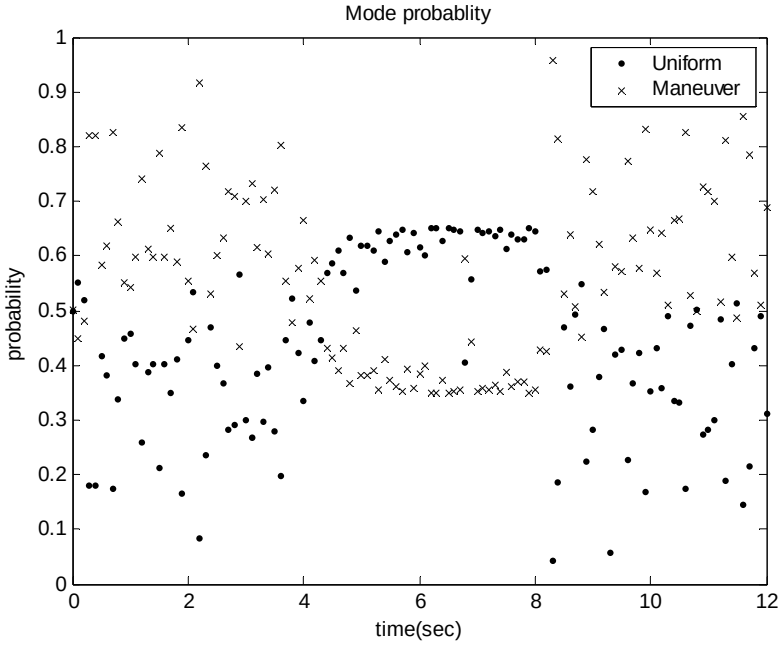

Figure 5.6 Mode Probability : $\pi_{11} = 0.8, \pi_{12} = 0.2, \pi_{21} = 0.2, \pi_{22} = 0.8$

the mode changes have different characteristics. As expected, the difference between marginal values of each mode probability decreases as transition probabilities between two modes increases. In Figure 5.6, two mode probabilities during uniform motion (0 ~ 4 seconds and 8 ~ 12 seconds) are almost mixed, which makes it delicate to distinguish the current mode from the figure. In addition, transient response speed changes. In the first simulation, the actual mode change occurs at 4 seconds but the conversion point is at 4.8 seconds. That is to say, there is 0.8 second time delay. However, this time delay is vanishing as different transition probabilities are adapted. In the second and third simulations (Figure 5.5 and Figure 5.6), time delays are 0.4 and 0.1 seconds, respectively. Even though these three simulations assumed that there is no measurement noise, it helps us understand the influence of transition probabilities. The next chapter deals with the case when there exists measurement noise, and it shows that the result is more complex and more uncertain.

5.3.3 Simulation with measurement noise

The IMM algorithm applied the same situation as above, except with the addition of measurement noise. Measurement noise prevents perfect tracking and introduces RMS position error into the estimation of true target position. As mentioned in the previous chapter, the mode probabilities shown in Figure 5.8 do not follow the same pattern as in Figure 5.4. A criterion of tracking performance is RMS position error which is defined as the difference between actual

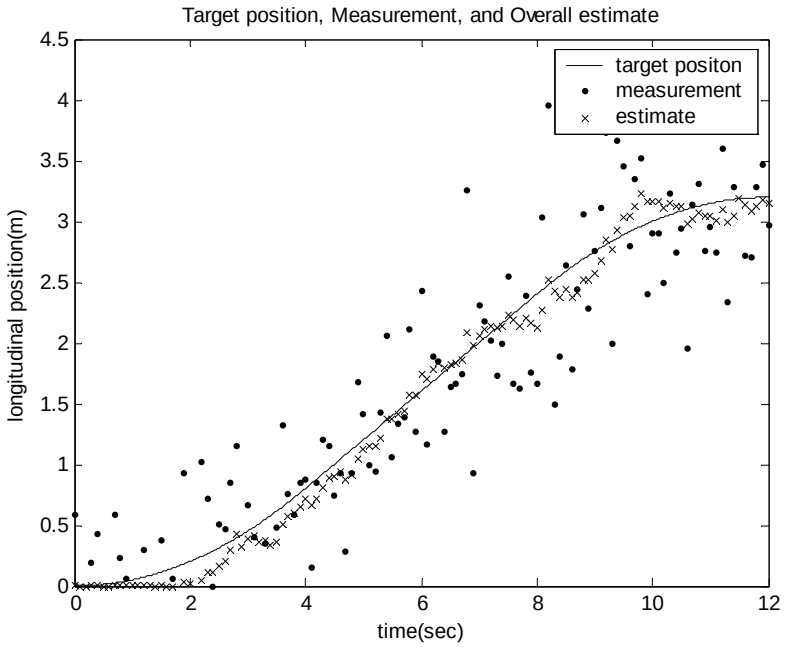

Figure 5.7 Target position, measurement, and overall estimate in longitudinal target tracking (with measurement noise)

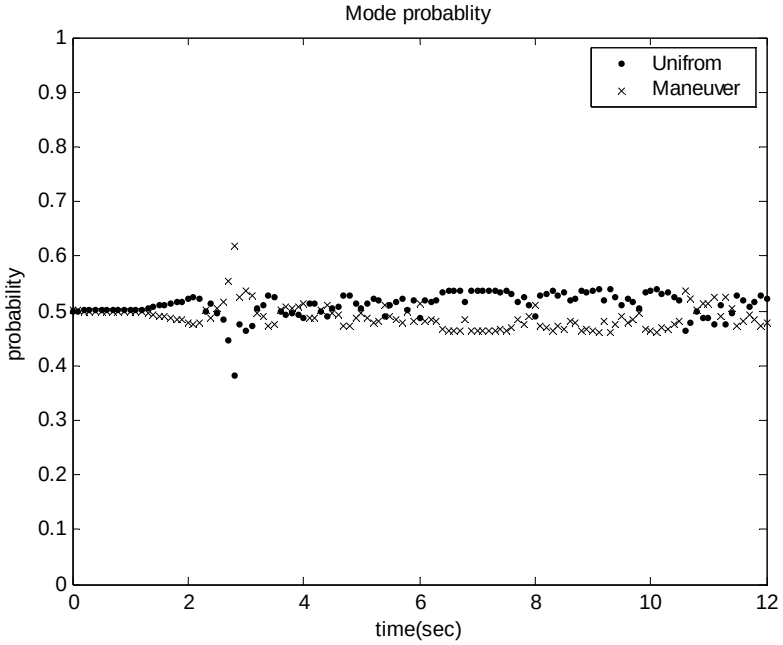

Figure 5.8 Mode probability in longitudinal target tracking (with measurement noise)

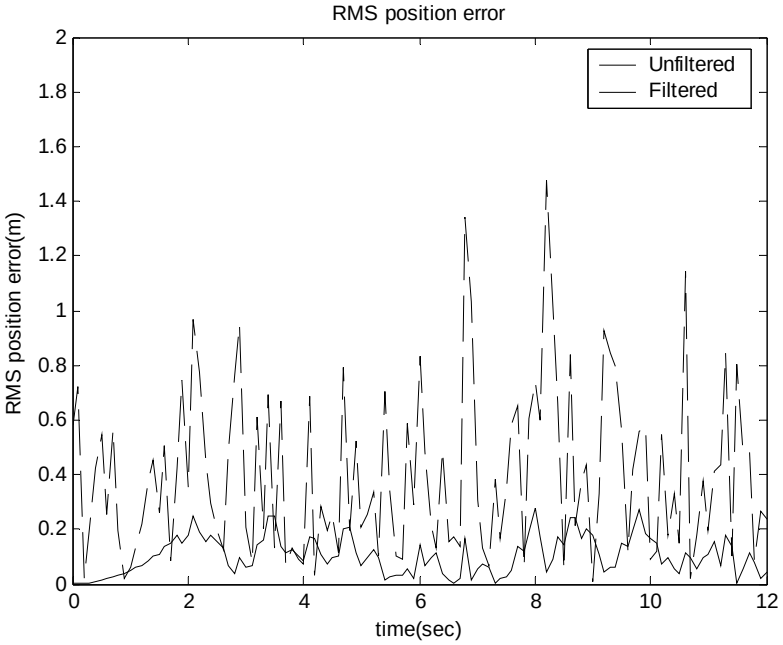

Figure 5.9 RMS position error in longitudinal target tracking (with measurement noise)

target position and estimated target position using the filter. For the first two seconds in Figure 5.7, the estimated values stay at zero and it is due to the difference between the target's actual initial condition and the model's initial condition. Another important result is that this state space model fits well for the target's uniform motion. Almost perfect target tracking during the target's uniform motion is achieved. This result can be checked in terms of RMS position error in Figure 5.9 as well. Overall, the filtered estimates have 25% of the RMS position error associated with unfiltered values.

5.4 Simulation Results

The same Matlab code for the IMM algorithm in chapter 4 is used to simulate the planar motion target tracking. Figure 5.10 shows the trajectory of the target and Figure 5.11 displays the longitudinal and lateral velocities of the target. These plots show that a relatively slow lane change from the first lane to the second lane, which is in front of the Adaptive Cruise Control (ACC) vehicle, occurs at 3.5 seconds. The maximum longitudinal acceleration is $2m/s^2$, and the maximum lateral acceleration is $1m/s^2$. Since the process noise covariance is determined based on the maximum acceleration increment over a sampling time, the maximum acceleration is crucial in target modeling. When slightly different process noise covariance is adopted, it can be verified that RMS position error blows up.

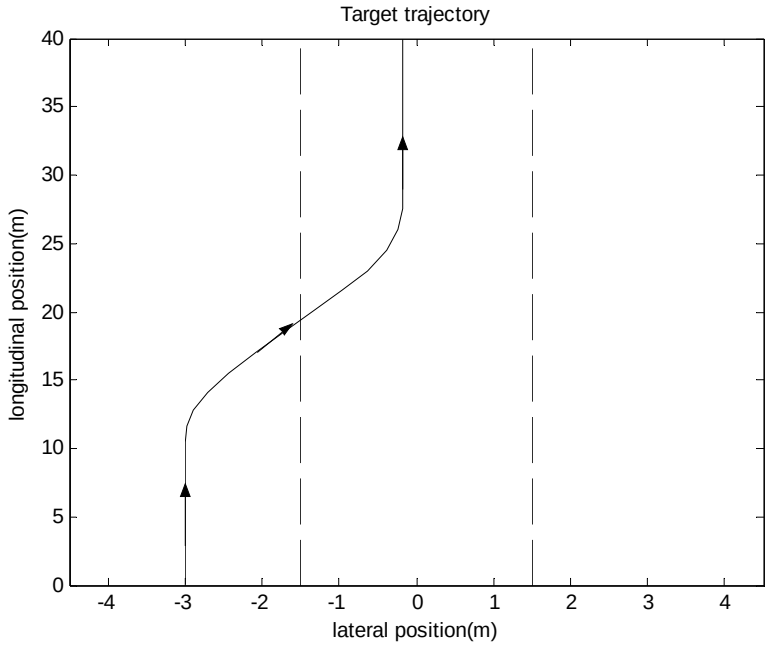

Figure 5.10 Trajectory of the planar motion target

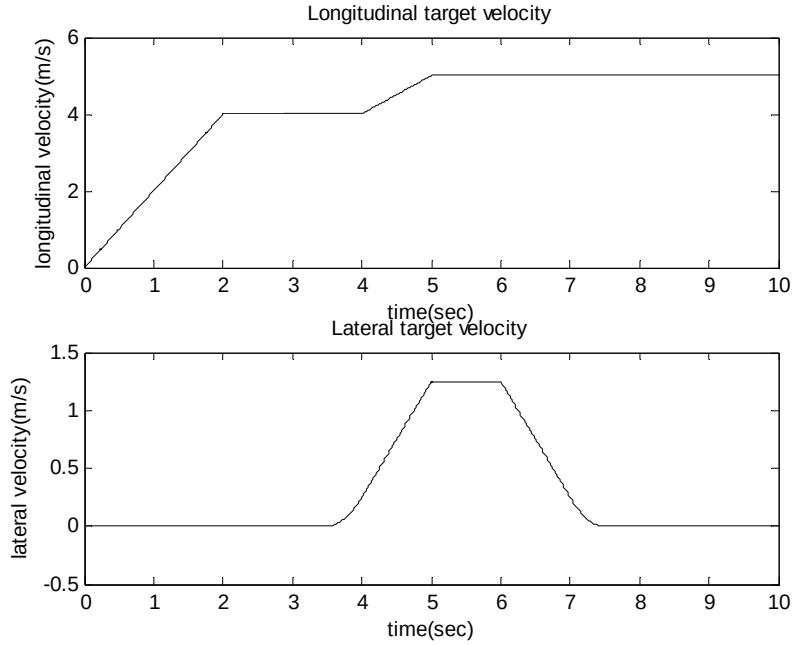

Figure 5.11 Target velocity

In the following simulation, sampling time T is set at 0.1 seconds and sensor scanning time is 0.02 seconds. Process noise covariance for each model is determined by the following matrix form:

$$Q_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times 0.5^2 \tag{5.13}$$

5.4.1 Single sensor estimation results

Target tracking is performed using sensor 1. The simulation results are shown in Figures 5.12, 5.13 and 5.14. In Figure 5.12, estimates during the first several seconds have much error. It can also be checked from the fact that the unfiltered RMS position error in Figure 5.14 is smaller than the filtered for the first 3 seconds. It is not only due to the difference between the target's actual initial condition and the model's initial condition but also due to the assumption in the sensor model. According to equation (5.8), measurement error covariance is a function of the azimuth θ. Actually, the field of view of radar is very narrow ($\pm 5^\circ$) and multiple sensors should be used to cover $\pm 90^\circ$. However, in sensor modeling, it is assumed that a single sensor covers $\pm 90^\circ$. This assumption causes large measurement error

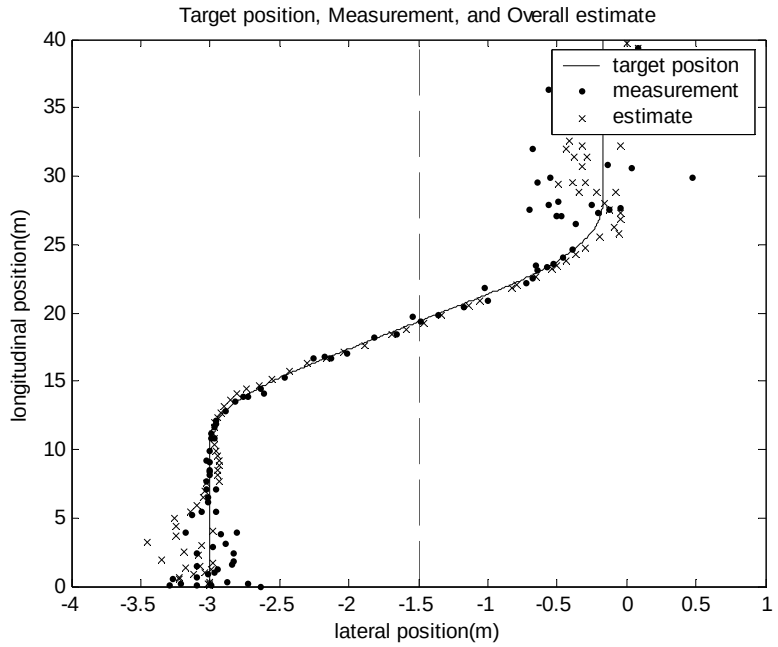

Figure 5.12 Target position, measurement, and overall estimate in longitudinal/lateral target tracking (with single sensor)

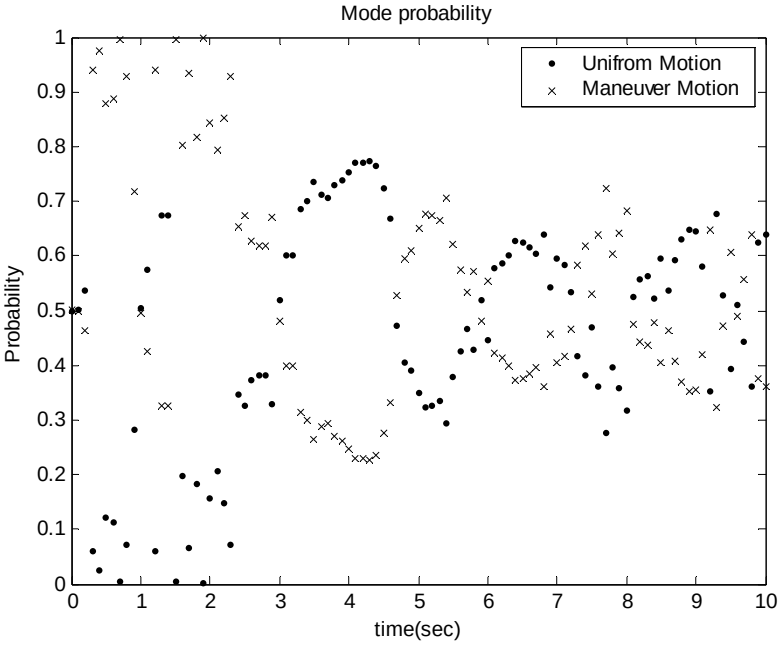

Figure 5.13 Mode probability in longitudinal/lateral target tracking (with single sensor)

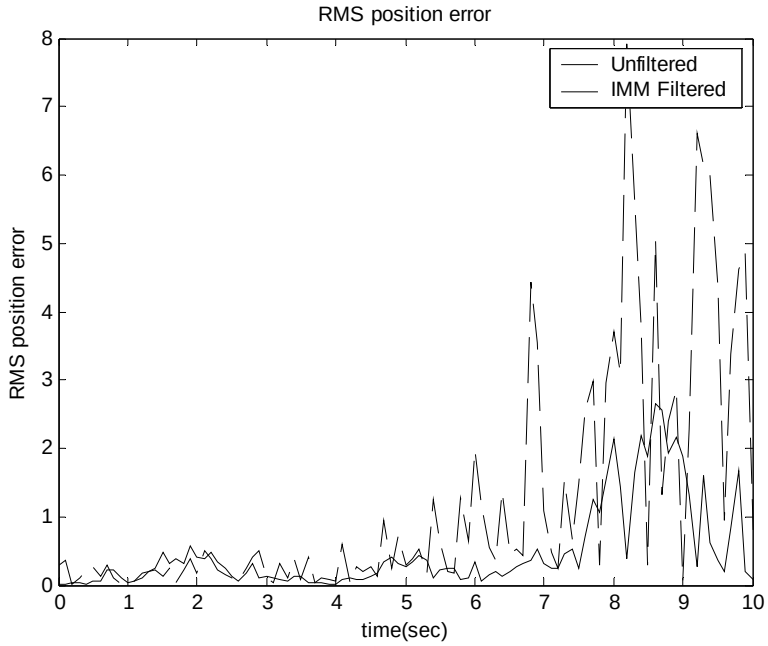

Figure 5.14 RMS position error in longitudinal/lateral target tracking (with single sensor)

covariance when azimuth is large and affects RMS position error. In the same sense, estimates are increasing as time goes by because the distance between the target and ACC vehicle is getting larger. As both longitudinal acceleration and lateral acceleration are considered, total mode changes occur six times at 2, 3.5, 4, 5, 6, and 7.5 seconds. The change of two accelerations makes the mode probability analysis more complicated. The fact that the target is moving with constant velocity after 7.5 seconds can be barely checked from Figure 5.13. In spite of complex mode probabilities, overall RMS position error is decreased by 28%. The maximum RMS position error is 2.7 m at 8.7 seconds.

5.4.2 Three sensor estimation results

Three different sensors are used to perform the final simulation of the IMM algorithm. Sensor 1 and sensor 2 have the same measurement error covariance and sensor 3 has a smaller one. As shown in Figures 5.15b, 5.16b, and 5.17b, measurement error increases as the distance of target and ACC vehicle increases. The maximum RMS position errors of unfiltered measurements using each sensor are 6.7m, 5.2m, and 2.6m, respectively. These values are large enough to threaten the safety of the driver and the vehicle if there is no filtering. After the IMM filtering and data fusion, RMS position error definitively decreases by less than 0.5m.

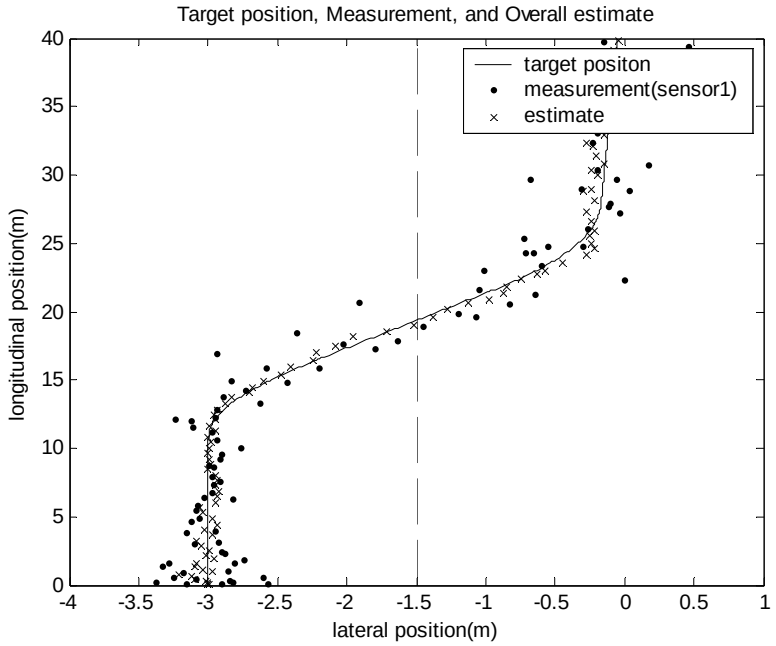

Figure 5.15a Target position, measurement using sensor 1, and overall estimate in longitudinal/lateral target tracking

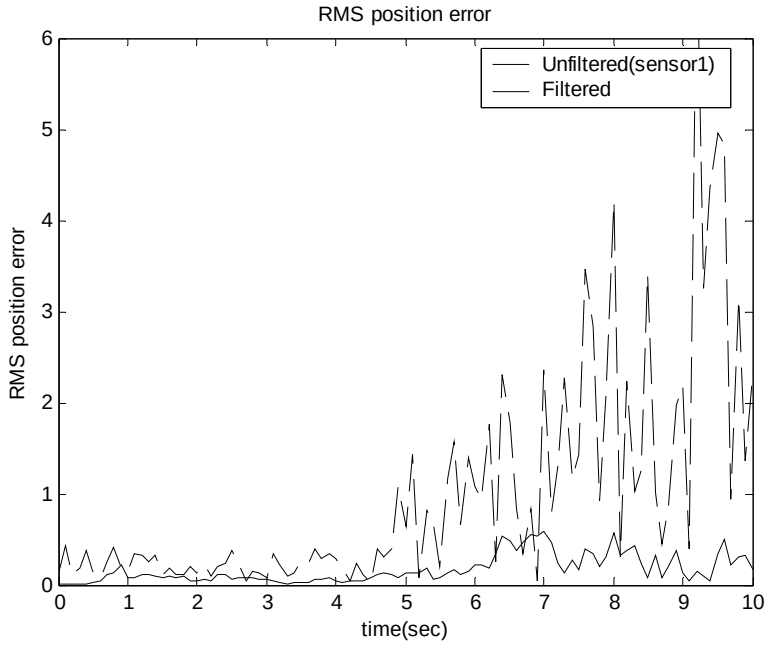

Figure 5.15b RMS position error in longitudinal/lateral target tracking (unfiltered measurement using sensor 1 vs. filtered estimate using three sensors)

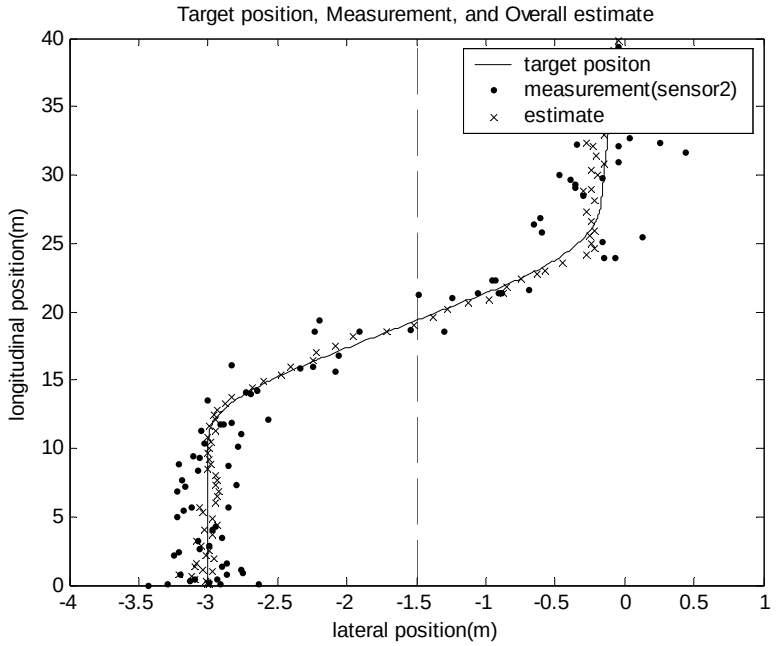

Figure 5.16a Target position, measurement using sensor 2, and overall estimate in longitudinal/lateral target tracking

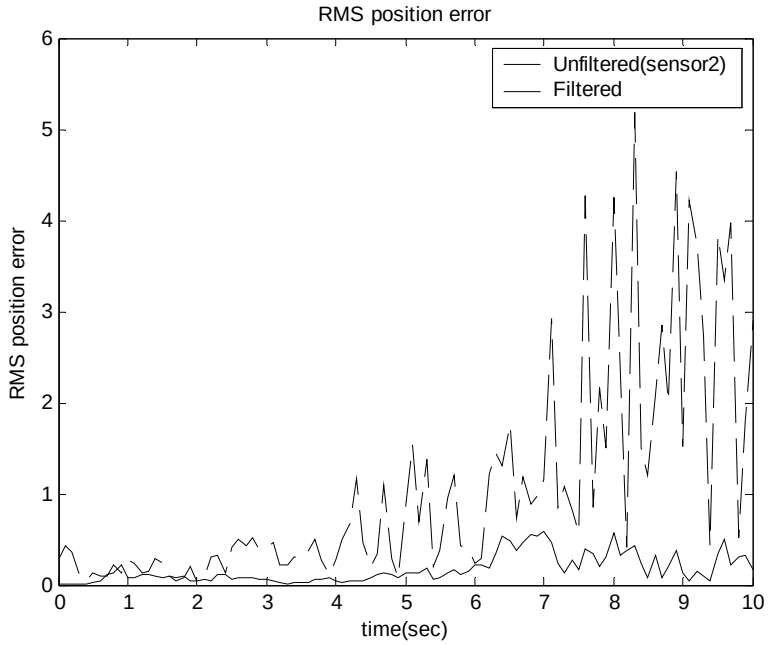

Figure 5.16b RMS position error in longitudinal/lateral target tracking (unfiltered measurement using sensor 2 vs. filtered estimate using three sensors)

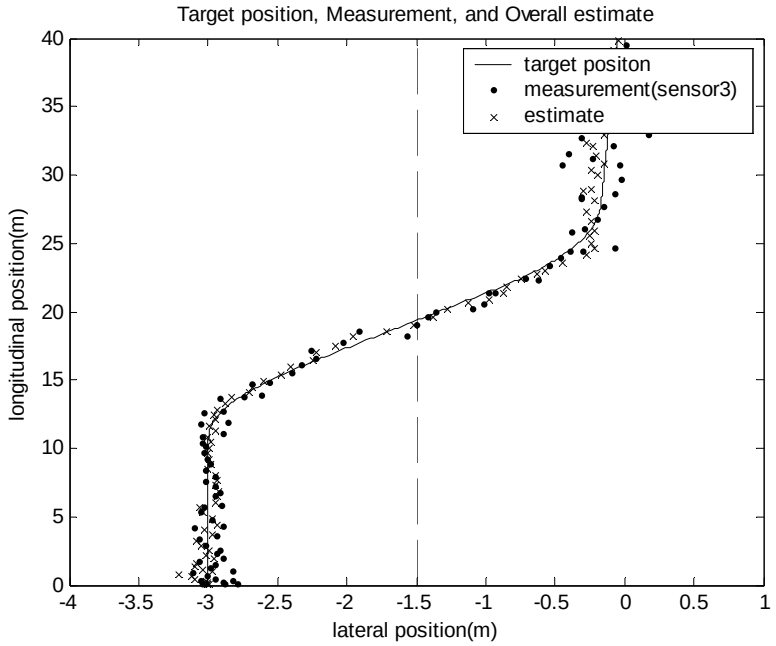

Figure 5.17a Target position, measurement using sensor 3, and overall estimate in longitudinal/lateral target tracking

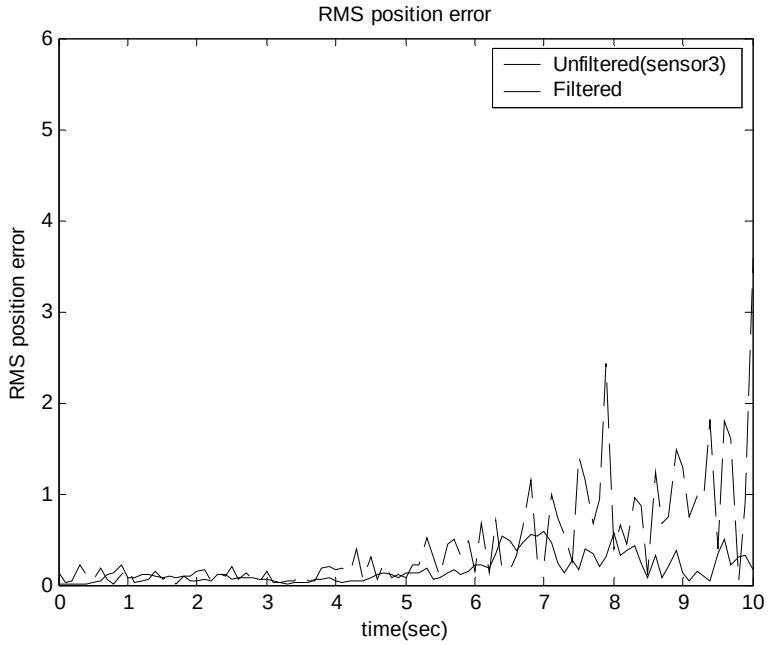

Figure 5.17b RMS position error in longitudinal/lateral target tracking (unfiltered measurement using sensor 3 vs. filtered estimate using three sensors)

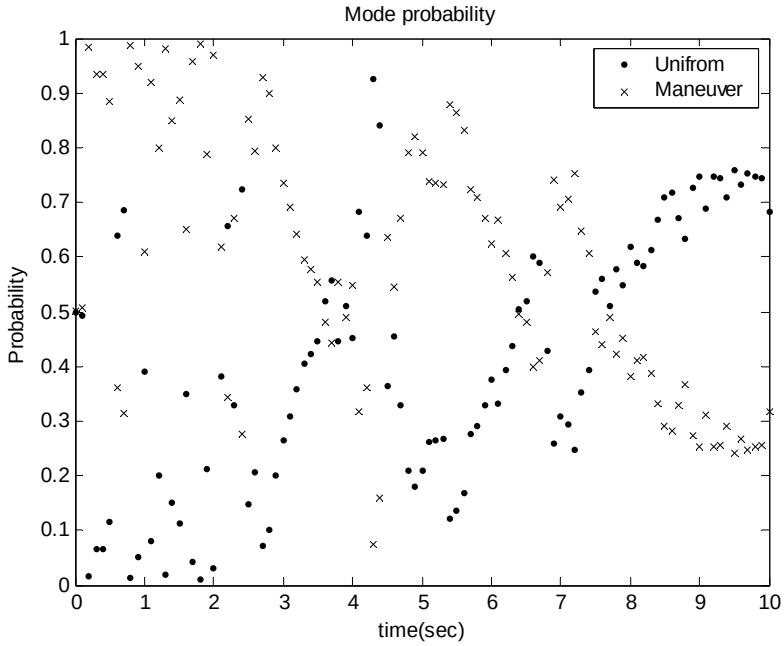

Figure 5.18 Mode probability in longitudinal/lateral target tracking (with three sensors)

when compared with single sensor estimation (max. RMS position error = 2.7m in Figure 5.5), sensor fusion with three sensors provides a more accurate estimation (max. RMS position error = 0.8m).

Chapter 6

Conclusion and Future Work

Applied optimal estimation techniques such as the Kalman filter allow one to obtain quality state estimates for a stochastic system corrupted by input and measurement noise. When more than one sensor is availble, the quality of the state estimates can be further increased with the PDA method as it assigns a weight to the measurement residual for each sensor measurement. With the use of multiple and redundant sensors, the PDAF can be constructed as a set of sequential Kalman filters and the accuracy of the fused estimates is maintained even with the permanent or intermittent loss of sensor measurement data.

Longitudinal control of vehicles in leader-follower mode is simple to implement and requires small computational ressources. Lateral control, however, is more difficult to achieve due to the non-linear equations of motion goverened by the vehicle dynamics. The PDA method can then be applied to the expanded, three-dimensional vehicle model to obtain fused estimates for the lateral and angular states. Additional sensors, such as magnetic road markers embedded in certain freeways, DGPS, and accelerameters, can also be used to achieve lateral control.

Highway safety, traffic, and congestion issues have motivated a significant amount of research in the area of vehicle and highway automation. However, this research cannot become viable without reliable fault diagnostics and fault-handling systems. Both semi-automated vehicles (such as the cases of adaptive cruise control and lane-keeping algorithms) as well as fully automated vehicles on dedicated automated highway systems have been studied. Further research should be conducted towards the development of a fault diagnostic system that can monitor the health of the sensors and actuators on an automated vehicle and identify the source of any fault that occurs.

Fault diagnostic design consists of two primary tasks: fault detection and fault identification. Fault detection can generally be performed by setting a threshold value that indicates the maximum allowable size of the measurement residual before a fault is declared. Stochastic change detection algorithms such as the Kalman Filter and the Probabilistic Data Association Filter (PDAF) are quite commonly used for fault detection. For increased robustness, adaptive thresholds and parameter estimation algorithms (PAA) are also used in order to explicitly account for modeling error.

Identification of faults, however, is a more complex problem since each fault must have a unique effect upon the residuals in order to perform correct identification. Many of the residual generation design techniques enforce this condition or the stronger condition that the fault effects on the residuals are independent of each other. Hypothesis testing, heuristics, and pattern recognition are useful techniques for fault identification. Fuzzy logic and knowledge-base methods can also be applied.

The challenge of radar health monitoring, for example, should be studied in order to obtain increased radar reliability. The use of an inexpensive and redundant sensor framework is extremely valuable in the radar health monitoring task. Another important problem to be considered is the loss of targets on sharp curves and on grades since a typical range and range-rate sensor such as a radar may lose track of targets on sharp curves or return contaminated measurements.

To reducing the target tracking error, the sensor fusion technique and the Interacting Multiple Model (IMM) algorithm can be combined. Since the Kalman filter based on a single state space model has a defect in the case that a target changes its mode, the IMM algorithm using more than two different models is inevitable. Even though sensor fusion and IMM algorithm are totally different techniques, these can cooperate to provide the optimal estimates. By comparing the simulation results of unfiltered measurements and filtered estimates, it shows that the RMS position error is reduced by 28% by using a single sensor and the IMM algorithm. The error reduction is greater when three sensor data are fused. The advantage in using the IMM algorithm is not only error reduction but also mode prediction. When the mode probability data can be analyzed accurately, the motion of a target can be predicted, which prevents collision between the leading vehicle and following vehicle.

This study is focused on a single target tracking problem. However, in many applications, the IMM algorithm is commonly used in multi target tracking. So the future work should be extended to the multiple target tracking with multiple sensors. In addition, if the fault modes are modeled and put into the IMM algorithm, fault detection and fault diagnosis can also be achieved.

Bibliography

[2.1] Welch, G. and Bishop, G. An Introduction to the Kalman Filter, TR 95-041 Depart ment of Computer Science.

[2.2] Bar-Shalom, Y. and Li, R.L., *Estimation and Tracking : Principles, Techniques, and Software*, Artech House, Norwood MA, 1993

[2.3] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J., "Interacting Multiple Model Methods in Tracking: A Surbey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no.1, pp 103-123, Jan. 1998.

[3.1] Bar-Shalom, Y., & Tse, E., "Tracking in a Cluttered Environment with Probabilistic Data Association," *Automation*, vol. 11, Sept. 1975, pp. 451-460.

[3.2] Fortmann, T.E., Bar-Shalom, Y., & Scheffe, M., "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," IEEE *Ocean Engineering*, vol. OE-8, July. 1983, pp. 173-184.

[3.3] Pao, L., & Frei, C.W., "A Comparison of Parallel and Sequential Implementations of a Multisensor Multitarget Tracking Algorithm," Proceedings of the American Control Conference, Seattle, WA, June 1995, 1683-1687.

[3.4] Bar-Shalom, Y., "Tracking Methods in a Multitarget Environment," IEEE Transaction *Automated Control*, vol. AC-23, Aug. 1978, pp.618-626.

[3.5] Bar-Shalom, Y., *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House,

Norwood, MA, 1990.

[4.1] Gillepsie, T.D., *Fundamentals of Vehicle Dynamics*, Society of Automotive Engineers, Warrendale, PA, 1992, p. 196.

[4.2] "Vehicle Dynamics Terminology," SAEJ670E, Society of Automotive Engineers, Warrendale, PA, 1992, Appendix A.

[4.3] Cole, D.E., *Elementary Vehicle Dynamics*, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 1971.

[4.4] Wong, J.Y., *Theory of Ground Vehicles*, Wiley-Interscience, Ottawa, Ontario, 1978, p. 216.

[4.5] Gillepsie, T.D., *Fundamentals of Vehicle Dynamics*, Society of Automotive Engineers, Warrendale, PA, 1992, pp. 198-199.

[4.6] Nordeen, D.L., and Cortese, A.D., "Force and Moment Characteristics of Rolling Tires," SAE 640028 (713A), 1963, p. 13.

[4.7] Steeds, W., *Mechanics of Road Vehicles*, Iliffe & Sons, London, England, 1960.

[4.8] Kosecka, J., Blasi, R., Taylor, C.J., & Malik, J., "Vision-Based Lateral Control of Vehicles," Department of Electrical Engineering, University of California at Berkeley, CA, 1997

[5.1] Caveney, D. and Hedrick, J.K., "Multiple target tracking in the adaptive cruise control environment using multiple models and probabilistic data association," *In Proceedings of ASME IMECE / Design Engineering-23274*, New york, NY, 2001.

[5.2] Li, R.L. and Bar-Shalom, Y., "Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking," *IEEE Trans. Control Syst. Tech.*, Vol.1, no.3, pp. 186-194, Sept. 1993.