

# A Continuum of Models for Stochastic Estimation

20 June 2000

Jeffery Layne, Ph.D. and Scott Weaver Ph.D..

United States Air Force

AFRL/SNAT

2241 Avionics Circle

WPAFB, Ohio 45433-7318

## ABSTRACT

In this paper, we investigate a recursive multiple model tracking approach similar to the Generalized Pseudo-Bayesian 1 (GPB1) [1] approach. However, here we consider a continuum of models rather than the discrete set that is usually implemented in the GPB1 method. By doing so better models are available to improve tracker performance and solve the bias problem inherent in most multiple model approaches.

## 1 Introduction

In tracking systems, we don't have precise knowledge of the system model. For this reason, crude models such as constant velocity, constant acceleration, and constant turn rate models have been developed. However, to capture scenarios that are not represented by the given model, much uncertainty is added to the model in the form of large process noise. To reduce this uncertainty, researchers have implemented combinations of many different models in multiple model approaches. However, our research has shown that most designs still require large uncertainties to capture the wide possibilities of target maneuvers. Consequently, tracking systems tend to ride the measurements (that is, the estimate error is about the same as the measurements) as little weight is placed on the model's prediction and much weight is given to the measurements.

However, by choosing a set of models that lie upon some continuum of models (as proposed in this paper) the uncertainty in each model may be reduced,

## Report Documentation Page

<b>Report Date</b> 20062000	<b>Report Type</b> N/A	<b>Dates Covered (from... to)</b> -
<b>Title and Subtitle</b> A Continuum of Models for Stochastic Estimation	<b>Contract Number</b>	
	<b>Grant Number</b>	
	<b>Program Element Number</b>	
<b>Author(s)</b> Layne, Jeffery; Weaver, Scott	<b>Project Number</b>	
	<b>Task Number</b>	
	<b>Work Unit Number</b>	
<b>Performing Organization Name(s) and Address(es)</b> United States Air Force AFRL/SNAT 2241 Avionics Circle WPAFB, Ohio 45433-7318	<b>Performing Organization Report Number</b>	
<b>Sponsoring/Monitoring Agency Name(s) and Address(es)</b> Director, CECOM RDEC Night Vision and Electronic Sensors Directorate, Security Team 10221 Burbeck Road Ft. Belvoir, VA 22060-5806	<b>Sponsor/Monitor's Acronym(s)</b>	
	<b>Sponsor/Monitor's Report Number(s)</b>	
<b>Distribution/Availability Statement</b> Approved for public release, distribution unlimited		
<b>Supplementary Notes</b>		
<b>Abstract</b>		
<b>Subject Terms</b>		
<b>Report Classification</b> unclassified	<b>Classification of this page</b> unclassified	
<b>Classification of Abstract</b> unclassified	<b>Limitation of Abstract</b> UNLIMITED	
<b>Number of Pages</b> 14		

allowing the filter to appropriately give more weight to the filters estimate and reducing the overall estimate error. Further, by having the models densely spaced the system does not have to interpolate between sparsely spaced models thus improving the performance when operating between system models. This is one problem that this paper addresses.

Another problem we have found with multiple model approaches is due to “non-symmetry” of the models. For example, suppose we have 3 constant velocity models with acceleration biases of  $-1g$ ,  $0g$ , and  $+1g$ . Further suppose that the current maneuver is  $.6g$ . In this situation models  $0g$  and  $1g$  compete and the likelihood of the  $-1g$  models are not necessarily zero. Consequently, these models will bias the result towards the result obtained for the  $-1g$  and  $0g$  accelerations. However, by implementing a continuum of models we maintain symmetry about the true maneuver. A similar phenomenon was first observed in [2] where a large number of models compete thus reducing performance as an increased number of models are added.

Further too few model leave modeling gaps and higher uncertainty. The problem studied here is to reduce the biases of the IMM constant when compared to the Stochastic Estimation using a Continuum of Models

## **2 Algorithm development**

The continuum-model algorithm derivation is similar to the derivation of the GPB1 algorithm. The difference is that the GPB1 processes each of its (finite number of) models separately, whereas in the continuum-model algorithm the infinite number of Kalman filter models are evaluated together, taking advantage of an assumed structure that relates all the models in the continuum.

In the development that follows, we apply the same procedures used in developing the GPB1 algorithm except a continuum of models is employed. The steps follow the development presented in [1].

### **2.1 Constant position model with velocity bias**

In this continuum-model algorithm derivation we assume knowledge of the truth model with the exception of the object's time-varying velocity. The system truth

model is assumed to be a known constant position model with and unknown velocity bias given as

$$x(k) = x(k-1) + T v_k + \omega(k-1) \quad \omega \sim N(0, Q), \quad T \in \mathbb{R} \quad (1)$$

$$z(k) = H x(k) + v(k-1) \quad v \sim N(0, R), \quad H \in \mathbb{R} \quad (2)$$

where  $\omega$  and  $v$  are zero-mean, Gaussian, white noise and  $T$  is the time step between measurements updates.

By implementing a continuum of (constant-position with velocity bias) Kalman filter models whose velocities span across all possible target velocities  $v_k \in [-\infty, \infty]$ , we know that one of the unaccountably infinite numbers of models will match the truth model exactly.

We will demonstrate that when propagating the current state estimate (and associated covariance) through the continuum of Kalman filter models, the state estimate for a given time-step will remain in the form

$$\hat{x}(k-1, v_{k-1} | k-1) = a(k-1 | k-1) v_{k-1} + b(k-1 | k-1) \quad (3)$$

where  $a(k-1 | k-1)$ ,  $b(k-1 | k-1) \in \mathbb{R}$  are the slope and intercept respectively of the affine function of  $v_{k-1}$ . Because  $a(k-1 | k-1)$  and  $b(k-1 | k-1)$  are independent of  $v_{k-1}$ , the form given in (3) allows us to analytically describe  $\hat{x}(k-1, v_{k-1} | k-1)$  through the computation of  $a(k-1 | k-1)$  and  $b(k-1 | k-1)$  without requiring knowledge of  $v_{k-1}$ . This compact description allows us to perform state estimation (for the next time-step) through each of the infinite number of models in the continuum simultaneously, simply by computing  $a(k | k)$  and  $b(k | k)$ . Similarities can readily be seen between the form given in (3) and the system model given in (1).

## 2.2 Propagation of State Estimate

Using the standard Kalman filter notation and derivation, we recall that the propagation of the state estimate from time  $k-1$  to time  $k$  applied to the system model (1) and (2) is

$$\hat{x}(k, v_k | k-1) = \hat{x}(k-1, v_{k-1} | k-1) + T v_k \quad (4)$$

where  $\hat{x}(k-1, v_{k-1} | k-1)$  is available from the previous time-steps computation. By letting

$$a(k | k-1) = T, \quad (5)$$

$$b(k | k-1) = \hat{x}(k-1, v_{k-1} | k-1) \quad (6)$$

we rewrite (4) as

$$\hat{x}(k, v_k | k-1) = a(k | k-1) v_k + b(k | k-1) \quad (7)$$

where the velocity assumed to be in affect on the previous time-step has been incorporated into  $b(k|k-1)$ . Therefore after propogation, the state estimate is still an affine function of the current velocity bias,  $v_k$ .

### 2.3 Update of state estimate

Recall that a measurement update of the state estimate at time  $k$  applied to the system model is

$$\hat{x}(k, v_k | k) = \hat{x}(k, v_k | k-1) + W(k)[z(k) - H \hat{x}(k, v_k | k-1)] \quad (8)$$

and substituting twice using (7) results in

$$\begin{aligned} \hat{x}(k, v_k | k) = & a(k | k-1) [1 - W(k)H] v_k \\ & + b(k | k-1) + W(k) [z(k) - H b(k | k-1)] \end{aligned} \quad (9)$$

where  $W(k)$  is the Kalman gain. Therefore, after measurement update of the state estimate, it is still an affine function of the velocity bias with slope and intercept

$$a(k | k) = a(k | k-1) [1 - W(k)H] \quad (10)$$

$$b(k | k) = b(k | k-1) + W(k)[z(k) - H b(k | k-1)] \quad (11)$$

respectively, yielding the same form assume in (3). This tells us by induction that if the initial state estimate is an affine function of the velocity bias then it will always remain so after propogation and update through the continuum of filters. Now we show how the error covariance estimate also retains this affine structure through the propogation and update steps.

## 2.4 Propagation and update of error covariance estimate

Propagation and update of the covariance  $P(k-1|k-1) \in \mathbb{R}$  is simpler than that just performed for the state estimate because not only is the covariance independent of the current velocity (as was  $a(k-1|k-1)$  and  $b(k-1|k-1)$ ) it is independent of velocity at any time.

Recall that the propagation of the state error covariance estimate to time  $k$  applied to the system is given by the following:

$$P(k | k-1) = P(k-1 | k-1) + Q \quad (12)$$

and the measurement update of the state error covariance at time  $k$  is given by

$$P(k | k) = P(k | k-1) - W(k)^2 S(k) \quad (13)$$

where the innovation covariance at update time is

$$S(k) = H^2 P(k | k-1) + R \quad (14)$$

and the Kalman Gain is

$$W(k) = P(k | k-1) H S(k)^{-1} \quad (15)$$

$$= \frac{P(k | k-1) H}{H^2 P(k | k-1) + R} \quad (16)$$

After update, the state error covariance is given by

$$P(k | k) = P(k | k-1) - \frac{(P(k | k-1) H)^2}{H^2 P(k | k-1) + R} \quad (17)$$

which is clearly independent of velocity.

This derivation of the propagation and update of the error covariance estimate tells us by induction that if the initial error covariance on a continuum of filters is independent of velocity then it will remain so. We follow along the same procedure as GPB1 as this architecture does destroyed constant nature of the state error covariance at some time

k. Hence by induction if covariance is constant (independent of the velocity) then it will remain so.

## 2.6 Match filtering

The likelihood function to the continuum of velocity biases as a function of the measurement at time k is given by

$$\Lambda(k, v_k) = p[z(k) | v_k, Z(k-1)] \quad (18)$$

$$= p[z(k) | v_k, \hat{x}(k-1, v_{k-1} | k-1), P(k-1|k-1)] \quad (19)$$

where  $Z(k) = \{z(k), \dots, z(0)\}$  is the set of all measurement through time k. The best estimated mean of  $z(k)$  using  $\hat{x}(k-1, v_{k-1} | k-1)$ ,  $P(k-1|k-1)$ , and the model used to propagate them, is  $H\hat{x}(k, v_k | k-1)$  giving the Gaussian probability density function.

$$\Lambda(k, v_k) = \frac{1}{\sqrt{2\pi S(k)}} \exp\left[-\frac{[z(k) - H\hat{x}(k, v_k | k-1)]^2}{2S(k)}\right] \quad (20)$$

$$= \frac{1}{\sqrt{2\pi S(k)}} \exp\left[-\frac{[z(k) - H_a(k|k-1)v_k - H_b(k|k-1)]^2}{2S(k)}\right] \quad (21)$$

## 2.7 Probability update for $v_k$

The following is a recursion that updates the probability density function for a target-exhibiting behavior from a model in the continuum with velocity  $v_k$

$$\mu(v_k) = p[v_k | Z(k)] \quad (22)$$

$$= p[v_k | z(k), Z(k-1)] \quad (23)$$

$$= \frac{1}{c} p[z(k) | v_k, Z(k-1)] p[v_k | Z(k-1)] \quad (24)$$

$$= \frac{1}{c} \Lambda(k, v_k) \int_{-\infty}^{\infty} p[v_k | v_{k-1}, Z(k-1)] p[v_{k-1} | Z(k-1)] dv_{k-1} \quad (25)$$

where the last equation makes use of the total probability theorem and c is the normalization constant. Let the probability  $\mu(v_{k-1}) = p[v_{k-1} | Z(k-1)]$  be given by the following Gaussian probability density function

$$\mu(v_{k-1}) = \frac{1}{\sqrt{2\pi\sigma_{\mu}^2(k-1)}} \exp\left[-\frac{[v_{k-1}-m_{\mu}(k-1)]^2}{2\sigma_{\mu}^2(k-1)}\right] \quad (26)$$

where  $m_{\mu}(k-1)$  and  $\sigma_{\mu}(k-1)$  are the mean and variance respectively of the probably density function for  $v_{k-1}$ . Also, let the transition probability  $p[v_k | v_{k-1}, Z(k-1)]$  to be a Gaussian with a probability density given by

$$p[v_k | v_{k-1}, Z(k-1)] = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left[-\frac{[v_k - v_{k-1}]^2}{2\sigma_p^2}\right] \quad (27)$$

where  $\sigma_p$  is the covariance associated with model transitions. Substituting Equations (26) and (27) into (25) yields

$$\mu(v_k) = \frac{1}{c_1} \Lambda(k, v_k) \int_{-\infty}^{\infty} \exp\left[-\frac{[v_k - v_{k-1}]^2}{2\sigma_p^2}\right] \exp\left[-\frac{[v_{k-1}-m_{\mu}(k-1)]^2}{2\sigma_{\mu}^2(k-1)}\right] dv_{k-1} \quad (28)$$

$$= \frac{1}{c_1} \Lambda(k, v_k) \int_{-\infty}^{\infty} \exp\left[-\frac{[v_k - v_{k-1}]^2}{2\sigma_p^2} - \frac{[v_{k-1}-m_{\mu}(k-1)]^2}{2\sigma_{\mu}^2(k-1)}\right] dv_{k-1} \quad (29)$$

$$= \frac{1}{c_1} \Lambda(k, v_k) \int_{-\infty}^{\infty} \exp\left[-\frac{[v_k^2 - 2v_{k-1}v_k + v_{k-1}^2]}{2\sigma_p^2} - \frac{[v_{k-1}^2 - 2v_{k-1}m_{\mu}(k-1) + m_{\mu}^2(k-1)]}{2\sigma_{\mu}^2(k-1)}\right] dv_{k-1} \quad (30)$$

$$= \frac{1}{c_1} \Lambda(k, v_k) \exp\left[-\frac{v_k^2}{2\sigma_p^2} - \frac{m_{\mu}^2(k-1)}{2\sigma_{\mu}^2(k-1)}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{[-2v_{k-1}v_k + v_{k-1}^2]}{2\sigma_p^2} - \frac{[v_{k-1}^2 - 2v_{k-1}m_{\mu}(k-1)]}{2\sigma_{\mu}^2(k-1)}\right] dv_{k-1} \quad (31)$$

$$= \frac{1}{c_1} \Lambda(k, v_k) \exp \left[ -\frac{v_k^2}{2\sigma_p^2} - \frac{m_\mu^2(k-1)}{2\sigma_\mu^2(k-1)} \right] \int_{-\infty}^{\infty} \exp \left[ -\left( \frac{\sigma_p^2 + \sigma_\mu^2(k-1)}{2\sigma_p^2 \sigma_\mu^2(k-1)} \right) v_{k-1}^2 + \left( \frac{2 m_\mu(k-1) \sigma_p^2 + 2 v_k \sigma_\mu^2(k-1)}{2\sigma_p^2 \sigma_\mu^2(k-1)} \right) v_{k-1} \right] dv_{k-1} \quad (32)$$

where  $c_1$  is the normalizing constant. From [3] we find

$$\int_{-\infty}^{\infty} \exp[-\alpha^2 x^2 + \beta x] dx = \frac{\pi}{\sqrt{\alpha}} \exp \left[ \frac{\beta^2}{4\alpha^2} \right], \quad \alpha > 0 \quad (33)$$

Let

$$\alpha^2 = \left( \frac{\sigma_p^2 + \sigma_\mu^2(k-1)}{2\sigma_p^2 \sigma_\mu^2(k-1)} \right) \quad (34)$$

$$\beta = \left( \frac{2 m_\mu(k-1) \sigma_p^2 + 2 v_k \sigma_\mu^2(k-1)}{2\sigma_p^2 \sigma_\mu^2(k-1)} \right) \quad (35)$$

Using Equation (33)-(35) and substituting Equation (21), we find that Equation (32) becomes

$$\mu(v_k) = \frac{1}{c_5} \exp \left[ -\frac{\left( \frac{H a(k|k-1) (z(k) - Hb(k|k-1)) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + 2 m_\mu(k-1) S(k)}{\left( H^2 a^2(k|k-1) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + S(k) \right)} \right)^2}{2 \frac{S(k) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right)}{\left( H^2 a^2(k|k-1) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + S(k) \right)}} \right] \quad (36)$$

where  $c_5$  is the normalizing constant. Proof of Equation (36) is given in Appendix 1. Note by induction if the initial probability  $\mu(v_k)$  is Gaussian it will remain Gaussian with a mean and variance propagated by the following equations

$$m_\mu(k) = \left( \frac{H a(k|k-1) (z(k) - Hb(k|k-1)) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + 2 m_\mu(k-1) S(k)}{\left( H^2 a^2(k|k-1) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + S(k) \right)} \right) \quad (37)$$

$$\sigma_{\mu}^2(k) = \left( \frac{S(k) \left( \sigma_p^2 + \sigma_{\mu}^2(k-1) \right)}{\left( H^2 a^2(k|k-1) \left( \sigma_p^2 + \sigma_{\mu}^2(k-1) \right) + S(k) \right)} \right) \quad (38)$$

respectively. Note that  $m_{\mu}(k)$  is the estimate of the process input  $v_k$

## 2.8 Overall error state and error covariance estimates

The combined state estimate is obtained by the following

$$\hat{x}(k, v_{k-1}|k) = \int_{-\infty}^{\infty} \hat{x}(k, v_{k-1}|k) m(v_k) dv_k \quad (39)$$

$$= \int_{-\infty}^{\infty} (a(k|k) v_k + b(k|k)) m(v_k) dv_k \quad (40)$$

$$= a(k|k) m_{\mu}(k) + b(k|k) \quad (41)$$

The combined covariance is obtained by a Gaussian Mixture of a continuum of probability density function, i.e. from [1, page 48] we find

$$\hat{P}(k|k) = \int_{-\infty}^{\infty} P(k|k) m(v_k) dv_k + \int_{-\infty}^{\infty} \hat{x}(k, v_k|k) m(v_k) dv_k - \hat{x}^2(k|k) \quad (42)$$

$$= P(k|k) + \int_{-\infty}^{\infty} (a(k|k) v_k + b(k|k))^2 m(v_k) dv_k - \hat{x}^2(k|k) \quad (43)$$

$$= P(k|k) + \int_{-\infty}^{\infty} a^2(k|k) v_k^2 m(v_k) dv_k + 2 a(k|k) b(k|k) m_{\mu}(k) + b^2(k|k) - \hat{x}^2(k|k) \quad (44)$$

$$= P(k|k) + \int_{-\infty}^{\infty} a^2(k|k) (v_k - m_{\mu}(k))^2 m(v_k) dv_k + a^2(k|k) \int_{-\infty}^{\infty} (2 m_{\mu}(k) v_k - m_{\mu}^2(k)) m(v_k) dv_k + 2 a(k|k) b(k|k) m_{\mu}(k) + b^2(k|k) - \hat{x}^2(k|k) \quad (45)$$

$$= P(k|k) + a^2(k|k) \sigma_{\mu}^2(k) + a^2(k|k) m_{\mu}^2(k) + 2 a(k|k) b(k|k) m_{\mu}(k) + b^2(k|k) - \hat{x}^2(k|k) \quad (46)$$

### 3 Simulation Results

In this section, we present an empirical analysis of the continuum model Kalman filter illustrating its superior qualities over a conventional IMM algorithm.

Here the truth data is the same for both algorithms. Everything such as noise covariance, sample time, and constant position with velocity bias are set the same for both algorithms to provide a fair comparison. The IMM was implemented with three model with differing biases, namely [-10 0 10].

Consider the results shown in Figures 1-2.. It is clear that the continuum model Kalman filter outperforms the IMM by a factor of 1.6:25. This fact is demonstrated further in Table 1 showing the RMS error for both algorithms.

### 4 Concluding Remarks

In this paper, we investigated a recursive multiple model tracking approach similar to the Generalized Pseudo-Bayesian 1 (GPB1) [1] approach. Here we consider a continuum of models rather than the discrete set that is usually implemented in the GPB1 method. By doing so we have provided better models to improve tracker performance and solved the bias problem inherent in most multiple model approaches.

Empirical results have shown conclusively that the continuum model Kalman filter is capable of out performing the IMM, especially while operation the IMM between fixed models velocities.

By defining a continuum of models we avoid problems frequently associated with multiple model techniques that use a finite number of models.

### Appendix 1 – Proof of Equation 36

Using Equation (34)-(36), we find that Equation (33) becomes

$$\mu(v_k) = \frac{1}{c_2} \Lambda(k, v_k) \exp \left[ -\frac{v_k^2}{2\sigma_p^2} - \frac{m_\mu^2(k-1)}{2\sigma_\mu^2(k-1)} \right] \exp \left[ \frac{\left( 2m_\mu(k-1) \sigma_p^2 + 2v_k \sigma_\mu^2(k-1) \right)^2}{2\sigma_p^2 \sigma_\mu^2(k-1)} \right] \left( \frac{2\sigma_p^2 \sigma_\mu^2(k-1)}{4(\sigma_p^2 + \sigma_\mu^2(k-1))} \right) \quad (47)$$

$$= \frac{1}{c_2} \Lambda(k, v_k) \exp \left[ -\frac{v_k^2}{2\sigma_p^2} - \frac{m_\mu^2(k-1)}{2\sigma_\mu^2(k-1)} \right] \exp \left[ \frac{\left( m_\mu(k-1) \sigma_p^2 + v_k \sigma_\mu^2(k-1) \right)^2}{2\sigma_p^2 \sigma_\mu^2(k-1) (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (48)$$

$$= \frac{1}{c_2} \Lambda(k, v_k) \exp \left[ -\frac{v_k^2}{2\sigma_p^2} - \frac{m_\mu^2(k-1)}{2\sigma_\mu^2(k-1)} \right] \exp \left[ \frac{\sigma_\mu^4(k-1) v_k^2 + 2\sigma_p^2 \sigma_\mu^2(k-1) m_\mu(k-1) v_k + m_\mu^2(k-1) \sigma_p^4}{2\sigma_p^2 \sigma_\mu^2(k-1) (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (49)$$

$$= \frac{1}{c_2} \Lambda(k, v_k) \exp \left[ \frac{-\sigma_p^2 \sigma_\mu^2(k-1) v_k^2 + 2\sigma_p^2 \sigma_\mu^2(k-1) m_\mu(k-1) v_k - m_\mu^2(k-1) \sigma_p^2 \sigma_\mu^2(k-1)}{2\sigma_p^2 \sigma_\mu^2(k-1) (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (50)$$

$$= \frac{1}{c_2} \Lambda(k, v_k) \exp \left[ -\frac{v_k^2 - 2 m_\mu(k-1) v_k + m_\mu^2(k-1)}{2 (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (51)$$

where  $c_2$  is the normalizing constant. Substituting Equation (21) into (51) yields

$$\mu(v_k) = \frac{1}{c_3} \exp \left[ -\frac{[Ha(k|k-1)v_k - (z(k) - Hb(k|k-1))]^2}{2S(k)} \right] \exp \left[ -\frac{v_k^2 - 2 m_\mu(k-1) v_k + m_\mu^2(k-1)}{2 (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (52)$$

$$= \frac{1}{c_3} \exp \left[ -\frac{H^2 a^2(k|k-1) v_k^2 - 2 H a(k|k-1) (z(k) - Hb(k|k-1)) v_k + (z(k) - Hb(k|k-1))^2}{2S(k)} \right] \exp \left[ -\frac{v_k^2 - 2 m_\mu(k-1) v_k + m_\mu^2(k-1)}{2 (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (53)$$

$$= \frac{1}{c_4} \exp \left[ -\frac{\left( H^2 a^2(k|k-1) (\sigma_p^2 + \sigma_\mu^2(k-1)) + S(k) \right) v_k^2 - 2 \left( H a(k|k-1) (z(k) - Hb(k|k-1)) (\sigma_p^2 + \sigma_\mu^2(k-1)) + 2 m_\mu(k-1) S(k) \right) v_k}{2S(k) (\sigma_p^2 + \sigma_\mu^2(k-1))} \right] \quad (54)$$

$$= \frac{1}{c_4} \exp \left[ - \frac{v_k^2 - 2 \frac{\left( H a(k|k-1) (z(k) - Hb(k|k-1)) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + 2 m_\mu(k-1) S(k) \right)}{\left( H^2 a^2(k|k-1) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + S(k) \right)} v_k}{2 \frac{S(k) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right)}{\left( H^2 a^2(k|k-1) \left( \sigma_p^2 + \sigma_\mu^2(k-1) \right) + S(k) \right)}} \right] \quad (55)$$

Completing the square we obtain Equation (36). Proof is complete.

## References

- [1] Y. Bar-Shalom and Y. Li, Estimation and Tracking: Principles; Techniques; and Software, Artech House Inc., 1993
- [2] X. Li and Y. Barshalom, Multiple-model estimation with variable structure, "*IEEE Transactions on Automatic Control*", Vol 41, No. 4, pp 478-493, April 1996.
- [3] P. Peebles, Probability, Random Variables, and Random Signal Principles, McGraw Hill, 1987

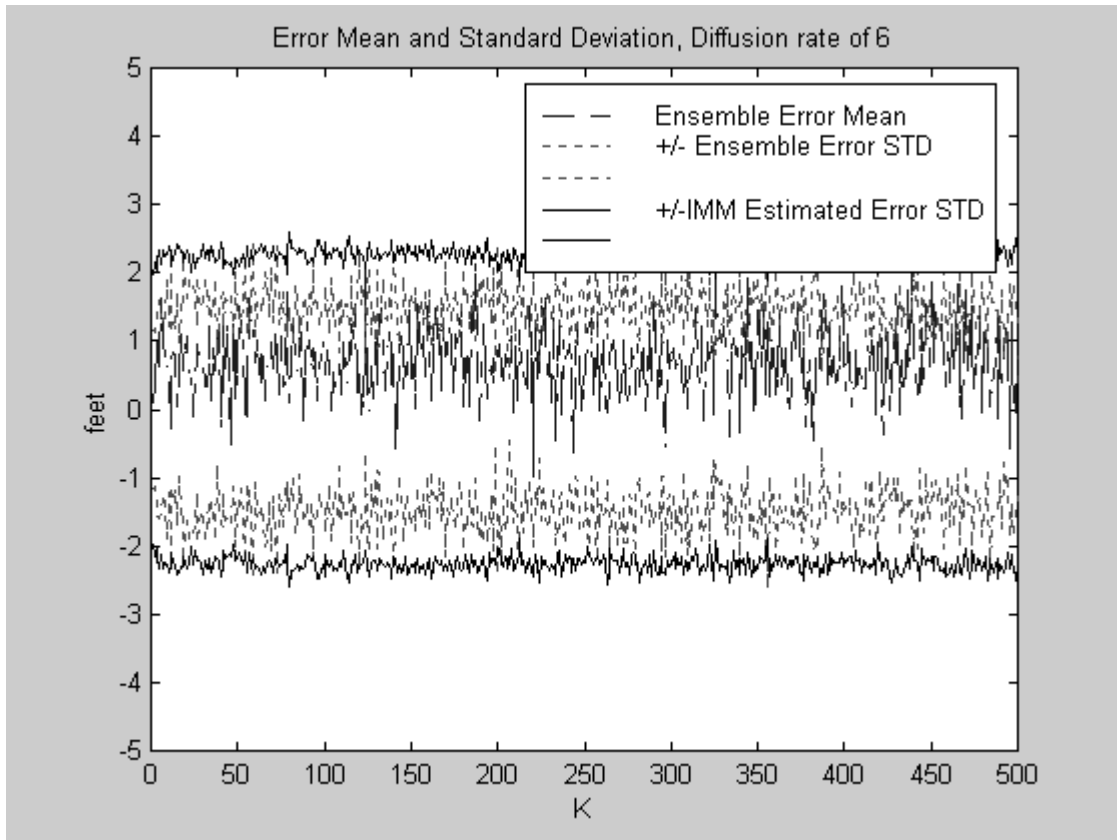


Figure 1- The ensemble statistical results, over 100 Monte Carlo runs, for a conventional IMM with three diffusion models (or velocity biases), namely  $[-10 \ 0 \ 10]$ . Note that there is an obvious bias in the result.

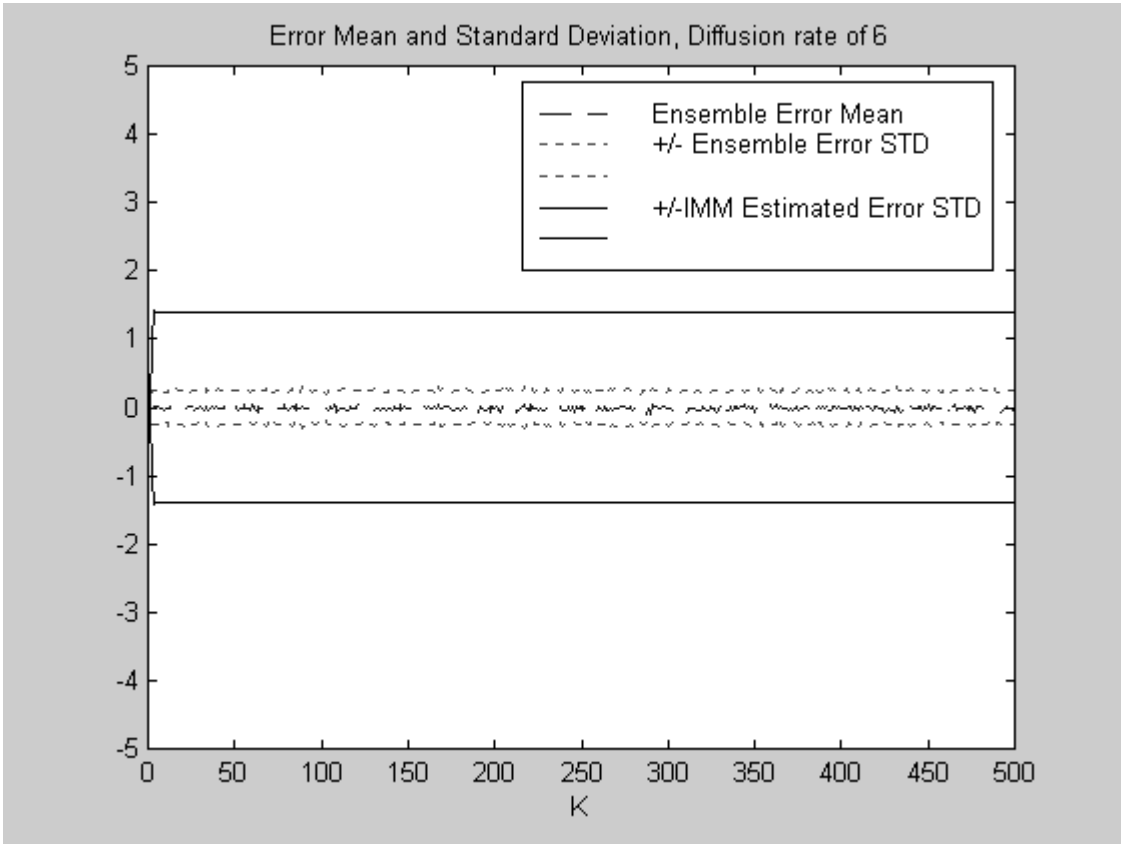


Figure 2- The ensemble statistical results for 100 Monte Carlo runs of the continuum model filter. There is no bias in the result.

Table 1: RMS error value for both algorithms the continuum model algorithm and the IMM

Algorithm	RMS
Continuum Model Kalman Filter	.25
Interacting Multiple Model Filter (IMM)	1.66