Multi model estimators

Karthik Gopalakrishnan

1 Introduction

The Kalman filter is the optimal filter for linear models with Gaussian, zero mean uncorrelated noise. In its most basic form, this filter can perform the estimation when the system dynamics is time invariant or if its time evolution is known apriori (by using the appropriate A and C matrix for every time instant). Consider the following set of equations where the system can evolve according to different mode dynamics

$$x_{t+1} = A_j x_t + W_t \quad j \in M \tag{1}$$

$$y_t = C_j x_t + R_t \quad j \in M \tag{2}$$

where M is the set of all possible mode index.

This is a linear hybrid system where the mode switches within the set M and the dynamics is linear for every mode. The estimation problem is as follows: Given measurements from a system with multiple modes of operation, how do we estimate the state and the current mode of operation.

This is a very relevant problem in the context of vehicle tracking. Suppose we wish to track the location of an aircraft using a radar. The aircraft can be in several possible modes like climb, cruise, descent or a coordinated turn. Each of these maneuvers is governed by separate equations and we wish to know the correct location of the aircraft during all the possible stages of flight.

The optimal filter for multi model estimation is computationally expensive. It requires tracking the exponentially growing mode histories and is not practically feasible. Thus approximate, non-optimal estimators were developed. In this report, I would compare the performance of two such methods- the Interacting Multi Model (IMM) estimator and the Generalized Pseudo Bayesian estimator of order 1 (GPB1) in the context of aircraft tracking.

For this report, the literature that I referred to is summarized:

- Papers on the Interacting Multi Model algorithm [2], [3]
- Survey of Interacting Multi Model algorithms [6]
- Papers on GPB filter [4]
- Survey describing dynamic models to describe a targets motion [5]
- Comparison between the IMM estimator and optimal estimator [1]

Some of the question that I had while starting out the project are listed below.

- How do the IMM and GPB1 estimators compare
- What are appropriate performance metrics for estimation of hybrid systems
- How important is the set of models that we provide to the estimators. If the aircraft performs maneuvers described by models from set M, how will the filter perform when provided with another set of models R

In the following sections, I will first summarize the two estimation algorithms. After that the results of the simulations are described.

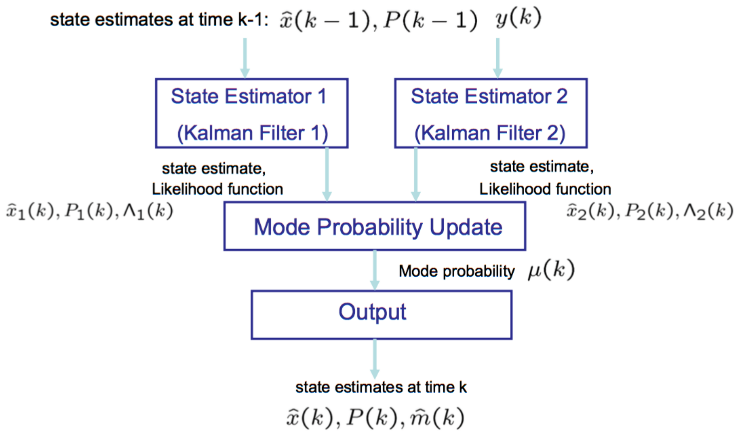


Figure 1: Flowchart for GPB of order 1 when there are two modes. Note that the covariance is denoted by P where as I would be using Q in the report. Source : Prof Jianghai Hu

2 The estimator

2.1 The Generalised pseudo Bayesian estimator

The idea behind the GPB estimator is to approximate the following term that comes in the optimal estimator

$$\Pr[x(k+1)|Y^{k+1}] = \sum_{l=1}^{r^{k+1}} \mathcal{N}[x(k+1); \hat{x}^{M^{k+1}=l}(k+1|k+1), Q^{M^{k+1}=l}(k+1|k+1)] \\ * \Pr\{M^{k+1} = l|Y^{k+1}\} \tag{3}$$

where r is the number of modes and the sequence of mode history

$$\{M^{k+1} = l\} = \{M(1) = i_1, M(2) = i_2, ..., M(k+1) = i_{k+1}\} \tag{4}$$

This has an exponentially growing number of terms with time.

A first order approximation for this is used in the GPB1 algorithm.

$$\Pr[x(k+1)|Y^{k+1}] = \sum_{l=1}^{r} \mathcal{N}[x(k+1); \hat{x}^{M(k+1)=l}(k+1|k+1), Q^{M(k+1)=l}(k+1|k+1)] \\ * \Pr\{M(k+1) = l|Y^{k+1}\} \tag{5}$$

A second order approximation will involve conditioning on two state histories M(k+1) and M(k). In this project, I will focus on the GPB1 algorithm.

This expression (5) intuitively explains the broad idea for solving the multi model estimation problem. We will use a bank of r Kalman filters in parallel, each tuned to a particular system dynamics. Outputs from each of these filters need to be combined in an appropriate fashion to come up with a single estimate. This is summarized in Fig (1)

Let π denote the transition probability matrix between the states and $\mu_j(k)$ denote the probability of mode j being the true state of the system at time k. We describe *one cycle* of the GPB1 algorithm when the hybrid system is described by Equation (1), (2)

Given $\hat{x}(k-1|k-1)$, $Q(k-1|k-1)$ mode probabilities $\mu(k-1)$

Step 1: Filtration Run each of the r Kalman filter with the given initial condition (Nothing new; same as what we did in class)

For all $i = 1, ..., r$

Propagation:

$$\hat{x}_i(k|k-1) = A_i\hat{x}(k-1|k-1)$$
$$Q_i(k|k-1)_i = A_i^T Q(k-1|k-1)A_i^T + W(k-1)$$

Kalman gain:

$$K_i(k) = Q_i^T(k|k-1)C_i^T[C_iQ(k|k-1)C_i + R(k-1)]^{-1}$$

Measurement update:

$$\hat{x}_i(k|k) = \hat{x}_i(k|k-1) + K_i(k)(y(k) - C_i\hat{x}_i(k|k-1))$$

$$Q_i(k|k) = (I - K_i(k)C_i)Q_i(k|k-1)$$

Calculating likelihood functions. This is an indication of how well the measurement is explained by this model (this particular filter). The terms in the equation are the residue and the residue-covariance.

$$\Lambda_i = \mathcal{N}(y(k) - \hat{x}_i(k|k-1); 0; C_iQ(k|k-1)C_i + R(k-1))$$

Step 2: Updating mode probability In this step, the likelihood of every filter is merged with the knowledge of the Markovian transitions probability of the states. So even if a state s has a high likelihood at time k, it may not be the maximum likelihood estimate at time k because it is an unlikely transition from the state at time (k-1)

$$\mu_i(k) = \frac{1}{c_k}\Lambda_i(k)\sum_{j=1}^{j=r}\pi_{ij}\mu_j(k-1)$$

Step 3: Merging the output

$$\hat{x}(k) = \sum_{i=1}^{r}\mu_i(k)\hat{x}_i(k|k)$$

$$Q(k|k) = \sum_{i=1}^{n}\mu_i(k)[Q_i(k|k) + (\hat{x}(k) - \hat{x}_i(k|k))(\hat{x}(k) - \hat{x}_i(k|k))^T]$$

The mode estimate:

$$\hat{m}(k) = argmax \quad \mu_i(k)$$

2.2 IMM Estimator

The IMM estimator is similar to the GPB1 estimator except for one difference. The input for each for the Kalman filters is different and not equal to $\hat{x}(k-1)$ but rather $\hat{x}_i(k-1|k-1) \quad \forall i = 1, ..., r$. We define a few additional variables. $\hat{x}_{0i}(k-1|k-1)$ is the input given to each of the Kalman filters 1,..., r and $\mu_{ij}(k-1|k-1)$ is the mixing probability. One cycle of the IMM estimator algorithm is as follows:

Step 1: Interaction

For all $i, j \in$ {1,2,..., r }

$$\mu_{ij}(k-1|k-1) = \frac{1}{c_j}\pi_{ij}\mu_i(k-1)$$

where $c_j = \sum_i \pi_{ij}\mu_i(k-1)$ is the normalising factor.

For all modes i=1,...,r

$$\hat{x}_{0i}(k-1|k-1) = \sum_{j=1}^{r}\hat{x}_j(k-1|k-1)\mu_{ji}(k-1|k-1)$$

$$Q_{0i}(k-1|k-1) = \sum_{j=1}^{r} \mu_{ji}(k-1|k-1) * [Q_j(k-1|k-1)+$$
$$(\hat{x}_j(k-1|k-1) - \hat{x}_{0i}(k-1|k-1))(\hat{x}_j(k-1|k-1) - \hat{x}_{0i}(k-1|k-1))^T]$$

Step 2: Filtering Same as the GPSB1, usinf $\hat{x}_{0i}(k-1|k-1)$ and $Q_{0i}(k-1|k-1)$ as inputs for each filter

Step 3: Updating mode probability Same as the GPSB1

Step 4: Merging the output Same as GPSB1

3 Aircraft tracking example

Consider the case of a moving aircraft which we wish to track. The aircraft can be in one of the two modes: maneuvering (non - zero acceleration) or non-maneuvering (zero acceleration). We restrict the motion of the aircraft to two dimensions (x-y plane) for simplicity.

Let the following be the modes of operations be considered. We may or may not use all of them in each example

1. Uniform motion in a straight line
2. Constant rate turn with $\omega = 0.1 \quad rad/sec$
3. Constant rate turn with $\omega = -0.1 \quad rad/sec$
4. Constant rate turn with $\omega = 0.05 \quad rad/sec$
5. Constant rate turn with $\omega = -0.05 \quad rad/sec$

The dynamics for mode 1 with a generic noise term is

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_{k+1} = \begin{bmatrix} 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_t + Noise_{process} \tag{6}$$

The dynamics for mode 2 for any constant turn rate ω and a process noise is

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \frac{sin\omega T}{\omega} & 0 & -\frac{1-cos(\omega T)}{\omega} \\ 0 & cos(\omega T) & 0 & -sin(\omega T) \\ 0 & \frac{1-cos(\omega T)}{\omega} & 1 & \frac{sin\omega T}{\omega} \\ 0 & sin(\omega T) & 0 & cos(\omega T) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}_t + Noise_{process} \tag{7}$$

The measurement equation is the same for all modes with $C_j = I_{4X4} \quad \forall j \in M$.

We now look at examples that cover the cases of M=R, $M \subset R$ and $M \supset R$

Example 1 : R=M

Let the aircraft to be tracked perform two kinds of motion- uniform straight line and constant rate turn of $\omega = 0.1 rad/sec$. The discrete equations are used with $\Delta T = 1 sec$. The initial state is x=0, y=0, x velocity = 100m/s and y velocity =0. For the first 100 seconds, the aircraft is going in a straight line. At t=101, it starts a turn with $\omega = 0.1 rad/sec$ till t=150. From t=150 to t=200, it continues in its straight line trajectory.

M=R={straight line mode, $\omega = 0.1$ mode}

The process noise has a covariance given by W=diag(1,1,1,1) and the probability of mode shift is given by the transition matrix

$$\pi = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}$$

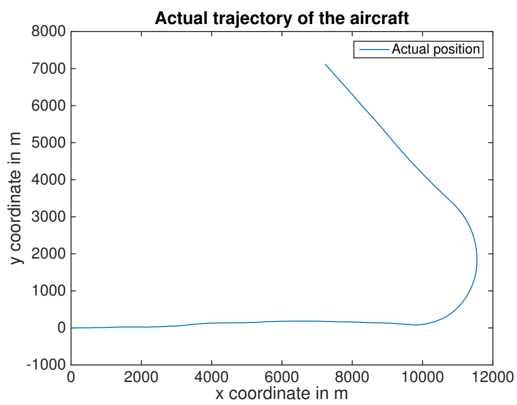
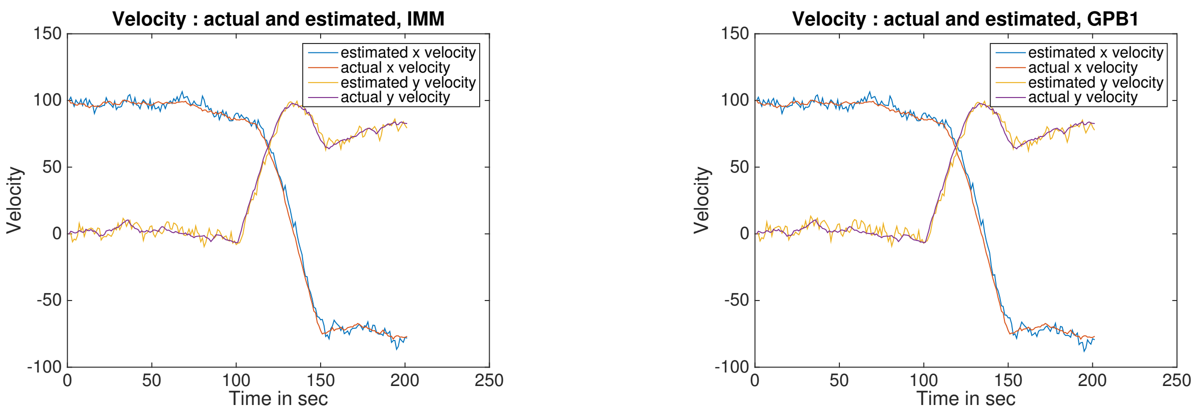


Figure 2: Trajectory of the aircraft. Straight flight from t=1 to 100 sec, omega=0.1 rad/sec turn from t=101 to 150 sec and straight line motion from t=151 to 200 sec



(a) IMM (b) GPB1

Figure 3: The velocity estimates by both methods seem to be good and similar to each other

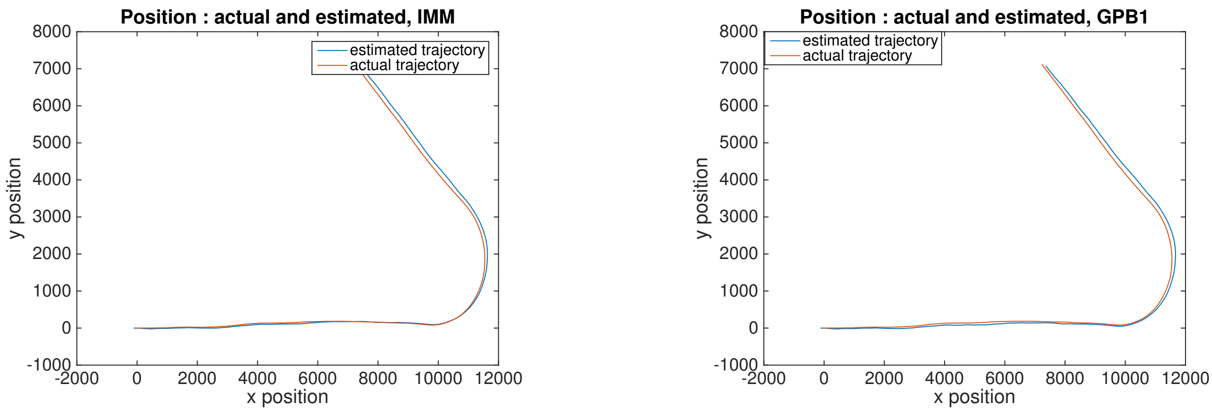
where the first row/column represent the uniform motion mode and the second row/column represent the omega=0.5 turn. The measurement noise has a covariance matrix R= diag$(100^2, 5^2, 100^2, 5^2)$. The velocity magnitude is fixed at 100m/s, so I wanted to have the standard deviation that was comparable to it (which is 5m/s). For the position variance, I have a much higher magnitude because the scale of distances covered by the radar is also much higher (in this case it is over 10 km). The performance of an estimation algorithm is evaluated using the following metrics

- The normalized mean square deviation of the estimate from the true state.

$$error = \frac{1}{T}\sum_{t=1}^{T} ||\hat{x}(t) - x_{true}(t)||_2$$

- The percentage of times the mode is estimated correctly.

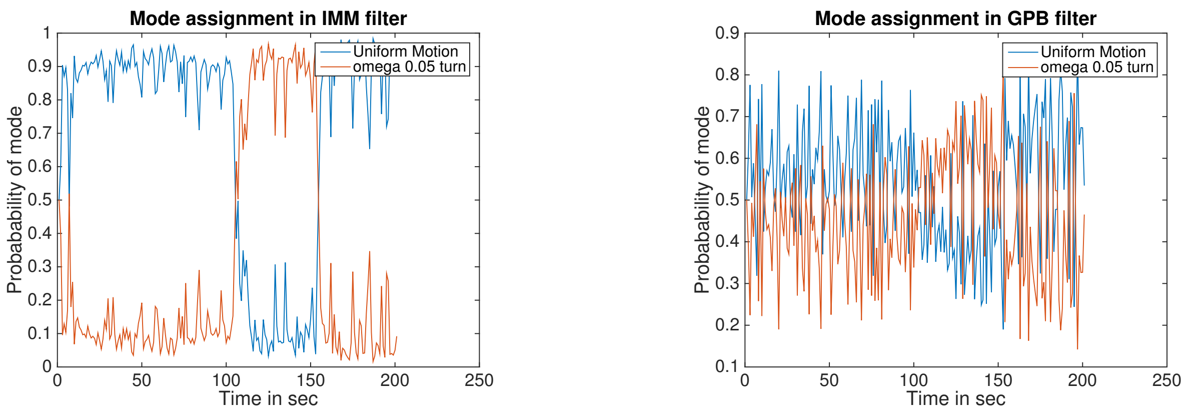
The true trajectory of the aircraft is shown in Fig (2)
Error Results:
Normalized error of IMM filter = 104.91
Normalized error of GPB1 filter = 129.15
Thus although the state estimates seem to be pretty similar to each other visually (Fig(3) and Fig(4)), the net estimation error of the GPB1 method is 46% higher than the IMM method (129.15 versus 104.91). The differences in the prediction error is better explained when we look at Fig(5) and Fig(6). Here we notice that the straight line motion and the turn mode is better identified by the IMM estimator in comparison to the GPSB1 estimator.



(a) IMM

(b) GPB1

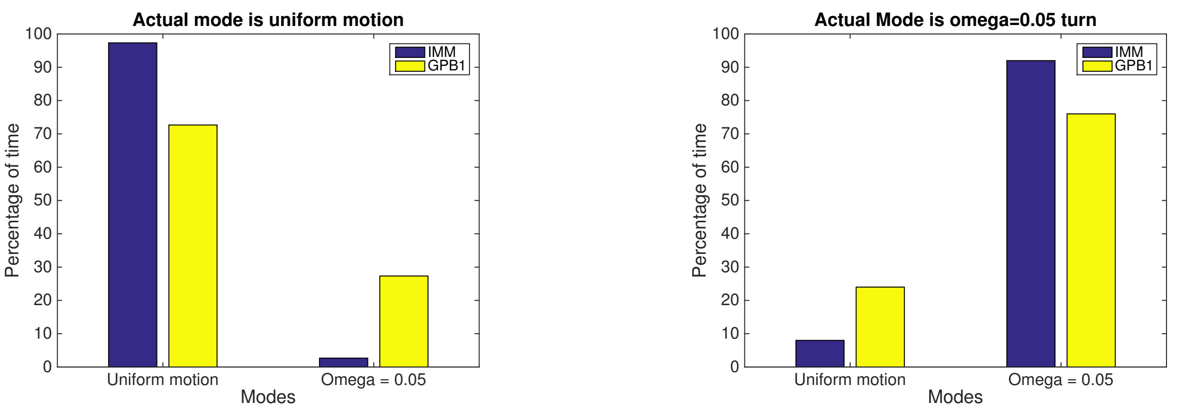
Figure 4: The position estimates by both methods seem to similar to each other on visual inspection



(a) IMM

(b) GPB1

Figure 5: The mode probability μ as a function of time for IMM and GPB1. We note that the IMM prediction is more stable and confident about the modes (has a higher probability value ascribed to the correct mode). Turn mode is active from t= 101 to t= 150 sec



(a) Uniform motion

(b) omega = 0.05 turn

Figure 6: Correct and incorrect prediction of mode in terms of percentage. The IMM does a better job of correctly predicting the current mode of the system. The y axis is the fraction of time that a mode is the Maximum Likelihood Estimate

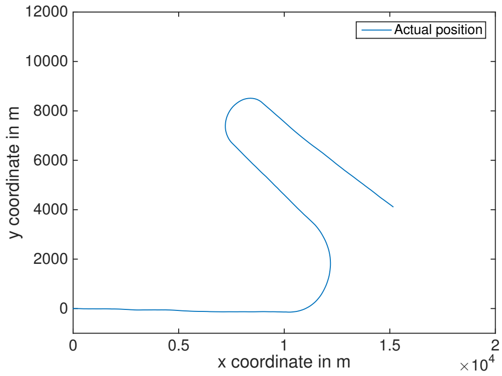
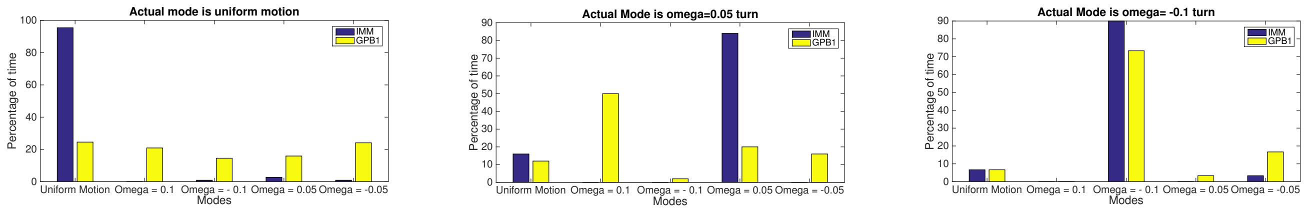


Figure 7: Trajectory of the aircraft for Example 2. Omega=0.05 turn from t=101 to t=150 and Omega = -0.1 turn from t=200 to t=230



(a) Uniform motion (b) 0.05 rad/sec turn (c) -0.1 rad/sec turn

Figure 8: The GPB1 does well only in the case on -0.1 rad/sc turn. Rest other mode estimates are mixed up. The IMM does better then the GPB1 in all modes of operation

Another interesting point is that the GPB1 performs much better when the measurement noise is lower. In such conditions, GPB1 performs similar to the IMM estimator. For example, when the measurement noise is given by R=diag$(10^2, 1, 10^2, 1)$, the GPB1 error and the IMM error are both $\tilde{1}01.5$ Also the mode prediction accuracy of the GPB1 increases to 95% for uniform motion and 90% for the turn. (compared to 70% and 75% respectively in the case of high noise)

Example 2 : R is a superset of M

M={straight line, $\omega = 0.05$, $\omega = -0.1$ }, R={straight line, $\omega = 0.1$, $\omega = -0.1$, $\omega = 0.05$, $\omega = -0.05$ }

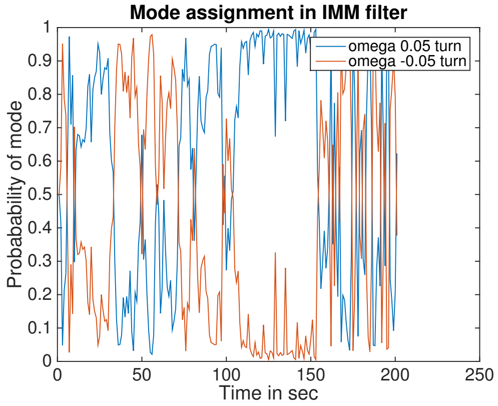
The trajectory of the flight is shown in Fig(7). The true mode of the flight is as follows: uniform motion from t=1 to t=100 s , $\omega = 0.05$ from t=101 to 150 s, uniform motion from t=151 to 200 s, $\omega = -0.1$ from t=201 to 230 s, uniform motion from t=231 to 300 s. The process and measurement noise is the same as Example 1.

IMM Error= 103.19 and GPB1 error = 111.47. As before the IMM error is lower. Also, since the error is normalized by time, on comparison with Example 1, we can say that adding additional flexibility in terms of modes can potentially reduce the state estimate error.

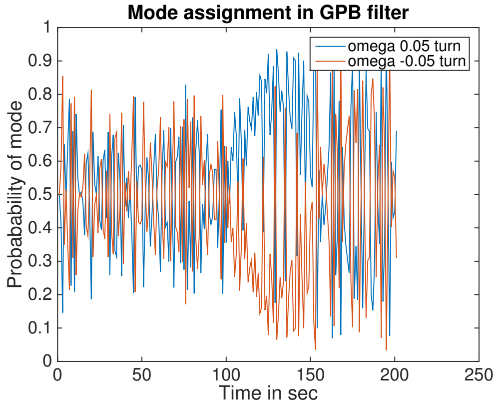
Additional modes in R need not however improve the mode prediction. The results in Fig(8) show that the IMM performs well in predicting the correct mode. The GPB1 gets confused between similar modes at times. In the case of the two turns, it is unable to distinguish the rate of the turn clearly. In the uniform motion case, the GPB1 considers the motion as a combination of all modes. Similar to Example 1, the GPB1 performance becomes closer to the IMM performance when the measurement noise is decreased.

Example 3 : R is a subset of M

Let us look at two sub cases here- $span\{R\} = M$ and $span\{R\} \subset M$. The actual flight trajectory (as in Figure (2) with straight line and $\omega = 0.05$ modes) , measurements, noise parameters and transition

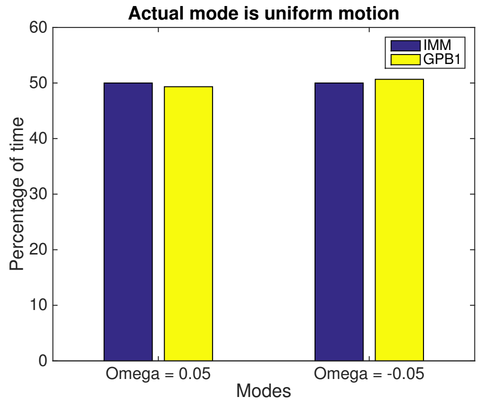


(a) IMM

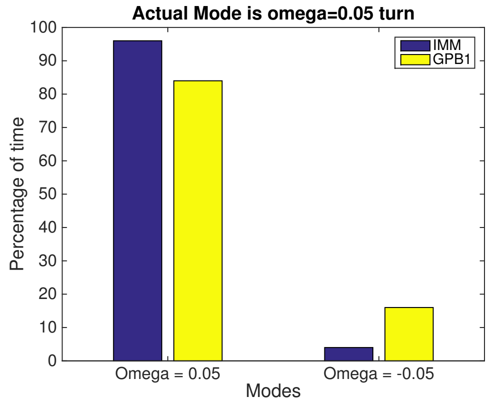


(b) GPB1

Figure 9: Plots of the mode probability μ. Time 101 to 150 sec is when the turn mode is active. For t=1 to 100, the modes keep switching between each other in order to capture the straight line motion



(a) Uniform motion



(b) omega = 0.05 turn

Figure 10: Correct and in-correction prediction of mode in terms of percentage. Note the 50% combination of bot modes that occurs to capture straight line motion

probabilities is the same as Example 1. The only difference is that the set of modes supplied to the estimator.

M={straight line mode, $\omega = 0.05$ mode}

R_1={ $\omega = 0.05$ mode, $\omega = -0.05$ mode}

In this case, we see that all the modes in M are not covered by the set of modes in R_1. Specifically, the straight line motion is not present in the set R_1 which is fed to the estimators. The result in this case is interesting. Although the straight line motion is not given to the estimator, it still works well. This is because the straight line motion can be captured by considering both the left and right turn modes with probability 0.5 each. This brings us to the notion of spanning modes. If the modes in set R_1 can be combined, in some probabilistic sense (this is a very loose argument; I am just explaining it intuitively) to describe all modes in M, we can still use the IMM/GPB1 estimator.

Figure (9) shows the mode probabilities when we use the set R_1 to track the aircraft. The estimator is able to track the straight lines by shifting between the two turn modes as seen in Fig(10).

Error when using R1:

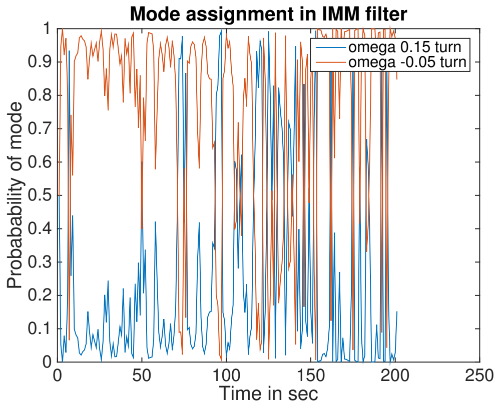
IMM estimation error = 112.8

GPB1 estimation error = 132.7

Let us now look at another example where R_2 spans the set M.

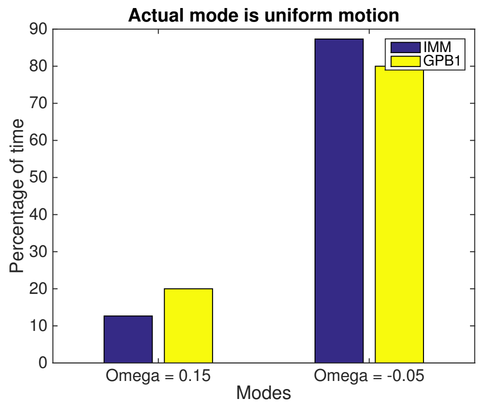
R_2= { $\omega = 0.15$ mode, $\omega = -0.05$ mode }

Error when using R2:
IMM estimation error = 125.76
GPB1 estimation error = 167.7
The probability plots for this case is shown in Fig (11)



(a) IMM (b) GPB1

Figure 11: Time 101 to 150 sec is when the turn mode is active. For t=1 to 100, the modes keep switching between each other in order to capture the straight line motion. Both the IMM and GPB1 have different probabilities of modes to help them capture the straight line. This is not unexpected as the IMM has an extra 'interaction' step before the parallel filtering.



(a) Uniform motion (b) omega = 0.05 turn

Figure 12: Correct and incorrect prediction of mode (by percentage). To capture uniform motion, I had naively expected a 3:1 ratio between the 0.15 and -0.05 modes. This is however incorrect because (a) The combination is non linear in omega since the A matrix has a sine and cosine term, and (b) I forgot to account for the modes having non-zero probabilities and still not being the MAP estimate. In the GPB1 case, the $\omega = 0.15$ mode consistently has a probability of about 0.3 in t=1 to t=100

Now take the case of R_3= $\{\omega = -0.15, \omega = -0.05\}$. We get large errors in the estimates. IMM error= 280.2 and GPB1 error = 280.7 In this particular case, the filter is stable, but has a large error. However, when the measurement noise is low, the filter fails.

Thus the conclusion for this section is that we would like to have the span of R equal to the modes in M. This seems to be intuitively sufficient for the filter to work, even though performance might be bad. But when the span of R is not equal to the modes in M, we can not have guarantees that the filter will be stable.

A case in point is the extreme conditions in R_4= $\{\omega = -0.75, \omega = -0.70\}$ and R_5= $\{\omega = 0.75, \omega = -0.70\}$. Using R_4 causes the filter to fail as it cannot capture the left turn that the aircraft takes. On the other hand, R_5 is stable, although it leads to pretty bad performance and very noisy estimates as shown in Figure (13)

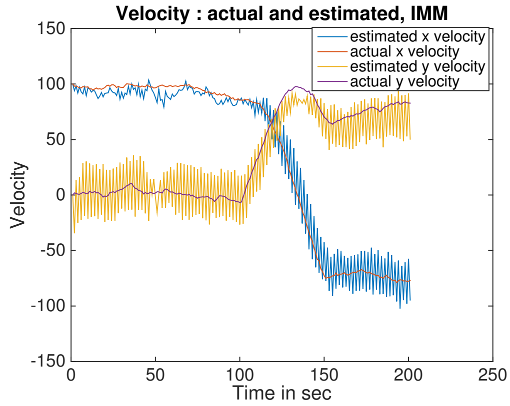


Figure 13: If modes in R are very different from modes in M (although they still span the set M), we might get poor quality estimates. The good news is that the filter is stable in spite of a bad guess for the set R

	R=M (from Example 1)	R_1	R_2	R_3	R_5
IMM	104.9	112.8	125.76	280.2	445.8
GPB1	129.15	132.7	167.7	280.7	470.6

Table 1: Comparison of the error when we have different R. As we move to the right columns , the modes of R are further away from those of M (although they can still be combined to span M). This explains the increasing error.

4 Discussion

In this project, I have explored the use of IMM and also compared it with the GPB1 algorithm. This is done considering a realistic scenario where we do not know all possible system modes that need to be supplied to the filter.

In case of a perfect match, that is when R=M, we observe the best performance of the estimator. This is obvious. Also the better performance of the IMM estimator over the GPB1 is seen. Visualization of the mode probability gives us some intuition of what modes the filter believes is active. The more interesting case is when we have R⊂M and R⊃M. When R has more modes than M, the GPB1 filter performs poorly in mode estimation. IMM is robust to such 'over specified' mode sets. This is a really useful property. When faced with unknown targets which we need to track, the filter performance should not be compromised just because we enumerated all likely modes in the set R. This gives a tracking system operator a lot more flexibility in tuning the filters. We note that the case of R⊂M can lead to failures. This can happen when modes in M cannot be expressed using available modes in R. When R spans the set M the filter will remain stable.

The performance gap between the IMM and the GPB1, in terms of the error is negligible when the measurement noise is very low. Although not shown in this report, we also observe the performance gap between the GPB1 and the IMM estimator tend towards zero when the measurement noise is really high (coefficient of variation is close to 1). When the noise is really high, it is very hard for both filters to do the estimation and they are equally worse. So the true performance benefits of the IMM over the GPB1 is observed when the noise levels are moderate.

Finally, I just wish to touch upon the question raised in class- Why does the GPB2 perform as well as the IMM. When there are r modes, the GPB1 algorithm uses r Kalman filters in parallel where as the GPB2 algorithm uses r^2 Kalman filters. Naturally GPB2 wouldperform better. When we use the complete GPB algorithm, at time instant t, we would need r^t Kalman filters. This is the optimal solution. In the literature, it is mentioned that the GPB2 performs similar to IMM. The performance of approximate methods (GPBand IMM) to the optimal is not compared. The reason for the good performance of the IMM is explained in literature [1]. In the paper,the authors derive an expression for the optimal estimator and explicitly identify the term that corresponds to the 'Interaction' step (the first step which ensures that each filter gets a different prior estimate) we see in the IMM. This feature of the IMM makes it a good approximation to the optimal multi model filter. I am however not sure if the increase in number of filters to r^r is the only thing that makes the GPB2 a good approximation.

References

[1] Yaakov Bar-Shalom, Subhash Challa, and Henk AP Blom. Imm estimator versus optimal estimator for hybrid systems. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(3):986–991, 2005.

[2] Yaakov Bar-Shalom, KC Chang, Henk Blom, et al. Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm. *Aerospace and Electronic Systems, IEEE Transactions on*, 25(2):296–300, 1989.

[3] Henk AP Blom and Yaakov Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *Automatic Control, IEEE Transactions on*, 33(8):780–783, 1988.

[4] Michael W Hofbaur. *Hybrid estimation of complex systems*, volume 319. Springer Science & Business Media, 2005.

[5] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1333–1364, 2003.

[6] Efim Mazor, Amir Averbuch, Yakov Bar-Shalom, and Joshua Dayan. Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(1):103–123, 1998.