Exploring PDAFs in target tracking

Kristian Stensgård, Leik Lima-Eriksen

November 25, 2020

1 Introduction

In this report, we will look at two different datasets containing RADAR measurements as well as the ground truth at different time steps, and the goal is to estimate the path by the use of RADAR measurements. Such RADAR measurements contain several measurements at each time step, where at most The first one is simply one generated from a data model with uncertainty and clutter added. The latter represents data collected from quite an agile boat accelerating and taking sharp turns in Trondheimsfjorden.

A probabilistic data association filter (PDAF) provides an optimal solution to these problems. The PDAF takes an MSEE approach to dealing with multiple measurements where it calculates a weighted average over all the measurements based on how likely they are to be the true measurement. In other words; each measurement, being either the target or clutter, is treated as a detection. Then for each detection, the likelihood for this detection being the target is calculated. The likelihoods are then used as weights when merging together the state estimates from all the measurements. The reader is referred to [1] for details regarding implementation of the mentioned filters.

2 Solution

2.1 Simulated data

The simulated dataset is made by constructing detection and clutter measurements by means of probabilistic sampling using an underlying true path. By inspecting Figure 1, one can observe that the target for the most part moves along straigh lines, and takes distinct turns with quite constant turn rate. Another observation is that the true detections come at a fixed interval. Altogether one would expect such a dataset to fit quite nicely with an IMM-PDAF with a CV-CT model. This comes from the fact that a combination of such models fits the data well; either it moves straight ahead (constant velocity, CV), or it takes turn with constant turnrate (CT).

From Figure 1 we see that we have a fair amount of clutter especially in the turns. However on the straights the clutter is not that prevalent. There are several important parameters involved in the tuning of an IMM-PDA filter. When tuning the parameters, it is useful to look at the NEES and the RMSE. We chose a 90% confidence interval for the NEES and would thus like the NEES to stay within these bounds 90% of the time. Also a low RMSE is desirable, but both the RMSE and NEES can have less desirable values if this reduces the peak errors. No significant peaks in the error results in a consistent filter, that on a new dataset would likely have a similar performance. Having large peaks at some times, means that the filter is unreliable

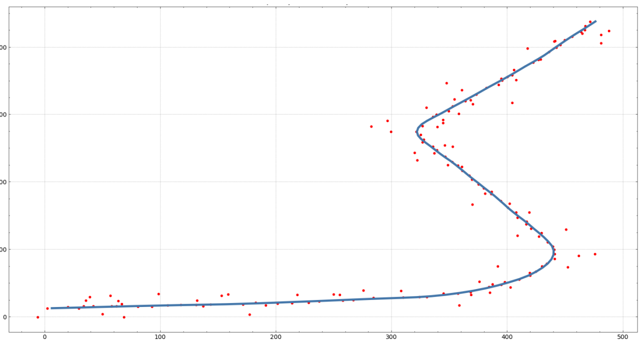

Figure 1: True trajectory (blue path) along with the closest measurements (red dots).

in these situations, which means that another tracking situation in the same environment could have resulted in track loss or poor performance.

The parameters that differ from an IMM is the P_D, the P_{FA}, and the gate size g. Simply put, P_D is a constant that specifies how large share of the samples contains a detection, and translates to the probability for getting a detection for each time new measurements arrive. For the P_{FA}, it specifies how many misdetections there will be for each unit volume cell. The P_D and the clutter intensity, P_{FA} are closely related. For the P_D it is reasonable to assume that about 90% of time we have a detection. This was approximated by calculating the ratio of samples which were within 45 metres from the ground truth at the same time step, and dividing by the number of time steps. By inspecting the samples, and playing it as a movie, see that the spread of the clutter is quite large and with a gate size that is small, one could argue that a low clutter intensity is correct. Typical values for P_{FA} is in the range of 10^{-6} to 10^{-2}. So a value of 10^{-5} was chosen. Since the target moves in a quite predictable manner, a small gate size could be used without sacrificing performance; one would never get in a situation where the gate fails to gate the detections. And having a small gate size would be quite beneficiary in especially the turns, as these contains a lot of nearby clutter which are way off the track and would just add noise to the state estimate.

The IMM-PDAF consists of two underlying state filters that are both EKF with respectively continuous velocity (CV) and constant turnrate (CT) as process models. These models are approximately the same if the variance is equal and the variance for the turn rate σ_ω^2 is zero. By using a non-zero σ_ω^2 and equal variance for the other entries in the Q-matrix, we should get mode switching when the vessel does a turn. This is due to the covariance ellipse of the CT-model being wider and therefore more prone to weigh a measurement as an indication of a turn if it happens to be transversal to the current velocity vector. The values for σ_{CT}^2 and σ_{CV}^2 were under tuning chosen to be 0.3^2 and a $\sigma_\omega^2 = (3 \cdot 10^{-3}\pi)^2$.

Another parameter that determines mode

[1]Brekke, Edmund. "Fundamentals of Sensor Fusion"

σ_{CV}	σ_{CT}	σ_ω	σ_z	P_{FA}	P_D	g
0.3	0.3	$3 \cdot 10^{-3}\pi$	2	$1 \cdot 10^{-}5$	0.8	2.5

Table 1: The parameters used in an IMM-PDAF on the simulated dataset.

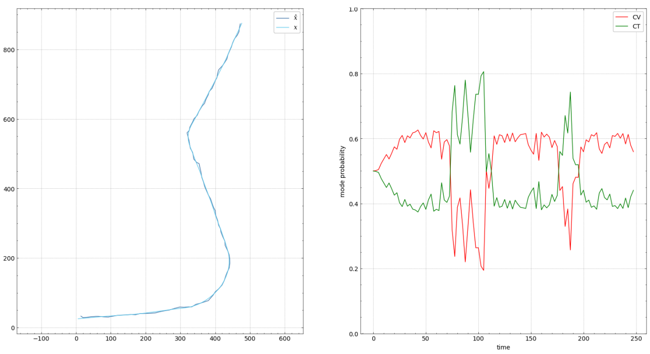

Figure 2: Left: the estimated track of the IMM dataset together with the ground truth. Right: mode probabilities

switching is the π-matrix. The values on the diagonal is the probability of staying in the same mode and the of diagonal elements are the probabilities for switching from mode i to j. The π-matrix is set with a probability of 0.9 on the diagonals and 0.05 on the off diagonals to only switch modes when we are certain of a change in behavior of the vessel such that a different process model is desirable to be used. The variance for the measurement model, σ_z^2, is chosen to have a significantly higher value than the ones for the process models, because we have so much noise in the measurements, this is seen clearly in Figure 1. The reader is referred to Table 1 for the parameters used in the solution.

From Figure 2, we see that the PDA track estimate follows the ground truth without having either large deviations or track loss. Also, the plot on the right in the figure reveals that the filter favors the CV model when the target is moving straight ahead, whereas it weighs CT highest when the target performs as turn. One can thus argue that the mode switching of the PDA works as intended.

The errors for position and velocity are shown in Figure 3. As stated earlier, we want to minimize the overall error, and also the magnitude of individual spikes such that we don't have a huge error at one particular brief moment. Apart from one spike at the beginning of the tracking, there are no huge errors for the position. The velocity is not as even but the values are quite acceptable. The RMS error is 3.14m for position and 0.694m/s for velocity. Since the position error is approximately same as σ_z, such an error is what we we should expect. The peak error for position is 10.84m and is due to the start position not being the actual ground truth. The peak velocity error is 1.81m/s.

The NEES along with the NEES for position and velocity is plotted in Figure 4. The figure shows very acceptable values for the confidence interval. However there are a few spikes that are quite prevalent and not

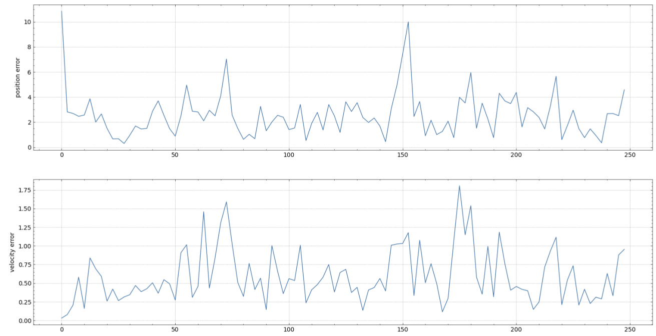

Figure 3: Error for the position and velocity as a function of time.

Figure 4: NEES for position, velocity and total plotted as function of time. With a 90% confidence interval

desirable.

2.2 Joyride data

The target in the joyride dataset is significantly more difficult to track - both in terms of getting no target loss and getting good enough and consistent performance. It includes sharp turns, turns with acceleration and subsequent turns in opposite direction. Also, at some extended periods there are no detections of the target. By combining all this knowledge, one can reason that as a bare minimum we need a continuous velocity (CV) process model which can handle both turning and accelerating at the same time. This equates to a CV-model with fairly large variance.

Such a state filter for the PDAF can be implemented through a pure EKF with a CV process model. Using the parameters summarized in Table 2, we get no track loss and the best possible performance from such a filter. However, only 67% of the NEES falls within a 90% confidence interval, which is far from ideal.

In order to improve the performance any further, we have to use a filter which models the process more closely at all times. Here, an IMM-PDAF is quite a

σ_{CV}	σ_z	P_{FA}	P_D	g
3	10	$1 \cdot 10^{-}5$	0.8	2.5

Table 2: The parameters used in an EKF-PDAF with CV process model on the joyride dataset.

good fit. From inspecting the dataset, one can observe that for extended periods the target does only one of the following: Either it moves with small changes in speed, it takes a turn, or it turns while making small adjustments to its speed. These three modes corresponds to respectively a CV, CT and a CV-high process model. Therefore, optimum performance is expected to be achieved when combining the three through an IMM.

This all boils down to the use of Bayes rule. If we e.g. know that the target has moved with constant velocity for quite some time, it is much more likely to continue to do so than to suddenly start turning. The same can be said when the target is in the middle of a turn. This is exactly what the IMM is based on. Therefore, one would expect the IMM-PDAF to deliver better performance than the EKF-PDAF. In many cases, a pure CV-CT process model would be sufficient. But since the target also accelerates while turning and changes turn rate while turning, the CV-high is expected to contribute to even better performance.

For the filter to work properly, it has to be correctly tuned to the given situation. A parameter that can be estimated to a certain degree, is the probability of detection P_D. By taking the same approach as for the simulated dataset, it seems that $\approx 80\%$ of the samples contain a true detection, and therefore this is our P_D.

Closely related to it, is the clutter intensity P_{FA}. A good guesstimate can be found by plotting the measurements for each time step as a movie. We observe that quite a few number of the measurements are clutter. Also, most of them originates far away from the target, and will be not even be considered by the filter as they are gated by the PDAF. Based on what was mentioned earlier about typical values for the P_{FA} and the observations from the plot, a value of $P_{FA} = 10^{-5}$ was chosen as a starting point. This also ended up being the most optimal value.

Furthermore, the process models have to be tuned. Ultimately, this boils down to finding well suited noise covariance values. Since the values corresponds to the amount of acceleration experienced, either linearly or in terms of turn, some sensible upper limits are easily decided. The target of interest is an agile boat, and so values should be in the order of 10^{-2} to $10^1 \approx 1G$. The values $\sigma_{CV}^2 = \sigma_{CT}^2 = 1^2$ and $\sigma_{CVH}^2 = 10^2$ was found to give the lowest possible NEES, as seen in Figure 8. $\sigma_\omega = 3 \cdot 10^{-3}$ was chosen to make the CT covariance ellipse have its shortes radius the same as CV and its longes radius the same as CV-high. When this relationship is present, one get the theoretically optimal switching between the three models, as they now model the different modes correctly.

Such a choice of values is a good compromise between being able to estimate both smooth turns and straight motion (CV and CT respectively) without letting measurement noise inflict the state estimate, while at the same time letting CV-high kick in when a quick change in motion happens. This is also exactly what we observe in Figure 8. Most of the time, the state is a mix of CV and CT since the turns are not perfect CT and the target never moves straight ahead. But at time instant 100 and 200, the target accelerates while turning, and so CV-high quickly kicks in and then disappears.

σ_{CV}	σ_{CVH}	σ_{CT}	σ_ω	σ_z	P_{FA}	P_D	g
1	10	1	$3 \cdot 10^{-3}\pi$	10	$1 \cdot 10^{-5}$	0.8	2.5

Table 3: Parameters used for CV-CT-CVhigh on joyride dataset.

One particularly interesting parameter in the case of the joyride is the covariance of the measurements σ_z^2. This indicates how much uncertainty exists in the detections. By plotting the ground truth together with the nearby measurements, we can see that on average the sensor is off by around 5-10 meters. Some of the detections are so off that they appear to be behind the vessel at some points. Both the extremes were tested as σ_z^2, and empirically $\sigma_z^2 = 10^2$ was found to give best performance on the dataset.

The transition matrix π also has to be decided. From the data, we see that we get a lot of measurements between each time the target changes mode, which is to be expected. However, this also tells us that the probability for changing mode should not be particularly high since we do not want to change mode often. At the same time, it should not be too difficult to change mode, as this would result in the mode changing too rarely. As a good compromise, the π matrix with diagonals of 0.9 and off-diagonals 0.05 was found to work best.

Lastly, the gate size g has to be decided. In essence, this is a multiplier in terms on standard deviations used on the track covariance to decide which measurements to consider by the PDAF. Intuitively, the value should lie in the range between 1.5 to 3.5 standard deviations to have high enough probability of gating a detection. Empirically we found that $g = 2.5$ worked best. This results in gating, statistically, 98.8% of the detections through the filter. By choosing a larger value, one would also include a lot of nearby clutter. On the other hand, a lower value resulted in track loss. This comes from the fact that transitions between models often happens such that the detected measurements now is far from the predicted state, and thus easily falls outside of the gate.

With the parameters as summarized in Table 3, we get no track loss, and 77% of our NEES falls within a 90% confidence interval. Albeit being significantly better than the 67% we got from the standard EKF-PDAF, neither does this filter work satisfactory. Because of the large spikes in NEES, there is the possibility that the filter would have gotten track loss during another tracking situation. Also, it behaves a bit weird with the spikes in the track around (6000, 2000) and (5700, 1800) and the uneven path at the start of the track. But at least it manages to regain its track when it is partly lost around (6200, 2300).

In Figure 7 the errors are plotted and it is clearly less desirable than for the simulated data. The velocity errors are greater in magnitude overall, but also have

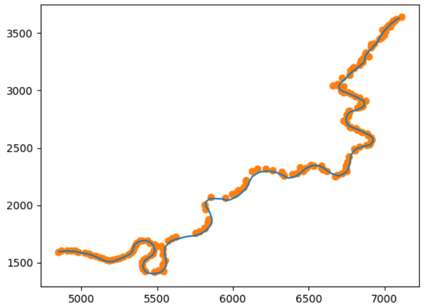

Figure 5: Plot of the true trajectory (blue) and nearby measurements (orange).

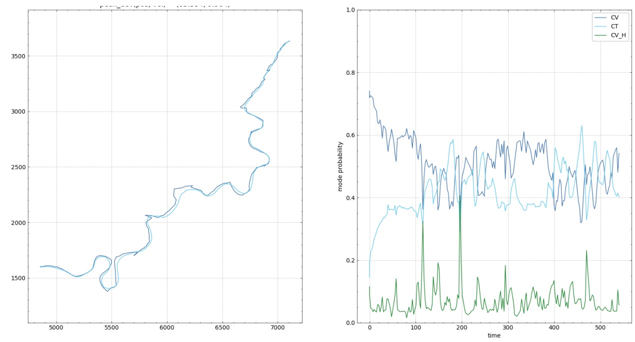

Figure 6: Left: the estimated track of the Joyride dataset together with the ground truth. Right: mode probabilities

more distinct spikes. The same is true for the position error. However, by comparing the position error to the estimated track it is clear that the error at time step 300 is due to missing measurements.

The RMS error for position is 25.9m and 4.127m/s for velocity. This is double of one standard deviation, σ_z, chosen for the measurement model. This is not satisfactory.

In Figure 9 the boats true path is plotted along with the estimated path and the measurements. Here it is possible to understand why some of the biggest errors in the tracking occur. The boat enters the figure in the upper right corner and follows the true path quite well. At (6200, 2300) the estimated path starts to deviate. From the plot we see that there is actually one measurement that is behind the previous one and together with the two following measurements, they make an almost straight line. This makes the estimated path have a heading that is quite far of the the actual heading of the boat. There also appear to be a few missing measurements after this, meaning the estimate continues on a path that deviates from the ground truth, making the error even greater. However, when a new measurement finally is present, the model quickly adjusts the heading and follows the ground truth very nicely.

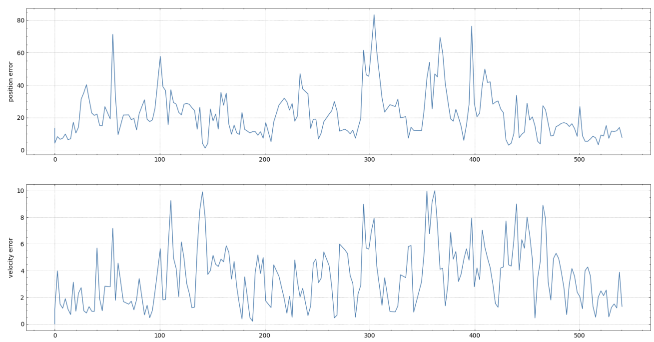

Figure 7: Error for the position and velocity as a function of time.

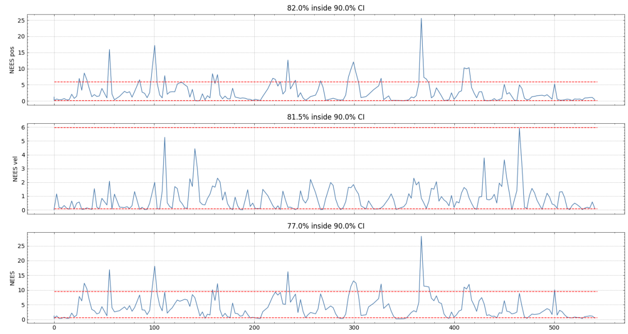

Figure 8: NEES for position, velocity and total plotted as function of time. With a 90% confidence interval

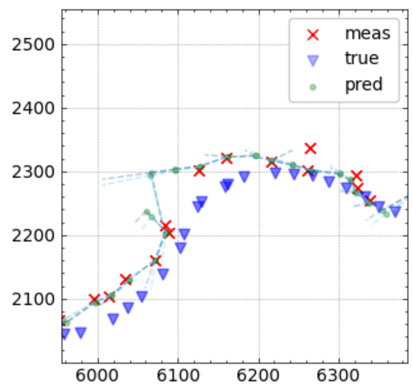

Figure 9: The predicted path along with the measurements and the ground truth for the boat.

3 Conclusion

As we have seen, the IMM-PDAF is a great tool for tracking targets in environments with a lot of error, assuming that we have Gaussian additive noise. The filter was found to give satisfactory performance on the simulated data, with $RMSE_{pos} = 3.14$m and $RMSE_{vel} = 0.694$m/s and 87% of NEES within a 90% confidence interval of the NEES. However, whereas it did not give track loss for the joyride dataset, the metrics were far from ideal. This PDAF yielded 77% inside 90% NEES confidential interval, $RMSE_{pos} = 25.9$m $RMSE_{vel} = 4.127$m/s It remains to explore other solutions on this dataset in order to get better performance. Either one can introduce IMU measurements on the target itself, or maybe use some sort of triangulation by the use of multiple radars in order to get a higher P_D and lower P_{FA}.