TTK4250 Sensor fusion
Graded assignment 1 IMM-PDAF

Written Fall 2020 By
Harald Minde Hansen and Peter Hatlebrekke Husebø

Abstract

This is the report for the Graded Assignment 1 in the course TTK4250 - Sensor fusion. The report consists of two sections, where section 1 answers Task 2 of the assignment and section 2 answers Task 3 of the assignment. We have no common conclusion for the whole assignment due to the restriction of four pages, but we make some comparisons with section 1 in section 2.

1 Tuning of the IMM-PDAF to the given data

In this section, we will discuss the tuning of parameters in the IMM-PDAF that we have made using a CV and a CT model with position measurements to the given data set in task 2 of the assignment. Our tuning parameters are the process disturbances of the models, the measurement noise, the detection probability, the false alarm intensity and the gate size.

1.1 Process disturbances vs. measurement noise

Process disturbances σ_a are errors in our model of the system, such as unmodelled dynamics from the environment the target moves in. In our IMM-PDAF, the two models are that the target moves in a straight line with constant velocity (CV) and that it turns with a constant turning rate (CT). For the CV model, we assume that the state change of the constant velocity is affected of some process disturbance which is assumed to be white noise. Integrating this state change to obtain the state, we get that the velocities for both x- and y-directions are independent Wiener processes. Since we assume the velocity to be constant, information about the acceleration, and hence the forces applied to the target, will not provide any further information about the movement of the target in the future. In the same manner, the CT model will not be affected by any knowledge about the torques. The measurement noise σ_z represents how certain we are on our measurements.

Setting the process disturbance large relative to the measurement noise implies that the predicted covariance in the EKF will be large relative to the measurement noise when calculating the innovation covariance for each sample. Since the innovation covariance is inverted and multiplied with the predicted covariance and the measurement Jacobian when calculating the Kalman gain, we will then get a larger Kalman gain under these circumstances. This means that we will not trust our model, but rather the measurements. Doing this, we risk to make our estimations diverge from the target as wrong measurements will be fully trusted and never corrected. Doing the opposite, i.e. setting the process disturbance small relative to the measurement noise will cause the updated steps in the EKF to trust the model rather than the measurements. This could cause an over-confident model, where we rely on a linear behaviour of our target. In our case, we observed that the ground truth trajectory moved in an almost straight line and turned with an almost constant turn rate from the ground truth trajectory in the left side of fig. 1, making it an almost linear movement. Hence, we found it reasonable to set the process disturbances to be less than the measurement noise. Yet, the turn rates seemed to change a bit during the CT modes, which caused us to set the disturbance for the CT model slightly larger than for the CV model. The measurement noise was tuned such that almost all measurements fell inside the ground truth trajectory $\pm 2\sigma_z$, as 95% of the measurements should be inside this interval.

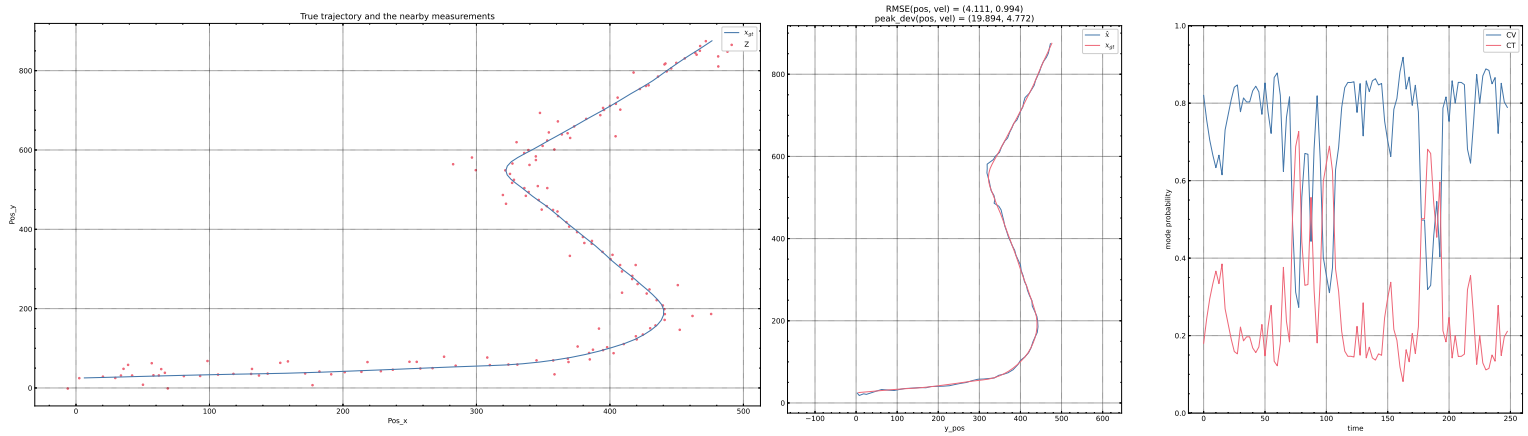
1.2 Detection probability, false alarm intensity and gate size

In the PDAF, the detection probability gives the probability for that the target actually gets detected for each time step. If we don't detect the target for a sample, we will simply use our model of the system as the estimate. Obviously, we want the detection of probability to be as large as possible, which can be done by lowering the detection threshold. This will unfortunately also cause the false alarm (clutter) probability to increase as well. We were given this value to be 0.90, and since this is within the range of the typical values from 0.50 to 0.95, we decided not to change this. The false alarm intensity was harder to tune, as we were not given an exact number of resolution cells N. This value is also said to suffer for huge uncertainties. For this reason, we ended up not adjusting this parameter neither, as it is difficult to say whether or not this will affect our tracking method. The validation gate for this 2D case is an ellipse which surrounds the space where measurements actually are considered. Setting this gate to be large, we would consider most of the measurements as true. With our probability of detection, we found it reasonable to set the validation gate to 5 standard deviations, such that as good as all measurements were considered. Setting it lower, we would be more certain about every measurement, but also miss out some measurements that actually existed.

1.3 Tracking data

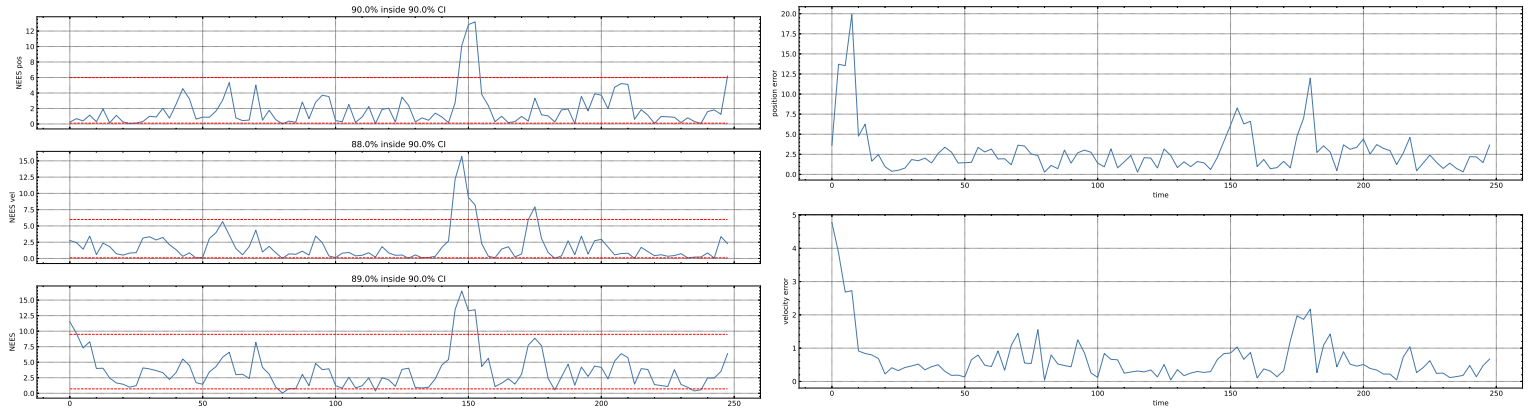
The IMM-PDAF works well according to the data extracted from fig. 2. As seen in b) the estimated trajectory follows the ground truth close all the time, with the largest discrepancy in the second turn (right turn). This is also seen in the position error plot, where it is increasing to about 10 meters. The normalized estimation error squared (NEES) is from a) seen to be 89% inside a 90% confidence interval. There is a peak about 6 units above the upper limit of the confidence interval after about 150 seconds. In this case the filter is overconfident, which makes it lose track. This is related to the same case where the estimated trajectory deviates from the ground truth. The positional Averaged normalized estimation error squared has a value of 3.70 with $CI = [3.55, 4.48]$. Thus we see that according to the NEES and ANEES the filter is in total very balanced, neither over- nor under-confident.

In the IMM-PDAF, we assume the measurements to be Gaussian distributed. Looking at the measurement samples against the ground truth trajectory, we observe that this is a rather fair assumption as it is almost the same amount of points of both sides with the vast majority inside 2 standard deviations from the ground truth. The exceptions are in the turns, where it seems to be an overweight of points on one side of the trajectory. Here, this assumption is violated, which can be some of the reason why we have the mentioned deviation in the second turn. Also, in the mixture reduction which is based on moment matching, wrong measurements may get larger weights than the correct ones, which can be seen in the deviation in the second turn, where the mixture reduction ends up choosing an outlier to be the correct one, even though there are better measurements that are omitted. Still, the filter works well in almost the whole trajectory, yet with a nice and almost linear behaviour of our target for this data set. The switching in the IMM also seems to be fine, where the probability for choosing the CT model is the largest in the two turns, while the CV probability is the largest on the straight lines.



a) b)

Figure 1: a) Ground truth and measurements b) Estimated trajectory and mode probabilities



a) b)

Figure 2: a) Total, position and velocity NEES b) Position and velocity error

2 Tuning of the PDAF/IMM-PDAF to the given Joyride data

In this section, we will discuss tuning of the PDAF on the data set Joyride, experimenting with use of EKF on a CV model and a CT model, in addition to a IMM-PDAF using CV-CT and CV-CT-CVhigh. The Joyride data set is much more challenging than the one we used in the previous task. As seen from the ground truth trajectory in fig.3, we observe a much more abrupt movement with frequent, fast turns. Here, we do not have long, approximately straight lines and slow turns with a constant turn rate. Hence, the assumptions of an almost linear CT and CV behaviour will be violated to a larger extent than earlier.

2.1 PDAF with EKF using the CV model

Using the EKF instead of an IMM provides easier overhead, but we will make the poor assumption that the target will move in an almost straight line with a constant velocity for all time samples. Therefore, we will not rely on our model to a large extent, which caused us to increase the process disturbance drastically, also relative to the measurement noise even though also the noise seemed to be larger in this data set. The measurement noise is set to be $\sigma_z = 10$ for all the following approaches, since this seemed to work reasonably well for our purposes. We sat the model disturbance to be $\sigma_a = 4$ to this noise for the CV model, and ended up with fig.3b). As this method is only suboptimal due to the poor assumptions, we did not expect an outstanding performance on this approach. However, we achieved a decent target tracking. We see that the largest discrepancies comes in the areas where the measurements have the most outliers (in particular where x holds the values 6000 to 6500) in the left of Fig.3:b), which agrees with the theory as we rely more in the measurements now.

2.2 PDAF with EKF using the CT model

As for the CV model, we again make an assumption that may be violated several times during the Joyride, as the turns are not done with a constant turning rate. Yet, we expect this model to perform better in areas where we have frequent fast turns. Setting the value of σ_ω to be too large, our estimates get stuck in angular circles with a large difference between them. This is because we now expect the turning rate to vary a lot. With our PDA with a given probability of detection P_D, we will have some samples where we assume that the measurements are not real. Therefore, we will use our predicted model with large changes in the turning rate, where we can end up with an angular circle. When we again get a measurement which we believe is from the target, the EKF will make us trust this measurement (with probability $1 - P_D$), and the estimate will perform a jump to this new measurement and maybe follow the next measurements, before the PDAF makes us use the model again (since we do not trust that all the measurements originates from the target), where we will create a new angular circle. Lower values of σ_ω will make the CT model to look more like a CV model, as we assume small variations in ω. The latter choice of σ_ω worked best for us, as it manages to track the target for low values. From fig.3b) we observe a pretty similar performance as for the CV model, but with a bit smaller

deviations in the turns (as e.g. when x holds the values 6000 to 6500), but slightly larger in the straight line, like in the beginning of the trajectory.

2.3 IMM-PDAF using CV-CT

Combining the two previous models in an IMM, we would expect the tracking to be better, as we could switch to the CV model on the almost straight lines, and the CT model in the turns. After some tuning, we ended up with the performance and switching shown in fig. 4a), with position and velocity error in fig. 5a) and its NEES in fig. 6a). We also obtained the $ANEES = 3.41$ with $CI = [3.68, 4.33]$. The ANEES tells that on an average the filter is barely underconfident, but looking at the NEES we see that it in some cases is overconfident. This makes it difficult for the model to stay close to the ground truth at all times, which is seen in the positional error in fig. 5a). When trying to cope with this by tuning the process noise of the models down it just got worse. This makes sense due to the fact that the ANEES said it was already barely underconfident.

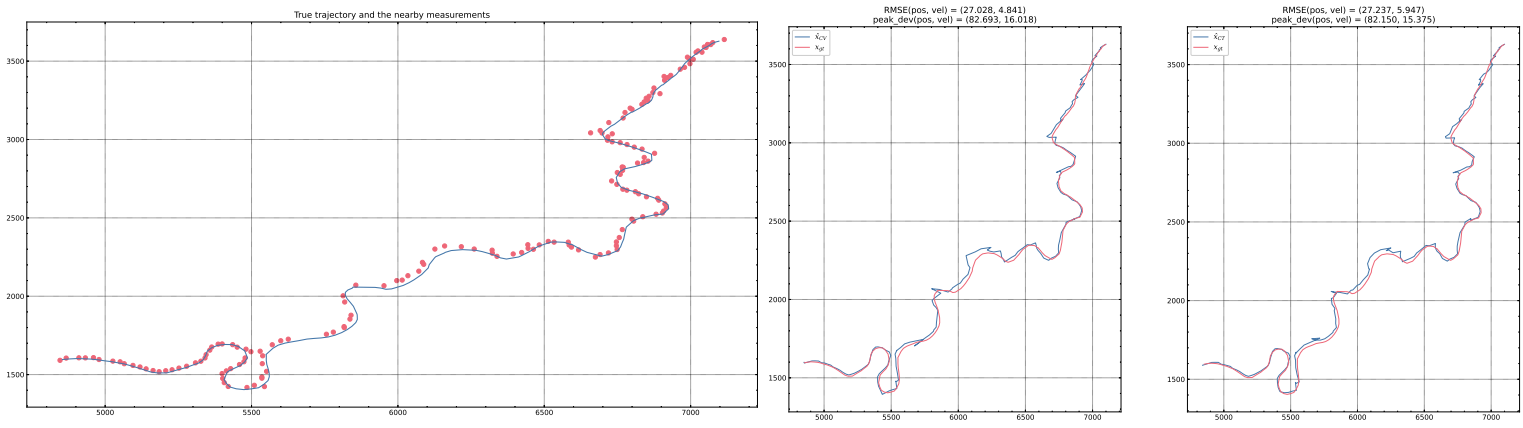
2.4 IMM-PDAF using CV-CT-CVhigh

Extending the previous IMM-PDAF to also include a CVhigh mode, we could hope for an even better performance where we switch to the CVhigh mode when an entire line is not straight, e.g. there is a longer distance between the measurements (given that the switching works optimally, of course, which is hard on this data set). We decided to set the transition probability matrix such that we could not go directly from CT to CVhigh and vice versa by setting these two elements to zero, since the CV-CT model worked quite well in the areas around the switching. Therefore, we will only allow CVhigh to be chosen when we are using the CV model and experience more disturbance. This increased the performance of our IMM-PDAF, since we didn't use the CVhigh mode in inappropriate areas. As one can see from the plots fig. 4b), we see that the CVhigh is chosen in some reasonable areas, and is not too trigger-happy. From the errors in fig. 5b) and NEES in fig. 6b), and the obtained $ANEES = 2.29$ with $CI = [3.68, 4.33]$ we can once more from the ANEES say that on average the filter is underconfident. And only 76.5% of the NEES inside $90\%CI$. Looking at fig. 6b) we see that it indeed is underconfident from time to time. It is therefore intuitively to think that this will be better if the process noise is tuned down. This also worked in practice, . We managed to get from $NEES = 76.5\%$ inside $90\%CI$ to $NEES = 87\%$ inside $90\%CI$. ANEES was changed from 2.29 to 3.03.

2.5 Assumptions and conclusion

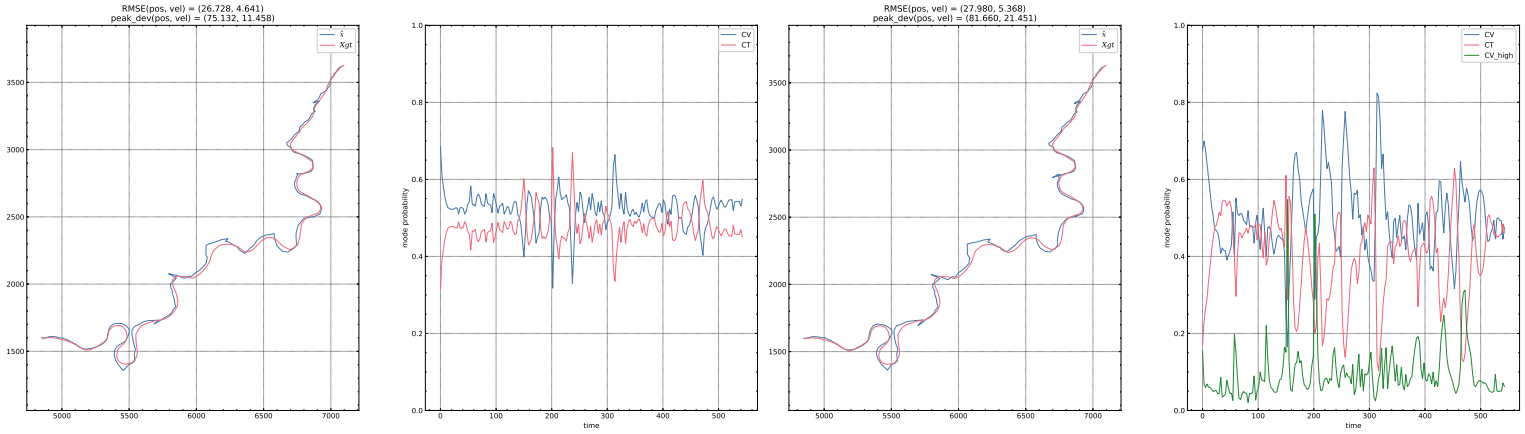
Analysing the trajectory, one could take a closer look on how ω changes over time by performing $atan2(y, x)$ operations over the whole ground truth trajectory, and whether or not σ_ω really is Gaussian distributed. This could help us to determine a realistic value of σ_ω, and to see to which extent the assumption of it being Gaussain distributed. An almost equivalent approach could be used to observe σ_a in the CV model. To limit this report and the amount of plots, we instead observe that we trajectory seems to overall have small, but frequent values for ω, causing us to set the σ_ω low.

We have assumed that the measurements are Gaussian distributed as discussed in the previous task. This also seems reasonable for this data set, even though we a have an overweight of measurement samples in some of the turns, as in the outliers where x holds the values 6000 to 6500. With respect to the performance, we managed to track the target for all four choices of PDAF. We got some errors, but due to the difficult data set, we think the PDAF worked tolerably fine.



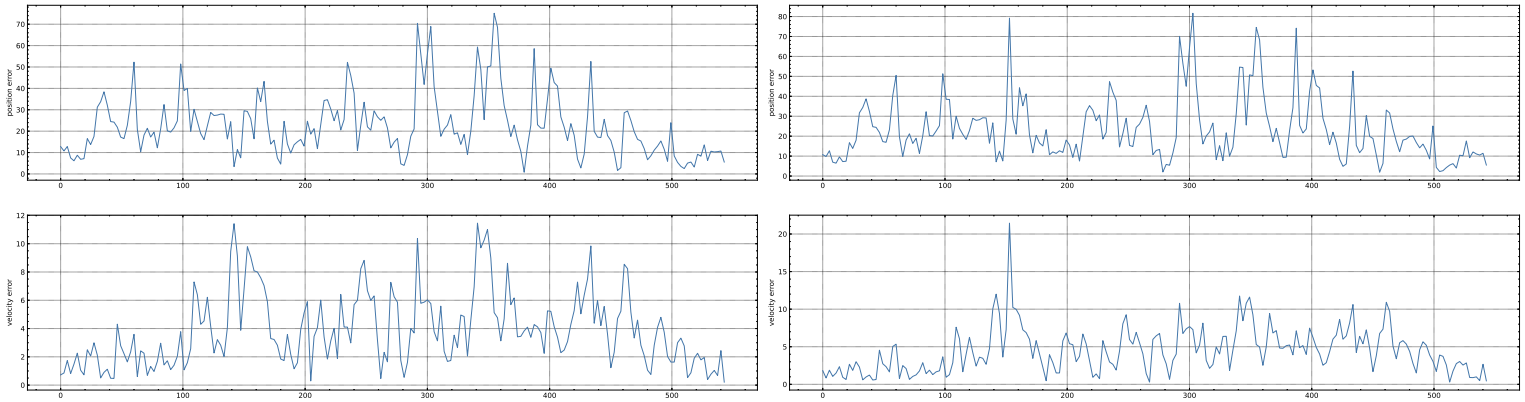
a) b)

Figure 3: a) Ground truth and measurements of joyride b) Estimated track by the use of EKF CV and CT



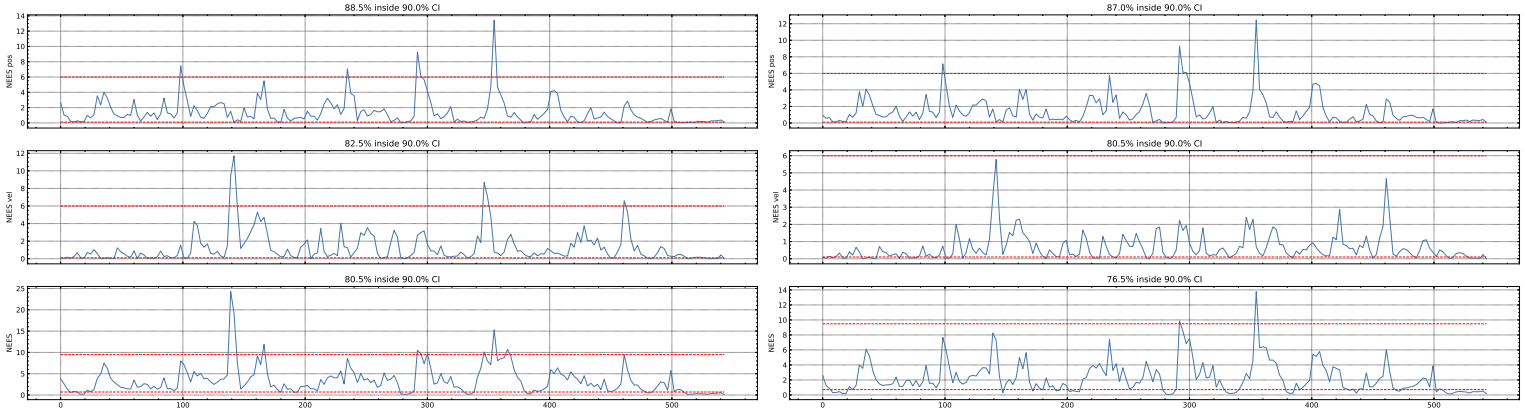
a) b)

Figure 4: a) Estimated trajectory and mode probability for IMM-PDA with CV-CT b) Estimated trajectory and mode probability for IMM-PDA with CV-CT-CVhigh



a) b)

Figure 5: a) Positional and velocity error for IMM-PDA with CV-CT b) positional and velocity error for IMM-PDA with CV-CT-CVhigh



a) b)

Figure 6: a) Total, position and velocity NEES for IMM-PDA with CV-CT b) Total, position and velocity NEES for IMM-PDA with CV-CT-CVhigh