TTK4250 IMM-PDA

Ludvig Løken Sundøen, Nadia Garson Wangberg

August 16, 2021

1 Abstract

In this assignment the Interacting Multiple Models algorithm (IMM) with the Extended Kalman filter (EKF) is combined with the Probabilistic data association filter (PDA) to create an IMM-PDA filter. IMM combines several modes, in our case a Constant Velocity process model (CV-model) and a Constant Turn-rate process model (CT-model). The PDA uses the IMM filter together with information such as the clutter intensity, detection probability and gate size. The final IMM-PDA filter is tuned to accurately fuse position sensor data with the process models for datasets such as the "Joyride" dataset.

2 Theory

2.1 The IMM-PDA

As explained in the abstract the IMM combines several modes. In our case two process models are combined with one measurement model into two Extended Kalman filters (EKF), meaning we get one EKF for each mode. The IMM uses these Kalman filters together with the transition matrix to estimate the state of our target.

What the IMM lacks however is handling misdetections and clutter, which the single-target tracking algorithm PDA deals with. The PDA is more than just filtering as it does not assume that the target always gives a measurement and also deals with measurements that does not come from the target (false alarms). Tuning the IMM-PDA is challenging as a lot of different parameters can be tuned. An understanding of all the parameters and how they affect the results is necessary in order to tune the IMM-PDA well.

2.2 Tuning Noise Parameters

Both the sensor and process model noise term can be tuned by changing their standard deviations σ_{sensor} and $\sigma_{process}$. This is important in order to get accurate Extended Kalman filters. One common way to tune these is by tuning while ensuring that the consistency conditions, NIS and NEES, are close to 1.

2.2.1 Measurement Noise σ_z

σ_z represents the standard deviation of the measurement, or the noise of our sensor. For our Joyride example σ_z was given to be between 6 and 10. σ_z can be tuned, by decreasing σ_z the measurements gets trusted more and by increasing it the process model is trusted more. σ_z can also be estimated by (1). The more data the better this estimate gets. σ_z can also sometimes be found from the sensors data-sheet, meaning someone has estimated it for us.

$$\sigma_z = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - x_{mean})^2} \tag{1}$$

2.2.2 Process Noise σ_a

σ_a represents the standard deviation of our acceleration, or the noise of our velocity process model. $\sigma_{a_{CV}}$ and $\sigma_{a_{CT}}$ denotes the standard deviation of the CV-model and CT-model respectively. A small σ_a means that we trust the process model. One could use physical conditions like $\sigma_{a_{max}}$ or an MLE to estimate it.

2.2.3 NIS and NEES

When tuning the noise covariances σ_a and σ_z consistency analysis can be helpful. The formula for the Normalized estimation error squared (NEES) and normalized innovation squared (NIS) can be found in equations (4.65) and (4.66) in Fundamentals of sensor fusion - Edmund Brekke. Both the NIS and NEES should preferably be 1, which means that the size of the error is consistent with the size of the covariance. If NEES is too large σ_a should be increased, if NIS is too large σ_z should be increased.

2.3 Tuning IMM Parameters

In addition to well tuned noise parameters the transition matrix Π_{ij} must be tuned. $\Pi_{ij} = Pr(s_k = 1|s_{k-1} = i)$ is the probability of transitioning from state i to state j. Where our states are CV and CT. Π_{ii} is often set to something between 0.7 and 0.95, as remaining in a state often is probable.

2.4 Tuning PDA Parameters

2.4.1 Clutter Intensity Λ

The PDA deals with false alarms μ, meaning measurements coming from clutter instead of the target. Clutter is often modelled as a poisson distribution where Λ is the intensity parameter $\mu(\phi) = Poiss(\phi; \Lambda)$. Estimating Λ is often inaccurate, so tuning it is required. Λ should be tuned high if there are a lot of false alarms, so a lot of measurements coming from clutter. It is typically a positive number close to zero.

2.4.2 Detection Probability P_D

P_D is the probability of detecting a target. It is typically between 0.5 and 0.95, but largely depends on the application. P_D should be set low if there is a lot of noise that could cause misdetections. It should also be set low if the target is hard to find and perceived as near invisible for the sensor.

2.4.3 Validation Gate Size g

The PDA assumes that the measurement from the target must be within some gate G, with center $\hat{z}$ and radius g. g is often a number larger than 3, this is important in order to be very sure that our true measurement is within the gate.

3 Task 2

We used values from the EKF in assignment 3 which was tuned for the same dataset. So the initialized values was $\sigma_{a_{CV}} = 0.35$, $\sigma_{a_{CT}} = 0.35$, $\sigma_\omega = \pi/15 \approx 0.209$, $\sigma_z = 3$ and $P_0 = diag([0, 0, 1, 1, 0])$ and $\mu_0 = [50, 50, 10, 10, \pi/4]$. All the other parameters was left to the default handed out values. The result was straight path with no turns. We then set $P_D \approx 1$ due to task stating that "missed detection removed". This is not necessarily a true assumption in the real world, and would likely be an over simplification.

At this point the IMM-PDA tracked fairly good, but with the average NEES values being a bit too low $ANEES_{pos} = 0.88$ $ANEES_{vel} = 0.44$, so σ_a and was tuned down for both the CV and CT model. This helped a bit but the model lost track during the second turn. This was a very abrupt turn and the NEES spiked to over 1000, it seemed like the CT model was unable to model it. This could be because the maneuvering index $\lambda = \frac{\sigma_v T^2}{\sigma_\omega}$ was too low. By turning σ_ω down, it regained track and the NEES values became $ANEES_{pos} = 0.92$ $ANEES_{vel} = 0.81$. Clutter intensity and gate size was also slightly changed to decrease the error. The final results can be seen in 1, 2, 3 and table 3.

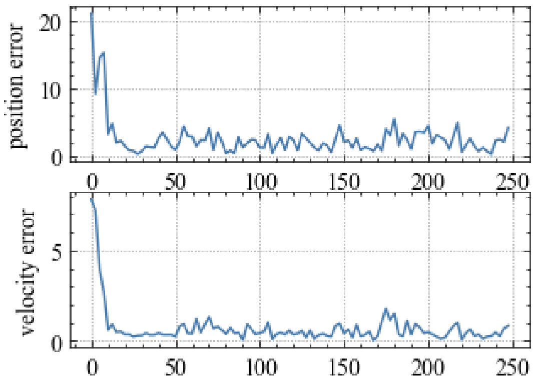

Figure 1: Estimation Error in task 2

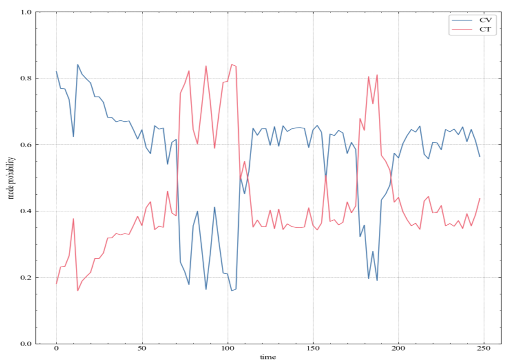

Figure 2: Mode probabilities in task 2

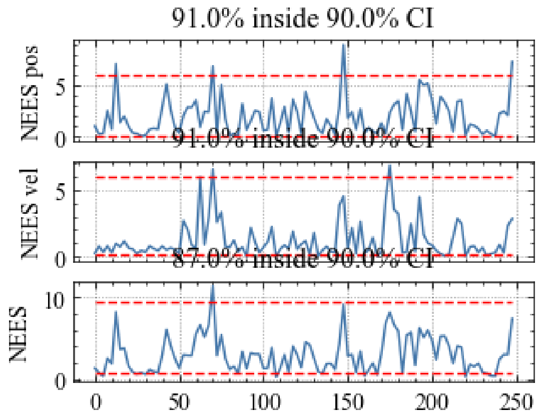

Figure 3: NEES in task 2

ANEES in position	1.98
ANEES in velocity	1.26
ANEES	3
NEESes in CI position	91%
NEESes in CI velocity	91%
NEESes in CI	87%
Position RMSE	28.09
Velocity RMSE	9.29

Table 1: Accuracy in task 2

4 Task 3

The parameters were initially set to the same as in task 2, but were later tweaked. The most distinctive change was the addition of an extra mode, a CV-process model with a higher variance. As can be seen in 5 CV High was not used that much, although, without it the IMM-PDA lost track. With the addition of a new mode, the transition matrix Π was expanded till a 3x3 matrix.

Starting out the most obvious parameter to tune was the initial state. The mean was set to the observed true starting position of the boat, which also meant that our confidence in the estimate was high, so the covariance was set low. Another approach could be to set the initial mean to the first measurement, this could work in scenarios when the true starting position is not known.

Tuning in reference to the consistency condition NEES was done to ensure model consistency. At first ANEES was too high, therefore σ_a was tuned up until ANEES approached 1 and the NEES occurrences inside the 90% CI

ANEES in position	5.36
ANEES in velocity	0.62
ANEES	7.12
NEESes in CI position	57.5%
NEESes in CI velocity	73.5%
NEESes in CI	73%
Position RMSE	285.00
Velocity RMSE	56.00

Table 2: Accuracy in task 3

became higher. The CI's could be seen in 6 As seen in 4, ANEES finally became $ANEES = 7.12$. Tuning σ_a higher after this point made the IMM-PDA lose track. This is likely because the IMM-PDA was made to trust the measurements more than the model due to the high process covariance. Those measurements it chose to follow where likely to be wrong at the moment it lost track. A solution could have been to increase σ_z, allowing σ_a to be tuned up and get a NEES closer to 1. σ_z was set to 6 as recommended from the task description. As we could not rely on the assumption of always detecting the target, the P_D was tuned down to $P_D = 0.95$. Finally the clutter intensity and gate size was tuned to minimize the error.

The IMM-PDA was rather difficult to tune for the Joyride dataset. A reason might be the boat's complex maneuvers that our simple CV and CT models might have been unable to fully capture. Maybe our process models were an over simplification.

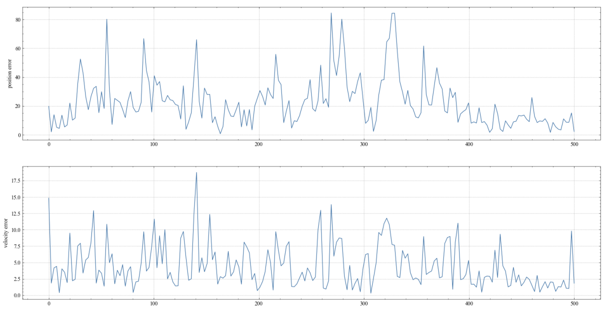

Figure 4: Estimation Error in task 3

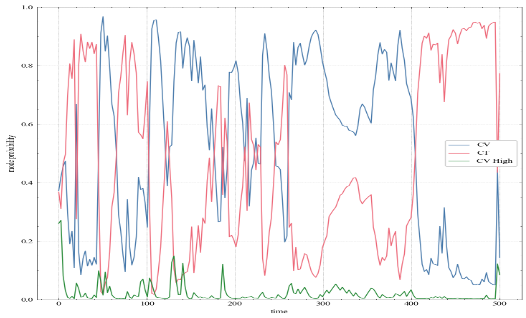

Figure 5: Mode probabilities in task 3

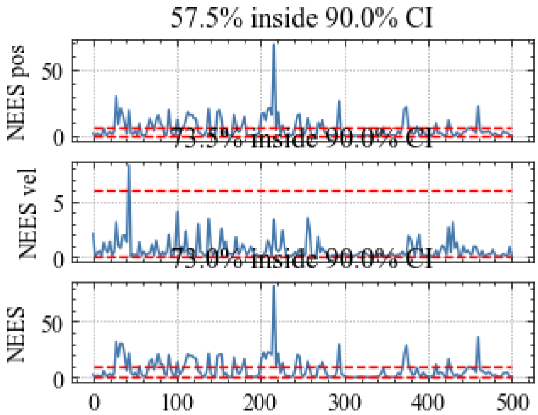

Figure 6: NEES in task 3

5 Reflection

Several approximations, which could have impacted the results has been done. IPDA is an extension of the PDA, it calculates a probability that a target exists instead of assuming existence. It has track confirmation and termination and could be a good way to improve results. The Extended Kalman filter linearities about the estimate and assumes gaussian noise, for highly non linear systems a particle filters could be an improvement. Such changes could be easily implemented due to the modularity of the system.