EE793 Target Tracking: Computer Exercise 4

Due: 02.06.2013, 23:59

In this fourth computer exercise (CE-4), you are going to implement multi target data association algorithms called as the global nearest neighbor (GNN) and joint probabilistic data association (JPDA) algorithms. For implementing these algorithms, you need to use the single target tracking code you wrote in Computer Exercise-2 as a backbone. What you wrote then was actually a code that handles multiple target tracking (although there was one true target, there might have been more than one confirmed targets) using single target tracking association methods like NN or PDA. These algorithms did not take into account that there might be more than one confirmed target and took care of them independently. Here, in this CE-4, the aim is to implement their multiple target versions.

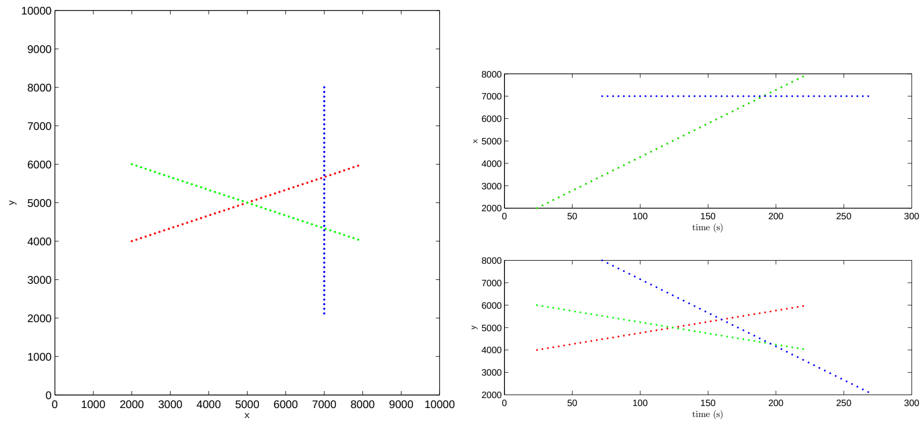
1 Global Nearest Neighbor (GNN) Algorithm

We are going to consider a three true target scenario in the region $0 \leq \mathsf{x}, \mathsf{y} \leq 10\,\text{km}$ as in CE-2. Below, the steps to be implemented are summarized. Some steps are exactly the multiple target versions of the steps in CE-2. You are required to modify your CE-2 implementation accordingly.

a) Download the true targets data from the web address `http://www.eee.metu.edu.tr/~umut/ee793/files/xdata.mat`.
When you load this mat file in Matlab, it is going to load into your workspace a three 5×50 size matrices named `x1data`, `x2data` and `x3data`. These are the data files corresponding to the true states of the three targets you are going to consider in this exercise. The first row of each matrix contains the time stamps for the true state vectors $[p_k^{\mathsf{x}}, p_k^{\mathsf{y}}, v_k^{\mathsf{x}}, v_k^{\mathsf{y}}]^T$ of each target in the subsequent 4 rows. Note that the targets data can start and end at different time values which are specified by the time stamp information. You can illustrate the positions of these targets on a 2-D $\mathsf{x} - \mathsf{y}$ plane as in Figure 1(a). However, such an illustration does not contain the important time information of when targets start and end. Also one cannot make judgments about whether the target trajectories intersect or not. Due to these, here, we are going to prefer the recent illustration technique in Multi TT (MTT) which plots the x and y components of the targets on separate figures with respect to time as in Figure 1(b). Create your time axis in Matlab as `t=0:4:300` seconds, i.e., $T = 4$ seconds in this exercise. Plot the targets positions as in Figure 1(b) which we will call as "MTT-illustration".

b) Clutter Generation: In this part, you are going to generate your own clutter in the exactly same way as in CE-2. We assume that the number of clutters are Poisson distributed with rate $\beta_{FA} = 10^{-7}$ (number of FA/area/scan). The spatial distribution of the FAs is going to be uniform in the region $0 < \mathsf{x}, \mathsf{y} < 10\,\text{km}$. Now, generate sets of such clutter representing the clutter sets we are going to receive at 76 time instants of $0 : 4 : 300$secs.

c) Measurement Generation: We measure the targets position with a measurement standard deviation of $50\,\text{m}$ for both x and y components. We assume that the detection



(a) Standard Illustration: Missing time data.

(b) MTT Illustration

Figure 1: Different illustrations for multi target tracking. We prefer (b) in this exercise.

probability is $P_D = 0.9$. Generate the target originated measurements. Now add the target originated data to the 76 sets of clutters considering the time stamps of the targets. Observe this overall measurements in the form of a MTT illustration as in Figure 1(b).

d) Multi TT with Single TT methods: Run your CE-2 code with these measurements which would do multi target tracking with single target tracking data association (NN and PDA). Observe the tracker results in the form of MTT-illustration: Plot only confirmed track estimated positions (when there are any) with respect to their time stamps as in Figure 1(b). You can give each confirmed track an identification number and choose different colors depending on these numbers.

e) Multi TT with GNN: Now check the notes on GNN and implement a GNN associator. This logic should replace your confirmed track association logic in your CE-2 code.

- You should do the gating and form a validation matrix as usual.
- Then using the log-likelihood terms specified in GNN lecture notes, form an assignment matrix whose rows correspond to measurements and columns corresponding to tracks and external sources. See the second matrix from above in Slide 18 of Lecture 4. You can take $\times = -\infty \equiv -10^{10}$ and $\beta_{EX} = \beta_{FA}$.
- Obtain the best associations by inputing your assignment matrix to `Auction.m` downloadable from the link `http://www.eee.metu.edu.tr/~umut/ee793/files/Auction.m`. See the additional notes in the m-file.
- Update the tracks with their associated measurements if any. If the track is not associated to any measurement, make only the prediction update, i.e., no measurement update.
- Send the unused measurements, which are nothing but the measurements associated to the external sources, to the initiation logic.

You can use the initiation logic you have implemented in CE-2. By running NN association and GNN association on your measurement data several times, try to observe significant differences between the results if any.

2 Joint Probabilistic Data Association (JPDA) Algorithm

This exercise uses the implementation of the previous one but just replaces GNN association with JPDA.

a) Multi TT with JPDA: Implement a JPDA association and update procedure. The update mechanism of JPDA is exactly the same as PDA that you implemented in CE-2. Only the association probabilities have to be calculated differently. The process you need is summarized below.

- You should do the gating and form a validation matrix as usual.
- Using the binary validation matrix, calculate the JPDA association probabilities. A code segment for the calculation of these probabilities from the validation matrix is given in the m-file `JPDA_ProbCalc.m` downloadable from `http://www.eee.metu.edu.tr/~umut/ee793/files/JPDA_ProbCalc.m`. The code in the m-file lists all possible hypotheses and calculates their probabilities and gives at the end $n_T \times (m+1)$ size probability matrix whose rows correspond to (current confirmed) tracks and columns correspond to measurements (n_T is the number of the current confirmed tracks and m is the number of measurements. The last column represents no measurement association probabilities for each track i.e., the hypothesis θ_0. Note that the code is a script and not self-running. You have to study it and make adaptations to your own data structures if you want to use it. See the additional notes in the m-file.
- Update the tracks with measurements in the same way as PDA using JPDA probabilities.
- Send the unused measurements, i.e., the measurements not in the gate of any track, to the initiation logic.

You can use the initiation logic you have implemented in CE-2. By running PDA association and JPDA association on your measurement data several times, try to observe significant differences between the results if any.