# Target Tracking: Computer Exercise 2

Due: 08.04.2013, Monday, 23:59

## 1 Track Initiation, Maintenance and Deletion

This computer exercise is about track handling in clutter and missed detections. We are going to consider a single target in the region $0 \leq \mathsf{x}, \mathsf{y} \leq 10\,\text{km}$.

**a) True Target Data Generation:** In this part you are going to create your target's true data. For this purpose consider the model

$$x_k \triangleq \begin{bmatrix} p_k^{\mathsf{x}} & p_k^{\mathsf{y}} & v_k^{\mathsf{x}} & v_k^{\mathsf{y}} \end{bmatrix}^T = \begin{bmatrix} I_2 & TI_2 \\ 0 & I_2 \end{bmatrix} x_{k-1} + \begin{bmatrix} \frac{T^2}{2} I_2 \\ TI_2 \end{bmatrix} a_k \tag{1}$$

where $I_2$ represents $2 \times 2$ size identity matrix. We will use $T = 1\,\text{s}$. The initial state is

$$x_0 \triangleq \begin{bmatrix} 5\,\text{km} & 5\,\text{km} & 25\,\text{m/s} & 25\,\text{m/s} \end{bmatrix}^T \tag{2}$$

Each element of $2 \times 1$ size acceleration noise $a_k$ is distributed with $\mathcal{N}(0, \sigma_a^2)$ where $\sigma_a = 2\,\text{m/s}^2$. Generate 100 seconds of the state trajectory for this target and observe the position and velocity components.

**b) Clutter Generation:** In this part, you are going to generate your own clutter. We assume that the number of clutter measurements are Poisson distributed with the rate $\beta_{FA} = 10^{-7}$ (number of FAs/area/scan) i.e., the probability that the number of FAs in any region $V$ is equal to $m_k$ is given as

$$P_{FA}(m_k) = \frac{(\beta_{FA} V)^{m_k} \exp(-\beta_{FA} V)}{m_k!} \tag{3}$$

The spatial distribution of the FAs is going to be uniform in the region $0 < \mathsf{x}, \mathsf{y} < 10\,\text{km}$. Now, generate 200 sets of such clutter representing the clutter sets we are going to receive at integer time instants in $[0, 199\,\text{s}]$.

**Hint:** Probably the easiest way to generate such a clutter is to select first the number of FAs $m_k \sim P_{FA}(.)$. This can be done in Matlab by `poissrnd(.)` with input $\beta_{FA} V$ where $V$ is the volume of the whole surveillance region. After selecting this number, one can generate uniform `x` and `y` values between $[0, 10000\,\text{m}]$ to obtain the coordinates of the noise using the commands:

```
>>xvalues=10000*rand(1,m_k);
>>yvalues=10000*rand(1,m_k);
```

Here, the command `rand(·)` generates uniform numbers between 0 and 1.

**c) Measurement Generation:** We measure the target position (`x` and `y` coordinates) with a measurement standard deviation of 20 m's for both `x` and `y` components. We assume that the detection probability $P_D = 0.9$. Generate the target originated measurements. Now add the target originated data to the 200 sets of clutters by selecting a random target start time (integer) $T_s$ distributed uniformly over the integers 0 and 50 seconds.

**Hint:** Note that the detection process can be simulated by generating a uniform random number $u \sim U(0, 1)$ for each measurement time and then by checking whether $u \lesseqgtr P_D$. If $u \leq P_D$, this means that the target is detected and if not, the target is not detected.

**d) Track Initiator:** In this part you are going to implement your track initiation algorithm. Implement an $M/N$-logic track initiator (a simple selection of parameters is the one we chose in class 2/2&2/3). You can use a simple nearest neighbor association for the track initiation:

- If there are more than one measurements in the gate of an initiator, assign the closest measurement to the initiator.
- If there is a single measurement falling into the gates of two or more initiators, assign the measurement to the oldest initiator.

Try running your track initiation algorithm with different FA rates $\beta_{FA}$ (e.g., $\beta_{FA} = 10^{-8}$ and $\beta_{FA} = 10^{-6}$) and see the effect. Comment on your results. You can find some instructions on how to implemented a general $M/N$ track initiator in the document "Notes on CE2".

**e) Tracker:** For the confirmed targets, implement single target trackers that use

- Nearest neighbor association.
- Probabilistic data association.

for each confirmed track. For track initiation and confirmation, you should use your initiator implemented in the previous step. Implement a confirmed track deletion logic and implement it as well. Observe your results with plots similar to those shown in the video file in the document "Notes on CE2". Try running your tracker with different FA rates $\beta_{FA}$ (e.g., $\beta_{FA} = 10^{-8}$ and $\beta_{FA} = 10^{-6}$) and see the effect. Comment on your results.

Note that with the tracker that you implemented in this exercise, although you implemented only a single target tracker, you actually have a multi target tracker which treats each target independently. So this is how you do primitive multi target tracking using a combination of single target trackers. Even if you introduce measurements of some other targets, your tracker can now track multiple targets. Actually, it might be true that, what you have implemented is currently used in very old radars all over the world today.