# Target Tracking: Notes on Computer Exercise 2

Umut Orguner

March 19, 2013

This document gives some notes on Computer Exercise 2. The algorithmic suggestions given below are optional and you can use any other algorithm to implement the corresponding methods. The first and second sections give ideas on how to implement an $M/N$ logic initiator and how to draw gate ellipsoids respectively. The formulae for the association probabilities are given in Section 3. Section 4 gives the link for an example run about how your algorithms would work when you implement them.

## 1 How to implement a general $M/N$-logic initiator

We consider a general $N_1/N_1\&M_2/N_2$-logic where $N_1 > 0$, $0 < M_2 < N_2$ are integers representing the $M/N$ logic parameters. This logic initializes a tentative track if

- we get all of the $N_1$ measurements (inside the gate) in the first $N_1$ scans and
- we get at least $M_2$ measurement (inside the gate) in the following $N_2$ scans (after the first $N_1$ scans).

and deletes it otherwise. Implementation of such an algorithm requires one to define a state such that once we know the current state, it is possible to calculate the next state based on the current measurements. For this purpose, we define the following state for the $M/N$ logic.

$$x_k^{M/N} = \left[\begin{array}{cccc} s_k & m_k & \bar{m}_k & n_k \end{array}\right]^T \tag{1}$$

where

$s_k \in \{0, 1, 2\}$: representing deleted, tentative and confirmed states respectively.
$m_k \in \{1, \ldots, N_1 + N_2\}$: total number of measurements collected since the initiator is created.
$\bar{m}_k \in \{1, \ldots, N_1 + N_2\}$: total number of measurements missed since the initiator is created.
$n_k \in \{1, \ldots, N_1 + N_2\}$: age of the initiator

Note that this is not the lowest dimension that state can have because obviously $\bar{m}_k = n_k - m_k$. A pseudo code of the $M/N$ track initiation logic can then be given given as follows.

**Algorithm 1** $M/N$***-logic***

- ***Initialization:*** *The first measurement* $y_k \neq \phi$ *of the tentative track arrives. Set*

$$x_k^{M/N} \triangleq \left[\begin{array}{cccc} 1 & 1 & 0 & 1 \end{array}\right]^T \tag{2}$$

- ***Recursive Update with*** $y_{k+i}$ ***for*** $i = 1, \ldots, N_1 + N_2$***:***
  - *Set age* $n_{k+i}$ *as* $n_{k+i} = n_{k+i-1} + 1$.
  - *Update* $m_{k+i}$ *and* $\bar{m}_{k+i}$ *as*

$$m_{k+i} = \begin{cases} m_{k+i-1}, & y_{k+i} = \phi \\ m_{k+i-1} + 1, & y_{k+i} \neq \phi \end{cases} \qquad \bar{m}_{k+i} = \begin{cases} \bar{m}_{k+i-1} + 1, & y_{k+i} = \phi \\ \bar{m}_{k+i-1}, & y_{k+i} \neq \phi \end{cases}$$

– *Decide on* $s_{k+i}$ *as*

*If* $n_{k+i} \leq N_1$ *(we are in the* $N_1/N_1$ *period)*
*If* $\bar{m}_{k+i} > 0$
*Set* $s_{k+i} = 0$ *(deleted)*
*Else*
*Set* $s_{k+i} = 1$ *(still tentative)*
*End of If*
*Else (we are in* $M_2/N_2$ *period)*
*If* $\bar{m}_{k+i} > (N_2 - M_2)$
*Set* $s_{k+i} = 0$ *(deleted)*
*ElseIf* $m_{k+i} \geq (N_1 + M_2)$
*Set* $s_{k+i} = 2$ *(confirmed)*
*Else*
*Set* $s_{k+i} = 1$ *(still tentative)*
*End of If*
*End of If*

■

Notice that in the above algorithm one should continuously check the state $s_k$ of the initiator and take it away from the initiator logic as soon as it is confirmed or deleted. In other words, the decision time can be earlier than $N_1 + N_2$.

## 2 Drawing Gates

One can visualize initiators and tracks by drawing their gates which are simply the regions

$$(y - \hat{y}_{k|k-1})^T S_{k|k-1}^{-1} (y - \hat{y}_{k|k-1}) = \gamma_G \tag{3}$$

where $\gamma_G$ is the gate threshold. Suppose we have the factorization $S_{k|k-1} = U^T U$ which can be obtained either with a Cholesky decomposition or singular value decomposition(`cholcov(.)`, `svd(.)` in Matlab). Once we obtain it, we have the equality for the gate boundary transformed into

$$\|U^{-T}(y - \hat{y}_{k|k-1})\|^2 \leq \gamma_G \tag{4}$$

Then we have the parametrization of the gate ellipsoid with the a variable $\theta \in [0, 2\pi]$ as

$$y = \hat{y}_{k|k-1} + \sqrt{\gamma_G} U^T \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix}^T \tag{5}$$

Hence a simple procedure to draw the gate ellipsoid is

**Algorithm 2** *Drawing the gate ellipsoid*

- *Factorize* $S_{k|k-1}$ *as* $S_{k|k-1} = U^T U$.
- *Generate 100 uniformly spaced* $\theta$ *values in* $[0, 2\pi]$.
- *Draw the following values* $y$ *onto the measurement space for all* $\theta$ *values.*

$$y = \hat{y}_{k|k-1} + \sqrt{\gamma_G} U^T \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix}^T \tag{6}$$

■

One issue with using `cholcov(.)` that it can sometimes complain and give an error about the symmetry of the input matrix. This is generally solved by adding a command similar to
$P_{k|k} = 0.5(P_{k|k} + P_{k|k}^{\mathrm{T}})$
after each Kalman measurement update into the code. This command would just symmetrize the updated matrix if it is not already symmetric which can be caused by numerical factors. This is one of the important dirty tricks that are generally used even if you are not going to use `cholcov(.)` at all and it would make your Kalman filter implementation numerically better in general.

## 3 Formulae for the Association Probabilities $p(\theta_i|Y_{0:k})$ in PDA

In the class, we have shown the measurements in the gate as $Y_k = \{y_k^i\}_{i=1}^{m_k}$. We had the following hypotheses about these measurements

$$\theta_0 = \{\text{All of } Y_k \text{ is FA i.e., no target originated measurement in the gate.}\}$$
$$\theta_i = \{\text{Measurement } y_k^i \text{ belongs to target, all the rest are FA.}\}$$

The the probabilities $p(\theta_i|Y_{0:k})$ can be calculated as

$$p(\theta_i|Y_{0:k}) \propto \begin{cases} (1 - P_D P_G)\beta_{FA}^{m_k} & i = 0 \\ P_D p(y_k^i|Y_{0:k-1})\beta_{FA}^{m_k-1} & \text{otherwise} \end{cases} \tag{7}$$

where

$$p(y_k^i|Y_{0:k-1}) = \mathcal{N}(y_k; y_{k|k-1}, S_{k|k-1}) \tag{8}$$

is the predicted measurement likelihood which uses the measurement prediction $y_{k|k-1}$ and the innovation covariance $S_{k|k-1}$ calculated by the Kalman filter.

Note that the probabilities in (7) must be normalized to unity. When the number of measurements $m_k$ in the gate is large, calculating the terms $\beta_{FA}^{m_k}$ and $\beta_{FA}^{m_k-1}$ can be a problem. For this reason the following equivalent form can be used.

$$p(\theta_i|Y_{0:k}) \propto \begin{cases} (1 - P_D P_G)\beta_{FA} & i = 0 \\ P_D p(y_k^i|Y_{0:k-1}) & \text{otherwise} \end{cases} \tag{9}$$

These formulas are equivalent to those given in the textbook in Section 6.6.1 and equation (6.32).

## 4 Example

There is a video file that I have formed from my implementation that you can watch to see how the tracker result looks like at the end. See the link below.

- Nearest neighbor tracker with 2/2&2/3 logic initiator.

At each step I show the current initiators (tentative tracks) and confirmed tracks in the system. I never delete the measurements, so the dots seen include also the older measurements collected.

For reference, a single run of the NN tracker with 2/2&2/3 logic takes in my computer (an Intel Core-2 Quad with 4GB RAM) about 0.6 seconds.