Target Tracking: Lecture 3
Single Target Tracking Issues

Umut Orguner
umut@metu.edu.tr
room: EZ-12 tel: 4425

Department of Electrical & Electronics Engineering
Middle East Technical University
Ankara, Turkey

Lecture Outline

- Track life
  - Gating
  - M/N logic based track life
  - Score based track life
  - Confirmed track deletion
- Single Target Data Association
  - Nearest neighbors (NN)
  - Probabilistic data association (PDA)

Basic ideas on track life

False alarms and clutter complicate target tracking even if we know that there is at most one target.

- **Track initiation:** Form **tentative tracks** and confirm only the persistent ones (ones getting data).
- **Track maintenance:** Feed the confirmed tracks with only relevant data. Closely related to **data association**. Measure "quality" if possible.
- **Track deletion:** Delete tracks with persistent absence of data or with low "quality".

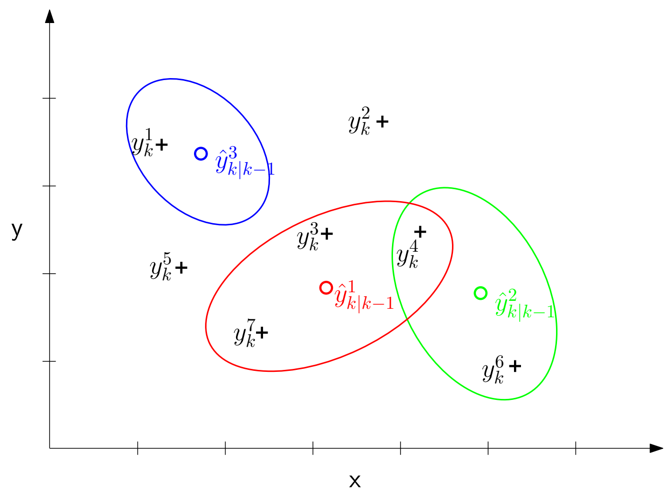
Curse of combinations!

We cannot consider all target-measurement combinations when the number of involved quantities is $\gtrsim 10$. The combinatorics and related probabilities becomes trouble in this case (even if most of the probabilities turn out to be almost zero in machine precision).

Gates and Gating

- **Gating** is the hard decisions made about which measurements are considered as valid (feasible) measurements for a target.
- The region in the measurement space that the feasible measurements for a target is allowed to be is called as target's **gate**.

Gating Illustration



$\hat{y}^i_{k|k-1}$: Predicted measurement position for the ith target

y^j_k : jth measurement

Gating Choices

Many different choices are possible which are basically a form of distance measuring between the predicted measurement and the measurements depending on the target uncertainty.

- Rectangular (low computation)

$$|y^{\mathsf{x}}_k - \hat{y}^{\mathsf{x}}_{k|k-1}| \lessgtr \kappa\sigma^{\mathsf{x}}_{k|k-1} \quad |y^{\mathsf{y}}_k - \hat{y}^{\mathsf{y}}_{k|k-1}| \lessgtr \kappa\sigma^{\mathsf{y}}_{k|k-1}$$

  where κ is generally ≥ 3.
- Ellipsoidal (most common)

$$(y_k - \hat{y}_{k|k-1})^T S^{-1}_{k|k-1}(y_k - \hat{y}_{k|k-1}) \lessgtr \gamma_G$$

  where γ_G is the gate threshold.

Ellipsoidal Gating Demystified

When the target state and measurement models are correct

$$\tilde{y}_k \triangleq y_k - \hat{y}_{k|k-1} \sim \mathcal{N}(0_{n_y}, S_{k|k-1})$$

Suppose $S_{k|k-1} = U_k U_k^T$ where U_k is invertible. Then,

$$\tilde{y}_k^T S^{-1}_{k|k-1}\tilde{y}_k = \tilde{y}_k^T U_k^{-T} U_k^{-1}\tilde{y}_k = \|U_k^{-1}\tilde{y}_k\|_2^2 \triangleq \|u_k\|_2^2$$

where

$$u_k \triangleq U_k^{-1}\tilde{y}_k \sim \mathcal{N}(0_{n_x}, I_{n_x})$$

Ellipsoidal Gating Demystified

When the target state and measurement models are correct

$$\tilde{y}_k \triangleq y_k - \hat{y}_{k|k-1} \sim \mathcal{N}(0_{n_y}, S_{k|k-1})$$

Suppose $S_{k|k-1} = U_k U_k^T$ where U_k is invertible. Then,

$$\tilde{y}_k^T S^{-1}_{k|k-1}\tilde{y}_k = \tilde{y}_k^T U_k^{-T} U_k^{-1}\tilde{y}_k = \|U_k^{-1}\tilde{y}_k\|_2^2 \triangleq \|u_k\|_2^2$$

where

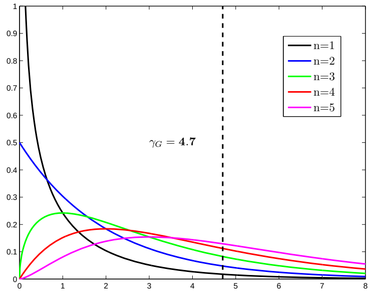
$$u_k \triangleq U_k^{-1}\tilde{y}_k \sim \mathcal{N}(0_{n_x}, I_{n_x})$$

Chi-square distribution

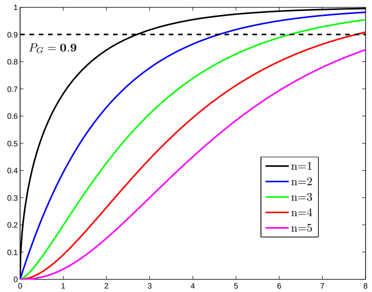
Sum of squares of n i.i.d. scalars distributed with $\mathcal{N}(0,1)$ is χ^2-**distributed with degrees of freedom** n.

Ellipsoidal Gating Demystified

Gate statistic $\tilde{y}_k^T S_{k|k-1}^{-1} \tilde{y}_k$ is $\chi^2_{n_y}$-distributed.



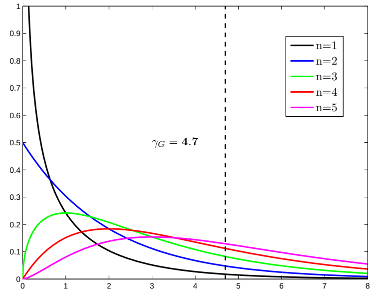
chi2pdf(.,n)



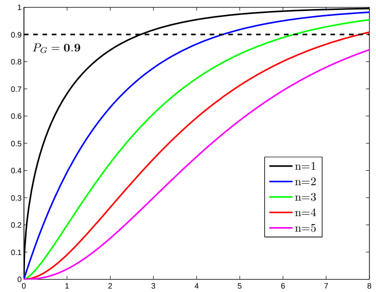
chi2cdf(.,n)

Ellipsoidal Gating Demystified

Gate statistic $\tilde{y}_k^T S_{k|k-1}^{-1} \tilde{y}_k$ is $\chi^2_{n_y}$-distributed.



chi2pdf(.,n)



chi2cdf(.,n)

Magic Command

Use γ_G=chi2inv(P_G,n_y) to get your unitless threshold.

Gating at $k = 1$

- Gating should be possible after getting the first measurement y_0 at time $k = 0$. In other words, at time $k = 1$, the gating should be applied on y_1.
- For this, $\hat{y}_{1|0}$ and $S_{1|0}$ should be calculated.
- With a single measurement, it might generally not be possible to form a proper state covariance $P_{0|0}$.
- Use measurement covariance and prior info about the target to form an initial state covariance $P_{0|0}$.

Gating at $k = 1$

Filter Initiation

Example-1: Cartesian state, position only measurements:

$$x_k \triangleq \begin{bmatrix} \mathsf{x}_k & \mathsf{y}_k & v_k^{\mathsf{x}} & v_k^{\mathsf{y}} \end{bmatrix}^T$$
$$y_k \triangleq \begin{bmatrix} \mathsf{x}_k & \mathsf{y}_k \end{bmatrix}^T + \begin{bmatrix} e_k^{\mathsf{x}} & e_k^{\mathsf{y}} \end{bmatrix}^T$$

where $e_k^{\mathsf{x}} \sim \mathcal{N}(0, r_{\mathsf{x}})$ and $e_k^{\mathsf{y}} \sim \mathcal{N}(0, r_{\mathsf{y}})$.

Suppose we got y_0, what would be $\hat{x}_{0|0}$ and $P_{0|0}$?

$$\hat{x}_{0|0} = \begin{bmatrix} \mathsf{x}_0 & \mathsf{y}_0 & 0 & 0 \end{bmatrix}^T$$
$$P_{0|0} = \mathrm{diag}\left(r_{\mathsf{x}}, r_{\mathsf{y}}, (v_{\max}^{\mathsf{x}}/\kappa)^2, (v_{\max}^{\mathsf{y}}/\kappa)^2\right)$$

This is called **single point initiation** [Mallick (2008)] in the literature.

Gating at $k = 1$

Filter Initiation

Example-1: An alternative is

- Do the first gating with $\|y_1 - y_0\| \le T v_{max}$.
- When the gate is satisfied, form the state estimate and covariance as

$$\hat{x}_{1|1} = \begin{bmatrix} \mathsf{x}_1 & \mathsf{y}_1 & \frac{\mathsf{x}_1 - \mathsf{x}_0}{T} & \frac{\mathsf{y}_1 - \mathsf{y}_0}{T} \end{bmatrix}^T$$

$$P_{1|1} = \begin{bmatrix} r_\mathsf{x} & 0 & r_\mathsf{x}/T & 0 \\ 0 & r_\mathsf{y} & 0 & r_\mathsf{y}/T \\ r_\mathsf{x}/T & 0 & 2r_\mathsf{x}/T^2 & 0 \\ 0 & r_\mathsf{y}/T & 0 & 2r_\mathsf{y}/T^2 \end{bmatrix}$$

This method is called as **two-point difference initiation** in the literature [Mallick (2008)].

Gating at $k = 1$

Filter Initiation

Example-2: Cartesian state, bearing only measurements:

$$x_k \triangleq \begin{bmatrix} \mathsf{x}_k & \mathsf{y}_k & v_k^\mathsf{x} & v_k^\mathsf{y} \end{bmatrix}^T$$
$$y_k \triangleq \arctan(\mathsf{y}_k/\mathsf{x}_k) + e_k^\phi$$

where $e_k^\phi \sim \mathcal{N}(0, r_\phi)$.

Suppose we got y_0, what would be $\hat{x}_{0|0}$ and $P_{0|0}$?

$$\hat{x}_{0|0} = \begin{bmatrix} \text{p2c}\left([\mathsf{r}_{\text{nominal}}, y_0]\right) & 0 & 0 \end{bmatrix}^T$$
$$P_{0|0} = \text{blkdiag}\left[\nabla\,\text{p2c} \times \text{diag}\left([(\mathsf{r}_{\max} - \mathsf{r}_{\min})/\kappa]^2, r_\phi\right) \times \nabla^T\,\text{p2c}, (v_{\max}^\mathsf{x}/\kappa)^2, (v_{\max}^\mathsf{y}/\kappa)^2\right]$$

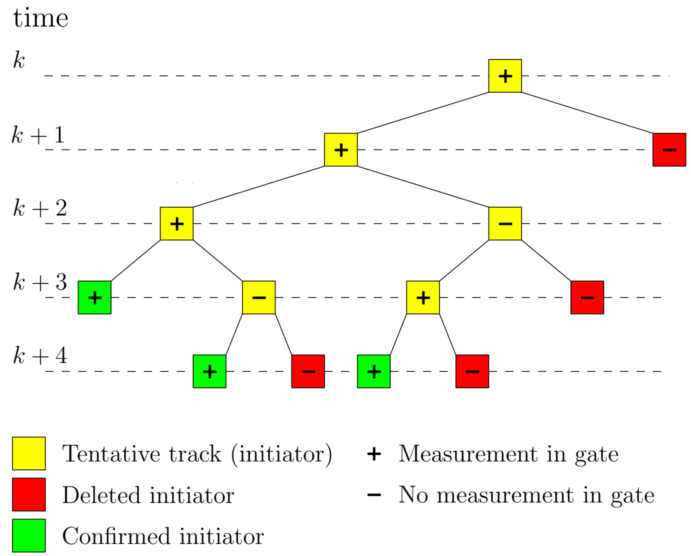
where p2c denotes the polar to Cartesian transformation.

Track Initiation & Deletion

- In general, each measurement arriving from the sensor should either
  - update a confirmed track
  - start a tentative track (initiator).
- Tentative tracks are generated to confirm or deny that a sequence of measurements comes from an actual target.
- There are two methods to confirm or delete a tentative track:
  - M/N **Logic:**
  - **Score**

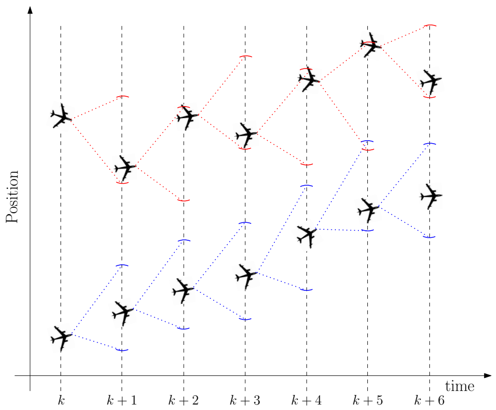
Track Initiation & Deletion: M/N logic

Example: "2/2&2/3" logic



Track Initiation & Deletion: M/N logic

- Simple counting based **low-computation** track initiation.
- Track (for not tentative but confirmed ones) deletion happens in the case of a predefined consecutive number of misses (no measurement in the gate).
- Independent of the track quality as long as gate criteria is satisfied.



These two tracks are equivalent according to M/N logic.

Track Initiation & Deletion: Score based approach (SPRT)

Suppose we have a measurement sequence $y_{0:k}$ (y_i can be ϕ empty set.). Consider two hypotheses about it.

$\mathcal{H}_0$: All measurements originate from FA

$\mathcal{H}_1$: All measurements originate from a single NT

Define the score (denoted as LPR_k meaning **log probability ratio**) of a (possibly tentative) track (or target) as

$$\mathrm{LPR}_k = \log \frac{P(\mathcal{H}_1|y_{0:k})}{P(\mathcal{H}_0|y_{0:k})}$$

Since $P(\mathcal{H}_0|y_{0:k}) = 1 - P(\mathcal{H}_1|y_{0:k})$, we have

$$P(\mathcal{H}_1|y_{0:k}) = \frac{\exp(\mathrm{LPR}_k)}{1+\exp(\mathrm{LPR}_k)}$$

Track Initiation & Deletion: Score based approach (SPRT)

How to calculate the score:

- With the initial data $y_0 \neq \phi$ (first data of an initiator)

$$\mathrm{LPR}_0 = \log \frac{p(y_0|\mathcal{H}_1)P(\mathcal{H}_1)}{p(y_0|\mathcal{H}_0)P(\mathcal{H}_0)} = \log \frac{\beta_{NT}}{\beta_{FA}} + C$$

  where β_{NT} and β_{FA} are new target and false alarms rates respectively.
  –P_D is the detection probability.
  –Setting $\mathrm{LPR}_0 = 0$ is also a common choice.

- With new measurement $y_k \neq \phi$

$$\mathrm{LPR}_k = \mathrm{LPR}_{k-1} + \log \frac{p(y_k|y_{0:k-1},\mathcal{H}_1)}{p(y_k|y_{0:k-1},\mathcal{H}_0)} = \mathrm{LPR}_{k-1} + \log \frac{P_D p_{k|k-1}(y_k)}{\beta_{FA}}$$

  where $p_{k|k-1}(y_k) \triangleq \mathcal{N}(y_k;\hat{y}_{k|k-1},S_{k|k-1})$ is the innovation (measurement prediction) likelihood.

Track Initiation & Deletion: Score based approach (SPRT)

How to calculate the score:

- With missing (no) measurement i.e., $y_k = \phi$

$$\mathrm{LPR}_k = \mathrm{LPR}_{k-1} + \log(1 - P_D P_G)$$

Making decisions based on LPR_k

- Design parameters:

  P_{FC}: The probability with which you can tolerate your tracker accept (confirm) false tracks. i.e., False track confirmation probability.
  P_{TM}: The probability with which you can tolerate your tracker delete (reject) true tracks. i.e., True track miss probability.

Track Initiation & Deletion: Score based approach (SPRT)

Making decisions based on LPR_k

- Finding thresholds for LPR_k

$$\gamma_{high} = \log \frac{1 - P_{TM}}{P_{FC}} = \log \frac{\text{Probability of accepting a true track}}{\text{Probability of accepting a false track}}$$

$$\gamma_{low} = \log \frac{P_{TM}}{1 - P_{FC}} = \log \frac{\text{Probability of rejecting a true track}}{\text{Probability of rejecting a false track}}$$

- Decision mechanism:
  - $LPR_k \geq \gamma_{high}$: Confirm the tentative track. i.e., Accept $\mathcal{H}_1$.
  - $LPR_k \leq \gamma_{low}$: Delete the tentative track. i.e., Accept $\mathcal{H}_0$.
  - $\gamma_{low} < LPR_k < \gamma_{high}$: Not enough evidence, continue testing, i.e., leave the tentative track as it is.

How to Delete Confirmed Tracks: M/N-logic case

Confirmed tracks are, most of the time, deleted because they do not get any gated measurements.

- In a scenario where one uses M/N-logic for track initiation, one can delete tracks if the track is **not updated** with a gated measurement for N_D scans.
- Another M/N logic can also be constructed also for confirmed track deletion.
- Example: 2/2&2/3 track deletion logic: Delete the confirmed track if the track is not updated for two consecutive scans and not updated in at least 2 of the three following scans.
- When KF's are not updated for several scans, the innovation covariances (determining the gate size) also become larger. A check on the gate size can also be useful in deletion.

How to Delete Confirmed Tracks: Score case

- Score calculation also takes into account the suitability of the target behavior to the model used for target state dynamics.
- If the target gets still some measurements inside the gate but the incoming measurements are quite wrongly predicted, a score based criterion can still be used for deleting confirmed tracks.
- Problem is that absolute score value becomes highly biased towards previous "good" measurements. It takes forever to delete a target that has long been tracked.
- Either use a sliding window or a fading memory

$$LPR_k = \sum_{i=k-N+1}^{k} \Delta LPR_i \qquad LPR_k = \alpha LPR_{k-1} + \Delta LPR_k$$

with a careful threshold selection (note that $\alpha < 1$).

How to Delete Confirmed Tracks: Score case

- What Blackman proposes in the book is to use the decrease from the maximum value reached. With this method:
  - Think of an hypothetical event that you would like to delete the target e.g. 5 consecutive "no measurements in the gate" events and calculate the decrease in the score that would happen

$$D_{LPR} = 5\log(1 - P_G P_D) \text{ (a negative quantity)}$$

  - Keep the maximum values reached by the scores of every track during the tracks lifetime. Call this value

$$M^j_{LPR,k} = \max_{0 \leq i \leq k-1} LPR^j_i$$

- Delete the track j if

$$LPR^j_k < M^j_{LPR,k} + D_{LPR}$$

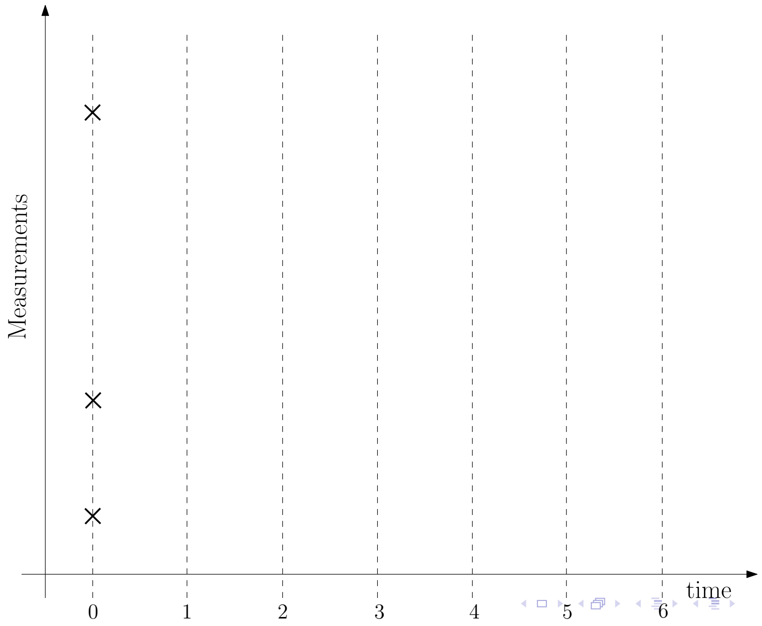
Higher level track initiation logic

- Now that we know how to process a sequence of measurements $y_{0:k}$ to decide whether it is a track or not.
- Generating such sequence of measurements is the job of a higher level track initiation logic.
- This level requires some data abstraction for the illustration of an example algorithm.
- We consider objects called **"Initiator"** each one of them corresponding to tentative tracks and keeping some level of its own history. An example can be

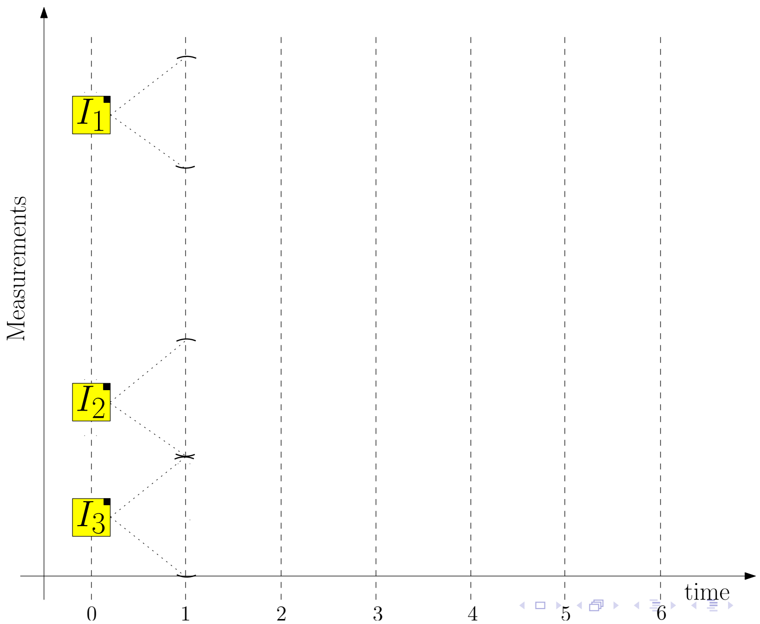
Initiator:

- Age
- State estimate
- State covariance
- Last update time
- M/N logic state
- Score

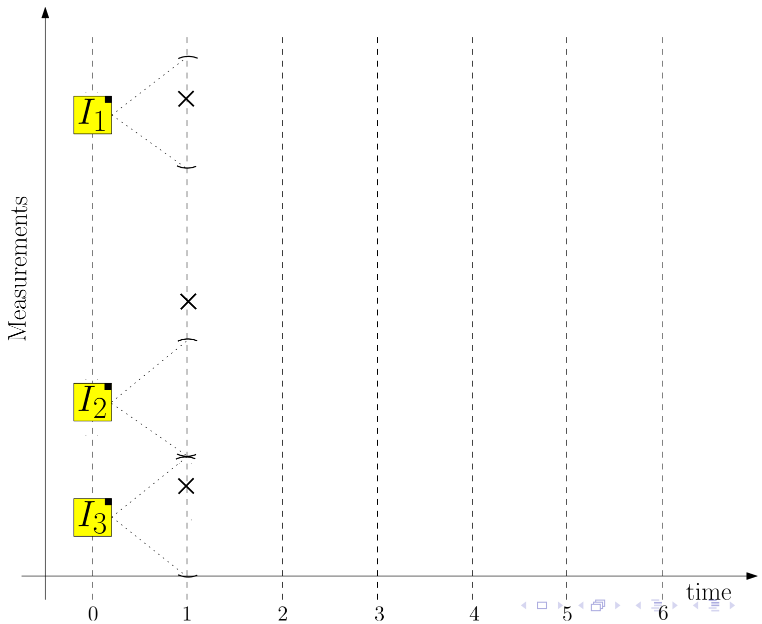
Higher level logic: Illustration with 2/2&2/3 logic



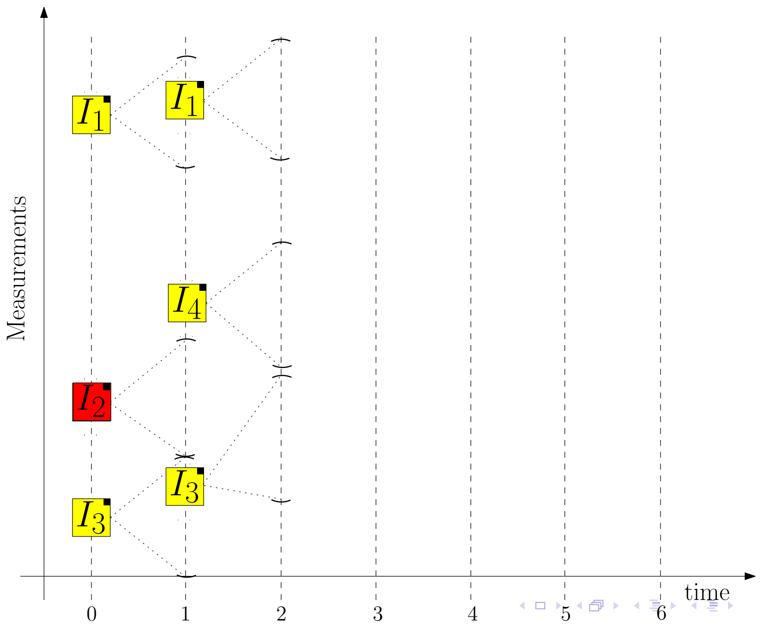
Higher level logic: Illustration with 2/2&2/3 logic



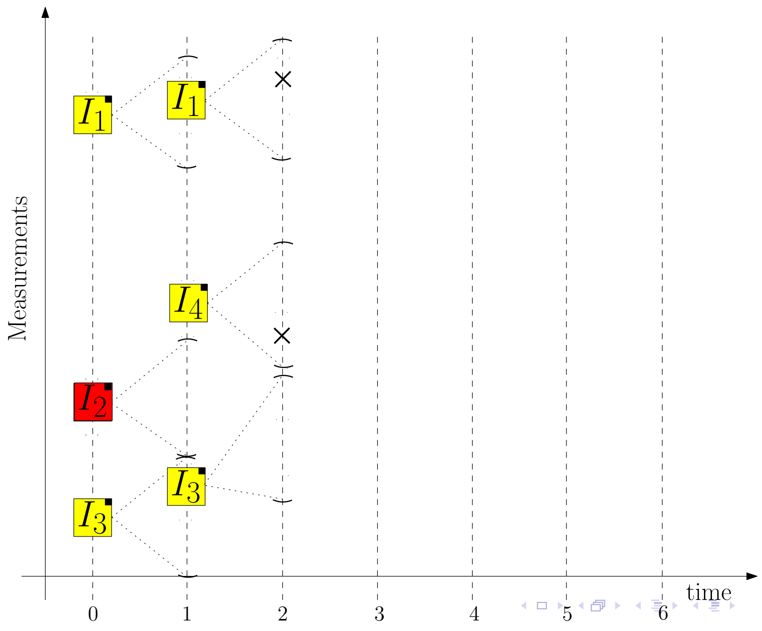
Higher level logic: Illustration with 2/2&2/3 logic



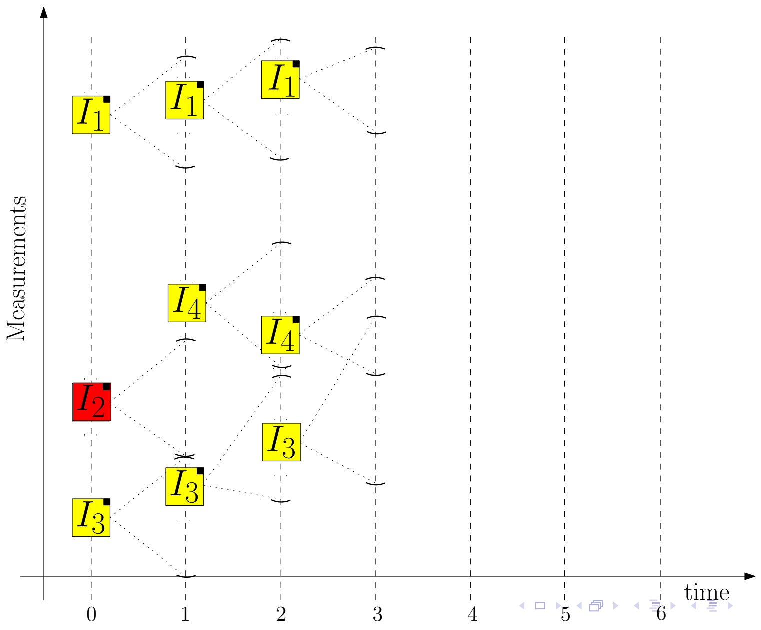
Higher level logic: Illustration with 2/2&2/3 logic



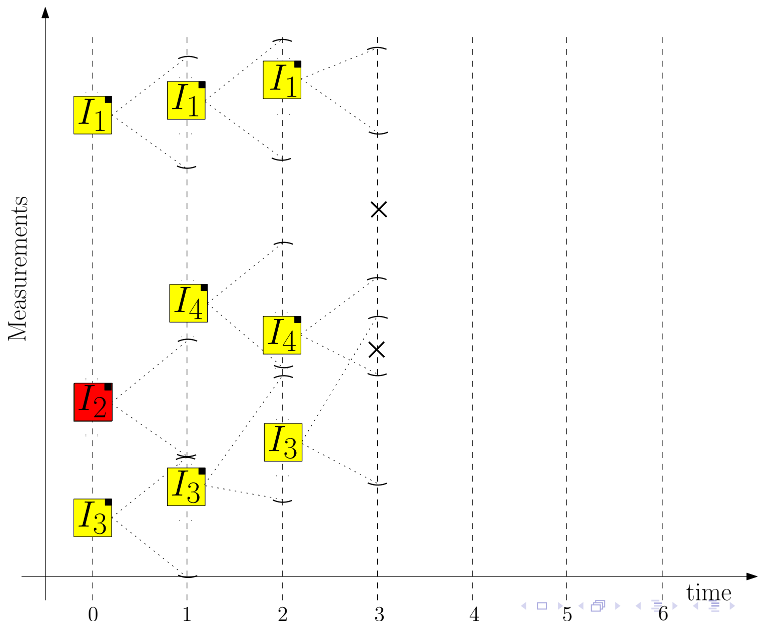
Higher level logic: Illustration with 2/2&2/3 logic



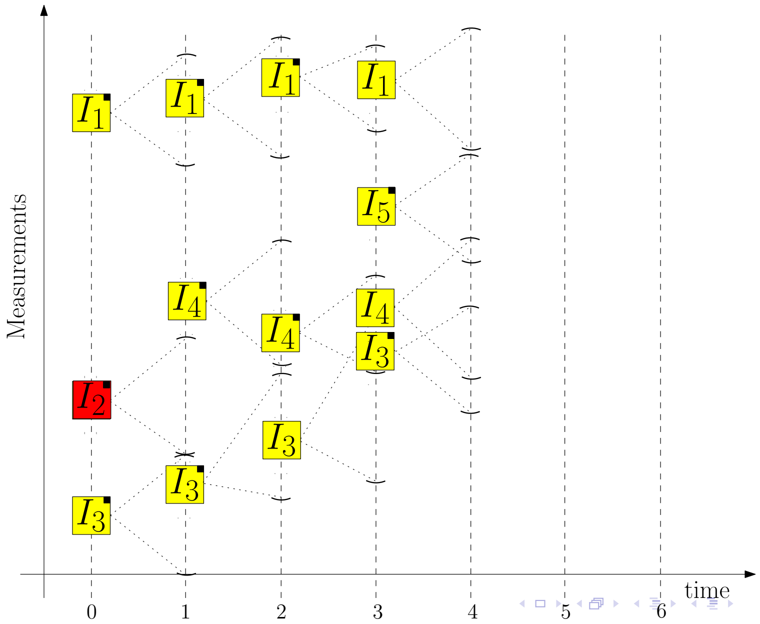
Higher level logic: Illustration with 2/2&2/3 logic



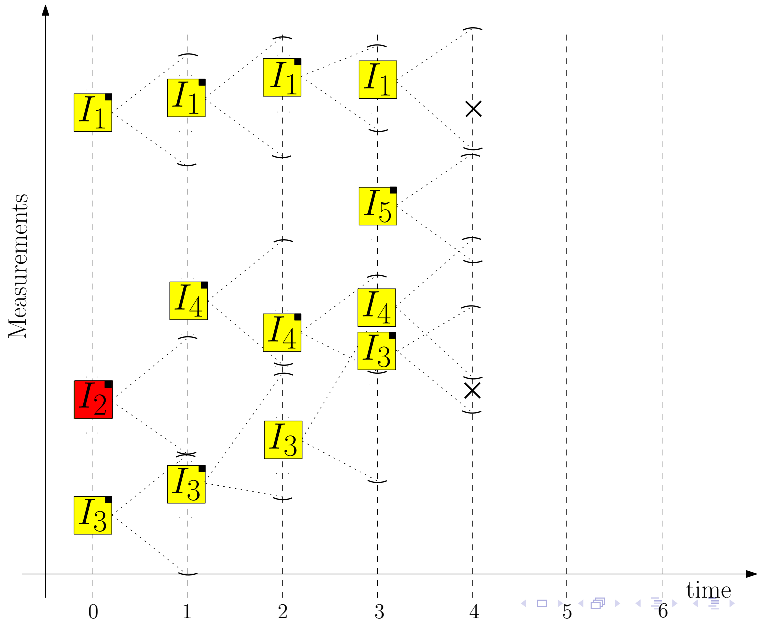
Higher level logic: Illustration with 2/2&2/3 logic



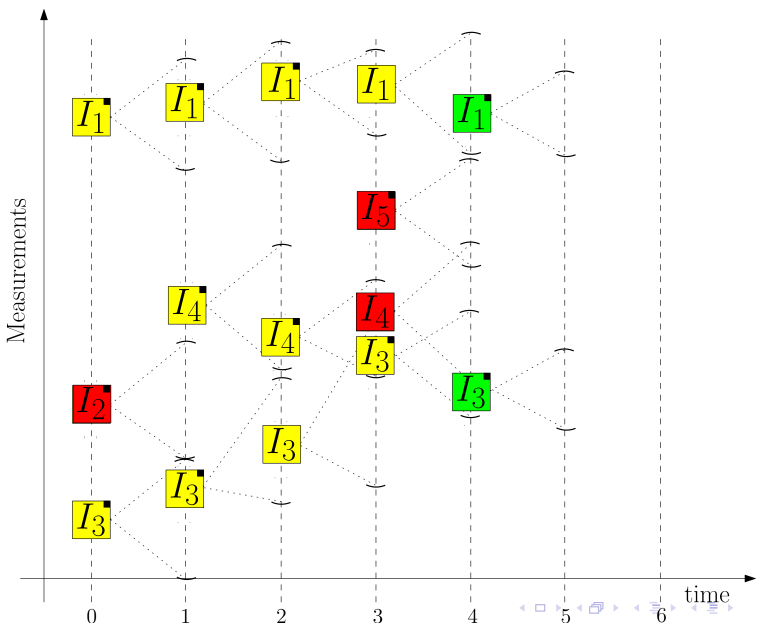
Higher level logic: Illustration with 2/2&2/3 logic



Higher level logic: Illustration with 2/2&2/3 logic



Higher level logic: Illustration with 2/2&2/3 logic



Higher level logic: Verbal Description

Higher level logic

- Time $k = 0$: Initialize tentative tracks from all measurements.
- Time $k > 0$:
  - Gate the measurements with the current initiators.
  - Make a simple association of gated measurements to the current initiators.
    - e.g., give a measurement in conflict to the oldest initiator.
    - e.g., assign a tentative track to the closest measurement in its gate.
    - Notice that with the tentative tracks, the association problem is not as critical as it is with confirmed tracks.
  - Process the current initiators with their associated measurements. Update their kinematic estimates, covariances, M/N-logic states, or scores etc..
  - Start new initiators from un-used measurements.
  - Check updated M/N-logic states or scores and confirm, delete those initiators (tentative tracks) satisfying the related criteria.

Higher level logic: Implementation Advice

In the implementation of exercises, you can use whatever type of programming you like. The following are general advice if you want some.

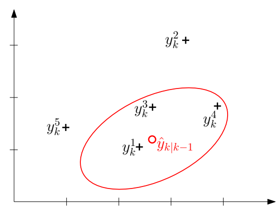
- Try to be as object oriented as possible.
- Define an initiator (or tentative) object (struct or class).
- e.g., Write the following functions
  - `Initator=Initializer(InitiatingMeasurement)` (constructor)
  - `GateDecisions=Gating(InitiatorArray,MeasurementSet)`
  - `AssociationDecisions=Associate(InitiatorArray...`
    `...,MeasurementSet,GateDecisions)`
  - `UpdatedInitiatorArray=Update(InitiatorArray...`
    `...,MeasurementSet,AssociationDecisions)`
  - `flags=CheckConfirm(InitiatorArray)`
  - `flags=CheckDelete(InitiatorArray)`

Single Target Data Association: NN

Nearest neighbor association rule: Choose the closest measurement to the measurement prediction in the gate in the sense that

$$i^* = \arg \min_{1 \le i \le m_k^G} (y_k^i - \hat{y}_{k|k-1})^T S_{k|k-1}^{-1} (y_k^i - \hat{y}_{k|k-1})$$

where m_k^G is the number of measurement in the gate.



- The target is updated with the selected measurement and the measurement is not further processed by either initiators or other targets (if any).
- The measurements outside the gate are certainly sent to higher level initiator logic to be processed.

Single Target Data Association: NN

Other measurements in the gate: If there are other measurements in the gate than the one selected by the NN rule, their transfer to the initiation logic is application dependent.

- If it is known that there might be closely spaced targets, and the sensor reports are accurate, then they can be sent to the higher initiation logic to be processed further i.e., to be started with tentative tracks.
- If the sensor reports are noisy, or the targets are extended with respect to sensor resolution, sometimes people choose to forget about them forever. An example is the rotor blades of the helicopter in the following video.
  `http://www.youtube.com/watch?v=iUx1msxDyw8`

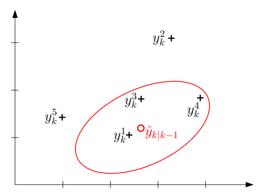
Single Target Data Association: PDA

Probabilistic data association: NN is a hard decision mechanism. Soft version of it is to not make a hard decision but use all of the measurements in the gate to the extent that they suit the prediction. Measurements in the gate are shown as $Y_k = \{y_k^i\}_{i=1}^{m_k}$. We have the following hypotheses about these measurements

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

for $i = 1, \ldots, m_k$. Then, the estimated density $p(x_k|Y_{0:k})$ can be calculated using total probability theorem as

$$p(x_k|Y_{0:k}) = \sum_{i=0}^{m_k} p(x_k|\theta_i, Y_{0:k}) \underbrace{p(\theta_i|Y_{0:k})}_{\triangleq \mu_k^i}$$



Single Target Data Association: PDA

$$p(x_k|\theta_i, Y_{0:k}) = \begin{cases} p(x_k|Y_{0:k-1}), & i = 0 \\ p(x_k|Y_{0:k-1}, y_k^i), & \text{otherwise} \end{cases}$$

In the special case of a Kalman filter

$$\hat{x}_{k|k}^i = \begin{cases} \hat{x}_{k|k-1}, & i = 0 \\ \hat{x}_{k|k-1} + K_k(y_k^i - \hat{y}_{k|k-1}), & \text{otherwise} \end{cases}$$

$$\Sigma_{k|k}^i = \begin{cases} \Sigma_{k|k-1}, & i = 0 \\ \Sigma_{k|k-1} - K_k S_{k|k-1} K_k^T, & \text{otherwise} \end{cases}$$

Note that the quantities $\Sigma_{k|k}^i$ and K_k, are the same for $i = 1, \ldots, m_k$.

Single Target Data Association: PDA

In the KF case, the overall state estimate $\hat{x}_{k|k}$ can be calculated as

$$\hat{x}_{k|k} = \sum_{i=0}^{m_k} \mu_k^i \hat{x}_{k|k}^i = \hat{x}_{k|k-1} + K_k(y_k^{eq} - \hat{y}_{k|k-1})$$

where

$$y_k^{eq} = \mu_k^0 \hat{y}_{k|k-1} + \sum_{i=1}^{m_k} \mu_k^i y_k^i$$

is the **equivalent** measurement.
The covariance $\Sigma_{k|k}$ corresponding to $\hat{x}_{k|k}$ is given by

$$\Sigma_{k|k} = \sum_{i=0}^{m_k} \mu_k^i \left[\Sigma_{k|k}^i + (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})^T\right]$$

$$= \sum_{i=0}^{m_k} \mu_k^i \Sigma_{k|k}^i + \underbrace{\sum_{i=0}^{m_k} \mu_k^i (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})^T}_{\text{spread of the means}}$$

Calculation of the probabilities μ_k^i

- First calculate

$$\tilde{\mu}_k^i = \begin{cases} (1 - P_D P_G)\beta_{FA}, & i = 0 \\ P_D p_{k|k-1}(y_k^i), & \text{otherwise} \end{cases}$$

  where $p_{k|k-1}(y_k) \triangleq \mathcal{N}(y_k; \hat{y}_{k|k-1}, S_{k|k-1})$ is the innovation (measurement prediction) likelihood.
- Then normalize $\tilde{\mu}_k^i$ to obtain μ_k^i.

$$\mu_k^i = \frac{\tilde{\mu}_k^i}{\sum_{i=0}^{m_k} \tilde{\mu}_k^i}$$

References

- Y. Bar-Shalom, "Dimensionless score function for multiple hypothesis tracking" *IEEE Transactions on Aerospace and Electronic Systems*, vol.43, no.1, pp.392–400, Jan. 2007.
- S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Norwood, MA: Artech House, 1999.
- Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles, Techniques.* Storrs, CT: YBS Publishing, 1995.
- M. Mallick, B. La Scala, "Comparison of single-point and two-point difference track initiation algorithms using position measurements", *Acta Automatica Sinica*, vol.34, no.3, pp.258–265, Mar. 2008.