

Target Tracking

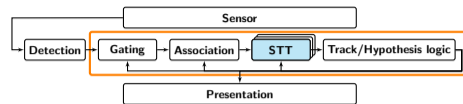
Le 3: Single Target Tracking

Gustaf Hendeby

Div. Automatic Control
Dept. Electrical Engineering
gustaf.hendeby@liu.se

- 1 Gating
- 2 Track logic
- 3 Data Association
- 4 Summary
- 5 Exercises

Summary: lecture 2



- Common tracking sensors: range, bearing, range-rate (Doppler shift),
Models are derived from physical relations, and assumptions about noise levels.
- Common motion target models: constant velocity, constant acceleration, coordinated turn
Coarse approximations to allow for reasonable target maneuvers, are derived from basic physical relations
- Mixtures are a common tool to cover several different possible behaviors
Maneuvering targets are commonly tracked using IMM filters, which approximate the complete filter bank solution

References on Single Target Tracking Topics

- S. K. Singh, M. Premalatha, and G. Nair. **Ellipsoidal gating for an airborne track while scan radar.** In *Proceedings of the International Radar Conference*, pages 334–339, Alexandria, VA, May 8–11 1995. (Gating)
- Y. Bar-Shalom, S. S. Blackman, and R. J. Fitzgerald. **Dimensionless score function for multiple hypothesis tracking.** *IEEE Transactions on Aerospace and Electronic Systems*, 43(1):392–400, Jan. 2007. (Track scores)
- Y. Bar-Shalom, F. Daum, and J. Huang. **The probabilistic data association filter.** *IEEE Control Systems Magazine*, 29(6):82–100, Nov. 2009. (PDA)

Filter Banks Revisited

Examples of algorithms

- *Generalized pseudo Bayesian* of depth n (GPB(n)) filter
- *Interacting multiple models* (IMM) filter
- *Range parameterized EKF* (RPEKF)

We will briefly recapitulate the idea, and exemplify with a bearings-only application.

Filter Banks Revisited

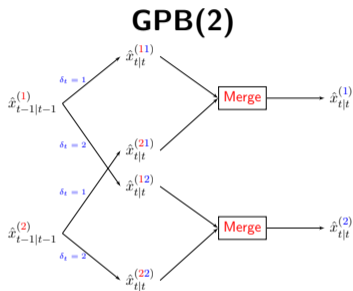
Examples of algorithms

- *Generalized pseudo Bayesian* of depth n (GPB(n)) filter
- *Interacting multiple models* (IMM) filter
- *Range parameterized EKF* (RPEKF)

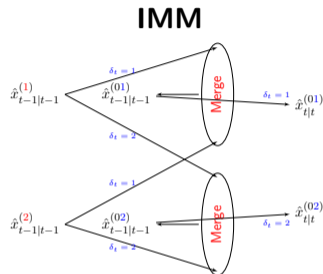
We will briefly recapitulate the idea, and exemplify with a bearings-only application.

(And I will try to clarify the relation between the GPB(2) and IMM.)

Clarifications: GBP(2) vs IMM



1. Fully expand the modes.
2. Merge, marginalize on oldest split



1. Compute mixing weights for merging after prediction, merge ancestors.
2. Predict and measurement update new nodes.

Clarifications: GPB(2) vs IMM

GPB(2)

$$\hat{x}_{t-1|t-1}^{(1)}$$

$$\hat{x}_{t-1|t-1}^{(2)}$$

1. Fully expand the modes.
2. Merge, marginalize on oldest split

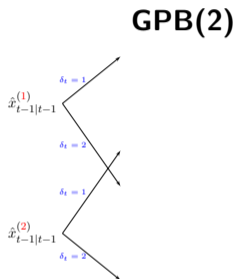
IMM

$$\hat{x}_{t-1|t-1}^{(1)}$$

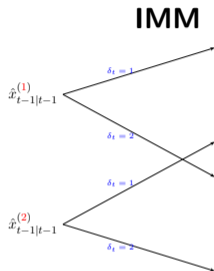
$$\hat{x}_{t-1|t-1}^{(2)}$$

1. Compute mixing weights for merging after prediction, merge ancestors.
2. Predict and measurement update new nodes.

Clarifications: GBP(2) vs IMM

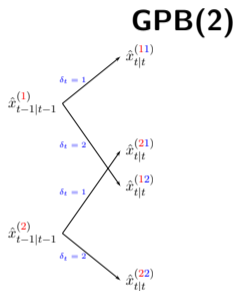


1. Fully expand the modes.
2. Merge, marginalize on oldest split

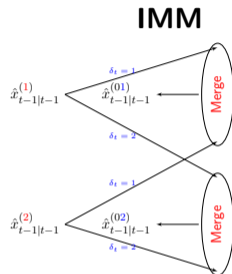


1. Compute mixing weights for merging after prediction, merge ancestors.
2. Predict and measurement update new nodes.

Clarifications: GBP(2) vs IMM

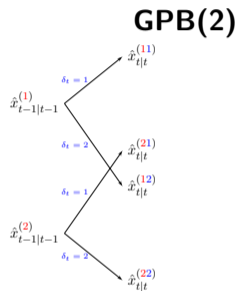


1. Fully expand the modes.
2. Merge, marginalize on oldest split

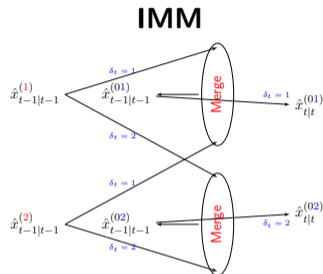


1. Compute mixing weights for merging after prediction, merge ancestors.
2. Predict and measurement update new nodes.

Clarifications: GBP(2) vs IMM

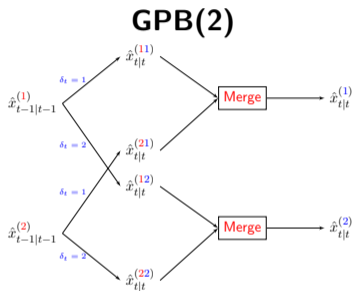


1. Fully expand the modes.
2. Merge, marginalize on oldest split

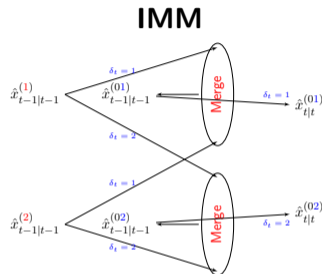


1. Compute mixing weights for merging after prediction, merge ancestors.
2. Predict and measurement update new nodes.

Clarifications: GBP(2) vs IMM



1. Fully expand the modes.
2. Merge, marginalize on oldest split

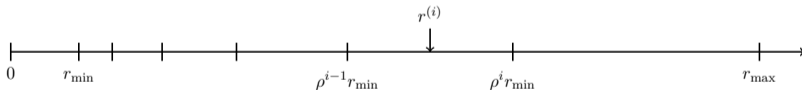


1. Compute mixing weights for merging after prediction, merge ancestors.
2. Predict and measurement update new nodes.

Range Parameterized EKF (RPEKF)

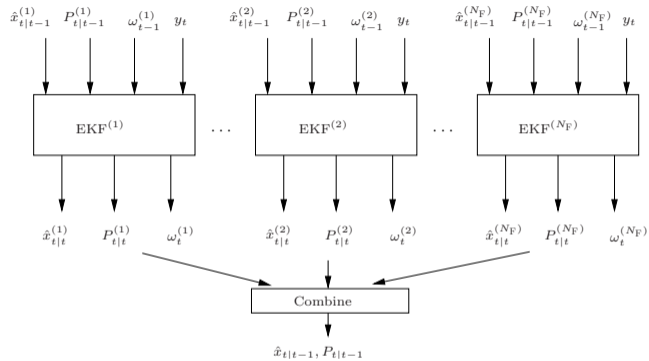
This is a filter bank problem, without mode transitions.

- Tracking with a bearing only sensor is difficult, as the range and hence the relative Cartesian position is unavailable.
- This is solved by out-maneuvering the target; however, how should the target be handled until enough information is gathered?
- Use filter bank to represent different possible range options, $r^{(i)}$, and let time determine the actual distance.

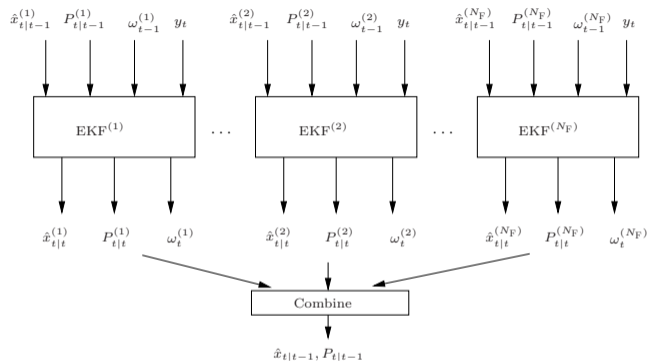


This is a **complete** filter bank, with a structure that limits the growth.

Range Parameterized EKF: illustration



Range Parameterized EKF: illustration



Practical concerns

- Filter pruning (divergence monitoring, range interval check etc)
- Re-start

Range Parameterized EKF: process noise tuning

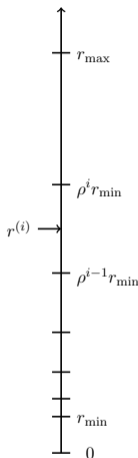
For an EKF the performance depends on the *coefficient of variation*, C_R . To have comparable performance for each filter, the same value should be used on each interval. Approx, $\sigma^{(i)}/r^{(i)}$, $i = 1, \dots, N$, where $r^{(i)}$ and $\sigma^{(i)}$ are the range and standard deviation for the different filters.

$$r^{(i)} = \frac{r_{\min}}{2}(\rho^i + \rho^{i-1})$$

$$\rho = \left(\frac{r_{\max}}{r_{\min}}\right)^{1/N}$$

$$C_R = \frac{\sigma^{(i)}}{r^{(i)}} = \frac{2(\rho - 1)}{\sqrt{12}(\rho + 1)}$$

Therefore, the variance for each interval is given as $\sigma^{(i)} = r^{(i)}C_R$.



Range Parameterized EKF: filter probabilities

The RPEKF uses the likelihood from each EKF, to recursively update its probability according to

$$\omega_t^{(i)} = p(y_t | x_t^{(i)}, i) \omega_{t-1}^{(i)}.$$

The prior distribution is assumed uniform, *i.e.* $\omega_0^{(i)} = \frac{1}{N}$, $i = 1, \dots, N$. However, if other information is available it could be used to enhance the performance.

Under a Gaussian assumption, the likelihood is given from the EKF as

$$p(y_t | x_t^{(i)}, i) \propto \frac{1}{\sqrt{\det(S_t^{(i)})}} \exp\left(-\frac{1}{2}(\epsilon_t^{(i)})^T (S_t^{(i)})^{-1} \epsilon_t^{(i)}\right)$$

$$S_t^{(i)} = H_t^{(i)} P_{t|t-1}^{(i)} (H_t^{(i)})^T + R_t$$

$$\epsilon_t^{(i)} = y_t - h(\hat{x}_{t|t-1}^{(i)})$$

$$(H_t^{(i)})^T = \nabla_x h^T(x) \Big|_{x=\hat{x}_{t|t-1}^{(i)}}$$

Range Parameterized EKF: filter update

The measurement update for each filter is given by the Kalman filter equations.

$$\begin{aligned}\hat{x}_{t|t}^{(i)} &= \hat{x}_{t|t-1}^{(i)} + K_t^{(i)} \epsilon_t^{(i)} \\ P_{t|t}^{(i)} &= P_{t|t-1}^{(i)} - K_t^{(i)} S_t^{(i)} (K_t^{(i)})^T \\ K_t^{(i)} &= P_{t|t-1}^{(i)} (H_t^{(i)})^T (S_t^{(i)})^{-1}\end{aligned}$$

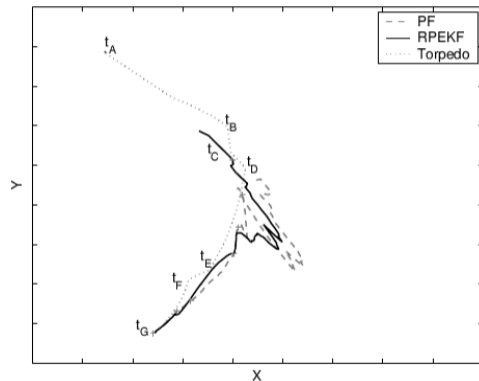
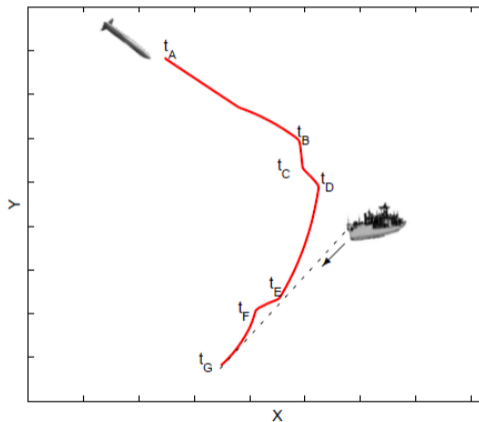
The combined estimate and covariance can now be expressed as:

$$\begin{aligned}\hat{x}_{t|t} &= \sum_i \omega_t^{(i)} \hat{x}_{t|t}^{(i)} \\ P_{t|t} &= \sum_i \omega_t^{(i)} \left(P_{t|t}^{(i)} + (\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})(\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})^T \right)\end{aligned}$$

where $P_{t|t}^{(i)}$ is the covariance and $\hat{x}_{t|t}^{(i)}$ the estimate for different filters.

Range Parameterized EKF and PF: torpedo example (1/3)

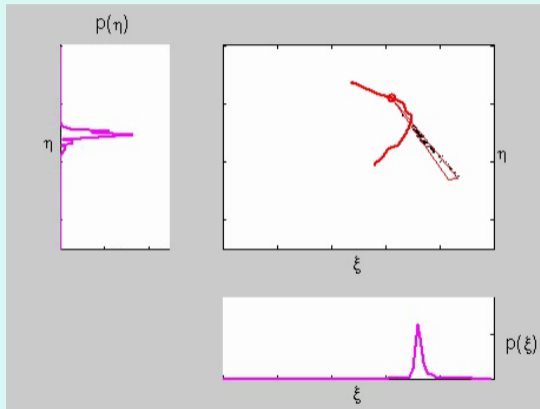
Passive sonar measurements (bearings-only) require maneuver for range observability.



Passive sonar data from SBUS Motala

Range Parameterized EKF and PF: torpedo example (2/3)

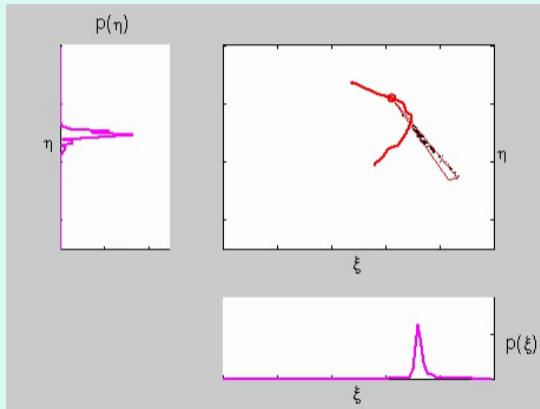
Bearings-only



- True trajectory not known (but constant course)
- PF
- RPEKF

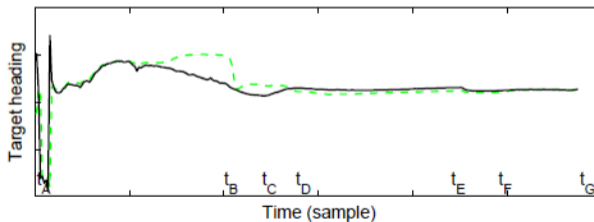
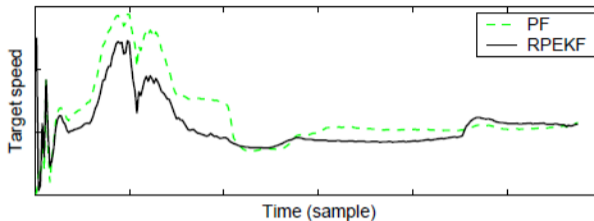
Range Parameterized EKF and PF: torpedo example (2/3)

Bearings-only

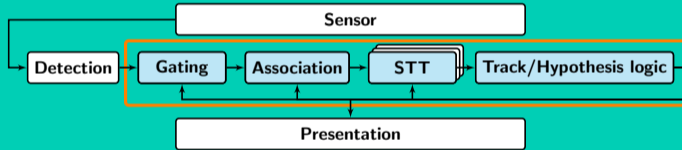


- True trajectory not known (but constant course)
- PF
- RPEKF

Range Parameterized EKF and PF: torpedo example (3/3)



Single Target Tracking



Single Target Tracking: a more complete story

So far (lecture 1–2)

- Exactly one target
- Exactly one true measurement per target in each time
- No clutter and/or outliers

Single Target Tracking: a more complete story

So far (lecture 1–2)

- Exactly one target
- Exactly one true measurement per target in each time
- No clutter and/or outliers

Additional complications

- At most one target (none or one)
- A measurement from the target with probability P_D
- Presence of clutter and/or outliers

Single Target Tracking: a more complete story

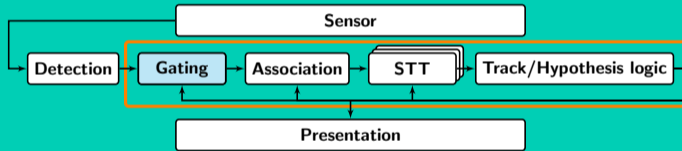
So far (lecture 1–2)

- Exactly one target
- Exactly one true measurement per target in each time
- No clutter and/or outliers

Additional complications

- At most one target (none or one)
⇒ **Target existence must be determined!**
- A measurement from the target with probability P_D
⇒ **The case of no actual measurement must be considered!**
- Presence of clutter and/or outliers
⇒ **False alarms must be detected and rejected!**

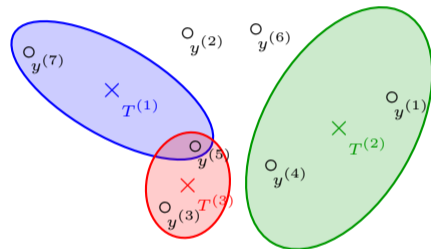
Gating



Gating

The purpose of gating is to remove measurements that are very unlikely to originate from a given target, that is:

- Reduce problem complexity, by minimizing the number of possible measurements to each target.
- Not remove (with low probability) valid measurements.
- Be a cheap operation
 - Rectangular gating.
 - Elliptical gating.



- *Cross: predicted measurement.*
- *Dot: obtained observation.*

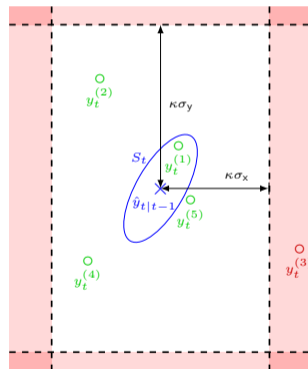
Gating: rectangular gating

Measurements are gated away when they are too far away from the expected measurement, given some distance measure often depending on the measurement and target uncertainty.

Rectangular (∞ -norm) gating

$$|y_t^x - \hat{y}_{t|t-1}^x| < \kappa \sigma_{t|t-1}^x, \quad |y_t^y - \hat{y}_{t|t-1}^y| < \kappa \sigma_{t|t-1}^y$$

where usually $\kappa \geq 3$, *i.e.*, the threshold is 3 standard deviations in each dimension.

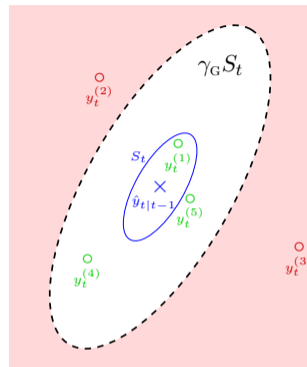


Gating: elliptical gating

Elliptical (Mahalanobis-norm) gating

$$(y_t - \hat{y}_{t|t-1})^T S_t^{-1} (y_t - \hat{y}_{t|t-1}) < \gamma_G$$

where γ_G is the gate threshold.

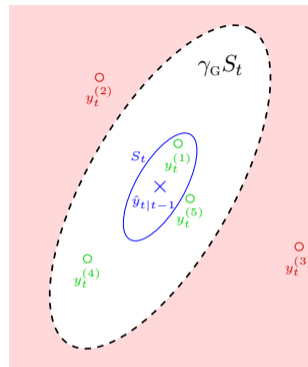


Gating: elliptical gating

Elliptical (Mahalanobis-norm) gating

$$(y_t - \hat{y}_{t|t-1})^T S_t^{-1} (y_t - \hat{y}_{t|t-1}) < \gamma_G$$

where γ_G is the gate threshold.



Reminder: χ^2 distribution

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

Elliptical Gating: motivation (1/2)

Assume a linear Gaussian state-space system, and using a correct Kalman filter, then

$$\epsilon_t = y_t - \hat{y}_{t|t-1} \sim \mathcal{N}(0_{n_y}, S_{t|t-1}).$$

$S_{t|t-1} \succeq 0$, hence a factorization $S_{t|t-1} = U_t^T U_t$ exist, such that

$$\epsilon_t^T S_{t|t-1}^{-1} \epsilon_t = \epsilon_t^T U_t^{-T} U_t^{-1} \epsilon_t = \|U_t^{-1} \epsilon_t\|_2^2 = \|\tilde{\epsilon}_t\|_2^2$$

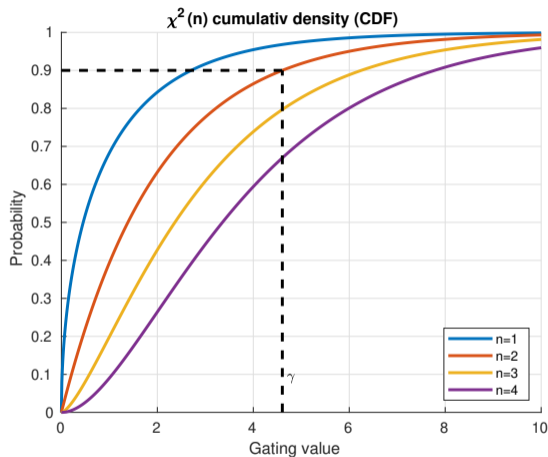
where

$$\begin{aligned}\tilde{\epsilon}_t &= U_t^{-1} \epsilon_t \sim \mathcal{N}(0_{n_y}, I_{n_y}) \\ \gamma &= \|\tilde{\epsilon}_t\|_2^2 = \sum_i \tilde{\epsilon}_{i,k}^2 \sim \chi^2(n_y).\end{aligned}$$

The probability to accept correct measurement in the gate becomes

$$P_G = \Pr(\gamma \leq \gamma_g) = \int_0^{\gamma_g} \chi^2(\gamma; n_y) d\gamma.$$

Elliptical Gating: motivation (2/2)



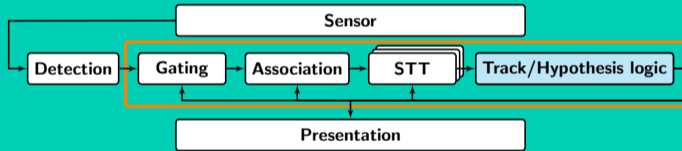
Elliptical gate threshold selection

$\text{gamma}_G = \text{chi2inv}(P_G, n_y)$ computes the threshold γ_G that gives the desired P_G .

Note: for 2-DoF ($n_y = 2$) we have

- $P_G = 0.9 \Rightarrow \gamma_G \approx 4.7$.
- $P_G = 0.99 \Rightarrow \gamma_G \approx 9.2$.

Track Logic



Track Logic

We will still consider a STT, but need some track logic.

Measurements and tracks (in a scan)

- *New target*: a measurement can originate from a new target.
- *Continuation*: a measurement can originate from an existing track.
- *False*: A measurement might be clutter.
- *Missing measurement*: There might not be a measurement corresponding to a track.

We will consider some methods for STT, (only one true target), but these can be generalized later on for MTT.

Track Logic

Does a target exist or not?

Given that measurements can either come from a target or be false observations, a method to determine if a target is present or not is required.

Track status:

- *Tentative*: Possibly a track, not fully decided yet.
- *Confirmed*: Considered being a track.
- *Deleted/Dead*: A track no longer present.

Track logics:

- *N/M logic*: Fast, but a bit *ad hoc*.
- *Track score based logic*: Slower, essentially a hypothesis test.

N/M Logic

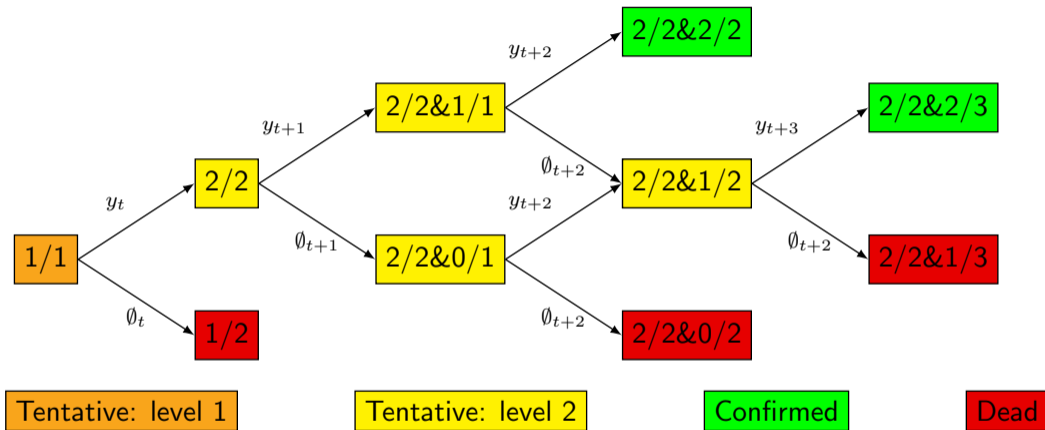
- Based on if a measurement gets gated with the measurement or not.
- A tentative track is initiated when there is no associated track.
- $2/2$ & $2/3$: A tentative track is confirmed when measurements have been associated with the track often enough (N/M). *E.g.*, often 2 times of the last 2 ($2/2$) followed by 2 of the following 3 ($2/3$).
- A track is deleted if it fails to get enough gated measurements (N/M). *E.g.*, missing 2 measurements the last 3 times ($2/3$).

N/M Logic

- Based on if a measurement gets gated with the measurement or not.
- A tentative track is initiated when there is no associated track.
- $2/2\&2/3$: A tentative track is confirmed when measurements have been associated with the track often enough (N/M). *E.g.*, often 2 times of the last 2 ($2/2$) followed by 2 of the following 3 ($2/3$).
- A track is deleted if it fails to get enough gated measurements (N/M). *E.g.*, missing 2 measurements the last 3 times ($2/3$).

Note

N/M logic only considers if any measurement is gated to the track or not. The quality of the track and association is disregarded. A practical trick is to delete a track when its uncertainty becomes too large.

N/M logic: example $"2/2&2/3"$ logic

Track Score

Track Score

The track score L_t is the log probability ratio that the obtained measurements are all false alarms, versus that they all come from a single target. A high value indicates the target exists.

Consider the two hypotheses:

\mathcal{H}_0 : \mathbb{Y}_t all originate from FA

\mathcal{H}_1 : \mathbb{Y}_t originate from a single target

If no measurement is obtained at time i , $y_i = \emptyset$.

The track score is the matching log probability ratio

$$L_t = \log \frac{\Pr(\mathcal{H}_1 | \mathbb{Y}_t)}{\Pr(\mathcal{H}_0 | \mathbb{Y}_t)}.$$

Track Score: properties

From this we get

$$\Pr(\mathcal{H}_1|\mathbb{Y}_t) = \alpha$$

$$\Pr(\mathcal{H}_0|\mathbb{Y}_t) = 1 - \Pr(\mathcal{H}_1|\mathbb{Y}_t) = 1 - \alpha$$

Hence

$$e^L = \frac{\alpha}{1 - \alpha} \Rightarrow \alpha = \frac{e^L}{1 + e^L}$$

- The probabilities of a track can be obtained from the track score

$$\Pr(\mathcal{H}_1|\mathbb{Y}_t) = \frac{e^{L_t}}{1 + e^{L_t}}.$$

- The track score can be updated recursively as

$$L_t = L_{t-1} + l_t,$$

where l_t is the contribution at time t to be derived later.

Track Score: initialization

When starting up a new (tentative) track from an observation y_0

$$L_0 = \log \frac{p(y_0|\mathcal{H}_1)P(\mathcal{H}_1)}{p(y_0|\mathcal{H}_0)P(\mathcal{H}_0)} = \log \frac{\beta_{NT}}{\beta_{FA}} + C$$

where β_{NT} and β_{FA} are new target and false alarms rates respectively. C effectively becomes tuning parameter combining the prior of targets and false alarms.

Setting $L_0 = 0$ is also a common choice in practice.

Track Score: score update

Observation: $y_t \neq \emptyset$

$$L_t = L_{t-1} + \log \frac{p(y_t | \mathbb{Y}_{t-1}, \mathcal{H}_1)}{p(y_t | \mathbb{Y}_{t-1}, \mathcal{H}_0)} = L_{t-1} + \underbrace{\log \frac{P_D P_G p(y_t | \mathbb{Y}_{t-1})}{\beta_{\text{FA}}}}_{l_t},$$

where $p(y_t | \mathbb{Y}_{t-1}) = \mathcal{N}(y_t; \hat{y}_{t|t-1}, S_{t|t-1})$ is the KF innovation likelihood.

No observation: $y_t = \emptyset$

$$L_t = L_{t-1} + \underbrace{\log(1 - P_D P_G)}_{l_t},$$

where $P_D P_G$ is the probability the target is detected and pass the gate.

Note: Usually the track gate is such that one can assume $P_G \approx 1$.

Track Score Logic (1/2)

Handling of tentative tracks

- $L_t \geq L_{\text{high}}$: Confirm the tentative track, *i.e.*, decide \mathcal{H}_1 .
- $L_t \leq L_{\text{low}}$: Delete the tentative track, *i.e.*, decide \mathcal{H}_0 .
- $L_{\text{low}} < L_t < L_{\text{high}}$: The track remains tentative, *i.e.*, no decision possible yet.

Track Score Logic (2/2)

Handling of confirmed tracks

- Delete if $L_t \leq L_{\text{del}}$, but L_t tends to grow over time as old goodness accumulates (cf. “integrator windup”)
- Several solutions exist:
 - Use a sliding window of length N , *i.e.*, $L_t = \sum_{\tau=t-N}^t l_\tau$
 - Use a forgetting factor, *i.e.* $L_t = \lambda L_{t-1} + l_t$, with $0 < \lambda < 1$
 - Cap the L_t to a reasonable value L_{max} , *i.e.*,
$$L_t = \min(L_t + l_t, L_{\text{max}})$$
 - Limit how much a track may drop from its maximum, *i.e.*,
$$L'_{\text{del}} = \max_{\tau} (L_{\tau}) - L_{\text{del}}$$

Track Score Logic: tuning

- Design parameters:

P_{FC} The probability of confirming false tracks

P_{TM} The probability of rejecting (deleting) true tracks

- Thresholds

$$L_{\text{high}} = \log \frac{1 - P_{TM}}{P_{FC}} = \log \frac{\text{Pr}(\text{Accept a true track})}{\text{Pr}(\text{Accept a false track})}$$

$$L_{\text{low}} = \log \frac{P_{TM}}{1 - P_{FC}} = \log \frac{\text{Pr}(\text{Reject a true track})}{\text{Pr}(\text{Reject a false track})}$$

Track Management: an overview

When a scan arrives

1. Gate incoming measurements with current tracks.
2. Associate the gated measurements with the current tracks.
3. Apply the track logic to the track–measurement combinations, and update the status accordingly.
4. Update the track filters with the measurements.
5. Create new tentative tracks from the unused measurements (a measurement is considered used even if it happens to be associated with a track that died).

Track Management: initialization (1/2)

To initialize a tentative track; the initial state, $\hat{x}_{0|0}$ and covariance matrix, $P_{0|0}$, must be obtained from the measurement, y_0 :

One-point initialization

- y_0 is informative enough to obtain $(\hat{x}_{0|0}, P_{0|0}) = h^{-1}(y_0, S_0)$.
- y_0 is not informative enough, but nominal values can be used to fill in the blanks.

E.g., the target position is obtained from the measurement and the velocity is assumed to be 0 with large uncertainty.

Track Management: initialization (2/2)

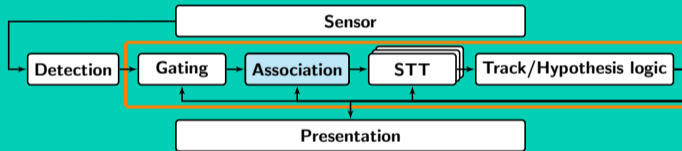
To initialize a tentative track; the initial state, $\hat{x}_{0|0}$ and covariance matrix, $P_{0|0}$, must be obtained from the measurement, y_0 :

Two-point initialization

Two measurements are collected before the track is properly initialized. The first gate should then be based on how much the measurement can reasonably move between two samples.

E.g., the measurement is position, and the gate is then equal to how much the target can reasonably move between measurements. Based on two consecutive measurements, position and velocity can be computed to initialize the state.

Data Association



Data Association

STT: We need to associate measurement to a track:

- **Hard decision**

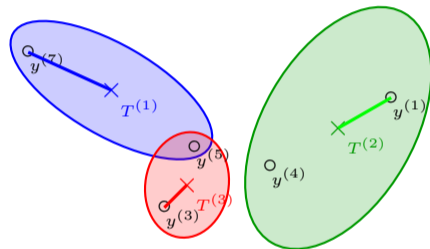
An explicit decision is made how to match tracks and measurements, and only those associations are considered further. (Sometimes several possibilities are considered in parallel though.)

E.g., nearest neighbor (NN) association.

- **Soft decision**

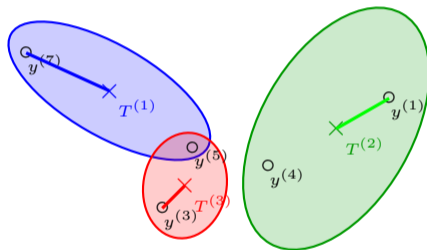
Instead of a matching a single measurement to a track, several measurements are associated to the track and all contribute to the end result.

E.g., probabilistic data association (PDA).



Nearest Neighbor (NN) Association

Pick the measurement is closes to the predicted measurement. Either Euclidian distance or probabilistic, the gating calculations can be reused.



NN is a hard decision mechanism for which measurement to use for a track.

Probabilistic Data Association (PDA)

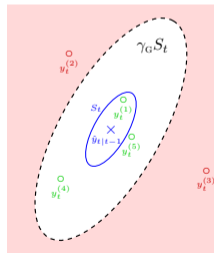
A soft association alternative is to use all of the measurements in the gate, weighted with how well fit the prediction. Measurements in the gate are shown as $Y_t = \{y_t^{(i)}\}_{i=1}^{m_t}$. We have the following hypotheses about these measurements

$$\mathcal{H}^{(0)} = \{\text{All of } Y_t \text{ is FA, i.e., no target originated measurement in the gate.}\}$$

$$\mathcal{H}^{(i)} = \{\text{Measurement } y_t^{(i)} \text{ belongs to target, all the rest are FA.}\}$$

for $i = 1, \dots, m_t$. Then, the estimated density $p(x_t | \mathbb{Y}_t)$ can be calculated using total probability theorem as

$$p(x_t | \mathbb{Y}_t) = \sum_{i=0}^{m_t} p(x_t | \mathcal{H}^{(i)}, \mathbb{Y}_t) \underbrace{p(\mathcal{H}^{(i)} | \mathbb{Y}_t)}_{\omega_t^{(i)}}$$



Probabilistic Data Association: KF algorithm (1/2)

$$p(x_t | \mathcal{H}^{(i)}, \mathbb{Y}_t) = \begin{cases} p(x_t | \mathbb{Y}_{t-1}), & i = 0 \\ p(x_t | \mathbb{Y}_{t-1}, y_t^{(i)}), & \text{otherwise} \end{cases}$$

In the special case of a Kalman filter

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

$$P_{t|t}^{(i)} = \begin{cases} P_{t|t-1} & i = 0 \\ P_{t|t-1} - K_t S_{t|t-1} K_t^T, & \text{otherwise} \end{cases}$$

Note that the quantities $P_{t|t}^{(i)}$ and K_t , are the same for $i = 1, \dots, m_t$.

Probabilistic Data Association: KF algorithm (2/2)

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

$$\begin{aligned}\hat{x}_{t|t} &= \sum_{i=0}^{m_t} \omega_t^{(i)} \hat{x}_{t|t}^{(i)} = \hat{x}_{t|t-1} + \sum_{i=1}^{m_t} \omega_t^{(i)} K_t (y_t^{(i)} - \hat{y}_{t|t-1}) \\ &= \hat{x}_{t|t-1} + K_t \left(\underbrace{\omega_t^{(0)} \hat{y}_{t|t-1} + \sum_{i=1}^{m_t} \omega_t^{(i)} y_t^{(i)}}_{=y_t^{\text{eq}}} - \hat{y}_{t|t-1} \right)\end{aligned}$$

where y_t^{eq} is an *equivalent* measurement.

The covariance $P_{t|t}$ corresponding to $\hat{x}_{t|t}$ is given by $(\hat{x}_{t|t}^{(0)} = \hat{x}_{t|t-1}, P_{t|t}^{(0)} = P_{t|t-1})$

$$P_{t|t} = \sum_{i=0}^{m_t} \omega_t^{(i)} (P_{t|t}^{(i)} + (\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})(\hat{x}_{t|t}^{(i)} - \hat{x}_{t|t})^T)$$

It remains to compute $\omega_t^{(i)}$.

PDA: brief derivation of association weights (1/3)

Given m_t measurements

$$\omega_t^{(i)} = \Pr(\mathcal{H}_t^{(i)} | \mathbb{Y}_t) = \Pr(\mathcal{H}_t^{(i)} | y_t, m_t, \mathbb{Y}_{t-1}), \quad i = 0, 1, \dots, m_t.$$

Applying Bayes' rule yields

$$\omega_t^{(i)} \propto p(y_t | \mathcal{H}_t^{(i)}, m_t, \mathbb{Y}_{t-1}) \Pr(\mathcal{H}_t^{(i)}, m_t | \mathbb{Y}_{t-1})$$

The probability density for correct measurement is given by

$$p(y_t^{(i)} | \mathbb{Y}_{t-1}) = P_G^{-1} \mathcal{N}(\epsilon_t^{(i)}; 0, S_t).$$

Hence, the first factor is

$$p(y_t | \mathcal{H}_t^{(i)}, m_t, \mathbb{Y}_{t-1}) = \begin{cases} \mathcal{V}_t^{-(m_t-1)} P_G^{-1} \mathcal{N}(\epsilon_t^{(i)}; 0, S_t), & i = 1, \dots, m_t \\ \mathcal{V}_t^{-m_t}, & i = 0, \end{cases}$$

where \mathcal{V}_t is the volume of the validation region (the gate).

PDA: brief derivation of association weights (2/3)

The second factor is calculated as

$$\Pr(\mathcal{H}_t^{(i)} | m_t, \mathbb{Y}_{t-1}) = \Pr(\mathcal{H}_t^{(i)} | m_t)$$

$$= \begin{cases} \frac{P_D P_G P_{FA}(m_t - 1) / m_t}{P_D P_G P_{FA}(m_t - 1) + (1 - P_D P_G) P_{FA}(m_t)}, & i = 1, \dots, m_t \\ \frac{(1 - P_D P_G) P_{FA}(m_t)}{P_D P_G P_{FA}(m_t - 1) + (1 - P_D P_G) P_{FA}(m_t)}, & i = 0, \end{cases}$$

where $P_{FA}(m_t)$ is the probability of obtaining m_t false measurements in the validation region (the gate).

PDA: brief derivation of association weights (3/3)

Two common assumptions are considered for $P_{\text{FA}}(m_t)$ namely a parametric model with a **Poisson density** with parameter $\beta_{\text{FA}} \mathcal{V}_t$ or a non-parametric model using a **diffuse prior**.

1. Poisson density

$$P_{\text{FA}}(m_t) = \frac{(\beta_{\text{FA}} \mathcal{V}_t)^{m_t} e^{-\beta_{\text{FA}} \mathcal{V}_t}}{m_t!}, \quad m_t = 0, 1, \dots$$

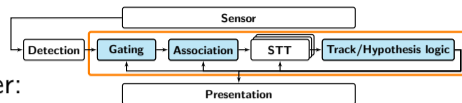
2. Diffuse prior

$$P_{\text{FA}}(m_t) = \frac{1}{M}, \quad m_t = 0, 1, \dots, M - 1,$$

where M is as large as needed.

Summary

Summary



- Gate to improve complexity in presence of clutter:
 - Rectangular: cheap but crude
 - Ellipsoidal: more correct
- Track logic determines if there is an object present or not.
 - State-machine for confirming target, based on gated measurements.
 - Score based logic, based on a hypothesis test.
- Different association strategies exist (so far for STT):
 - Nearest neighbor (NN) association.
A hard decision to use the "closest" measurement.
 - Probabilistic data association (PDA).
A soft decision where all measurements in the gate are combined.

Exercises

About the Exercises (1/2)

Purpose

- Provide hands on experience of common tools used in multi-target tracking.
- Help build a small toolbox of methods to get started with target tracking problem.

For full details see the course homepage:

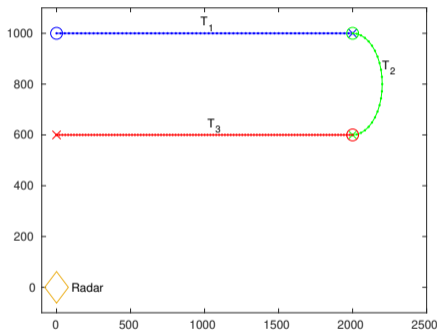
- <https://mtt.edu.hendeby.se/exercise.html>

About the Exercises (2/2)

- You can write your solutions in the language of your choice. (*I should be able to read and follow the solutions though...*)
- Feel free to use available packages, unless otherwise stated.
- The exercises are individual, but it is okay to discuss among each other.
- Hand in all produced code, and a *short* report (it should be possible to follow the report), outlining what you have done and your results.
- The exercises will incrementally lead up to a simplified MTT solution. Results from the earlier exercises will be reused in later exercises.
- Put your code and report in a git-repo, that you give me access to.

Exercise 1: part 1

Make a small simulator for the state trajectory:



- Simulate trajectory.
- Generate measurement:
 - P_D
 - P_{FA}
 - clutter
- Set up methods for visualization and performance evaluation.

Exercise 1: part 2

Get the hang of single target tracking (STT):

- Data:
 - Simulated data.
 - Provided realistic data.
- Filters:
 - EKF with CV model for different agility.
 - IMM for varied agility.
- NN-association and gating to handle false alarms.
- Monte Carlo RMSE evaluation of performance.

Exercise 1: part 3

Implement different track logics

- N/M logic.
- Track score logic.
- Evaluate on provided measurements.

Gustaf Hendeby

www.liu.se