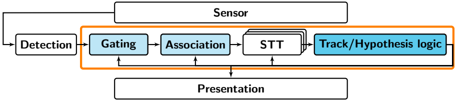
Target Tracking
Le 7: Selected topics

Gustaf Hendeby and Rickard Karlsson

Div. Automatic Control
Dept. Electrical Engineering
gustaf.hendeby@liu.se,
rickard.g.karlsson@liu.se

LINKÖPING UNIVERSITY

Summary: lecture 5-6



Multi-Hypotheses Tracker

- The conseptual MHT given by Reid 1979
- The Hypothesis Oriented MHT (HO-MHT)
  - Use k-best solutions to the assignment problem (Murty's method)
  - Find N_h-best hypothesis, generating as few hyps. as possible
- Track Oriented MHT (TO-MHT)
  - Maintain tracks, create hypotheses when needed.
  - Less tracks than global hypotheses.
- Presentation of the current state is not trivial.
- MATLAB and Python frameworks for MTT
- Guest lecture Arriver: radar, vision sensor fusion, machine learning, data association, cpu vs performance

Selected Topics

Today's lecture will focus on several different topics.

- Purpose is to highlight some problems/applications
- The ambition is an overview with references
- Examples: TkBD, T2T fusion, group tracking, and ETT

However, for some topics like ETT and group tracking there might be simularities.

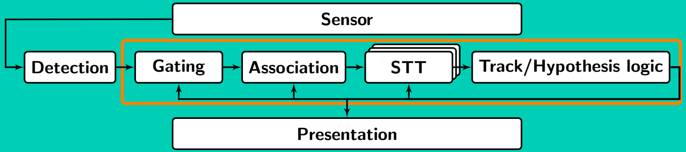
References on Multiple Target Tracking Topics (1/2)

- Performance Evaluation
  - M. Guerriero, L. Svensson, D. Svensson, and P. Willett. Shooting two birds with two bullets: How to find minimum mean OSPA estimates. In *13st International Conference on Information Fusion*, Edinburgh, UK, July 2010.
- Track-to-Track Fusion
  - J. K. Uhlmann. Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion*, 4(3):201–215, 2003.
  - J. Nygårds, V. Deleskog, and G. Hendeby. Safe fusion compared to established distributed fusion methods. In *IEEE Intemational Conference on Multisensor Fusion and Integration for Intelligent Systems*, Baden-Baden, Germany, Sept. 2016.
  - B. Noack, J. Sijs, and U. D. Hanebeck. Inverse covariance intersection: New insights and properties. In *20st International Conference on Information Fusion*, Xi'an, China, July 2017.

References on Multiple Target Tracking Topics (2/2)

- Track Before Detect
  - Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets.
    IEE Proc on Radar Sonar Navigation, 153(4):345–351, Aug. 2006.
  - B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina. A tutorial on bernoulli filters: Theory, implementation and applications.
    IEEE Transactions on Signal Processing, 61(13):3406–3430, July 2013.
- Extended Target Tracking
  - K. Granström, L. Svensson, S. Reuter, Y. Xia, and M. Fatemi. Likelihood-based data association for extended object tracking using sampling methods.
    IEEE Transactions on Intelligent Vehicles, 3(1), Mar. 2018.
  - K. Granström, M. Baum, and S. Reuter. Extended object tracking: Introduction, overview and applications.
    Journal of Advances in Information Fusion, 12(1), Dec. 2017.
- BOOK: B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications.*
  Artech House. 2004.

Performance Evaluation



LINKÖPING UNIVERSITY

Single Target Tracking: root mean square error (RMSE)

- A common performance measure for estimation is the *(root) mean square error* ((R)MSE). Given M estimates $\hat{x}_{1:T}^{(i)}$ of the matching ground truth $x_{1:T}^{0(i)}$,

$$\mathsf{MSE}(\hat{x}_t) = \frac{1}{M}\sum_{i=1}^{M} \hat{\|}\hat{x}_t^{(i)} - x_t^{0(i)}\|^2.$$

- The MSE combines the variance and bias of the estimate, $\mathsf{MSE}(\hat{x}_t) = \mathsf{var}\,\hat{x}_t + b_t^2$.

Single Target Tracking: RMSE performance bound

Cramér-Rao lower bound (CRLB)

The CRLB offers a fundamental performance bound for unbiased estimators and can be found as

$$\text{cov}(x_t - \hat{x}_{t|t}) \succeq P_{t|t}^{\text{CRLB}},$$

where $P_{t|t}^{\text{CRLB}}$ is the CRLB, given by the EKF around the true state (parametric CRLB) and inverse intrinsic accuracy replacing all noise covariances.

It is also possible to construct a posterior CRLB.

Note: The CRLB can be used when setting sensor requirements and in system design.

Normalized Estimation Error Squared (NEES)

- NEES provides a consistency estimate of an estimator,

$$\mathsf{NEES}(\hat{x}_t) = \frac{1}{M}\sum_{i=1}^{M}(\hat{x}_t^{(i)} - x_t^{0(i)})^T (P_t^{(i)})^{-1}(\hat{x}_t^{(i)} - x_t^{0(i)}).$$

- Given a Gaussian assumption and correct tuning, $\mathsf{NEES}(\hat{x}_t) \sim \chi^2(n_x)$

$< \boldsymbol{n_x}$ conservative estimate, *i.e.*, the estimate is better than indicated with the P.

$\approx \boldsymbol{n_x}$ the estimated covariance matches what is observed.

$> \boldsymbol{n_x}$ optimistic estimate, *i.e.*, the estimate is worse than indicated with the P.

Note: A $\chi^2(n_x)$ distribution has mean n_x and variance $2n_x$.

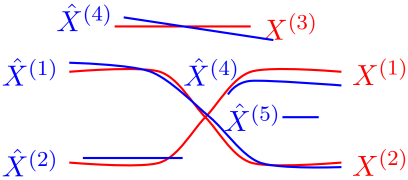
Multi-Target Tracking: performance

Multi-target tracking performance is a problem of relating elements of two different sets:

$$\{X^{(1)}, \dots, X^{(N)}\} \overset{\varphi: n \leftrightarrow m}{\longleftrightarrow} \{\hat{X}^{(1)}, \dots \hat{X}^{(M)}\}$$

How to handle:

- Inconsistent number of targets? $N \neq M$
- Match estimated track to ground truth track? φ
- Label switches? φ changes over time



How to judge the tracking result (blue tracks), compared to the ground truth (red tracks)? The number of tracks does not match, and the labels are different. . .

Multi-Target Tracking: performance criteria

Important properties:

- RMSE/NEES per target; how accurate are estimated tracks?
- Time to start track; how long does it take to confirm a new track?
- Track consistency; are the tracks kept together over time?

Multi-Target Tracking: OSPA (1/2)

- *Optimal subpattern assignment* (OSPA) is an extension of RMSE to the multi-target setting.
- Two sets of tracks $X = \{x^{(i)}\}_{i=1}^{N}$ (ground truth) and $\hat{X} = \{\hat{x}^{(i)}\}_{i=1}^{M}$ (estimated tracks).
- Is local, in the sense that is does not take label switches into consideration.
- Cardinality (number of targets) mismatch is penalized.
- Superseded by *generalized OSPA* (GOSPA), which has a slightly different carnality handling?

Multi-Target Tracking: OSPA (2/2)

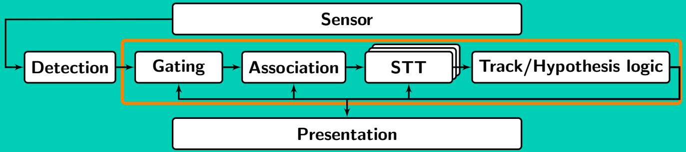
OSPA metric

Given two sets of tracks $\hat{X}$ and X, a metric $d(x, \hat{x})$, and a cost for incorrect targets c,

$$\tilde{d}_p^{(c)}(X, \hat{X}) = \left(\frac{1}{N} \min_{\theta} \sum_i d^{(c)}(x^{(i)}, \hat{x}^{(\theta(i))})^p + c^p |M - N|\right)^{\frac{1}{p}},$$

where $d^{(c)}(x, \hat{x}) = \min\big(d(x, \hat{x}), c\big)$ is a version of the chosen norm that saturates at c.

Track-to-Track Fusion



Track-to-Track (T2T) Fusion

- Consider a network of stand alone nodes performing target tracking.
- Estimates are passed around, which can lead to double use of data.
- How to efficiently combine measurements in a sound way?

Track-to-Track Fusion: independent estimates

Sensor Fusion Formula

Independent estimates $\{(\hat{x}^{(i)}, P^{(i)})\}_i$ we can combine these using the fusion formula:

$$\hat{x} = P \sum_i (P^{(i)})^{-1} \hat{x}^{(i)}$$
$$P^{-1} = \sum_i (P^{(i)})^{-1}.$$

This will give an over-confident estimate in case the estimates are not independent. In case of dependent estimates, more elaborate methods are needed.

Track-to-Track Fusion: dependent measurements (1/3)

Covariance Intersection (CI)

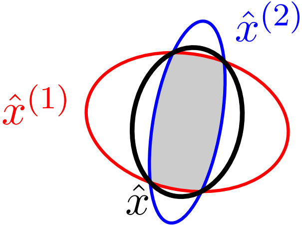
A conservative estimate of combined estimate of several estimates $\{(\hat{x}^{(i)}, P^{(i)})\}_i$ with unknown correlations:

$$\hat{x} = P \sum_i \omega^{(i)} (P^{(i)})^{-1} \hat{x}^{(i)}$$
$$P^{-1} = \sum_i \omega^{(i)} (P^{(i)})^{-1},$$

where $\sum_i \omega^{(i)} = 1$ are chosen as to minimize P under some norm, usually $\operatorname{tr}(P)$ or $\det(P)$.

Track-to-Track Fusion: covariance intersection illustration

- The covariance of the fused estimate will be within the intersection between the two covariances.
- Covariance intersection will choose the "smallest" P, covering the intersection.



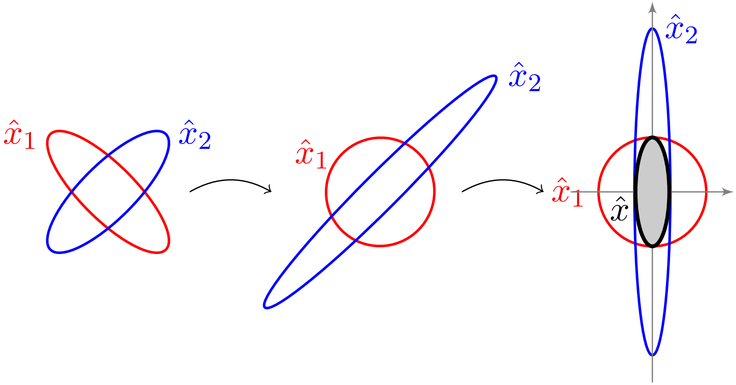
Track-to-Track Fusion: dependent measurements (2/3)

Safe Fusion

An easy to compute, but not completely conservative method to fuse two estimates with unknown dependencies.

1. SVD: $P^{(1)} = U_1 D_1 U_1^T$.
2. SVD: $D_1^{-1/2} U_1^T P^{(2)} U_1 D_1^{-1/2} = U_2 D_2 U_2^T$.
3. Transformation matrix: $T = U_2^T D_1^{-1/2} U_1^T$.
4. State transformation: $\hat{\hat{x}}_1 = T\hat{x}^{(1)}$ and $\hat{\hat{x}}_2 = T\hat{x}^{(2)}$.
   The covariances of these are $\operatorname{cov}(\hat{\hat{x}}_1) = I$ and $\operatorname{cov}(\hat{\hat{x}}_2) = D_2$.
5. For each component $i = 1, 2, \ldots, n_x$, let
$$[\hat{\hat{x}}]_i = [\hat{\hat{x}}_1]_i, \quad [D]_{ii} = 1 \qquad \text{if} \quad [D_2]_{ii} \geq 1,$$
$$[\hat{\hat{x}}]_i = [\hat{\hat{x}}_2]_i, \quad [D]_{ii} = [D_2]_{ii} \text{if} \quad [D_2]_{ii} < 1.$$
6. Inverse state transformation:
$$\hat{x} = T^{-1}\hat{\hat{x}}, \quad P = T^{-1} D^{-1} T^{-T}$$

Track-to-Track Fusion: safe fusion illustration



- The two estimates are transformed to become as independent as possible.
- Extract the best information in each direction.

Track-to-Track Fusion: dependent measurements (3/3)

Inverse Covariance Intersection (ICI)

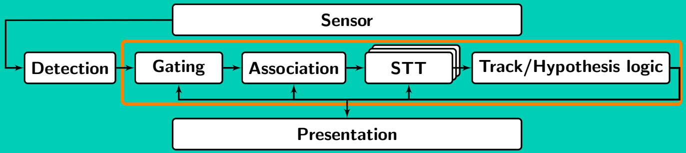
Conservative fusion method of two estimates under unknown dependencies given some (not completely known) structure.

$$\hat{x} = P\Big(\big((P^{(1)})^{-1} - \omega P_c^{-1}\big)\hat{x}^{(1)} + \big((P^{(2)})^{-1} - (1-\omega)P_c^{-1}\big)\hat{x}^{(2)}\Big)$$
$$P^{-1} = (P^{(1)})^{-1} + (P^{(2)})^{-1} - P_c^{-1}$$
$$P_c = \omega P^{(1)} + (1-\omega)P^{(2)}$$

Where ω is chosen to minimize some norm of P, *e.g.*, $\mathrm{tr}(P)$ or $\det(P)$.

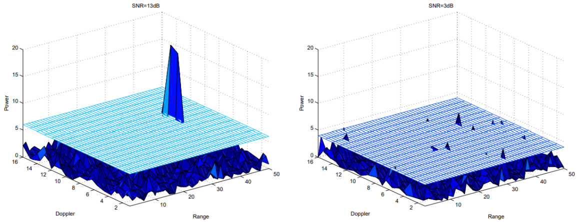
- The worst case common information, P_c, is estimated (mild structural assumptions).
- Fuse the estimates, taking the estimated common information into consideration.

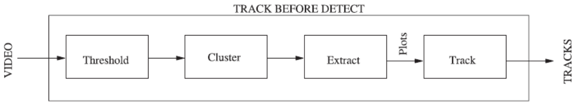
Track Before Detect (TkBD)



Track Before Detect: SNR motivation

General TkBD concept: simultaneous detection and tracking





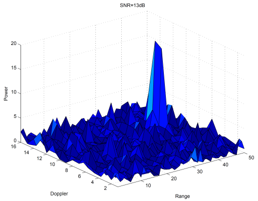
- High SNR: traditional detection works
- Low SNR: traditional detections will not work
- Note: do not want to lower the threshold too much!
- CFAR

Track Before Detect: Radar example

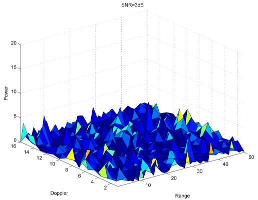
The Track Before Detect concecpt will be described in a longer example. It contains several topics:

- Track Before Detect principle
- Radar model
- Motion model
- Target birth/death model
- For a Bayesian filtering context
- Extend Target Tracking (ETT)

Track Before Detect: idea



(a) SNR=13 dB. A high SNR makes it easy to detect the point target.



(b) SNR=3 dB. A low SNR makes the target hard to detect in a cluttered environment.

- Radar example (but also applies for images).
- Assume one target.
- Consistent motion model.
- Threshold detector vs simultaneous detection and tracking
- Stealthy targets

Track Before Detect: assumptions and methods

Basically we assume that we can use:

- Data over several scans
- Prohibit or penalize deviations from straight line motion
- Assume one target (or sufficiently separated)

There are many ways to achieve TkBD:

- Batch-algorithms
- Hough transform
- Dynamic Programming
- Bayesian filtering

Solution for tracking of stealthy targets:

Unthresholded info via simultaneous detection and tracking.

Track Before Detect: Bayesian concept (1/2)

First study a 2D image, with position, velocity and intensity as states

$$x_t = \begin{pmatrix} X_t & Y_t & \dot{X}_t & \dot{Y}_t & I_t & m_t \end{pmatrix}^T$$

We also need to consider the mode of existance (m) of a target, with birth/death according to:

$$P_b = P(m_t = 1 | m_{t-1} = 0)$$
$$P_d = P(m_t = 0 | m_{t-1} = 1),$$

which will give a Markov transition matrix.

Track Before Detect: Bayesian concept (2/2)

Dynamics:

CV-model or similar.

Observation model:

$$y_t^{(i,j)} = \begin{cases} h^{(i,j)}(x_t) + e_t^{(i,j)}, & \text{if target present} \\ e_t^{(i,j)}, & \text{if target absent} \end{cases}$$

where $h^{(i,j)}(x_t)$ is the target intensity contribution in resolution cell (i, j).
For a 2D point target we consider a Gaussian for describing this:

$$h^{(i,j)}(x_t) \propto I_t \cdot e^{-\frac{(i\Delta_x - X_t)^2 + (j\Delta_y - Y_t)^2}{2\sigma^2}}$$

Basically, we now have all that is needed to write down this as a Bayesian formulation, which can be solved with for instance a PF.

Track Before Detect: radar modeling (1/2)

Now consider a radar tracking stealthy targets:

- Instead of thresholding, the entire radar video signal is used, *i.e.* the received power, $P(r^{(j)}, d^{(k)}, b^{(l)})$, $\forall j, k, l$.
- The measurements consist of the power levels in $N_r \times N_d \times N_b$ sensor cells, where N_r, N_d, and N_b are the number of range, Doppler, and bearing cells.

For each range-Doppler-bearing cell, $(r^{(j)}, d^{(k)}, b^{(l)})$, the received power in the measurement relation is given by

$$y_{P,t}^{jkl} = \left|y_{A,t}^{jkl}\right|^2 = |A_t^{jkl} \cdot h_A^{jkl}(x_t) + e_t^{jkl}|^2,$$

where $j = 1, \ldots, N_r,\ k = 1, \ldots, N_d,\ l = 1, \ldots, N_b$.

Track Before Detect: radar modeling (2/2)

$$h_A^{jkl}(x_t) = \exp -\frac{(r^{(j)} - r_t)^2}{2R}\lambda_r - \frac{(d^{(k)} - d_t)^2}{2D}\lambda_d - \frac{(b^{(l)} - b_t)^2}{2B}\lambda_b.$$

The constants R, D, and B are related to the size of the range cell, the Doppler cell, and the bearing cell. Losses are represented by the constants λ_r, λ_d, and λ_b. The noise is defined by

$$e_t^{jkl} = e_{I,t}^{jkl} + \imath \cdot e_{Q,t}^{jkl},$$

which is complex Gaussian, where $e_{I,t}^{jkl}$ and $e_{Q,t}^{jkl}$ are independent, zero-mean white Gaussian with variance σ_e^2, for the in-phase and quadrature-phase, respectively.

It is possible to derive a rather complicated likelihood function.

Track Before Detect: tracking filter (1/2)

Estimation model

$$\begin{aligned} x_{t+1} &= f(x_t, m_t, w_t) \\ y_t &= h(x_t, m_t) + e_t, \end{aligned}$$

where m_t is target precense or not. Typically, given by a Markov probability for birth/death events.

This has the impact on the measurement model:

$$y_t = \begin{cases} e_t, & \text{if } m_t = 0 \\ h(x_t) + e_t, & \text{if } m_t = 1. \end{cases}$$

Track Before Detect: tracking filter (2/2)

For the radar model we have

$$y = h(x) + e = \begin{pmatrix} \varphi \\ \theta \\ r \\ \dot{r} \end{pmatrix} + e = \begin{pmatrix} \mathrm{atan2}(\mathsf{y}/\mathsf{x}) \\ \mathrm{atan2}(\mathsf{z}/\sqrt{\mathsf{x}^2+\mathsf{y}^2}) \\ \sqrt{\mathsf{x}^2+\mathsf{y}^2+\mathsf{z}^2} \\ \frac{\mathsf{x}v^{\mathsf{x}}+\mathsf{y}v^{\mathsf{y}}+\mathsf{z}v^{\mathsf{z}}}{\sqrt{\mathsf{x}^2+\mathsf{y}^2+\mathsf{y}^2}} \end{pmatrix} + e$$

Now possible to use a particle filter. For a specific problem, one has to calculate relevant likelihoods etc.

Track Before Detect: extended targets (1/2)

A spatial distribution model for extended objects is assumed, $p(\tilde{x}_t|x_t)$, which can be interpreted as a generator of a point source $\tilde{x}_t$ from an extended target with its center and orientation given by the state vector x_t.

Receiving a measurement from a source $\tilde{x}_t$ somewhere on the target leads to a likelihood conditioned on a specific source $\Lambda(x_t) = p(y_t|\tilde{x}_t)$. Using this model the total likelihood is obtained as

$$p(y_t|x_t) = \int p(y_t|\tilde{x}_t)p(\tilde{x}_t|x_t)\, d\tilde{x}_t.$$

Track Before Detect: extended targets (2/2)

$$p(y_t|x_t) = \int p(y_t|\tilde{x}_t)p(\tilde{x}_t|x_t)\, d\tilde{x}_t.$$

- Point Target:

$$p(\tilde{x}_t|x_t) = \delta(\tilde{x}_t - x_t).$$

- Point Sources:

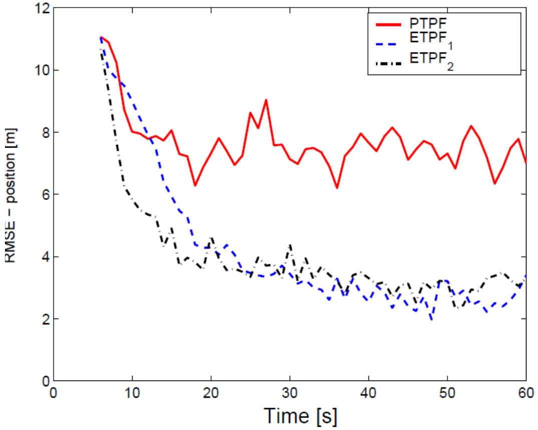
$$p(\tilde{x}_t|x_t) = \sum_{i=1}^{M} \Lambda(x_t^{(i)})\delta(\tilde{x}_t - x_t^{(i)}).$$

- Extended Target:

$$p(y_t|x_t) \approx \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} p(y_t|\tilde{x}_t^{(i)}),$$

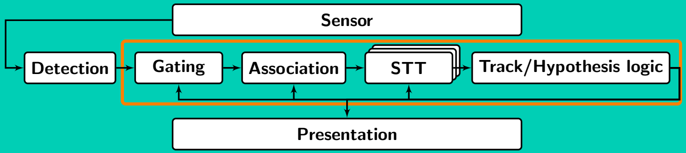
with $\tilde{x}^{(i)}$, independently drawn according to $p(\tilde{x}_t|x_t)$ for $i = 1, \ldots, \tilde{M}$.

Track Before Detect: summary

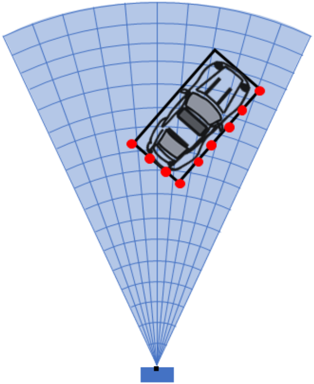


- TkBD can be used for extended targets
- Position RMSE for point targets and two extended targets
- Computational intensive
- Motion model must correspond to true target
- Multiple targets will be complicated
- Possible to track for low SNR

Extended target Tracking (ETT)



Extended Target Tracking



From MATLAB Sensor fusion and tracking toolbox.

When the sensor resolution becomes higher than the target size:

- Target cannot be modeled as points anymore.
- One measurement per target does not hold any more.
- Measurement could be correlated.
- Options to deal with this:
  - Cluster the measurements before applying a regular tracker.
  - Take the target extent into consideration (estimate it).

    The simplest extension is a point target with an estimated geometric shape, like the length (see TkBD).

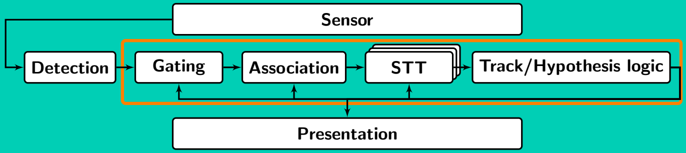
Extended Target Tracking: measurement clustering

- A standard MTT is a point target tracker.
- It assumes that every track can be detected at most once by a sensor in a scan.
- If detections are not clustered, the tracker generates multiple tracks per object.
- Clustering returns one detection per cluster, at the cost of having a larger uncertainty

Extended Target Tracking: extension modeling

- **Geometry:** Need to specify a model for the extended object: rectangular, ellipsoidal, star convex etc.
- **Dynamics:** Each extended object must have some motion model, for instance coordinated turn about its pivot.
- ETT handles multiple detections per object and sensor without the need to cluster detections, at the cost of more advanced association and a more complex model.

Group Tracking



LINKÖPING UNIVERSITY

Group Tracking

Standard tracking:

- A target is a “single point”
- We receive at most one measurement for each target

Group tracking:

- Tracking a group of targets that moves in a similar way
- An extended target could be seen as a similar problem

Note: extended target tracking and group tracking could sometimes be the same.

Group Tracking: dynamic model

Consider the bulk model (B) and the individual targets x, according to:

$$B_{t+1} = f^B(B_t, w_t)$$
$$x_{t+1}^{(i)} = f^{(i)}(x_t^{(i)}, w_t^{(i)}),$$

where we assume $i = 1, \ldots, N_{tg}$. Usually $f^{(i)} = f$.

Note: The bulk is the center or the mean position, orientation etc. Everything can be implemented by extending the state vector.

Group Tracking: observation model

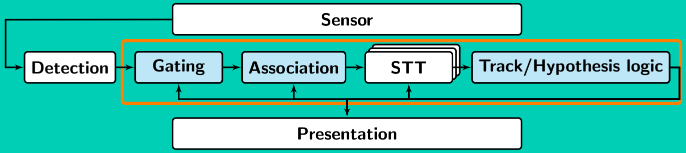
The observation cannot originate from multiple sources. Each measurement is from a target or clutter

$$y_t^{(j)} = h\big(\Psi(x_t^{(i)}, B_t)\big) + e_t,$$

where Ψ be a nonlinear transformation.

Now proceed with association etc.

Summary Classic Target Tracking

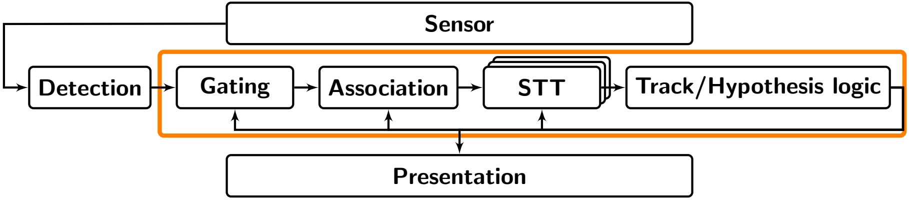


LINKÖPING UNIVERSITY

Summary Multi-Target Tracking Course: basis

Problem formulation:

Multi-target tracking is the problem of decide how many targets are present and how they move, given measurements with imperfections.



Components in classical multi-target tracking solutions.

Summary Multi-Target Tracking Course: single target tracking

Single target tracking

- Filters
  - (Extended/Unscented) Kalman type filter
  - Particle filter
  - Filter banks (IMM, GBP, RPEKF, ...)
- Motion models: $x_{t+1} = f(x_t) + v_t$
  - Constant velocity
  - Constant acceleration
  - Coordinated turn
  - Switched models for maneuvering targets
- Observation models: $y_t = h(x_t) + e_t$
- Clutter
- Missed detections

Summary Multi-Target Tracking Course: multi-target tacking

Multi-target tracking

- Classic methods (GNN, JPDA, MHT):
  - Differ in the association method used.
  - Track logic for initiation and termination.

Summary Multi-Target Tracking Course: extensions

- Track Before Detect: raw observations are used for simulataneous detection and tracking in **poor SNR**.
- Performance measures
  - Root mean square error (RMSE)
  - Normalized estimation error square (NEES)
  - Cramér-Rao lower bound (CRLB)
  - Optimal subpattern association (OSPA): multi-target
- Extended target and group tracking
- Various examples of tracking applications from research and industry

Gustaf Hendeby and Rickard Karlsson