U.S. NAVAL RESEARCH LABORATORY

A Crash Course in Basic Single-Scan Target Tracking (Abridged)

David Frederic Crouse

Radar Division, Washington, DC

April 2018

Distribution A: Approved for Public Release

U.S. NAVAL RESEARCH LABORATORY

INTRODUCTION

2 / 85

Radar Sensor Technology XXII, edited by Kenneth I. Ranney, Armin Doerry,
Proc. of SPIE Vol. 10633, 106330P · © 2018 SPIE · CCC code: 0277-786X/18/$18
doi: 10.1117/12.2307567

What is Target Tracking?

Different Impressions Obtained from the Literature:

- A control systems problem to point an antenna towards an object of interest.
- The prediction of the future state of a dynamical system based on measurements and models.
- The act of connecting a vehicle's consecutive positions over time.
- A problem that was solved by Rudolph E. Kálmán in 1960.

What is Target Tracking?

Target Tracking Is:

- An aid to reduce the workload of radar operators.
- A process of finding objects of interest where humans couldn't discern them.
- An optional part of a radar/sonar system.
- An indispensable part of a radar system.
- A critical part of a missile control system or of a counter-missile system.
- A trivial connecting of the dots.
- Something that people can do better than the computer.
- Something that the computed can do better than people.

What is Target Tracking?

Target Tracking Is:

- An aid to reduce the workload of radar operators.
- A process of finding objects of interest where humans couldn't discern them.
- An optional part of a radar/sonar system.
- An indispensable part of a radar system.
- A critical part of a missile control system or of a counter-missile system.
- A trivial connecting of the dots.
- Something that people can do better than the computer.
- Something that the computed can do better than people.

The difficulty and utility of target tracking methods depend on the application.

What Areas of Study Are Relevant?

What Areas of Study Are Relevant?

U.S. Naval Research Laboratory

5 / 85

Resources

- Getting started can be difficult.
- No comprehensive textbooks on tracking exist.
- Some useful books:
  - (Bar-Shalom, Li, and Kribarajan): *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*
  - (Crassidis, Junkins) *Optimal Estimation of Dynamic Systems*
  - (Bar-Shalom, Willett, Tian) *Tracking and Data Fusion: A Handbook of Algorithms*
  - (Blackman, Popoli) *Design and Analysis of Modern Tracking Systems*
  - (Maybeck) *Stochastic Models, Estimation, and Control*
  - (Stone, Streit, Corwin, Bell) *Bayesian Multiple Target Tracking*
  - (Challa, Moreland, Mušicki, Evans) *Fundamentals of Object Tracking*
  - (Mahler) *Statistical Multisource-Multitarget Information Fusion*

U.S. Naval Research Laboratory

6 / 85

Resources

- The International Conference on Information Fusion by the International Society of Information Fusion (ISIF) is the most relevant to target tracking, especially networked/multistatic tracking.
  - ISIF http://www.isif.org
  - Fusion 2018, Cambridge England: http://fusion2018.org
    Fusion 2019, Ottawa Canada.
- The Tracker Component Library (TCL) offers over 1,000 free, commented Matlab routines related to Tracking, Coordinate Systems, Mathematics, Statistics, Combinatorics, Astronomy, etc.
  - https://github.com/USNavalResearchLaboratory/TrackerComponentLibrary
  - Description of library given in
    D. F. Crouse, "The Tracker Component Library: Free Routines for Rapid Prototyping," IEEE Aerospace and Electronic Systems Magazine, vol. 32, no. 5, pp. 18-27, May. 2017.

Overview

MATHEMATICAL PRELIMINARIES

Mathematical Preliminaries

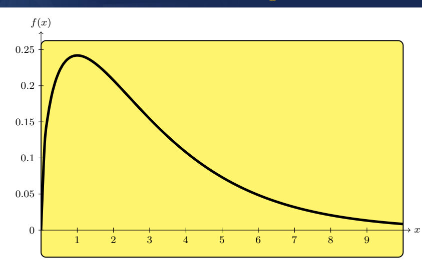
Useful Mathematical Tools

1. Univariate and Multivariate Taylor Series Expansions
   - Given in the unabridged version of the slides.
2. Useful Probability Distributions.
3. Cubature Integration.
4. The Cramér-Rao Lower Bound.

Probability Distributions

- The four most prevalent probability distributions in target tracking tend to be:
  1. The Multivariate Gaussian Distribution.
     - Usual assumed noise distribution; discussed in the unabridged slides.
  2. The Central Chi-Square Distribution.
  3. The Binomial Distribution.
  4. The Poisson Distribution.
- In the TCL, functions relating to these and many other distributions are given in "Mathematical Functions/Statistics/Distributions."
- For the above distributions, see GaussianD, ChiSquareD, BinomialD, and PoissonD in the TCL.

Probability Distributions: Chi-Squared



- The central chi-squared distribution with k degrees of freedom is

$$\chi^2(x, d_x) = \frac{x^{\frac{k}{2}-1}e^{-\frac{x}{2}}}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)} \tag{1}$$

where Γ is the gamma function.
- Plotted is $\chi^2(x, 3)$.
- Confidence regions of a desired % are easily determined using Gaussian approximations, Mahalanobis distances, and chi-squared statistics.

Probability Distributions: Chi-Squared

- Given a Gaussian PDF estimate of a target, a point $\mathbf{x}$, is within the first pth-percentile if

$$(\hat{\mathbf{x}} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\hat{\mathbf{x}} - \boldsymbol{\mu}) < \gamma_p \quad (2)$$

where γ_p depends on p and on d_x, the dimensionality of x.

	Confidence Region p				
d_x	0.9	0.99	0.999	0.9999	0.99999
1	2.7055	6.6349	10.8276	15.1367	19.5114
2	4.6052	9.2103	13.8155	18.4207	23.0259
3	6.2514	11.3449	16.2662	21.1075	25.9017
6	10.6446	16.8119	22.4577	27.8563	33.1071
9	14.6837	21.6660	27.8772	33.7199	39.3407

Values of γ_p for p and d_x.

- Use ChiSquareD.invCDF in the TCL to determine γ_p.

Probability Distributions: Chi-Squared

- The chi-squared distribution plays a role in assessing covariance consistency.
- The covariance is consistent if it realistically models the error.
- The Normalized Estimation Error Squared (NEES) is the simplest of multiple methods for assessing consistency.

$$\text{NEES} \triangleq \frac{1}{N d_x} \sum_{i=1}^{N} (\hat{\mathbf{x}}_i - \mathbf{x}_i) \mathbf{P}_i^{-1} (\hat{\mathbf{x}}_i - \mathbf{x}_i) \quad (3)$$

  - $\hat{\mathbf{x}}_i$ and $\mathbf{P}_i$ are estimated mean and covariance from ith random trial.
  - $\mathbf{x}_i$ true value from ith random trial.
- If estimator is unbiased, covariance is always correct and errors truly Gaussian, then the NEES is $\frac{1}{N d_x}$ time a central chi-squared random variable with $N d_x$ degrees of freedom.
- The function calcNEES in the TCL can be useful.

Probability Distributions: Binomial

- Consider constant false alarm rate (CFAR) detector with a given P_{FA} per cell, such as the ones given by the CACFAR or OSCFAR functions in the TCL.
- Grid of N cells (e.g. in range and range-rate).
- Probability of n false alarms is binomially distributed.

$$\Pr\{n\} = \binom{N}{n} P_{\text{FA}}^{n} \left(1 - P_{\text{FA}}\right)^{N-n} \quad (4)$$

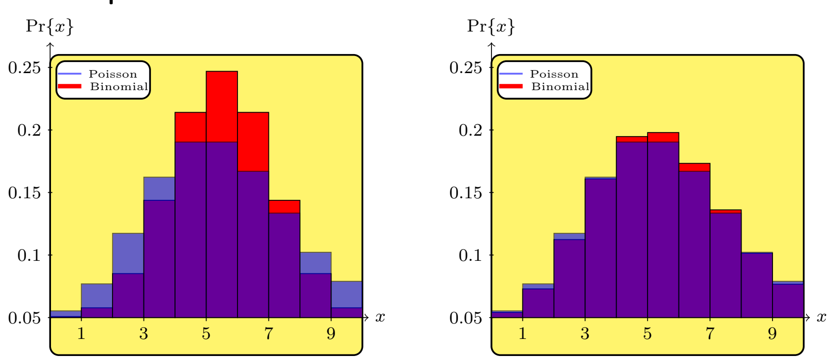
with mean

$$\tilde{\lambda} = N P_{\text{FA}} \quad (5)$$

- The binomial distribution is almost never used in trackers.
- It is approximated by a Poisson distribution with the same $\tilde{\lambda}$.
  - The asymptotic equivalence is derived in the unabridged slides.

Probability Distributions: Poisson

Example:



(a) N=10 (b) N=50

- Both plots, $\tilde{\lambda} = 5$ for both distributions.
- At $N = 1000$, the binomial and Poisson plots look the same.

Cubature Integration

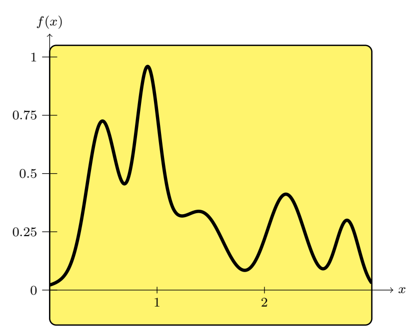
- Many integrals cannot be solved analytically with a finite number of terms.
  - Try to evaluate a Fresnel integral:

$$C(z) = \int_0^z \cos\left(\frac{\pi t^2}{2}\right) dt \qquad (6)$$

- Quadrature integration is a technique for efficient numerical evaluation of univariate integrals.
- Cubature integration is multivariate quadrature integration.
- The TCL has many functions related to cubature integration in "Mathematical Functions/Numerical Integration" and "Mathematical Functions/Numerical Integration/Cubature Points."

Cubature Integration: Why?

Numerically integrate the function from 0 to 2.

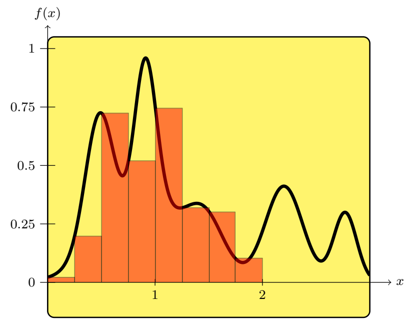


- Evaluate

$$\int_0^2 f(x)\, dx = ? \qquad (7)$$

Cubature Integration: Why?

Numerically integrate the function from 0 to 2.



- Basic calculus solution: A Riemann sum:

$$\int_0^2 f(x)\,dx \approx \sum_{k=0}^{N-1} f\left(k\Delta_x\right)\Delta_x \qquad \text{where } 2 = N\Delta_x. \quad (8)$$

Cubature Integration: Theory

- Riemann sums converge very slowly.
- The idea in quadrature/cubature is the relation

$$\int_{\mathbf{x}\in\mathbb{S}} \mathbf{f}(\mathbf{x})w(\mathbf{x})\,d\mathbf{x} = \sum_{i=0}^{N} \omega_i \mathbf{f}(\mathbf{x}_i), \qquad (9)$$

  is *exact* for a particular weighting function w for *all* polynomials f up to a certain order and approximate for other functions f.
  - $\mathbb{S}$ is a region, such as $\mathbb{R}^n$ or the surface of a hypersphere.
  - Unlike a Riemann sum, N is finite.
  - Cubature weights ω_i and points $\mathbf{x}_i$ depend on w and the order.
- Efficient: For a fifth-order integral with a multivariate Gaussian weighting function, one can choose points such that $N = 12$.

On Solving Integrals

- Many parts of target tracking involve solving difficult multivariate integrals.
- Many algorithms fall into one of two categories:
  1. Use cubature integration for the integrals.
  2. Use a Taylor series expansion to turn the problem polynomial and solvable.
- This comes up again and again.

The CRLB

- The Cramér-Rao Lower Bound (CRLB) is a lower bound on the variance (or covariance matrix) of an unbiased estimator.
- The CRLB and a posterior CRLB (PCRLB) are widely used to asses tracker performance.
- Under certain conditions, the CRLB states

$$\mathrm{E}\left\{(\mathbf{x}-\mathbf{T}(\mathbf{z}))(\mathbf{x}-\mathbf{T}(\mathbf{z}))'\right\} \geq \mathbf{J}^{-1} \tag{10}$$

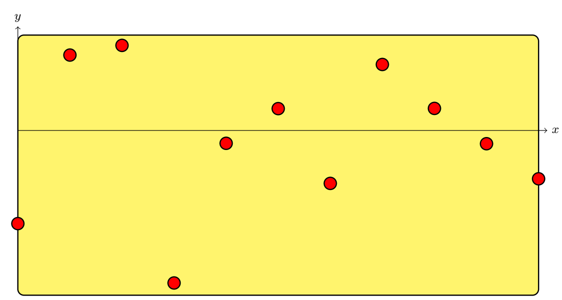
  - A matrix inequality refers to sorted eigenvalues.
  - $\mathbf{x}$ is the quantity being estimated.
  - $\mathbf{T}(\mathbf{z})$ is the best unbiased estimator.
  - $\mathbf{J}$ is the Fisher information matrix.
  - The expectation is taken over the conditional PDF $p(\mathbf{z}|\mathbf{x})$ if $\mathbf{x}$ is deterministic.
- The Fisher information matrix has two equivalent formulations:

$$\mathbf{J}^B = -\mathrm{E}\left\{\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}' \ln\left(p(\mathbf{z}|\mathbf{x})\right)\right\} \tag{11}$$

$$= \mathrm{E}\left\{\left(\nabla_{\mathbf{x}} \ln\left(p(\mathbf{z}|\mathbf{x})\right)\right)\left(\nabla_{\mathbf{x}} \ln\left(p(\mathbf{z}|\mathbf{x})\right)\right)'\right\} \tag{12}$$

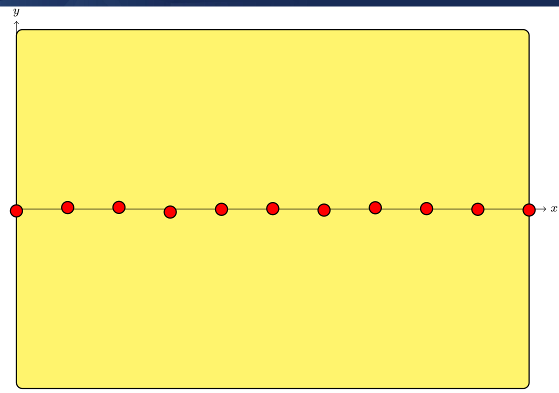
MEASUREMENTS AND NOISE

Measurements and Noise



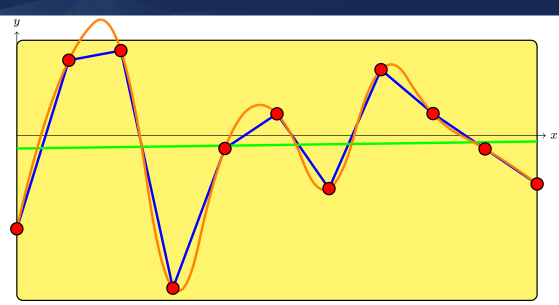
- Are these points false alarms or a possible track over time?
- Are they accurate measurements that are far apart?
- Are false alarms very unlikely or highly likely?

Measurements and Noise



- Are these points false alarms or a possible track over time?
- These are the same points as before at a different scale.
- Measurements are inherently *noisy*.
- Knowledge of measurement noise level determines scale.

Measurements and Noise



- The blue line is "connect-the-dots." The orange line just adds interpolation.
- The blue/orange lines are only good if the points are very accurate.
- The green line is much more reasonable if the points are inaccurate.
- The noise level determines the best fit.

BAYES' THEOREM AND THE LINEAR KALMAN FILTER UPDATE

Bayes' Theorem

- Given a PDF $p(\mathbf{x})$ representing the target state estimate at a particular time.
- Given a measurement $\mathbf{z}$ and a conditional PDF of the measurement $p(\mathbf{z}|\mathbf{x})$.
- Bayes' theorem states that

$$\overbrace{p(\mathbf{x}|\mathbf{z})}^{\substack{\text{posterior}\\\text{distribution}}} = \frac{\overbrace{p(\mathbf{z}|\mathbf{x})}^{\substack{\text{measurement}\\\text{distribution}}} \overbrace{p(\mathbf{x})}^{\substack{\text{prior}\\\text{distribution}}}}{\underbrace{p(\mathbf{z})}_{\text{normalizing constant}}} \tag{13}$$

- The value $p(\mathbf{z})$ is essentially a normalizing constant.

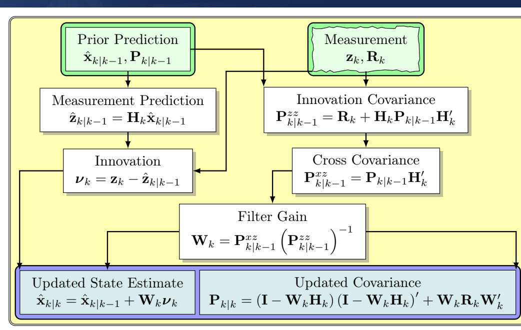
$$p(\mathbf{z}) = \int_{\mathbf{x}\in\mathbb{S}} p(\mathbf{z}|\mathbf{x})p(\mathbf{x})\,d\mathbf{x} \tag{14}$$

where $\mathbb{S}$ is whatever space $\mathbf{x}$ is in (For discrete variables, the integral becomes a sum).

Bayes' Theorem and Joint Distributions

- Bayes' theorem underlies all rigorous measurement update algorithms in tracking.
- The Kalman filter measurement update is just Bayes' theorem applied to a linear/Gaussian measurement model assuming a Gaussian prior.
- Notation change for standard tracking:
  - The "prior" subscript will be replaced by "$k|k-1$" to indicate that one has an estimate of a current (step k) state given prior (step $k-1$) information.
  - The "posterior" subscript will be replaced by "$k|k$" to indicate that one has an estimate of a current state given current information.

Bayes' Theorem: Linear Gaussian Distributions



- The discrete measurement update step of the Kalman filter with common notation/terminology.
- The updated covariance estimate has been reformulated in *Joseph's form* for numerical stability.
- See KalmanUpdate in "Dynamic Estimation/Measurement Update" in the TCL.

Bayes' Theorem: Why Use Approximations?

- The Kalman filter update is optimal for measurements that are linear combinations of the target state.
- However, why not just apply Bayes' theorem more precisely?
- Bayes' theorem is again:

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \tag{15}$$

  - Just multiply two known functions and normalize the result.
- Bayes' theorem is trivial. Why not always do it optimally?

Bayes' Theorem: Why Use Approximations?

- Bayes' theorem is just normalized multiplication. Why not just discretize space and do everything almost optimally on a grid?
- Simplest "optimal" Bayesian filter:
  1. Discretize the entire estimation space
  2. Evaluate probabilities on a discrete grid for given distributions
  3. Multiply matrices of probabilities to get posterior; normalize
- It is simple.
- With parallelization over GPUs, couldn't it be done quickly?

U.S. NAVAL RESEARCH LABORATORY

Bayes' Theorem: Why Use Approximations?

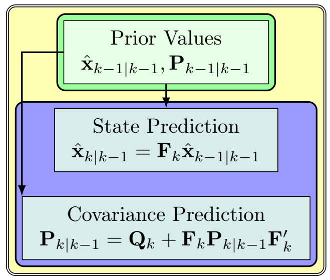
- Why the brute-force grid approach is seldom done:
  - One target 3D position and velocity in $50\,\text{km}$ cube all directions about sensor, speed in any direction to Mach 4 $(1372, \text{m/s})$, discretized to $5\,\text{m}$ and 1m/s.
  - Grid for single probability density function (PDF) is more than 2×10^{22} in size (we need two).
  - As floating doubles, one grid requires more than 82 zettabytes of RAM (1 ZB=1 trillion GB).
  - 64GB RAM stick $\approx \$255$ so cost $\approx \$330$ trillion ($\$660$ trillion for two grids, US GDP $\approx \$53$ trillion).
  - Computing power to multiply two grids in $1\,\text{ms}$ is ≈ 20 exaflops.
  - Most powerful supercomputer (Tianhe-2, China) 33.85 petaflops. We need 612 of them.
- A smarter approach would be to use some type of adaptive grid or set of points.
  - This is the basis of particle filters (to be discussed later).
- The Kalman filter is much faster than the most efficient particle filter.

U.S. Naval Research Laboratory

U.S. NAVAL RESEARCH LABORATORY

THE LINEAR KALMAN FILTER PREDICTION

The Linear Kalman Filter Prediction Summary



- The stochastic dynamic models describe prediction when the initial state $\mathbf{x}$ is deterministic.
- The prediction step of the standard Kalman filter is derived in the unabridged slides and handles random $\mathbf{x}$.
- See the discKalPred function in the TCL.

LINEAR INITIAL STATE ESTIMATION AND THE INFORMATION FILTER

Linear Initial State Estimation

Two common approaches to starting the filter are

1. One-point initiation.
   - See the functions in "Dynamic Models/One-Point Initialization" in the TCL.
2. Using an information filter.
   - See infoFilterUpdate and infoFilterDiscPred in the TCL.
   - This is discussed in the unabridged slides.

Linear Initial State Estimation

- One-point initiation is the simplest approach:
  - The initial state and covariance matrix are

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} \hat{\mathbf{z}}_{\text{Cart}} \\ \mathbf{0}_{d_x - d_z} \end{bmatrix} \quad (16)$$

$$\hat{P}_{0|0} = \begin{bmatrix} \mathbf{R}_{\text{Cart}} & \mathbf{0}_{d_z, d_x - d_z} \\ \mathbf{0}_{d_x - d_z, d_z} & \text{diag}([\sigma_1^2, \sigma_2^2, \ldots, \sigma_{d_x - d_z}^2]) \end{bmatrix} \quad (17)$$

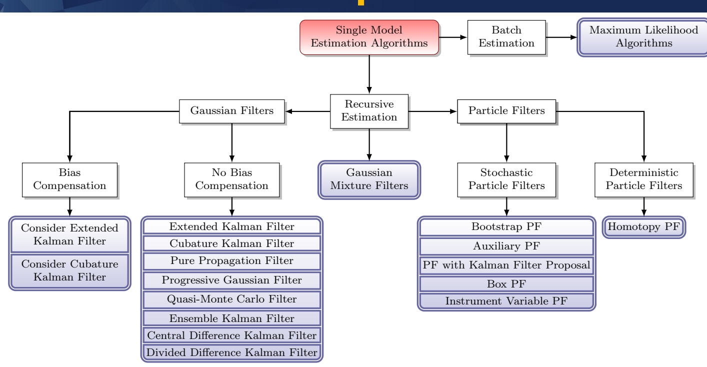
  where
    - d_x and d_z are the dimensionalities of the state and the Cartesian-converted measurement.
    - $\sigma_1^2, \ldots, \sigma_{d_x - d_z}^2$ are large variances based on the maximum velocity, acceleration, etc of the target.
- Known position, other components "uninformative".
- Updates and predictions can then be done using the standard Kalman filter.
- A rule of thumb for σ_i is to use the maximum value of the value of the moment divided by 2 or 3.

U.S. NAVAL RESEARCH LABORATORY

NONLINEAR MEASUREMENT UPDATES

U.S. NAVAL RESEARCH LABORATORY

Nonlinear Measurement Updates



- Measurement updates are possible without Cartesian conversion.
- Major nonlinear filtering algorithms shown.
- We focus on the Extended Kalman Filter and variants of the cubature Kalman filter (which include the "unscented" KF).
- See EKFUpdate and cubKalUpdate in the TCL.

U.S. Naval Research Laboratory

U.S. NAVAL RESEARCH LABORATORY

Nonlinear Measurement Updates

- The Kalman filter arose from a Bayesian update given that a linear measurement and the state are jointly Gaussian.
- Approximating a *nonlinear* measurement

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{w} \tag{18}$$

where $\mathbf{w}$ is Gaussian, as jointly Gaussian with the state, one still has the same basic update equations as the Kalman filter

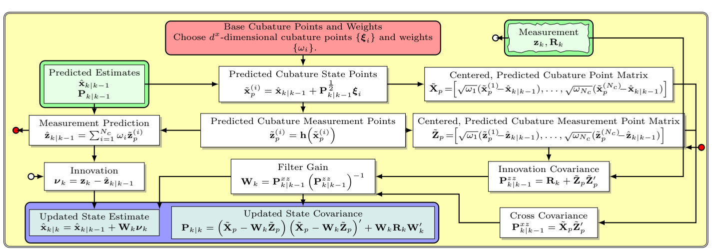
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}^{xz}_{k|k-1}\left(\mathbf{P}^{zz}_{k|k-1}\right)^{-1}\left(\mathbf{z} - \hat{\mathbf{z}}_{k|k-1}\right) \tag{19}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}^{xz}_{k|k-1}\left(\mathbf{P}^{zz}_{k|k-1}\right)^{-1}\mathbf{P}^{zx}_{k|k-1} \tag{20}$$

but the quantities $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}^{zz}_{k|k-1}$, $\mathbf{P}^{xz}_{k|k-1}$ are now integrals.

U.S. NAVAL RESEARCH LABORATORY

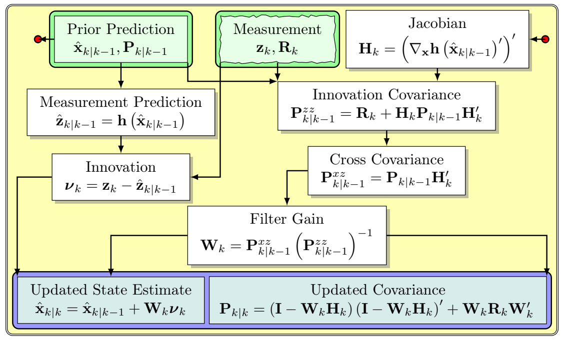
Nonlinear Measurement Updates: CKF



- The simplest solution to the nonlinear integrals is to use cubature integration, shown above.
- The square root is a lower-triangular Cholesky decomposition.
- The vector formulation above requires all cubature weights be positive, but allows for Joseph's form to be used.
- A Joseph's formulation supporting negative cubature weights is probably impossible.

U.S. NAVAL RESEARCH LABORATORY

Nonlinear Measurement Updates: EKF



- An alternative approach is to use a Taylor series expansion of the nonlinear function.
- The result is the extended Kalman filter (EKF), shown above.

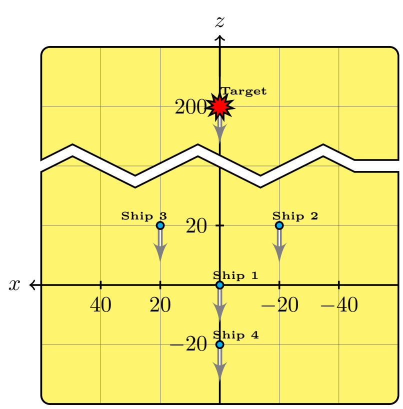
U.S. NAVAL RESEARCH LABORATORY

DIRECT FILTERING VERSUS MEASUREMENT CONVERSION

Filtering Versus Measurement Conversion

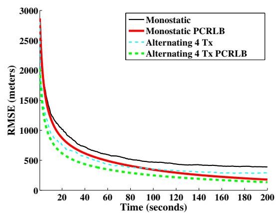
- Two common approaches for basic tracking exist:
  1. Cartesian converting measurements (and covariances) and using a linear filter.
  2. Directly using measurements in a nonlinear filter.
- These shall be compared in a simple example.

Filtering Versus Measurement Conversion

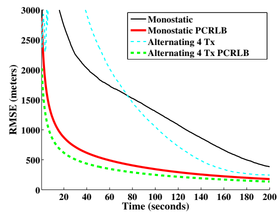


- A flat Earth.
- All ships on the surface traveling $-10\,\text{m/s}$ in the negative z direction.
- The target initially at an altitude of $7\,\text{km}$ going $100\,\text{m/s}$.
- Radars on ships pointed 15° up from the horizontal.
- $\tilde{q} = 0.4802\,\text{m}^2/\text{s}^3$
- Measurements every $T = 0.5\,\text{s}$.
- Tracks initialized via an information filter with 2 converted measurements.
- $\mathbf{R}^{\frac{1}{2}} = \text{diag}([10\,\text{m}, 10^{-2}, 10^{-2}])$.

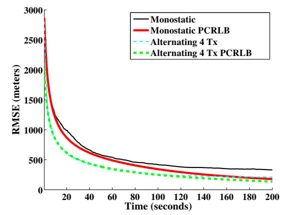
Filtering Versus Measurement Conversion: RMSE



(a) Converted Measurements, RMSE



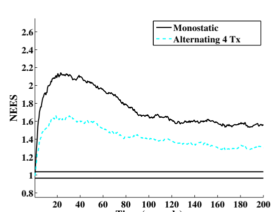
(b) EKF, RMSE



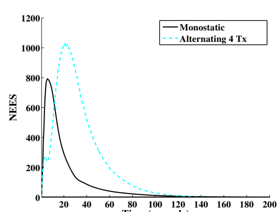
(c) CKF, RMSE

- The positional RMSE error of three different tracking algorithms. The CKF used 5th order points.
- The CKF has the best RMSE performance.

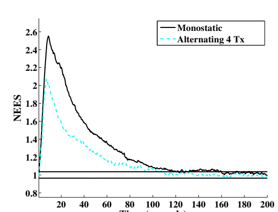
Filtering Versus Measurement Conversion: NEES



(a) Converted Measurements, NEES



(b) EKF, NEES



(c) CKF, NEES

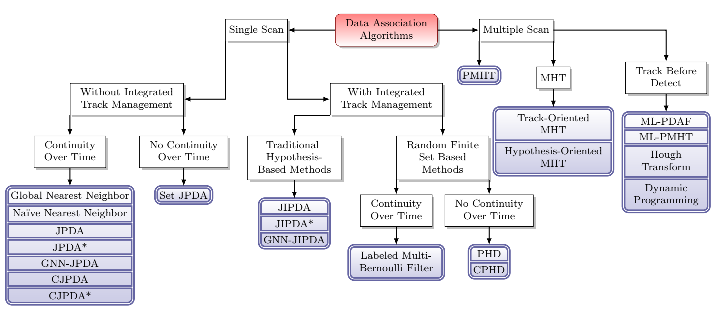
- The NEES of three different tracking algorithms.
- The EKF is bad; the CKF is the best over time; converted measurements are initially the best.

U.S. NAVAL RESEARCH LABORATORY

DATA ASSOCIATION

U.S. NAVAL RESEARCH LABORATORY

Data Association



- Common algorithms for assigning measurements to targets shown.
- We focus on non random finite set (RFS)-based single scan approaches.

Data Association

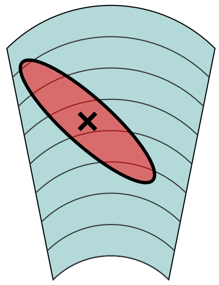
Topics considered are:

1. The Likelihood Function.
2. Naïve Nearest Neighbor, the Score Function, and Global Nearest Neighbor (GNN)
3. Probabilistic Data Association (PDA) and Joint Probabilistic Data Association (JPDA) variants

The Likelihood Function

- Consider one known target with a Gaussian prediction $\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$ with a 100% detection probability and with N_M measurements present.
- Which measurement should be assigned to the target?
- Single-scan data association algorithms make this decision based only on the current state prediction $\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$.
- Multiple scan data association look at multiple sets of measurements.

The Likelihood Function



- Let $\mathbf{H}^p$ be a matrix so $\mathbf{H}^p\mathbf{x}$ extracts the position components of a Cartesian state.
- Given Cartesian-converted measurements $\mathbf{z}_1^{\text{Cart}}, \ldots, \mathbf{z}_{N_M}^{\text{Cart}}$ one might assign the ith one such that

$$i = \arg\min_i \left\| \mathbf{H}^p\mathbf{x} - \mathbf{z}_i^{\text{Cart}} \right\|^2 \quad (21)$$

- This is usually bad:
  - Measurements are more accurate in range than cross range.
    - Cross-range becomes worse farther away from sensor, as illustrated (monostatic).
  - The shape of the uncertainty region of the state can matter.
    - Target ellipse crosses multiple range cells in image.

The Likelihood Function

- One cannot convert the state to the measurement coordinate system and use a similar l_2 norm.
  - Mixing units (e.g. range, angle, and even range rate) makes no sense.
- Valid distance measures can be derived from likelihood functions and likelihood ratios.
  - Another reason that measurement covariance matrices matter.
- Let $\mathbf{Z}^{k-1}$ be the set of all measurements up to discrete time $k-1$ and $\mathbf{\Theta}^{k-1}$ be the information of which measurements are assigned to the track up to time $k-1$.
- A valid cost function is the likelihood $p(\mathbf{z}|\mathbf{Z}^{k-1}, \mathbf{\Theta}^{k-1})$.

The Likelihood Function

- Written out, the likelihood of the ith measurement:

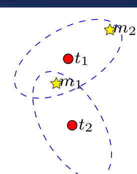
$$p(\mathbf{z}_i|\mathbf{Z}^{k-1},\boldsymbol{\Theta}^{k-1}) \triangleq \tilde{\Lambda}\left(\theta^i\right) = \left|2\pi\mathbf{P}^{zz,i}_{k|k-1}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right)'\left(\mathbf{P}^{zz,i}_{k|k-1}\right)^{-1}\left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right)} \quad (22)$$

- $\mathbf{P}^{zz,i}_{k|k-1}$ depends on the covariance matrix $\mathbf{R}_i$ of the ith measurement.
- Taking the negative logarithm of the likelihood and dropping the normalizing constant terms and $1/2$ scale factor one has a *Mahalanobis distance*:

$$-\log\left(\tilde{\Lambda}\left(\theta^i\right)\right) \propto \left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right)'\left(\mathbf{P}^{zz,i}_{k|k-1}\right)^{-1}\left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right) \quad (23)$$

- From the mathematics section, we know that Mahalanobis distances can be used for chi-squared testing to determine whether measurements can even be considered valid.
  - The exclusion of measurements from possible assignments is gating.

Naïve Nearest Neighbor



- For multiple targets, one is tempted to assign the highest likelihood measurement to each target.
- In the above scenario, both targets would be assigned to measurement m_1.
- Naïve nearest neighbor leads to track coalescence and ultimately, needless track loss.
- A practical algorithm must assign measurements jointly across targets, accounting for missed detections.
- Naïve nearest neighbor is one of the options in singleScanUpdate in the TCL.

The Score Function

- We want to derive a cost function (a score function) that can be used for multiple target assignment.
- The exponential of the score function derived in the unabridged slides here is computed in makeStandardLRMatHyps and makeStandardCartOnlyLRMatHyps in the TCL.

The Score Function

- Under many standard assumptions, the marginal change in the log-likelihood for assigning a measurement is

$$\Delta\Lambda_{t,i} = \begin{cases} \ln\left(P_D^t \frac{\mathcal{N}\left\{\mathbf{z}_i, \hat{\mathbf{z}}^t_{k|k-1}, \mathbf{P}^{zz,i,t}_{k|k-1}\right\}}{\lambda}\right) & \text{if } i \neq 0 \\ \ln(1-P_D^t) & \text{if } i = 0 \end{cases} \quad (24)$$

  - $\hat{\mathbf{z}}^t_{k|k-1}$ is the predicted measurement from the tth target,
  - $\mathbf{P}^{zz,i,t}_{k|k-1}$ is the innovation covariance the for ith measurement and tth target.
- The term $\Delta\Lambda_{t,i}$ is the marginal *score function* for single-frame assignment.
- Summing the marginals for a full target-measurement assignment, one forms the full score function $\Lambda(\theta)$ for a scan.

The Score Function

- When using a converted measurement filter, the units of $\mathcal{N}\left\{\mathbf{z}_i, \hat{\mathbf{z}}^t_{k|k-1}, \mathbf{P}^{zz,i,t}_{k|k-1}\right\}$ are in Cartesian coordinates, but the units of λ are usually in the radar's local coordinates.
- The proper conversion of λ to Cartesian coordinates yields a different λ at every point.
  - Cartesian λ is higher closer to the sensor.
- The Cartesian version of λ given λ in the measurement coordinate system is

$$\lambda_x = \frac{1}{|\mathbf{J}(\mathbf{y})|}\lambda_y \tag{25}$$

- In the TCL, necessary Jacobians are in "Coordinate Systems/Jacobians/Converted Jacobians" and include calcRuvConvJacob and calcPolarConvJacob, among others.

GNN Assignment

- One could assign measurements to targets and false alarms by choosing the assignment θ that maximizes the score function.
- How many valid assignments are there for m measurement and N_T targets?

$$N_{\text{hyp}} = \underbrace{\sum_{l=0}^{\min(m,N_T)}}_{\text{Sum over the number of targets observed}} \overbrace{\binom{N_T}{l}}^{\text{Choose which targets are observed}} \underbrace{\binom{m}{l}}_{\text{Choose which measurements are not false alarms}} \overbrace{l!}^{\text{Assign the measurements to the targets}} \tag{26}$$

- Suppose there are 3000 measurements and targets, and no false alarms or missed detections.
  - There are $3000! \approx 4.14 \times 10^{9130}$ hypotheses.
  - This is about one googol (10^{100}) raised to 91.3.

GNN Assignment

- There are $3000! \approx 4.14 \times 10^{9130}$ hypotheses, but only $3000^2 = 9 \times 10^6$ marginal hypotheses (values of $\Delta\Lambda_{t,i}$).
- The efficient solution is formulated as a GNN assignment (2D assignment) problem:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \sum_{i=1}^{N_R}\sum_{j=1}^{N_C} \Delta\Lambda_{i,j} x_{i,j} \tag{27}$$

$$\text{subject to } \sum_{j=1}^{N_C} x_{i,j} = 1 \quad \forall i \qquad \text{Every target is assigned to an event.} \tag{28}$$

$$\sum_{i=1}^{N_R} x_{i,j} \leq 1 \quad \forall j \qquad \text{Not every event is assigned to a target.} \tag{29}$$

$$x_{i,j} \in \{0,1\} \quad \forall x_{i,j} \qquad \text{Equivalent to } x_{i,j} \geq 0 \quad \forall x_{i,j} \tag{30}$$

- $N_R = N_T$ and $N_C = N_T + m$, number of measurements plus missed detection hypotheses.

GNN Assignment

- Each target gets its own missed detection hypotheses; costs for other targets' hypotheses are $-\infty$.
- To use the algorithm note that the cost matrix takes the form

$$\mathbf{C}_l \triangleq \begin{bmatrix} \overbrace{\begin{matrix}\Delta\Lambda_{1,1} & \dots & \Delta\Lambda_{1,m} \\ \Delta\Lambda_{2,1} & \dots & \Delta\Lambda_{2,m} \\ \vdots & \ddots & \vdots \\ \Delta\Lambda_{N_T,1} & \dots & \Delta\Lambda_{N_T,m}\end{matrix}}^{\text{Assignment Costs}} & \overbrace{\begin{matrix}\Delta\Lambda_{1,0} & -\infty & \dots & -\infty \\ -\infty & \Delta\Lambda_{2,0} & \dots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ -\infty & -\infty & \dots & \Delta\Lambda_{N_T,0}\end{matrix}}^{\text{Missed Detection Costs}} \end{bmatrix}. \tag{31}$$

- 2D assignment is a binary integer programming problem.
- A polynomial time solution is implemented as assign2D and kBest2DAssign in the TCL

The PDA and JPDA Algorithms

- The GNN algorithm is a maximum-likelihood approach.
- An alternative is to use the expected value over all possible target-measurement assignments.
- For a single target, the expected value and the covariance of the estimate are called *probabilistic data association* (PDA).
- For multiple targets, it is called Joint Probabilistic Data Association (JPDA).
- Variants of the PDA and JPDA are implemented in singleScanUpdate in the TCL.

The PDA and JPDA Algorithms

- For the tth target, the JPDA update is

$$\mathbf{x}^t_{k|k} = \mathrm{E}\left\{\mathbf{x}^t_k \left|\mathbf{Z}, I_p\right.\right\} = \sum_{i=0}^{m} \beta^{i,t}\hat{\mathbf{x}}^{t,i}_{k|k} \tag{32}$$

$$\mathbf{P}^t_{k|k} = \mathrm{E}\left\{\left(\mathbf{x}^t_k - \hat{\mathbf{x}}^t_{k|k}\right)\left(\mathbf{x}^t_k - \hat{\mathbf{x}}^t_{k|k}\right)'\middle|\mathbf{Z}, I_p\right\} \tag{33}$$

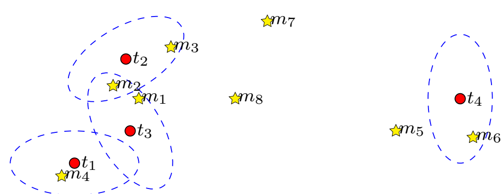
$$= \sum_{i=0}^{m} \beta^{i,t}\left(\mathbf{P}^{t,i}_{k|k} + \left(\mathbf{x}^{t,i}_k - \hat{\mathbf{x}}^t_{k|k}\right)\left(\mathbf{x}^{t,i}_k - \hat{\mathbf{x}}^t_{k|k}\right)'\right) \tag{34}$$

  - $\beta_{i,t}$ is the probability of assigning measurement i to target t (0 is a missed detection).
  - Superscripts of i and t indicate measurement and target hypotheses.
  - I_p is information on the (assumed Gaussian) prior estimates.
- The literature often uses a simpler expression for $\mathbf{P}^t_{k|k}$ that is not quadratic in form and subject to finite precision errors.

The PDA and JPDA Algorithms

- Assumptions going into the PDA/JPDA are that the prior distributions on all targets are Gaussian.
- The covariance cross terms between targets are not zero, but are omitted.
- The hardest part of the PDA/JPDA is the computation of the β values.

The PDA and JPDA Algorithms

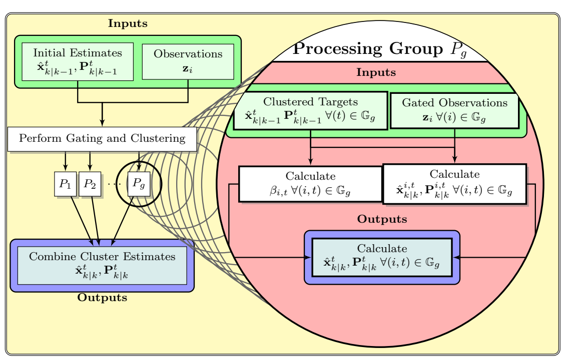


- Gating and clustering are important parts of a large-scale JPDA implementation.
- In the above figure, measurements are said to gate with a target if in the ellipse overlaps them.
  - In practice, use a chi-squared test on the Mahalanobis distance.
- There are three clusters of targets and measurements.
  1. Target t_1 is in a cluster with m_4.
  2. Targets t_2 and t_3 (linked by m_2) cluster with m_1, m_2, and m_3.
  3. Target t_4 is in a cluster with m_6.

The PDA and JPDA Algorithms: Gating and Clustering

- Brute-force gating and likelihood evaluation is implemented in the TCL via the makeStandardLRMatHyps and makeStandardCartOnlyLRMatHyps functions.
- Clustering can be computationally efficiently performed using disjoint sets, an obscure Computer Science data structure.
- Disjoint sets for clustering are implemented in the DisjointSetM and DisjointSet classes in the TCL; DisjointSet keeps track of only targets in clusters; DisjointSetM keeps track of targets and measurements in clusters.

The PDA and JPDA Algorithms: Gating and Clustering



- An illustration go how separate clusters can be processed independently.
- $\mathbb{G}_g$ is the set of targets and measurements in the gth cluster.

The PDA and JPDA Algorithms: Computing β

- When the β terms must be computed exactly, two approaches shall be considered:
  1. Via brute-force evaluation of all joint association events.
  2. Via matrix permanents.
- The matrix permanent approach is faster, but brute force is necessary to derive some JPDAF variants.

The PDA and JPDA Algorithms: Computing β

- Consider a matrix of likelihoods with $\Delta\tilde{\Lambda}_{t,i} = e^{\Delta\Lambda_{t,i}}$, non-normalized assignment probabilities:

$$\mathbf{C} \triangleq \begin{bmatrix} \overbrace{\begin{matrix} \Delta\tilde{\Lambda}_{1,1} & \dots & \Delta\tilde{\Lambda}_{1,m} \\ \Delta\tilde{\Lambda}_{2,1} & \dots & \Delta\tilde{\Lambda}_{2,m} \\ \vdots & \ddots & \vdots \\ \Delta\tilde{\Lambda}_{N_T,1} & \dots & \Delta\tilde{\Lambda}_{N_T,m} \end{matrix}}^{\tilde{\mathbf{C}}:\text{ Assignment Likelihoods}} & \overbrace{\begin{matrix} \Delta\tilde{\Lambda}_{1,0} & 0 & \dots & 0 \\ 0 & \Delta\tilde{\Lambda}_{2,0} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Delta\tilde{\Lambda}_{N_T,0} \end{matrix}}^{\text{Missed Detection Likelihoods}} \end{bmatrix} \tag{35}$$

- The normalized expression for the β terms can be rewritten directly in terms of likelihoods using elements of $\mathbf{C}$:

$$\beta_{j,k} = \Delta\tilde{\Lambda}_{j,k} \frac{\sum\limits_{\sigma\in\mathbb{P}^{N_T-1,N_T-1+m}} \prod_{\substack{n=1\\n\neq j}}^{N_T} c_{n,\sigma_n}}{\sum\limits_{\sigma\in\mathbb{P}^{N_T,N_T+m}} \prod_{n=1}^{N_T} c_{n,\sigma_n}} \tag{36}$$

The PDA and JPDA Algorithms: Computing β

- The expression simplifies to

$$\beta_{j,k} = \Delta\tilde{\Lambda}_{j,k} \frac{\text{perm}\left(\bar{\mathbf{C}}_{j,k}\right)}{\text{perm}\left(\mathbf{C}\right)} \tag{37}$$

  where $\bar{\mathbf{C}}_{j,k}$ is the matrix $\mathbf{C}$ after removing row j and column k.
- The matrix permanent cannot be evaluated in polynomial time unless P=NP.
- Efficient exponential complexity algorithms exist. In the TCL, the function perm implements an efficient algorithm.

The PDA and JPDA Algorithms

- Functions to explicitly compute the β values are implemented in the calc2DAssignmentProbs function in the TCL.
- Many techniques to approximate β values exist and are implemented in calc2DAssignmentProbsApprox in the TCL.
- Methods to do the complete PDA and JPDA update are given in singleScanUpdate in the TCL.
- However, one usually uses a variant of the JPDA algorithm rather than the JPDA algorithm itself.

The JPDA Algorithm: Coalescence

- Consider two targets whose states consist only of scalar position and have been stacked.
- Suppose that the joint PDF for the two targets is

$$p(\mathbf{x}) = \frac{1}{2}\delta\left(\mathbf{x} - \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) + \frac{1}{2}\delta\left(\mathbf{x} - \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) \tag{38}$$

- One target is located at $+1$ and one target is located at -1, but we do not know which.
- $\mathrm{E}\{\mathbf{x}\} = \mathbf{0}$, where no target is located.
  - Identity uncertainty causes track coalescence!
- Coalescence is not a "bias".
- Coalescence is the result of using the expected value given uncertain identity.

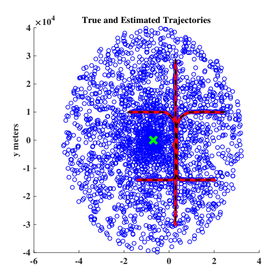
The JPDA Algorithm: Coalescence

- The Set JPDAF, the GNN-JPDA and the JPDA* can reduce coalescence.
- The GNN-JPDA is simple:
  1. Determine the measurement to use with a GNN filter, giving $\hat{\mathbf{x}}_{k|k}$.
  2. Compute $\mathbf{P}_{k|k}$ as in the JPDA, using the GNN estimate as the mean $\hat{\mathbf{x}}_{k|k}$.
- The hard assignment avoids coalescence.
- Computing $\mathbf{P}_{k|k}$ as a MSE matrix improves covariance consistency/reduces track loss.
- Available as an option in singleScanUpdate in the TCL with exact and approximate βs.

The JPDA Algorithm: Coalescence

- The brute-force computation of the βs had loops:
  1. Choose how many targets are observed.
  2. Choose which targets are observed.
  3. Choose which measurements originated from targets.
  4. Permute all associations of observed targets to target-originated measurements.
- The JPDA* is the same as the JPDA except in the innermost loop, only the maximum likelihood permutation is used.
  - Has the smoothing of the expected value.
  - The hard decision gets rid of identity uncertainty: Resistant to coalescence.
- Use calcStarBetasBF for the βs in the TCL. Available as an option in singleScanUpdate in the TCL.

The JPDA Algorithm: Example



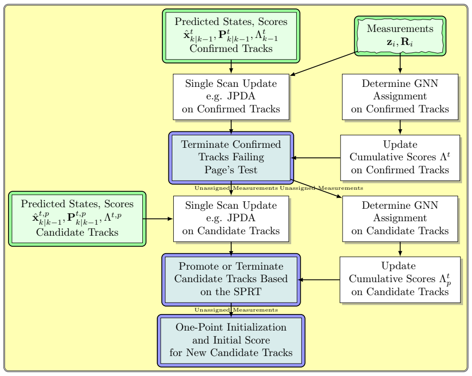
- A 2D example of the JPDA* including gating and clustering is given in demo2DDataAssociation in "Sample Code/Basic Tracking Example" in the TCL.
- A sample run is shown above. Tracks were started from two cued measurements.
- Estimated tracks: Red. True track: Dashed black. Detections: Blue. Very resistant to false alarms.

CASCADED LOGIC AND INTEGRATED TRACKERS

Cascaded Logic and Integrated Trackers

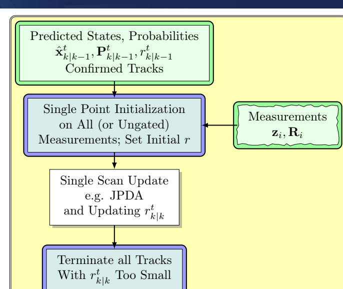
- The GNN and JPDA algorithms only update established tracks.
- Most practical systems require the ability to start and terminate tracks.
- Two main categories of algorithms exist for single-scan data association approaches:
  - Cascaded Logic Trackers
    - Confirmed-tracks, pre-tracks and hard decisions for initiation and termination.
  - Integrated Trackers
    - Lots of targets, each with a probability of existing.

A Cascaded Logic Tracker



- Multiple Types of cascaded logic trackers exist.
- There are confirmed tracks and candidate tracks.
  - Sometimes pre-tracks too.
- Scores usually updated via GNN assignments.
- Measurements not in GNN assignments go on to the next stage.
- Creation, promotion and deletion of tracks in purple-outlined boxes.

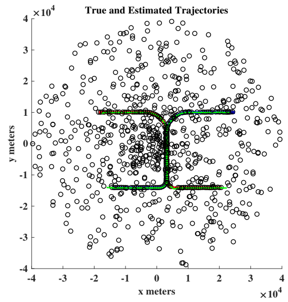
An Integrated Tracker



- Integrated trackers maintain a probability of target existence with each possible target.
- Usually, a track is not considered firm until its existence probability exceeds a threshold.
- A track is not terminated until its existence probability goes below a lower threshold.
- Measurement update implemented in the singleScanUpdateWithExistence function in the TCL.

U.S. NAVAL RESEARCH LABORATORY

An Integrated Tracker



- A rigorous derivation of the JIPDA class of filters is usually done using finite set statistics.
- A proper coverage of finite set statistics is beyond the scope of this presentation.

- An example of a minimal end-to-end GNN-JIPDAF in 2D is given in demo2DIntegratedDataAssociation in "Sample Code/Basic Tracking Examples" in the TCL.
- A plot of a run of the sample code with the detections and found tracks (green) and true tracks (red) is shown above for the simple two-target scenario.

U.S. NAVAL RESEARCH LABORATORY

SUMMARY

Summary I

- Gaussian approximations and Poisson clutter are widely used.
- Tracking algorithms need consistent measurement covariance matrices. Cross terms between range and range rate can matter.
- The Kalman filter comes from a Bayesian update of a linear dynamic model and a linear measurement.
- The EKF and CKF use Taylor series and cubature approximations to solve difficult integrals in an approximate nonlinear Kalman filter.
- Approaches to measurement conversion with consistent covariances include using Taylor series and cubature approximations to solve difficult integrals.

Summary II

- The GNN filter is a maximum likelihood filter for data association.
- The JPDA is an MMSE (expected value) filter for data association.
- One typically uses a variant of the JPDA, because the expected value is undesirable given target identity uncertainty.
- Cascaded logic and integrated additions to GNN and JPDA filter variants allow for track initiation and termination.
- Lots of free, commented Matlab code for tracking can be found at https://github.com/USNavalResearchLaboratory/TrackerComponentLibrary which is also http://www.trackercomponentlibrary.com