U.S. NAVAL RESEARCH LABORATORY

A Crash Course in Basic Single-Scan Target Tracking

David Frederic Crouse

Radar Division, Washington, DC

16 October 2017

U.S. NAVAL RESEARCH LABORATORY

INTRODUCTION

U.S. NAVAL RESEARCH LABORATORY

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.

U.S. NAVAL RESEARCH LABORATORY

What is Target Tracking?

Target Tracking Can Encompass:

1. Determining how many objects of interest are present given true/spurious measurements and missed detections.
   - Example measurements: Signal amplitude, range, angles, and range rate of detections, raw I and Q antenna outputs, time delay of arrival, received signal strength...
2. Creating a statistical representation of the state (position, velocity, dynamic model, etc.) of each object from measurements.
3. Quantifying the confidence in the target state.
4. Determining which measurements are from an object of interest.
5. Predicting what the object of interest might look like or do in the future.
6. Quantifying the confidence in predictions.

U.S. NAVAL RESEARCH LABORATORY

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.

U.S. NAVAL RESEARCH LABORATORY

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.

U.S. NAVAL RESEARCH LABORATORY

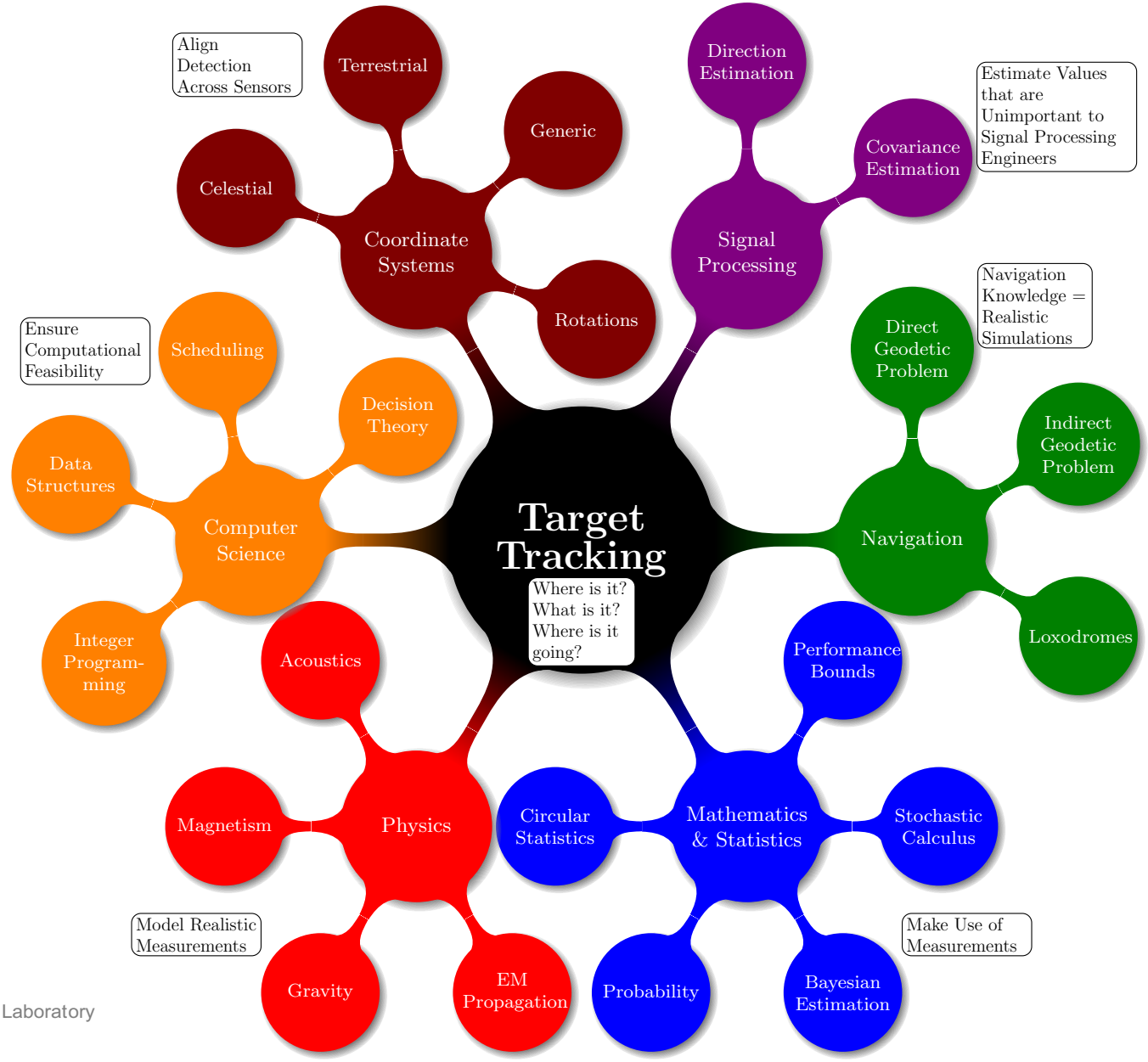
Target Tracking Vs. Automatic Target Tracking

- Before cheap, powerful computers:
  - Tracks plotted on a map by hand.
- Automatic target tracking: The computer...
  - Connects the dots.
  - Smooths the estimates.
  - Determines how many things are present.
  - Estimates properties that can't be plotted on a map.
- The term "automatic target tracking": Generally only used in old literature.

What Areas of Study Are Relevant?

U.S. NAVAL RESEARCH LABORATORY

What Areas of Study Are Relevant?



U.S. NAVAL RESEARCH LABORATORY

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

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.

U.S. NAVAL RESEARCH LABORATORY

Some Applications of Target Tracking

- Air traffic control.
- Satellite and debris collision avoidance.
- Analysis of particulate motion in particle accelerators.
- Analysis of the motion and interaction of bacteria.
- Range bin alignment and other aspects of ISAR for continual imaging of moving objects.
- Maritime domain awareness.
- Missile guidance and defense.
- Situational awareness for autonomous drone swarms.

U.S. NAVAL RESEARCH LABORATORY

Overview

1. Mathematical Preliminaries
2. Coordinate Systems
3. Measurements and Noise
4. Measurement Conversion
5. Bayes' Theorem and the Linear Kalman Filter Update
6. Stochastic Calculus and Linear Dynamic Models
7. The Linear Kalman Filter Prediction
8. Linear Initial State Estimation and the Information Filter
9. Nonlinear Measurement Updates
10. Square Root Filters
11. Direct Filtering Versus Measurement Conversion
12. Data Association
13. Integrated and Cascaded Logic Trackers
14. Dealing with Beams
15. Summary

U.S. NAVAL RESEARCH LABORATORY

MATHEMATICAL PRELIMINARIES

Mathematical Preliminaries

Useful Mathematical Tools

1. Univariate and Multivariate Taylor Series Expansions
2. Useful Probability Distributions.
3. Cubature Integration.
4. The Cramér-Rao Lower Bound

Taylor Series Expansions

- In Calculus II, one typically learns about Taylor and Maclaurin series expansions of continuous scalar functions:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{1}$$

$$f^{(n)}(a) \triangleq \left. \frac{df(x)}{dx^n} \right|_{x=a} \tag{2}$$

  - $a \neq$ is a Taylor series.
  - $a = 0$ is a Maclaurin series.
- Algorithms using multivariate Taylor series expansions often arise in tracking as alternatives to cubature techniques.

Taylor Series Expansions

- Arbitrary-order multivariate Taylor series expansions are very seldom given in textbooks.
- Consider a function of a d_x-dimensional variable $\mathbf{x}$:

$$F(t) \triangleq f(\mathbf{a} + t\mathbf{h}) \tag{3}$$

$$h \triangleq \mathbf{x} - \mathbf{a} \tag{4}$$

- Note $f(\mathbf{a}) = F(0)$ and $f(\mathbf{x}) = F(1)$
- The scalar Taylor series expansion of (3) at $x = 1$ is

$$F(1) = \sum_{n=0}^{\infty} \frac{F^{(n)}(0)}{n!} \tag{5}$$

where the chain rules gives

$$F^{(k)}(t) = \sum_{i_1=1}^{d_x} \cdots \sum_{i_k=1}^{d_x} \left(\frac{\partial^{\sum_{j=1}^{k} i_j}}{\prod_{j=1}^{k} \partial x_{i_j}} \right) f(\mathbf{a} + t\mathbf{h}) \prod_{j=1}^{k} h_{i_j} \tag{6}$$

Taylor Series Expansions

- After appropriate substitutions, one gets the full multivariate Taylor series expansion

$$f(\mathbf{x}) = \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{i_1=1}^{d_x} \cdots \sum_{i_k=1}^{d_x} \left(\frac{\partial^{\sum_{j=1}^{k} i_j}}{\prod_{j=1}^{k} \partial x_{i_j}} \right) f(\mathbf{a}) \prod_{j=1}^{k} \left(x_{i_j} - a_{i_j} \right) \tag{7}$$

- If $\mathbf{f}$ is a vector function, then just replace f with the vector notation $\mathbf{f}$.
- A first-order expansion of a vector function $\mathbf{f}$ can be written

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{a}) + \left(\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{a})' \right)' (\mathbf{x} - \mathbf{a}) \tag{8}$$

- Generally, only first and second order expansions are used in tracking applications.
- In the TCL, an arbitrary order multivariate polynomial for a Taylor series expansion can be obtained with the taylorSeriesPoly function.

Probability Distributions

- The four most prevalent probability distributions in target tracking tend to be:
  1. The Multivariate Gaussian Distribution.
  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.

U.S. NAVAL RESEARCH LABORATORY

Probability Distributions: Gaussian

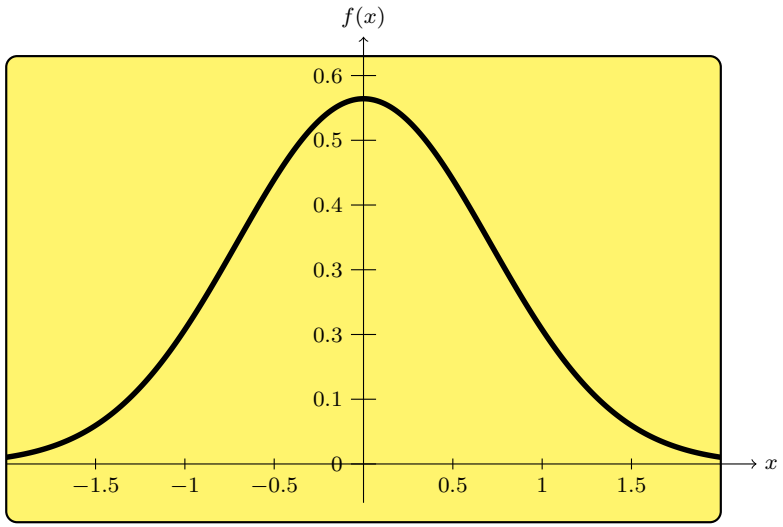
- Multivariate Gaussian distributions arise a lot: $\mathcal{N}\{\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$:

$$\mathcal{N}\{\mathbf{z}; \hat{\mathbf{z}}, \mathbf{R}\} = |2\pi\mathbf{R}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{z}-\hat{\mathbf{z}})'\mathbf{R}^{-1}(\mathbf{z}-\hat{\mathbf{z}})}. \tag{9}$$

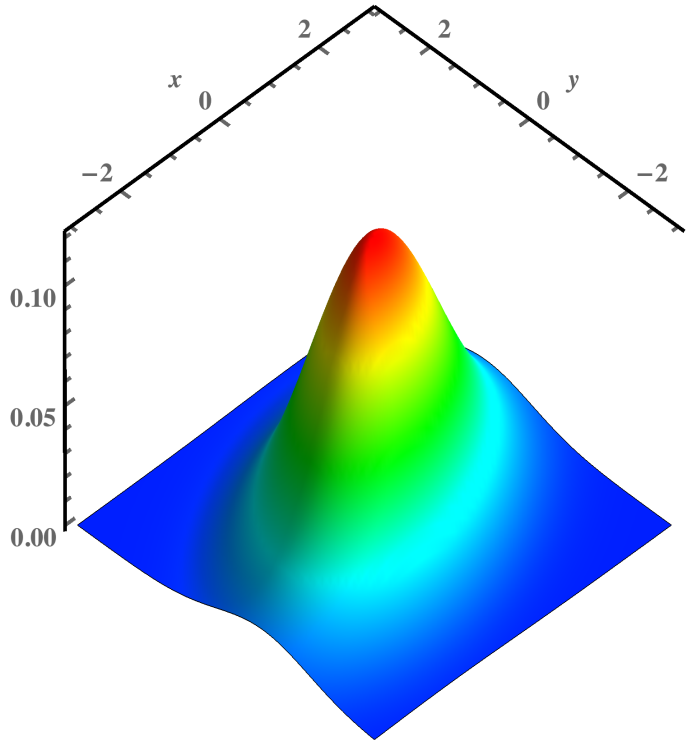
- Parameterized by mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
- Maximum entropy (most pessimistic) distribution when given only $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.
- Assuming noise is due to many small random contributions, a justification for the Gaussian approximation is from the central limit theorem.

  (The sum of N independent and identically distributed random variables approaches a Gaussian distribution as N gets large).
- Noise in *measurement domain* typically approximated Gaussian.
- Arises in certain areas with stochastic dynamic models.
- Is a *conjugate prior* distribution.

Probability Distributions: Gaussian



(a) Univariate



(b) Bivariate

- In (a), the Gaussian has mean 0 and variance $\sigma^2 = \frac{1}{2}$.
- In (b), the Gaussian has mean $\mathbf{0}$ and covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} 2 & 1/2 \\ 1/2 & 1 \end{bmatrix} \tag{10}$$

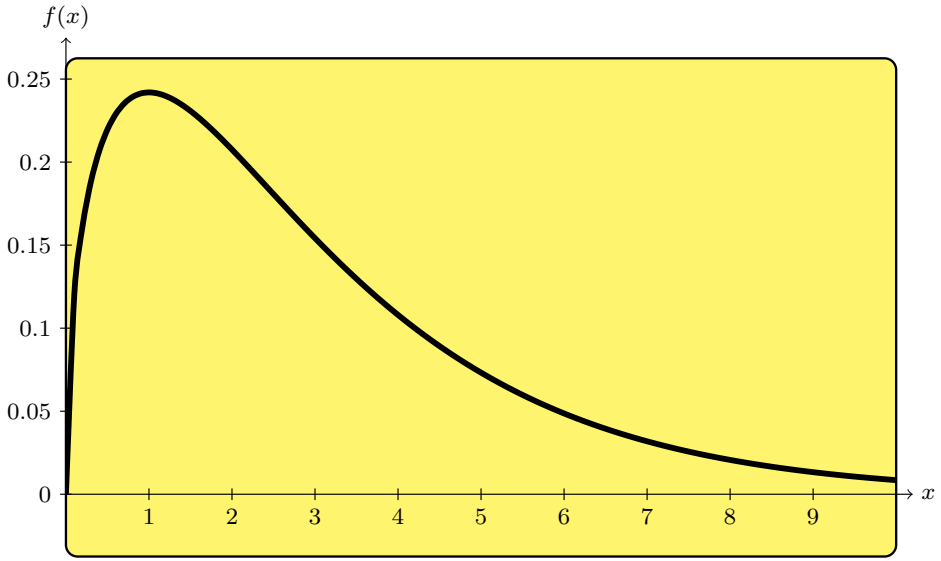
Probability Distributions: Chi-Squared

- The Mahalanobis distance of a sample $\hat{\mathbf{x}}$ is

$$D_M\left(\hat{\mathbf{x}}\right) \triangleq \left(\hat{\mathbf{x}} - \boldsymbol{\mu}\right)' \boldsymbol{\Sigma}^{-1} \left(\hat{\mathbf{x}} - \boldsymbol{\mu}\right) \tag{11}$$

- The Mahalanobis is *dimensionless*:
  - Can compare states having different units (e.g. position and velocity).
- The Mahalanobis distance shall be shown to arise in gating and is a part of the score function for data association.
- The Mahalanobis distance is used in assessing covariance realism via the normalized estimation error squared (NEES).
- If $\hat{\mathbf{x}}$ is drawn from a d_x-dimensional Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, then $D_M\left(\hat{\mathbf{x}}\right)$ is distributed central chi squared with d_x degrees of freedom, often written $\chi^2(d_x)$.

Probability Distributions: Chi-Squared



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

$$\chi^2(x, d_x) = \frac{x^{\frac{k}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{k}{2}} \Gamma\left(\frac{k}{2}\right)} \tag{12}$$

  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

$$(\hat{\mathbf{x}} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\hat{\mathbf{x}} - \boldsymbol{\mu}) < \gamma_p \tag{13}$$

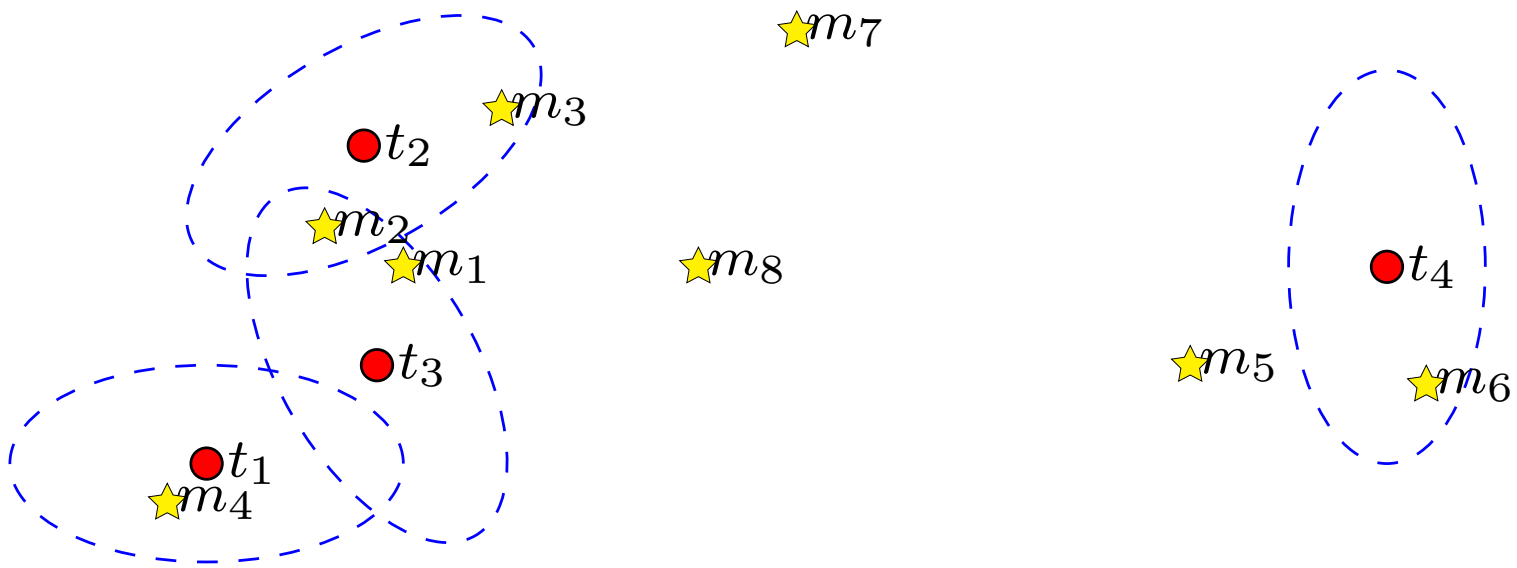
where γ_p depends on p and on d_x, the dimensionality of x.

	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



- Uncertainty regions are ellipses/ellipsoids.
- This type of test can be used for *gating*:
  - Gating: Eliminating targets/measurements for consideration for association to other targets/measurements.
  - Gating reduces computational complexity of measurement assignment.
- Above example: t for targets (with confidence ellipses) and m for measurements.

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.

$$\text{NEES} \triangleq \frac{1}{N d_x} \sum_{i=1}^{N} \left(\hat{\mathbf{x}}_i - \mathbf{x}_i\right) \mathbf{P}_i^{-1} \left(\hat{\mathbf{x}}_i - \mathbf{x}_i\right) \tag{14}$$

  - $\hat{\mathbf{x}}_i$ 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.

U.S. NAVAL RESEARCH LABORATORY

Probability Distributions: Chi-Squared

- The chi-squared test for covariance consistency with a confidence of p:
  - Is the NEES above the $p/2$ probability value of the chi-squared CDF?
  - Is the NEES below the upper $p/2$ probability value of the chi-squared CDF?
  - If so, then the covariance is consistent to p percent.
- Covariance consistency very important for data fusion, association, measurement filtering, and detecting errors in signal processing chain.
- In general, the NEES should be close to 1.

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.

$$\Pr\{n\} = \binom{N}{n} P_{\mathsf{FA}}^{n} \left(1 - P_{\mathsf{FA}}\right)^{N-n} \tag{15}$$

with mean

$$\tilde{\lambda} = N P_{\mathsf{FA}} \tag{16}$$

- The binomial distribution is almost never used in trackers.

Probability Distributions: Binomial

- We fix the mean $\tilde{\lambda}$ and vary N, which means that P_{FA} must change:

$$P_{\text{FA}} = \frac{\tilde{\lambda}}{N} \tag{17}$$

- The probability of n false alarms becomes

$$\Pr\{n\} = \binom{N}{n} \frac{\tilde{\lambda}}{N^n} \left(1 - \frac{\tilde{\lambda}}{N}\right)^{N-n} \tag{18}$$

- What is $\lim_{N\to\infty} \Pr\{n\}$?
- First note that

$$\lim_{N\to\infty} \binom{N}{n} \frac{\tilde{\lambda}}{N^n} = \lim_{N\to\infty} \frac{N^n + O(N^{n-1})}{n!} \frac{\tilde{\lambda}^n}{N^n} = \frac{\tilde{\lambda}^n}{n!} \tag{19}$$

- This leaves the term $\lim_{N\to\infty} \left(1 - \frac{\tilde{\lambda}}{N}\right)^{N-n}$

Probability Distributions: Binomial

- Rewrite the remaining term as

$$\lim_{N\to\infty}\left(1-\frac{\tilde{\lambda}}{N}\right)^{N-n}=\lim_{N\to\infty}\left(1-\frac{\tilde{\lambda}}{N}\right)^{N}\left(1-\frac{\tilde{\lambda}}{N}\right)^{-n} \tag{20}$$

- It can be seen that

$$\lim_{N\to\infty}\left(1-\frac{\tilde{\lambda}}{N}\right)^{-n}=1 \tag{21}$$

- The other term is written as a Taylor series expansion:

$$\lim_{N\to\infty}\left(1-\frac{\tilde{\lambda}}{N}\right)^{N}=\lim_{N\to\infty}\sum_{k=0}^{\infty}(-1)^k\frac{\tilde{\lambda}^k\prod_{j=0}^{k-1}(N-j)}{k!N^k} \tag{22}$$

$$=\lim_{N\to\infty}\sum_{k=0}^{\infty}(-1)^k\frac{\tilde{\lambda}^k(N^k+O(N^{k-1})}{k!N^k}=\sum_{k=0}^{\infty}(-1)^k\frac{\tilde{\lambda}^k}{k!}=e^{-\tilde{\lambda}} \tag{23}$$

- The final simplification is because the series is the Taylor series expansion of $e^{-\tilde{\lambda}}$.

Probability Distributions: Poisson

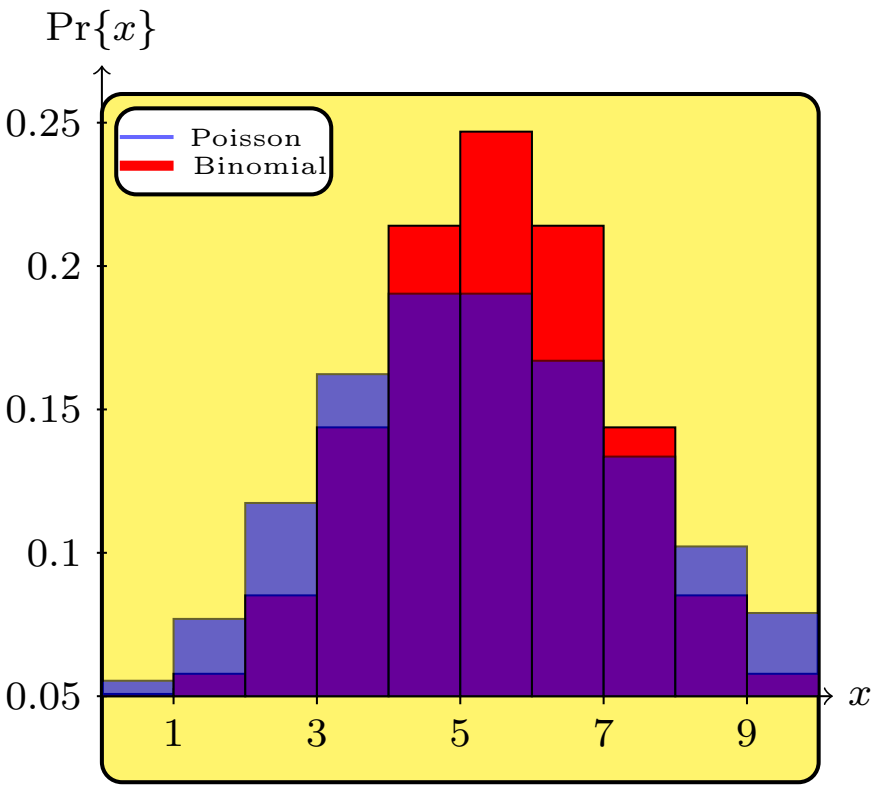
- The distribution resulting from holding $\tilde{\lambda}$ constant and taking $N \to \infty$ is thus Poisson:

$$\Pr\{n\} = e^{-\tilde{\lambda}} \frac{\tilde{\lambda}^n}{n!} \tag{24}$$

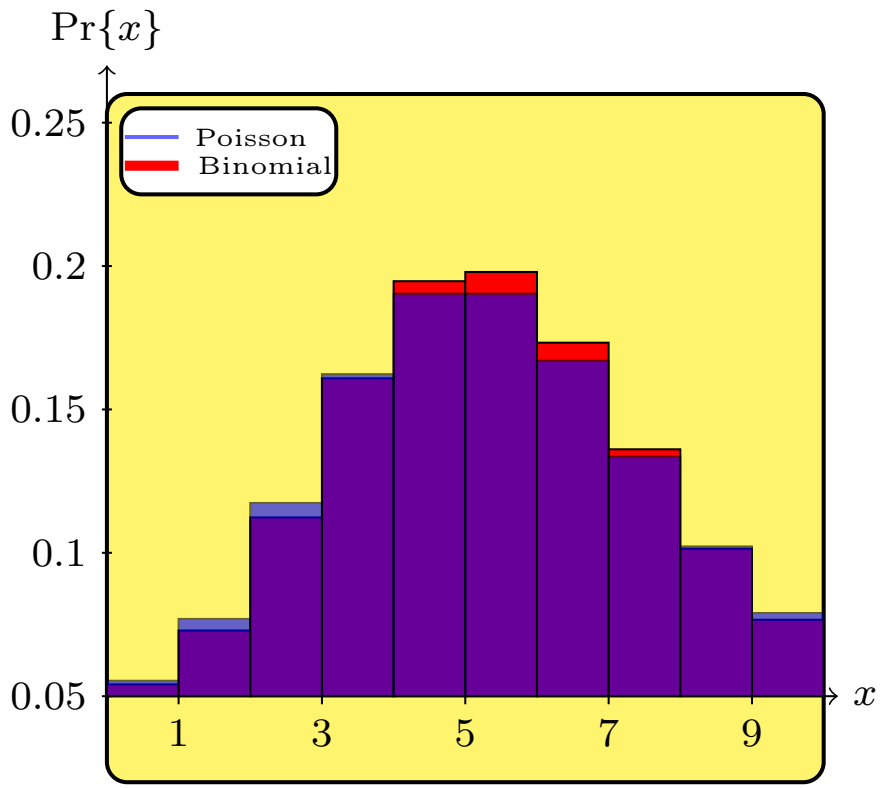
- Typically one splits $\tilde{\lambda} = \lambda V$.
- λ is a false alarm *density* and V is a volume.
  - For a range-range rate map, volume units might be $\mathrm{m}^2/\mathrm{s}^2$.
- Formulation of likelihood functions for target-measurement association simpler using Poisson approximation.
- Given λ, common likelihood formulations do not need to know V.

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

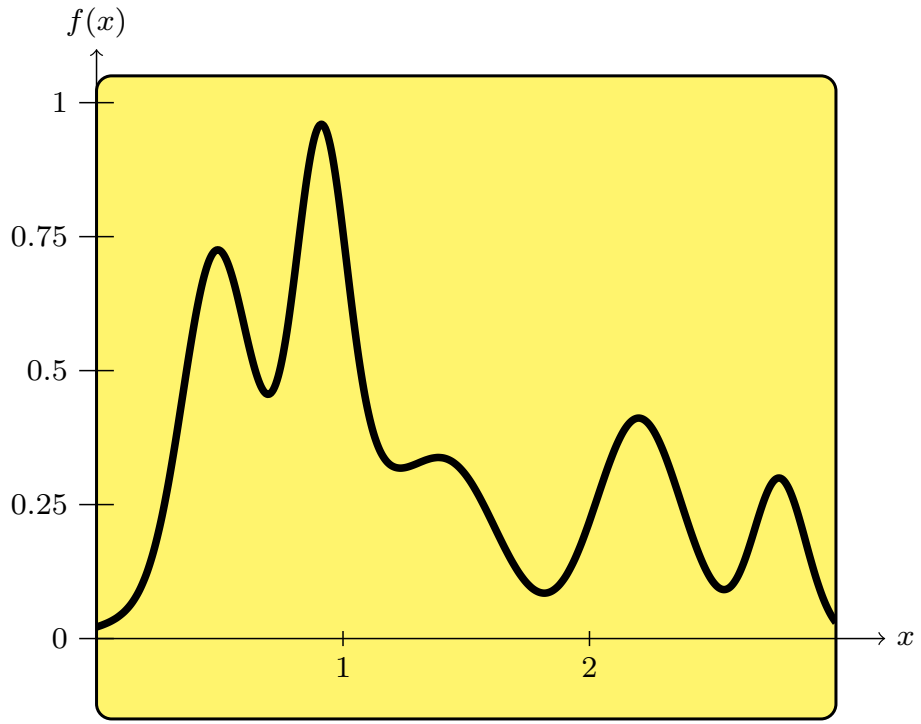
- Many integrals cannot be solved analytically with a finite number of terms.
  - Try to evaluate a Fresnel integral:

$$C(z) = \int_0^z \cos\left(\frac{\pi t^2}{2}\right) dt \tag{25}$$

- 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.

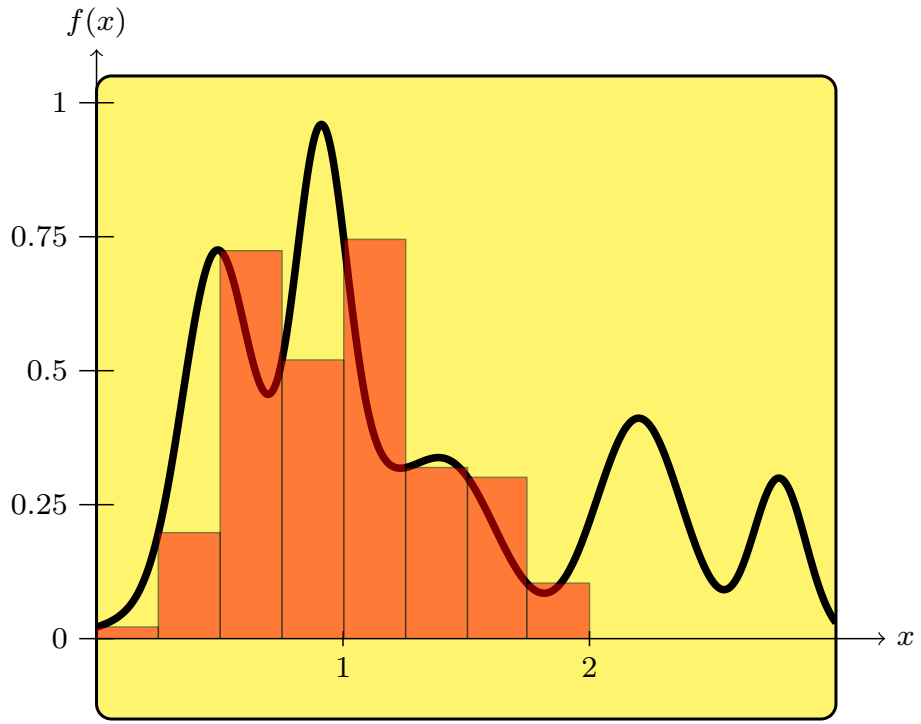


- Evaluate

$$\int_0^2 f(x)\,dx = ? \tag{26}$$

Cubature Integration: Why?

Numerically integrate the function from 0 to 2.



- Basic calculus solution: A Riemann sum:

$$\int_0^2 f(x)\,dx \approx \sum_{k=0}^{N-1} f\left(k\Delta_x\right)\Delta_x \qquad \text{where } 2 = N\Delta_x. \quad (27)$$

U.S. NAVAL RESEARCH LABORATORY

Cubature Integration: Why?

- Riemann sums converge very slowly.
- Finite precision errors for small Δ_x add up fast.
- Very inefficient in multiple dimensions.
- Not always practicable for:
  - Infinite integrals.
  - Integrals over weird shapes, such as a 4D hypersphere.
- Cubature integration produces *exact* solutions in some instances.
- Matlab's integral function uses adaptive quadrature.
- One can do much better than Matlab's function for certain classes of integrals.

Cubature Integration: Theory

- The idea in quadrature/cubature is the relation

$$\int_{\mathbf{x}\in\mathbb{S}} \mathbf{f}(\mathbf{x}) w(\mathbf{x})\, d\mathbf{x} = \sum_{i=0}^{N} \omega_i \mathbf{f}(\mathbf{x}_i), \tag{28}$$

  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 $N = 12$.

Cubature Integration: Theory

- In 1D, quadrature points can be systematically found.
- Let $w(x)$ be a weight function and $p_1(x), p_2(x), \ldots$ a sequence of polynomials orthonormal to $w(x)$:

$$\int_a^b w(x) p_m(x) p_n(x)\, dx = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

- All orthogonal polynomials can be expressed with a three-term recurrence:

$$p_n(x) = (a_n x + b_n)\, p_{n-1}(x) - c_n p_{n-1}(x) \tag{30}$$

  with $p_{-1}(x) = 0$, $p_0(x) = 1$, $a_n > 0$, and $c_n > 0$.
- One can often determine the pattern a_n, b_n, c_n using (29) and (30).

Cubature Integration: Theory

- The orthogonal polynomial $p_n(x) = k_n \prod_{i=1}^{n}(x - t_i)$ with $k_n > 0$ has n distinct real roots (t_i) between a and b.
- It has been proven that knowing p_n:

$$\omega_n = -\left(\frac{k_{N+1}}{k_N}\right) \frac{1}{p_{N+1}(t_n)\dot{p}_N(t_n)} \qquad n = 1, 2, \ldots, N \quad (31)$$

and the cubature points $\xi_n = t_n$, where

$$\dot{p}_n(t) \triangleq \left.\frac{p_n(x)}{dx}\right|_{x=t} \quad (32)$$

- Efficient methods of solving the problem utilizing matrix decompositions of the three-term recursion coefficients exist.
- 1D quadrature points can be extended into multiple dimensions using either a Gaussian product method, or the method of Smolyak (linCubPoints2MultiDim in the TCL).
- Specialized methods of generating cubature points also exist.

Cubature Integration

- For target tracking the most useful weighting function is

$$w(x) = \mathcal{N}\left\{\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right\} \tag{33}$$

- Points typically tabulated for a $\mathcal{N}\left\{\mathbf{0}, \boldsymbol{I}\right\}$ distribution.
  - Transform to $\mathcal{N}\left\{\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right\}$ as

$$\boldsymbol{\xi}_n \leftarrow \boldsymbol{\mu} + \boldsymbol{\Sigma}^{\frac{1}{2}} \boldsymbol{\xi}_n \tag{34}$$

    where $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\frac{1}{2}} \left(\boldsymbol{\Sigma}^{\frac{1}{2}}\right)'$ (lower-triangular Cholesky decomposition).

- Many formulae for cubature points are given in
  A. H. Stroud, Approximate Calculation of Multiple Integrals. Edgewood Cliffs, NJ: Prentice-Hall, Inc., 1971.

U.S. NAVAL RESEARCH LABORATORY

On Solving Integrals

- Many parts of target tracking involve solving difficult multivariate integrals.
- Many algorithms fall into one of two categories:
  1. Use cubature integral 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.
- Under certain conditions, the CRLB states

$$\mathrm{E}\left\{\left(\mathbf{x}-\mathbf{T}(\mathbf{z})\right)\left(\mathbf{x}-\mathbf{T}(\mathbf{z})\right)'\right\} \geq \mathbf{J}^{-1} \tag{35}$$

  - 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:

$$\mathbf{J}^{B} = -\mathrm{E}\left\{\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}'\ln\left(p(\mathbf{z}|\mathbf{x})\right)\right\} \tag{36}$$

$$= \mathrm{E}\left\{\left(\nabla_{\mathbf{x}}\ln\left(p(\mathbf{z}|\mathbf{x})\right)\right)\left(\nabla_{\mathbf{x}}\ln\left(p(\mathbf{z}|\mathbf{x})\right)\right)'\right\} \tag{37}$$

U.S. NAVAL RESEARCH LABORATORY

Using the CRLB

- An estimator is deemed "statistically efficient" if its accuracy achieves the CRLB.
- Most tracking problems exactly or approximately satisfy the assumed regularity conditions.
- The CRLB is an important tool in assessing the performance of an estimator.
- The CRLB can be used as an approximate measurement covariance matrix.
- The square root of the trace of the position components of the CRLB can be used as a lower bound on the position root mean squared error (RMSE) when localizing a target.
- Many common track filters are very close to or achieve the CRLB in simple situations.

Using the CRLB

- If $\mathbf{x}$ is not deterministic, take an expected value over $\mathbf{x}$ when computing the CRLB.
  - Equivalent to computing the CRLB with expectations over $p(\mathbf{x}, \mathbf{z})$ instead of $p(\mathbf{z}|\mathbf{x})$.
- In instances where the CRLB cannot be achieved, a bound called the the Bhattacharya bound is tighter.
- In the TCL, functions related to the CRLB (and Fisher information matrix) are in "Dynamic Estimation/Performance Prediction" as well as computePolyMeasFIM and directionOnlyStaticLocCRLB in "Static Estimation", and delayDopplerRateCRLBLFMApprox in "Mathematical Functions/Signal Processing."
- The proof of the multivariate CRLB is next, but is more for reference as it is long.

The CRLB: Long Proof

- The CRLB is derived assuming the regularity conditions:

$$\frac{\partial}{\partial x_i}\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})\,d\mathbf{z} = \int_{\mathbf{z}} \frac{\partial}{\partial x_i} p(\mathbf{z}|\mathbf{x})\,d\mathbf{z} \tag{38}$$

$$\frac{\partial}{\partial x_i}\int_{\mathbf{z}} T_i(\mathbf{z}) p(\mathbf{z}|\mathbf{x})\,d\mathbf{z} = \int_{\mathbf{z}} T_i(\mathbf{z}) \frac{\partial}{\partial x_i} p(\mathbf{z}|\mathbf{x})\,d\mathbf{z} \tag{39}$$

  where $T_i(\mathbf{z})$ is the unbiased estimator for the ith component of $\mathbf{x}$.

- The unbiasedness assumption of the estimator $T_i(\mathbf{z})$ means that

$$\mathrm{E}\left\{T_i(\mathbf{z}) - x_i\right\} = \int_{\mathbf{z}} \left(T_i(\mathbf{z}) - x_i\right) p(\mathbf{z}|\mathbf{x})\,d\mathbf{z} = 0 \qquad \forall i \in [1, d_x] \tag{40}$$

The CRLB's: Long Proof

- Differentiate both sides of the unbiasedness equation (using regularity conditions) with respect to x_i:

$$\int_{\mathbf{z}} (T_i(\mathbf{z}) - x_i) \frac{\partial}{\partial x_i} p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} - \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} = 0 \tag{41}$$

- Noting that a PDF integrates to 1:

$$\int_{\mathbf{z}} (T_i(\mathbf{z}) - x_i) \frac{\partial}{\partial x_i} p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} = 1 \tag{42}$$

- We now construct a substitution. Note that

$$\frac{\partial}{\partial x_i} \ln\left(p(\mathbf{z}|\mathbf{x})\right) = \frac{1}{p(\mathbf{z}|\mathbf{x})} \frac{\partial}{\partial x_i} p(\mathbf{z}|\mathbf{x}) \tag{43}$$

so

$$\frac{\partial}{\partial x_i} p(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x}) \frac{\partial}{\partial x_i} \ln\left(p(\mathbf{z}|\mathbf{x})\right) \tag{44}$$

The CRLB: Long Proof

- The substitution of the logarithm identity into the integral is

$$\int_{\mathbf{z}} \left(T_i(\mathbf{z}) - x_i\right) p(\mathbf{z}|\mathbf{x}) \frac{\partial}{\partial x_i} \ln\left(p(\mathbf{z}|\mathbf{x})\right)\, d\mathbf{z} = 1 \tag{45}$$

- However, consider the following integral with x_j for $j \neq i$:

$$\int_{\mathbf{z}} \left(T_i(\mathbf{z}) - x_i\right) p(\mathbf{z}|\mathbf{x}) \frac{\partial}{\partial x_j} \ln\left(p(\mathbf{z}|\mathbf{x})\right)\, d\mathbf{z} = \int_{\mathbf{z}} \left(T_i(\mathbf{z}) - x_i\right) \frac{\partial}{\partial x_j} p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} \tag{46}$$

$$= \frac{\partial}{\partial x_j} \int_{\mathbf{z}} T_i(\mathbf{z}) p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} - x_i \int_{\mathbf{z}} \frac{\partial}{\partial x_j} p(\mathbf{z}|\mathbf{x})\, d\mathbf{z} \tag{47}$$

$$= \frac{\partial}{\partial x_j} x_i = 0 \tag{48}$$

The CRLB: Long Proof

- Define the matrices:

$$\mathbf{Z} = \begin{bmatrix} T_1(\mathbf{z}) - x_1 \\ \vdots \\ T_{d_x}(\mathbf{z}) - x_{d_x} \\ \hline \frac{\partial}{\partial x_1} p(\mathbf{z}|\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_{d_x}} p(\mathbf{z}|\mathbf{x}) \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{Y}_1 \\ \hline \mathbf{Y}_2 \end{bmatrix} \tag{49}$$

- The regularity conditions say that

$$\mathrm{E}\{\mathbf{Z}\} = \mathbf{0} \tag{50}$$

The CRLB: Long Proof

- Given the regularity conditions:

$$\mathrm{Cov}\left\{\mathbf{Z}\right\} = \mathrm{E}\left\{\mathbf{Z}\mathbf{Z}'\right\} = \mathrm{E}\left\{\begin{bmatrix} \mathbf{Y}_1\mathbf{Y}_1' & \mathbf{Y}_1\mathbf{Y}_2' \\ \mathbf{Y}_2\mathbf{Y}_1' & \mathbf{Y}_2\mathbf{Y}_2' \end{bmatrix}\right\} \tag{51}$$

- Note that

$$\mathrm{E}\left\{\mathbf{Y}_1\mathbf{Y}_1'\right\} = \mathrm{Cov}\left\{\mathbf{T}(\mathbf{x})\right\} \tag{52}$$
$$\mathrm{E}\left\{\mathbf{Y}_1\mathbf{Y}_2'\right\} = \mathbf{I} \tag{53}$$
$$\mathrm{E}\left\{\mathbf{Y}_2\mathbf{Y}_2'\right\} = \mathbf{J} \tag{54}$$

where the identities with the logarithmic integrals led to (53) and $\mathbf{J}$ is the Fisher information matrix as given by

$$\mathbf{J} = \mathrm{E}\left\{\left(\nabla_{\mathbf{x}} \ln\left(p(\mathbf{z}|\mathbf{x})\right)\right)\left(\nabla_{\mathbf{x}} \ln\left(p(\mathbf{z}|\mathbf{x})\right)\right)'\right\} \tag{55}$$

The CRLB: Long Proof

- Simplifying

$$\mathrm{Cov}\left\{\mathbf{Z}\right\} = \begin{bmatrix} \mathrm{Cov}\left\{\mathbf{T}(\mathbf{x})\right\} & \mathbf{I} \\ \mathbf{I} & \mathbf{J} \end{bmatrix} \geq \mathbf{0} \tag{56}$$

where the inequality represents that the matrix has to be positive (semi-)definite.

- Use a matrix identity related to the Schur complement:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \tag{57}$$

with $\mathbf{A} = \mathrm{Cov}\left\{\mathbf{T}(\mathbf{x})\right\}$, $\mathbf{B} = \mathbf{I}$, $\mathbf{C} = \mathbf{I}$, $\mathbf{D} = \mathbf{J}$ to get $\mathrm{Cov}\left\{\mathbf{Z}\right\}$ as the leftmost matrix in (57).

The CRLB: Long Proof

- The matrix multiplying that is

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{J}^{-1} & \mathbf{0} \end{bmatrix} \tag{58}$$

- It is known that the product of a positive semidefinite matrix and a positive definite matrix is positive semidefinite, so one can write

$$\begin{bmatrix} \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathrm{Cov}\left\{\mathbf{T}(\mathbf{x})\right\} - \mathbf{J}^{-1} & \mathbf{I} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} \geq \mathbf{0} \tag{59}$$

- This finishes the proof that

$$\mathrm{Cov}\left\{\mathbf{T}(\mathbf{x})\right\} \geq \mathbf{J}^{-1} \tag{60}$$

- The proof of the equivalence to an expectation of second derivatives is omitted.

The CRLB and the PCRLB

- The Posterior CRLB (PCRLB) is a recursive formulation of the CRLB that is well suited for assessing recursive discrete-time estimation problems.
- A full proof of the PCRLB is given in:
  P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior Cramér- Rao bounds for discrete-time nonlinear filtering," IEEE Transactions on Signal Processing, vol. 46, no. 5, pp. 1386-1396, May 1998.
- Here, we use the results for a system with linear dynamics and additive noise of the form

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k \tag{61}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \tag{62}$$

where $\mathbf{v}_k \sim \mathcal{N}\{\mathbf{0}, \mathbf{Q}_k\}$ and $\mathbf{w}_k \sim \mathcal{N}\{\mathbf{0}, \mathbf{R}_k\}$.

The CRLB and the PCRLB

- The Fisher information matrix at time $k+1$ after a measurement update is

$$\mathbf{J}_{k+1} = \mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1}\mathbf{F}_k \left(\mathbf{J}_k + \mathbf{F}_k'\mathbf{Q}_k^{-1}\mathbf{F}_k\right)^{-1} \mathbf{F}_k'\mathbf{Q}_k^{-1} + \mathrm{E}\left[\left(\nabla_{\mathbf{x}_{k+1}} \mathbf{h}_{k+1}(\mathbf{x}_{k+1})'\right) \mathbf{R}_{k+1}^{-1} \left(\nabla_{\mathbf{x}_{k+1}} \mathbf{h}_{k+1}(\mathbf{x}_{k+1})'\right)'\right] \tag{63}$$

- The expectation is evaluated over all possible simulation tracks, often via Monte Carlo simulation.
- The initial $\mathbf{J}_0$ can be set to the CRLB for the first measurement.
- These expressions will be used later to evaluate estimator performance.

U.S. NAVAL RESEARCH LABORATORY

COORDINATE SYSTEMS

Coordinate Systems:

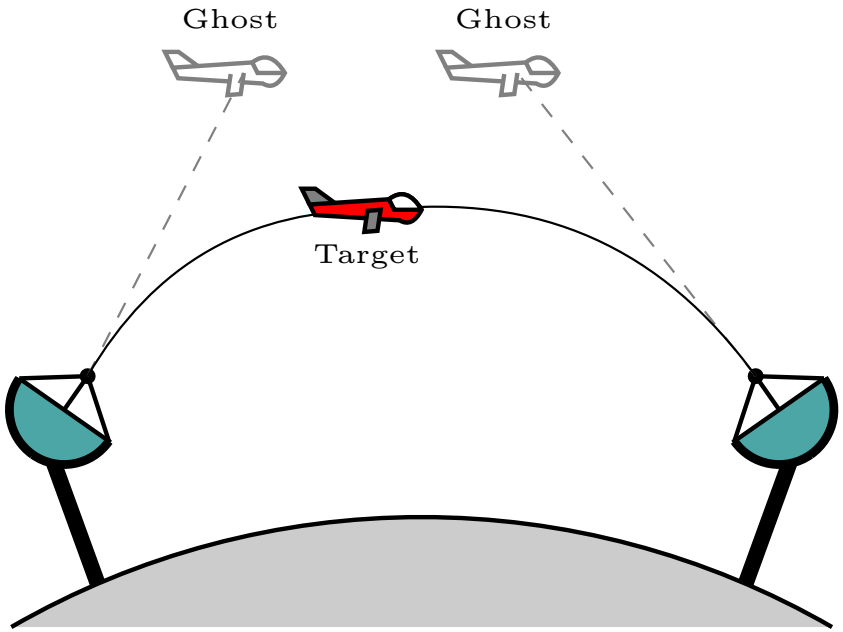
Classes of Coordinate Systems

1. Physical Coordinate Systems
   - Temporal and spatial quantities linked to global physical observations.
   - Examples: WGS-84, GCRS, ITRS, UTC.
2. Semi-Physical Coordinate Systems
   - Physical quantities linked to convenient mathematical models.
   - Examples: Latitude and longitude, geoid height, pressure altitude, UTM.
3. Purely Mathematical
   - Physical coordinate systems expressed in terms of mathematical.
   - Examples: Cartesian, spherical, ellipsoidal, r-u-v, polar.
4. Measurement Coordinate Systems
   - Local mathematical systems linked to physical quantities.
   - Systems corrupted by refraction and other physical effects.

Overview on coordinate systems given in:

D. F. Crouse, "An Overview of Major Terrestrial, Celestial, and Temporal Coordinate Systems for Target Tracking," Formal Report, Naval Research Laboratory, no. NRL/FR/5344–16-10,279, 10 Aug. 2016, 173 pages.

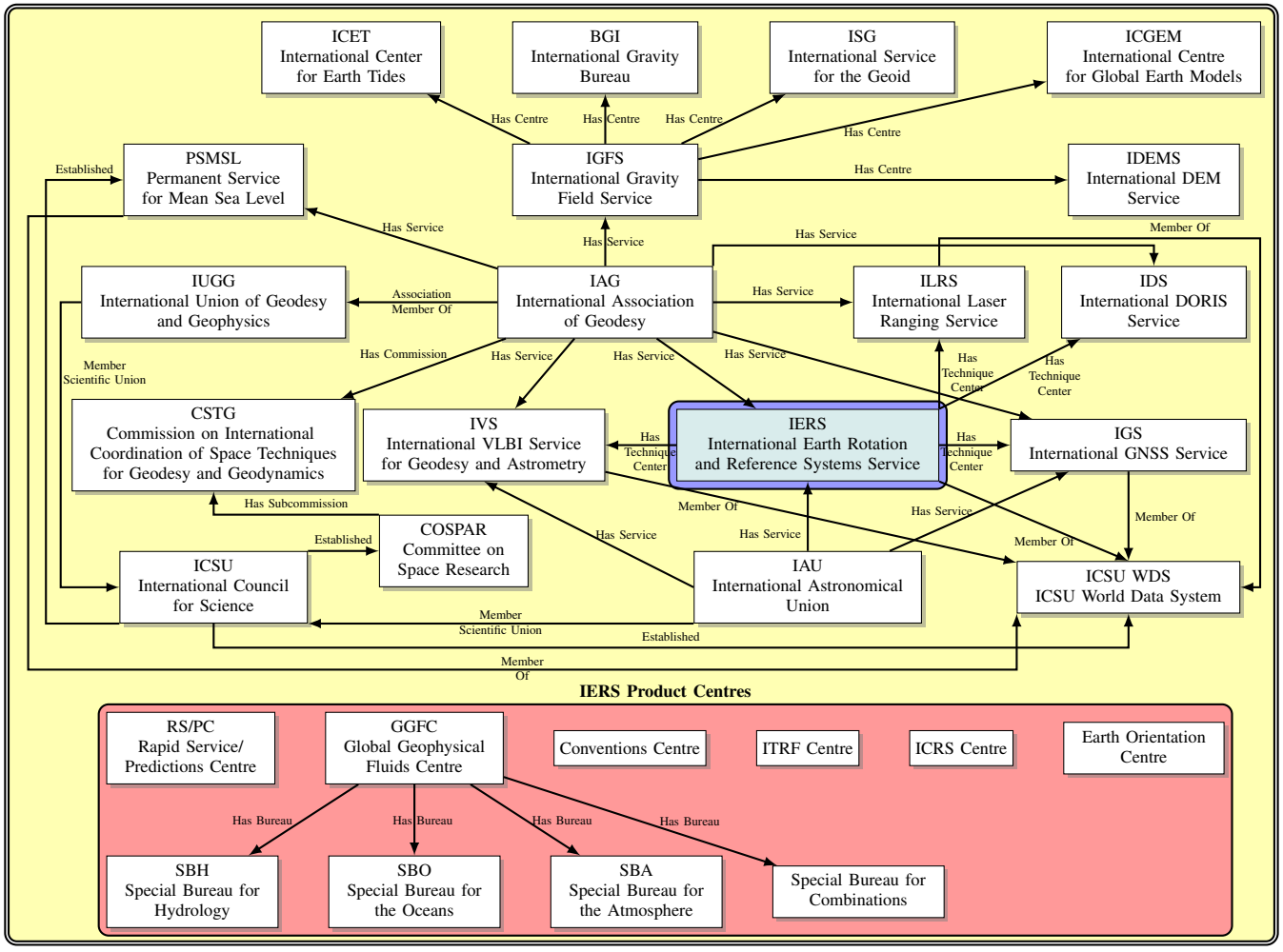
Physical Coordinate Systems 1



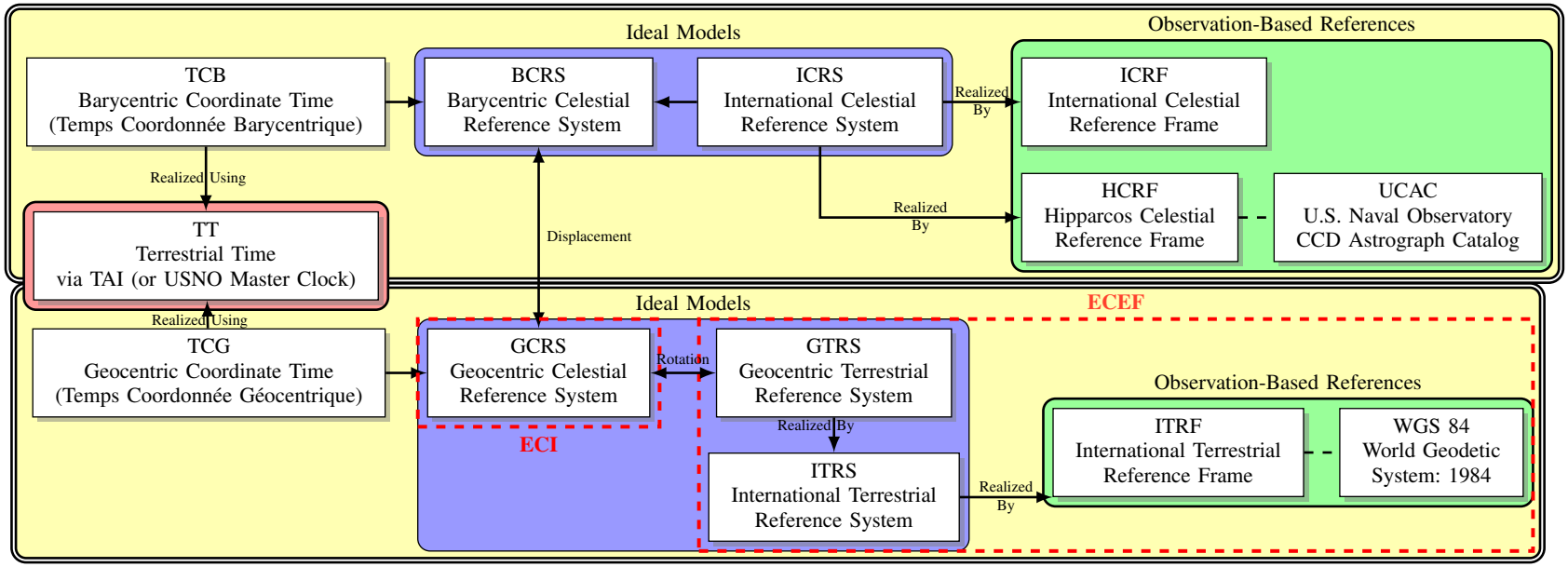
- Physical coordinate systems most important when using multiple sensors.
- Coordinate system establishment: "Sensor registration."
- Registration is not independent of measurement coordinate system effects.
- Poor registration/ignored measurement effects=Worse estimates: "Ghosting" of targets.

Physical Coordinate Systems 2

Internationally highly bureaucratically standardized



Physical Coordinate Systems 3



- Most useful systems are Earth-centered Earth-fixed' (for aircraft) and Earth-centered inertial (ECI) (for satellites).
- BCRS/ICRS best for tracking things far from Earth
  - For example, NASA and ESA's Solar & Heliospheric Observatory [SOHO] satellite.
- ECEF and ECI via rigorously defined physical systems.
- WGS-84 is used by the DoD and GPS.

Physical Coordinate Systems and the TCL

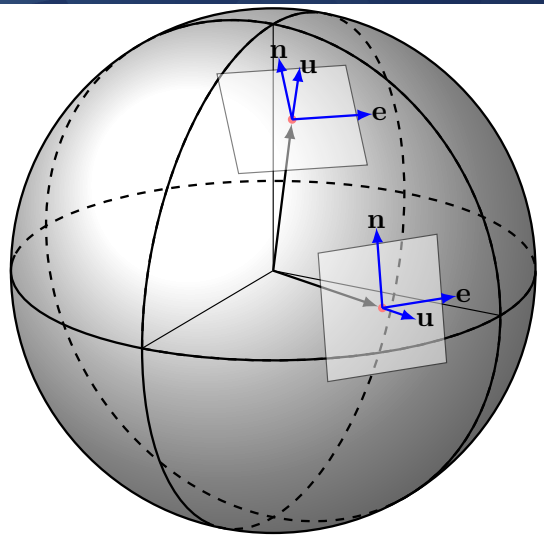
Terrestrial and Celestial Systems Conversions in the TCL					
CIRS2GCRS	CIRS2TIRS	ecliptic2ICRS	G2ICRS	GCRS2BCRS	GCRS2CIRS
GCRS2ITRS	GCRS2MOD	GCRS2TIRS	GCRS2TOD	ICRS2Ecliptic	ICRS2G
ICRS2J2000F	ITRS2GCRS	ITRS2TEME	ITRS2TIRS	J2000F2ICRS	MOD2GCRS
TEME2GCRS	TEME2ITRS	TIRS2CIRS	TIRS2GCRS	TIRS2ITRS	TOD2GCRS

Temporal Conversions in the TCL					
BesselEpoch2TDB	Cal2TAI	Cal2TDB	Cal2TT	Cal2UTC	GPS2TAI
GPS2TCG	GPS2TT	GPS2UTC	TAI2GAST	TAI2GMST	TAI2GPS
TAI2TCB	TAI2TCG	TAI2TDB	TAI2TT	TAI2UT1	TAI2UTC
TCB2TDB	TCG2GPS	TCG2TAI	TCG2TT	TCG2UTC	TDB2BesselEpoch
TDB2TCB	TDB2TT	TT2BesselEpoch	TT2Cal	TT2GAST	TT2GMST
TT2GPS	TT2LAST	TT2LAT	TT2LMST	TT2LMT	TT2TAI
TT2TCB	TT2TCG	TT2TDB	TT2UT1	TT2UTC	TT2YearFrac
UT12UTC	UTC2Cal	UTC2DayCount	UTC2GPS	UTC2TAI	UTC2TCB
UTC2TCG	UTC2TDB	UTC2TT	UTC2UT1	UTC2YearFrac	

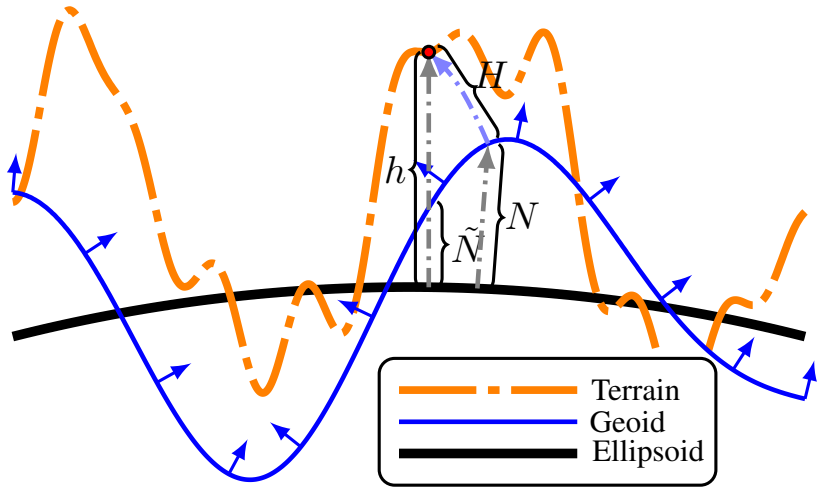
Temporal Helper Functions in the TCL			
numSecInYear	leapSecsOnDay	isLeapYear	dayOfWeek4Cal
getTimeZoneInfo	getTimeZoneList	cumLeapSec	JulDate2JulEpoch
JulEpoch2JulDate	fracDayOfMonth2HourMinSec	dayOfYear2MonthDay	

- See functions in TCL folder "Coordinate Systems/Celestial and Terrestrial Systems" and in "Coordinate Systems/Time", among others.

Semi-Physical Coordinate Systems I



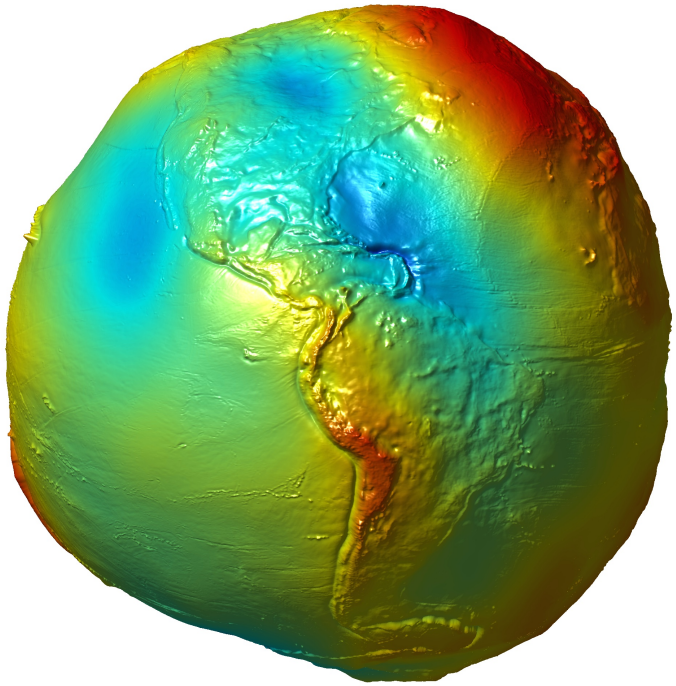
(a) ENU Axes



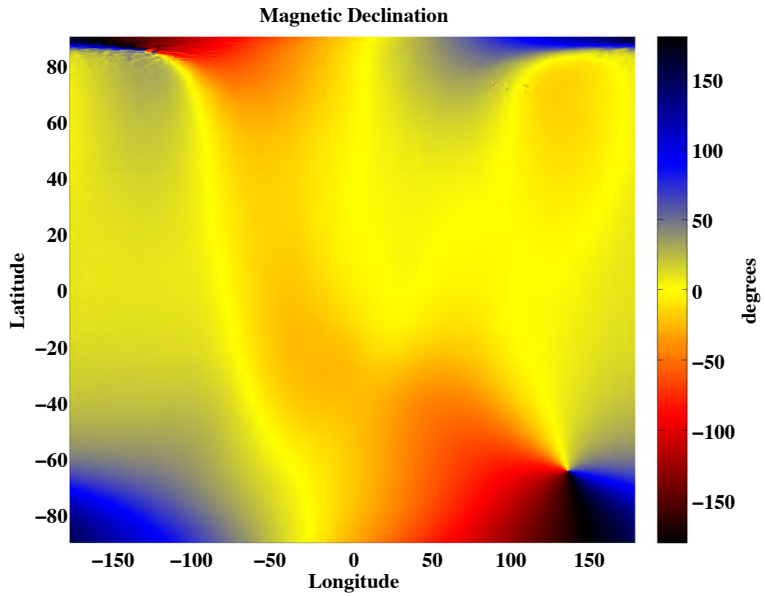
(b) Elevation and Geoid Height

- Ellipsoidal systems (latitude, longitude and altitude) as well as other maps (e.g. universal transverse mercator [UTM]).
- Local East-North-Up based on ideal ellipsoidal Earth.
- Elevation on maps often measured with respect to geoid, not reference ellipsoid.
- Ellipsoid height h differs from exact $(N + H)$ and approximate $(\tilde{N} + H)$ geoid (gravitational) height.

Semi-Physical Coordinate Systems II



(a) Exaggerated Earth Geoid



(b) Magnetic Field v.s True North

- Geoid: Theoretical surface of constant gravitational potential.
  - Highest geoid height: Top of Mt. Everest (Himalayas).
  - Farthest point from Earth's center: Top of Chimborazo (Andes).
- Magnetic coordinates considered semi-physical.

Semi-Physical Coordinate Systems and the TCL

Map-Related Functions in the TCL		
ellips2Cart.m	Cart2Ellipse	getENUAxes
geogHeading2uVec	uVec2GeogHeading	uDotEllipsoid
proj2Ellips	isInRefEllipsoid	uDotNumeric

Gravity-Related Cooridinate Functions in the TCL		
MSL2EllipseHelmert	ellips2MSLHelmert	getEGMGeoidHeight

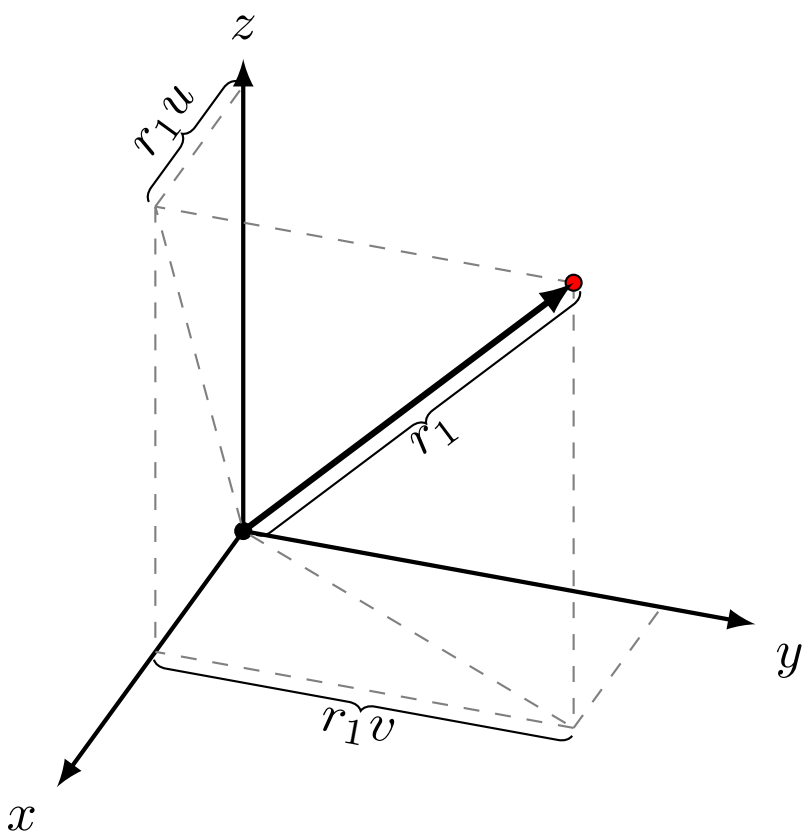
Magnetic-Related Conversions in the TCL		
CartCD2ITRS.m	geogHeading2Mag	ITRS2CartCD
ITRS2MagneticApex	ITRS2QD	magHeading2Geog
spherCD2SpherITRS	spherITRS2SpherCD	trace2EarthMagApex

Pressure-Related Conversions in the TCL	
orthoAlt4Pres	presTemp4OrthoAlt

- The above functions, and others, handle semi-physical systems in the TCL.

Mathematical Coordinate Systems

Common 3D Mathematical Systems: Monostatic r-u-v



- Monostatic range (one-way from receiver to the target) r_1 and direction cosines u and v (z-axis points in radar direction)

$$x = r_1 u, \tag{64}$$

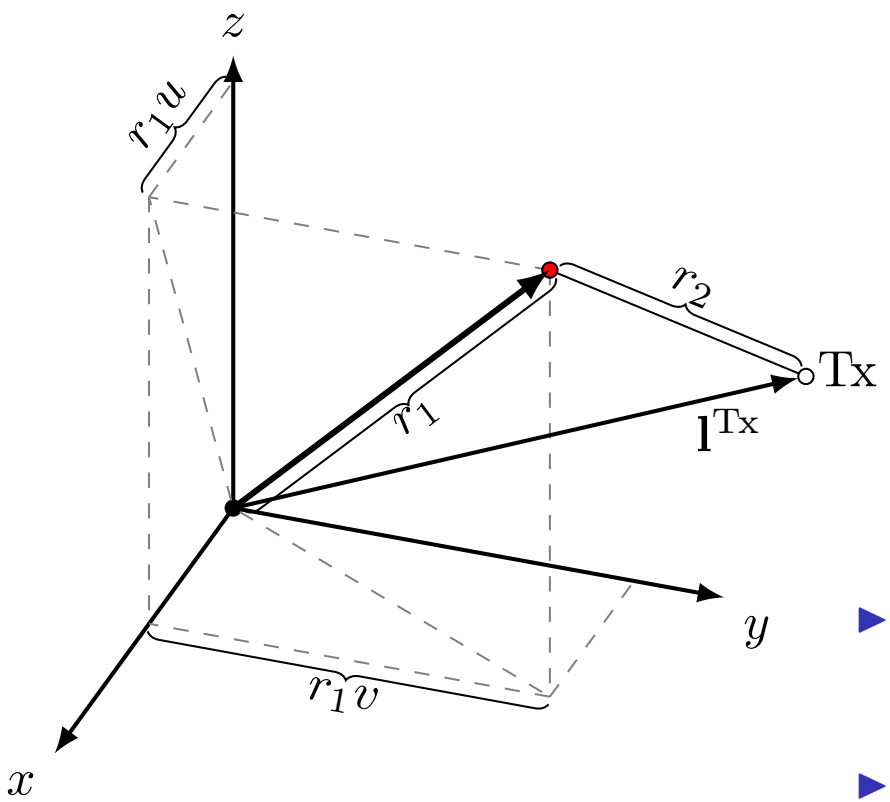
$$y = r_1 v, \tag{65}$$

$$z = r_1 \sqrt{1 - u^2 - v^2} \tag{66}$$

- Direction cosines tend to work better with stochastic models than spherical systems.
- See Cart2Ruv, ruv2Cart and ruv2Ruv among others in the TCL.

Mathematical Coordinate Systems

Common 3D Mathematical Systems: Bistatic r-u-v



- Bistatic range $r_B = r_1 + r_2$ and direction cosines u and v.

$$x = r_1 u, \tag{67}$$

$$y = r_1 v, \tag{68}$$

$$z = r_1 \sqrt{1 - u^2 - v^2} \tag{69}$$

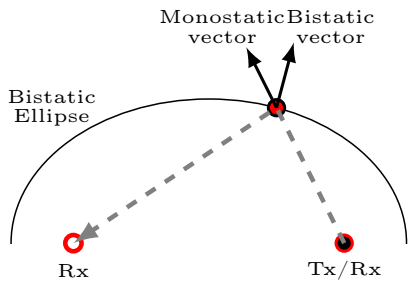
$$r_1 = \frac{r_B^2 - \left\|\mathbf{l}^{\mathrm{Tx}}\right\|^2}{2\left(r_B - \mathbf{u} \cdot \mathbf{l}^{\mathrm{Tx}}\right)}. \tag{70}$$

- $\mathbf{l}^{\mathrm{Tx}}$ is the transmitter location in the receiver's local coordinate system.
- A good system for the transmitter separated from the receiver.
- See Cart2Ruv, ruv2Cart and ruv2Ruv among others in the TCL.

Mathematical Coordinate Systems

Common Component: Range Rate

- In addition to measuring a position, velocity information is often available as a monostatic or bistatic range rate.



- Range rate provides a single component of a 3D velocity vector.
- Range rate is obtained from a Doppler shift measurement.
- See getRangeRate among others in the TCL.

Mathematical Coordinate Systems

- With $\mathbf{t}$ indicating a $3X1$ Cartesian position, the bistatic range is

$$r_B = \|\mathbf{t} - \mathbf{l}^{\mathrm{Rx}}\| + \|\mathbf{t} - \mathbf{l}^{\mathrm{Tx}}\| \tag{71}$$

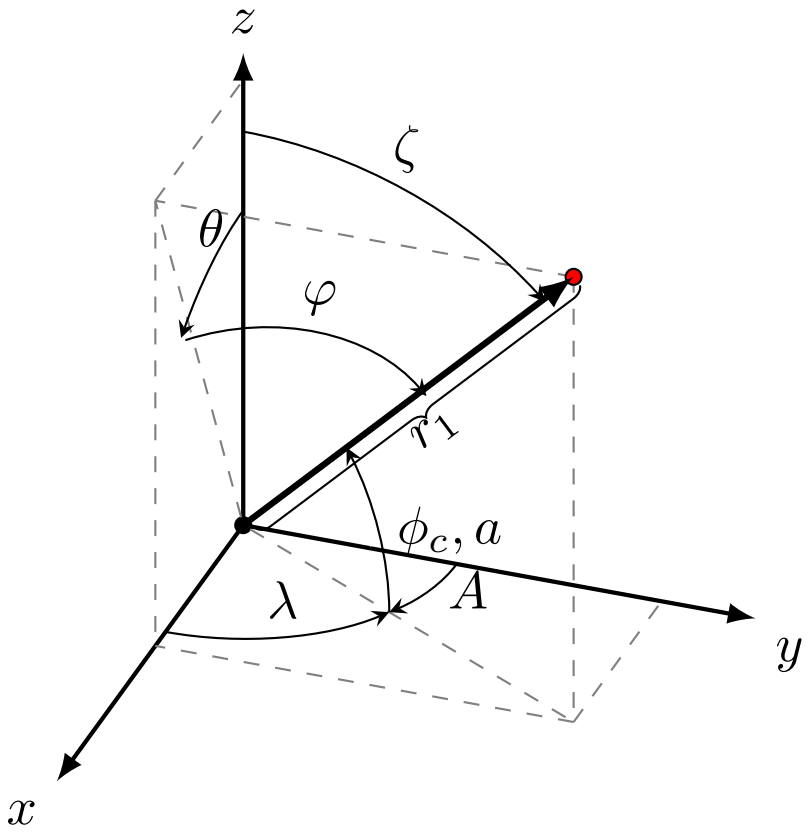
- Assuming Newtonian mechanics and no atmospheric refraction, the range rate is a straightforward derivative of range with respect to time:

$$\dot{r}_B \triangleq \frac{\partial r_B}{\partial \tau} = \frac{\left(\mathbf{t} - \mathbf{l}^{\mathrm{Rx}} + (\dot{\mathbf{t}} - \dot{\mathbf{l}}^{\mathrm{Rx}})\tau\right)'(\dot{\mathbf{t}} - \dot{\mathbf{l}}^{\mathrm{Rx}})}{\|\mathbf{t} - \mathbf{l}^{\mathrm{Rx}} + (\dot{\mathbf{t}} - \dot{\mathbf{l}}^{\mathrm{Rx}})\tau\|} + \frac{\left(\mathbf{t} - \mathbf{l}^{\mathrm{Tx}} + (\dot{\mathbf{t}} - \dot{\mathbf{l}}^{\mathrm{Tx}})\tau\right)'(\dot{\mathbf{t}} - \dot{\mathbf{l}}^{\mathrm{Tx}})}{\|\mathbf{t} - \mathbf{l}^{\mathrm{Tx}} + (\dot{\mathbf{t}} - \dot{\mathbf{l}}^{\mathrm{Tx}})\tau\|} \tag{72}$$

$$\dot{r}_B|_{\tau=0} = \left(\frac{\mathbf{t} - \mathbf{l}^{\mathrm{Rx}}}{\|\mathbf{t} - \mathbf{l}^{\mathrm{Rx}}\|} + \frac{\mathbf{t} - \mathbf{l}^{\mathrm{Tx}}}{\|\mathbf{t} - \mathbf{l}^{\mathrm{Tx}}\|}\right)'\dot{\mathbf{t}} - \left(\frac{\mathbf{t} - \mathbf{l}^{\mathrm{Rx}}}{\|\mathbf{t} - \mathbf{l}^{\mathrm{Rx}}\|}\right)'\dot{\mathbf{l}}^{\mathrm{Rx}} - \left(\frac{\mathbf{t} - \mathbf{l}^{\mathrm{Tx}}}{\|\mathbf{t} - \mathbf{l}^{\mathrm{Tx}}\|}\right)'\dot{\mathbf{l}}^{\mathrm{Tx}} \tag{73}$$

Mathematical Coordinate Systems

Common 3D Mathematical Systems: Spherical



- Multiple definitions (range, azimuth, elevation)

$$x = r_1 \cos(\lambda)\cos(\phi_c) \quad (74)$$

$$y = r_1 \sin(\lambda)\cos(\phi_c) \quad (75)$$

$$z = r_1 \sin(\phi_c) \quad (76)$$

or

$$x = r_1 \sin(\theta)\cos(\varphi) \quad (77)$$

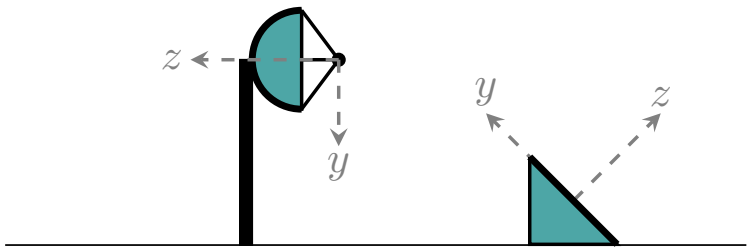
$$y = r_1 \sin(\varphi) \quad (78)$$

$$z = r_1 \cos(\theta)\cos(\varphi) \quad (79)$$

among others.

- See Cart2Sphere and spher2Cart among others in the TCL.

Mathematical Coordinate Systems: Rotations



- Local versus global coordinate system rotations.
- Textbooks devote many pages to transformations.
- Simplified by using
  S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 4, pp. 376-380, Apr. 1991.
  - Given two unit vectors in each system, obtain rotation matrix.
  - Implemented as findTransParam in the TCL.
  - Similar findRFTransParam to orient radar face with respect to reference ellipsoid.

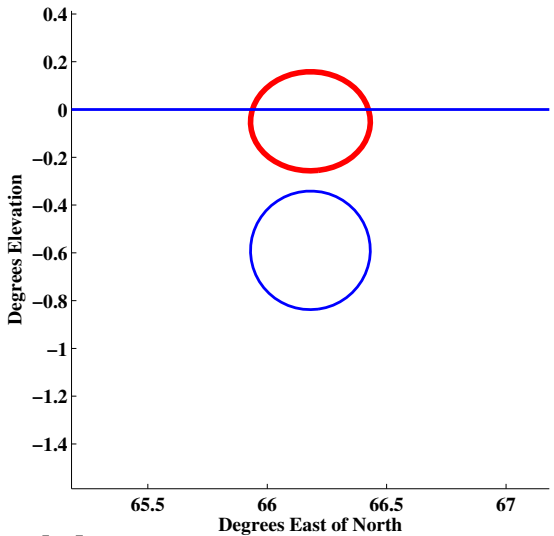
Mathematical Coordinate Systems and the TCL

Mathematical Coordinate System Functions in the TCL			
bipolar2Cart	Cart2Bipolar	Cart2EllipsHarmon	Cart2Hypersphere
Cart2Pol	Cart2Ru2D	Cart2Ruv	Cart2Sphere
ellips2Sphere	ellipsHarmon2Cart	findRFTransParam	findTransParam
getEllipsHarmAxes	getGlobalVectors	getLocalVectors	hypersphere2Cart
pol2Cart	polAng2U2D	ru2Cart2D	ruv2Cart
ruv2Ruv	spher2Cart	spher2Ellipse	spher2Sphere
spherAng2Uv	u2PolAng2D	uv2SpherAng	

Individual Components in the TCL			
getPolAngle	getRange	getRangeRate	getSpherAngle
getTDOA	getUDirection2D	getUVDirection	

- Many mathematical systems available.
- Many conversions bistatic.
- Jacobians and Hessians (not shown) available for systems commonly used in measurements.

Measurement Coordinate Systems



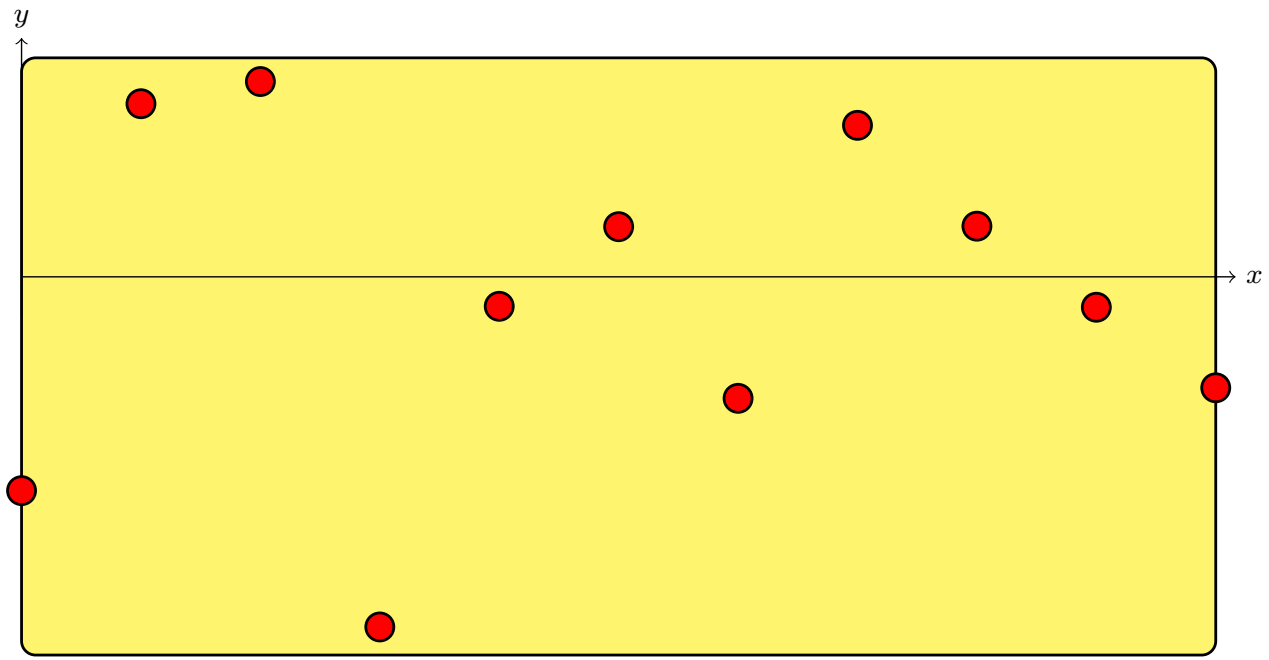
Sunrise near Hilo Hawaii, 1 June 2013. Standard refraction makes the sun over the horizon (red) while physically below the horizon (blue). Use readJPLEphem to read ephemeris data in the TCL.

- Measurements are corrupted from geometric quantities by physical effects: Atmospheric refraction, general relativity, etc.
  - Example: "Stationary" receivers move ± 15 cm during day due to solid-Earth tides.
- The GPS literature is a good source of very detailed physical perturbations.
- The TCL has a number of standard monostatic/bistatic refraction corrections in "Atmospheric Models/Standard Exponential Model."

U.S. NAVAL RESEARCH LABORATORY

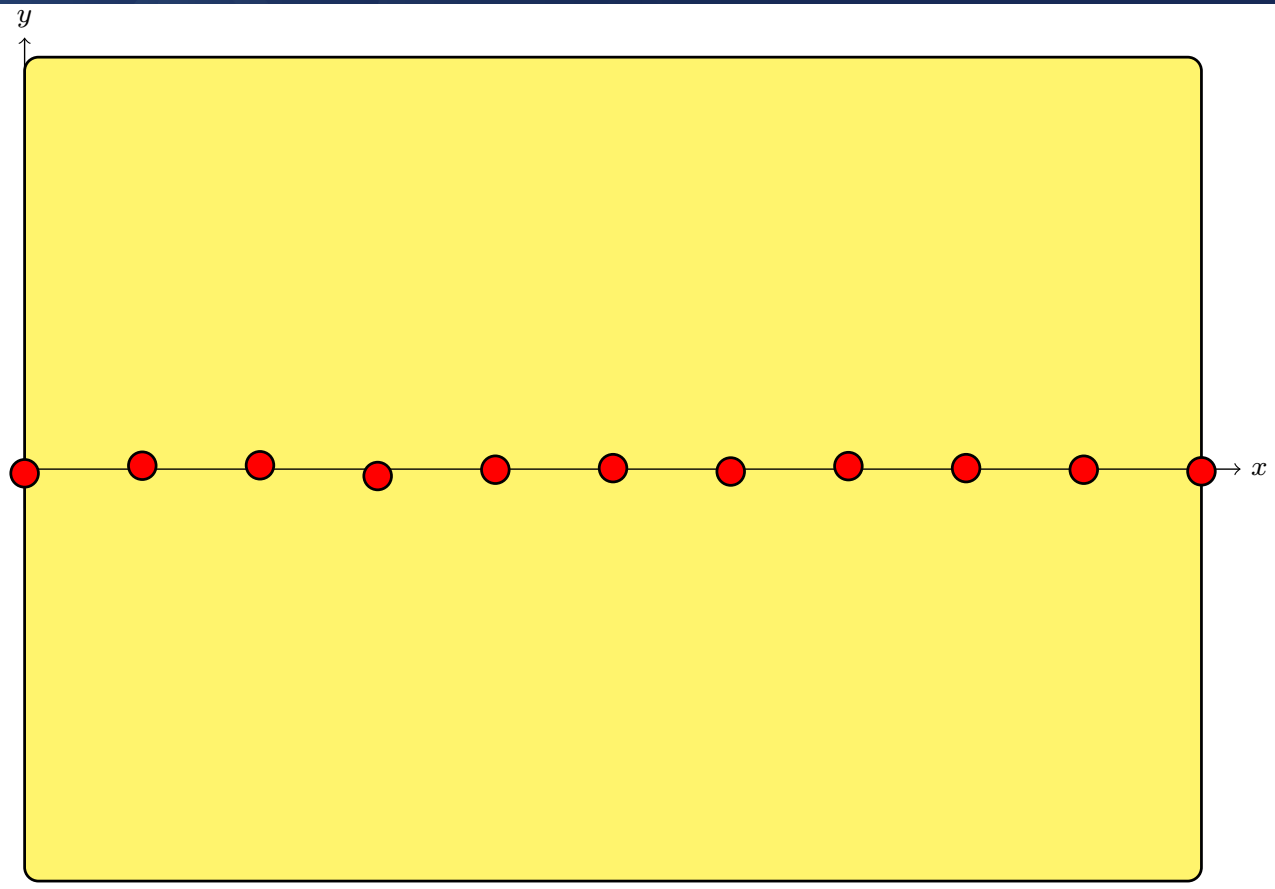
MEASUREMENTS AND NOISE

Measurements and Noise



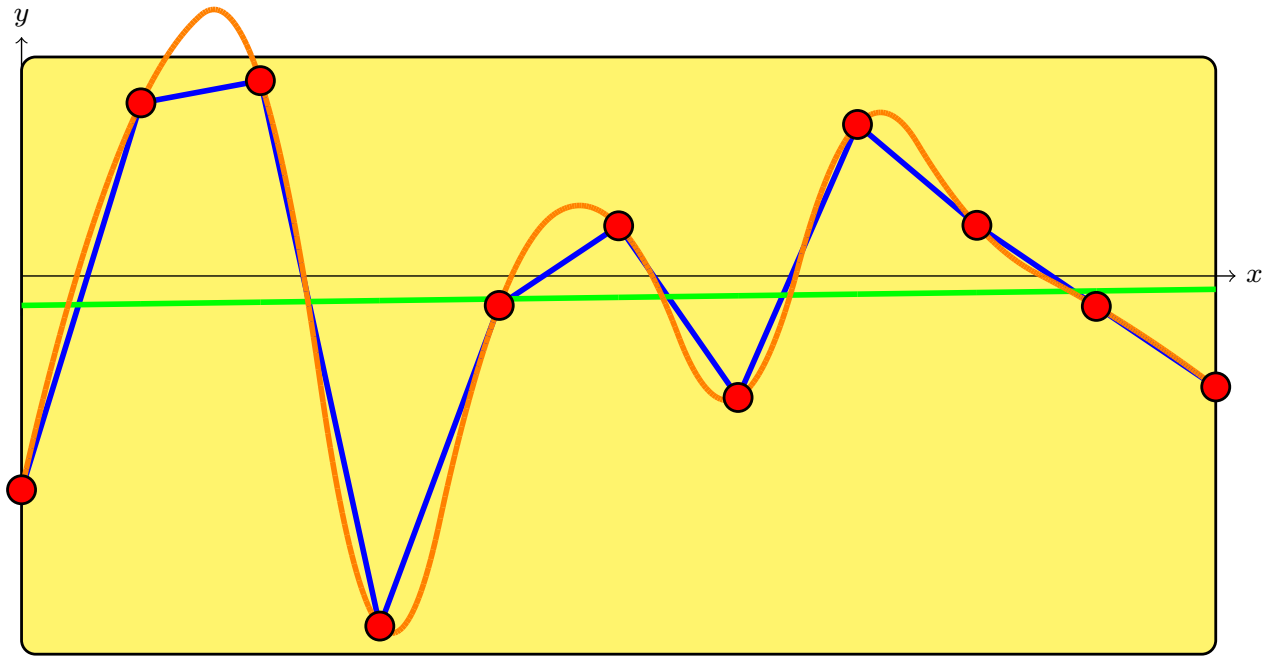
- Are these points false alarms or a possible track over time?
- Are they accurate measurements that are far apart?
- Are false alarms very unlikely or highly likely?

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.

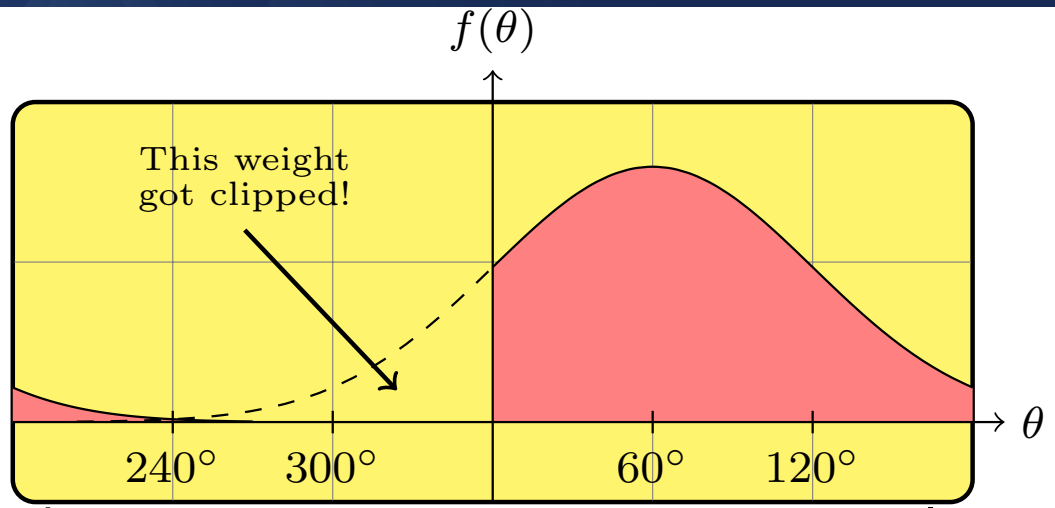
Measurements and Noise

- Information on measurement accuracy is important for
  - Discerning false alarms from tracks.
  - Estimating a target state more accurately than "connecting the dots".
- The noisy nature of the measurements necessitates the use of statistics to accurately estimate a target state.
- The noise is often approximated as additive zero-mean multivariate Gaussian in the local coordinate system of the measurement. For a measurement $\hat{\mathbf{z}}$ the true value $\mathbf{z}$ is thus distributed

$$\mathcal{N}\{\mathbf{z}; \hat{\mathbf{z}}, \mathbf{R}\} = |2\pi\mathbf{R}|^{-\frac{1}{2}} \, e^{-\frac{1}{2}(\mathbf{z}-\hat{\mathbf{z}})'\mathbf{R}^{-1}(\mathbf{z}-\hat{\mathbf{z}})}. \tag{80}$$

where $\mathbf{R}$ is the covariance matrix of the measurement.

Measurements and Noise: Circular Statistics

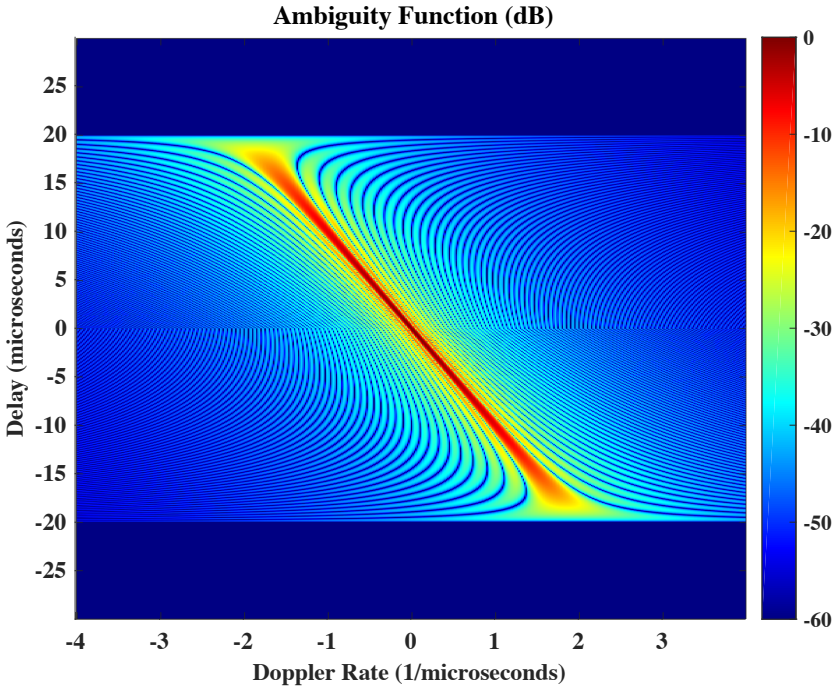


- Issues with Gaussian approximations on angular quantities.
- Solutions:
  - One could use circular statistics (often inconvenient).
  - One could wrap the distribution (not always feasible).
- Example wrapping solution:
  D. F. Crouse, "Cubature/ Unscented/ Sigma Point Kalman Filtering with Angular Measurement Models," in The 18th International Conference on Information Fusion, Washington, D.C., July 2015.
- The TCL has circular distributions and the wrapRange, angCircDiff, and wrap2Sphere functions.
  wrapping functions.

Measurements and Noise: Signal Processing

- Designers of tracking algorithms should be familiar with some signal processing because:
  - Detection algorithm designers seldom derive a covariance $\mathbf{R}$ (or other statistics) for their algorithms.
  - When $\mathbf{R}$ is given, it is often lacking cross terms (e.g. between range and range rate) even when the waveform leads to a true coupling.
  - Range and/or range-rate ambiguity can sometimes be solved directly in a multihypothesis tracker (MHT).
  - Understanding the measurement signal processing informs about when zero mean/Gaussian approximations beak down.

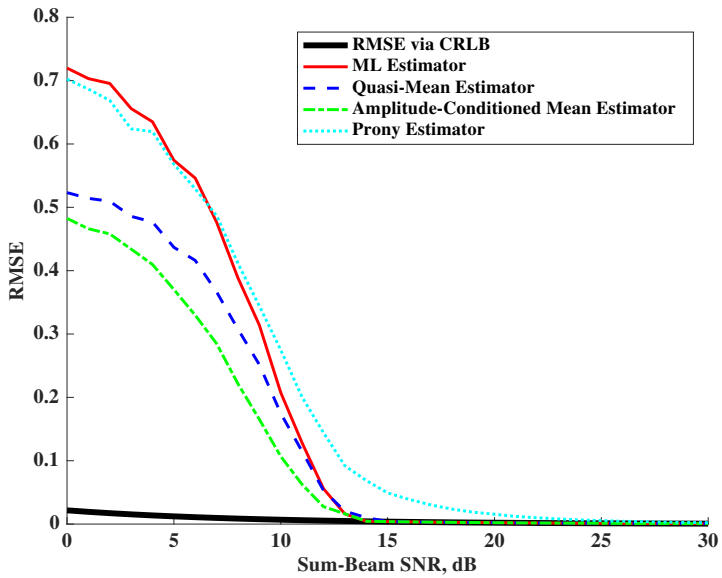
Measurements and Noise: Cross Correlations

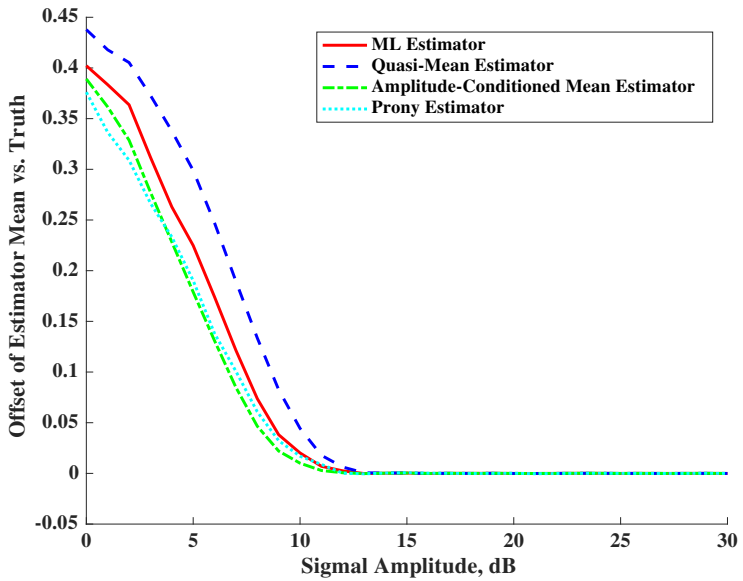


- The wideband ambiguity function for a 2 MHz linearly frequency modulated (LFM) chip lasting $20, \mu$s
- Detections will most likely be on the ridge.
- Errors in range $\rightarrow$ errors in range rate (Doppler) along the ridge $\rightarrow$ correlations between range and range-rate measurement components.

Measurements and Noise: Directional Estimation

Limits of Common Assumptions: Directional Estimation





- Rule of thumb for linear arrays: For direction cosine estimates below $13\,\mathrm{dB}$ signal to noise ratio (SNR), "biases" arise, measurement distribution decreasingly Gaussian, variance estimators from the Cramér-Rao Lower Bound (CRLB) are poor.

Measurements and Noise: Directional Estimation

- A statistically unbiased estimator of $\mathbf{x}$ given $\mathbf{z}$ is the expected value $\mathrm{E}\{\mathbf{x}\,|\mathbf{z}\}$.
- Suppose that one observes many independent trials $\mathbf{z}_1, \ldots, \mathbf{z}_n$, so n is large and the true value of $\mathbf{x}$ is $\mathbf{x}_{\text{true}}$
- One usually expects that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}\{\mathbf{x}\,|\mathbf{z}_i\} \to \mathrm{E}\{\mathbf{x}\,|\mathbf{z}_1, \ldots, \mathbf{z}_n\} = \mathbf{x}_{\text{true}} \qquad (81)$$

- That is not always true.
- Counterexample:
  - A linear array produces estimates of $u \in (-1, 1)$.
  - If the true value is at or near -1, noise will always make $\mathrm{E}\{\mathbf{x}\,|\mathbf{z}_i\} > -1$ unless the distribution is a delta function at -1
  - The mean of $n \to \infty$ numbers that are all > -1 will be > -1.

Measurements and Noise: Directional Estimation

- Though $\mathrm{E}\{\mathbf{x}\,|\mathbf{z}_i\}$ is unbiased in the strict statistical sense, it can be "biased" in how one expects it to function.
- Algorithms utilizing Gaussian approximations implicitly assume that that the average of multiple independent estimates approaches the true value.
- Directional estimates with low SNR and/or that are near the edges of the valid estimation region violate common approximations.
- Gaussian approximations ignoring such effects might work until the revisit rate becomes very fast.
  - Estimator's predicted error (via covariance matrix) approaches/ goes below magnitude of the "bias".

U.S. NAVAL RESEARCH LABORATORY

Measurements and Noise: Directional Estimation

- Understanding the limits of the signal processing origins of measurements allows one to choose more appropriate target tracking methods.
  - Traditional methods, such as the Kalman filter, become unreliable.
  - More sophisticated/ computationally-demanding techniques, such as particle filters, might become necessary.

U.S. NAVAL RESEARCH LABORATORY

MEASUREMENT CONVERSION

Measurement Conversion

- Cartesian conversion of noisy measurements is sometimes necessary.
- Gaussian measurement will be non-Gaussian in Cartesian coordinates.
  - Given the measurement function $\mathbf{z} = \mathbf{h}(\mathbf{t})$ over the range of $\mathbf{z}$ and the distribution of $\mathbf{z}$ is $p(\mathbf{z})$ then the exact PDF of the converted measurement is

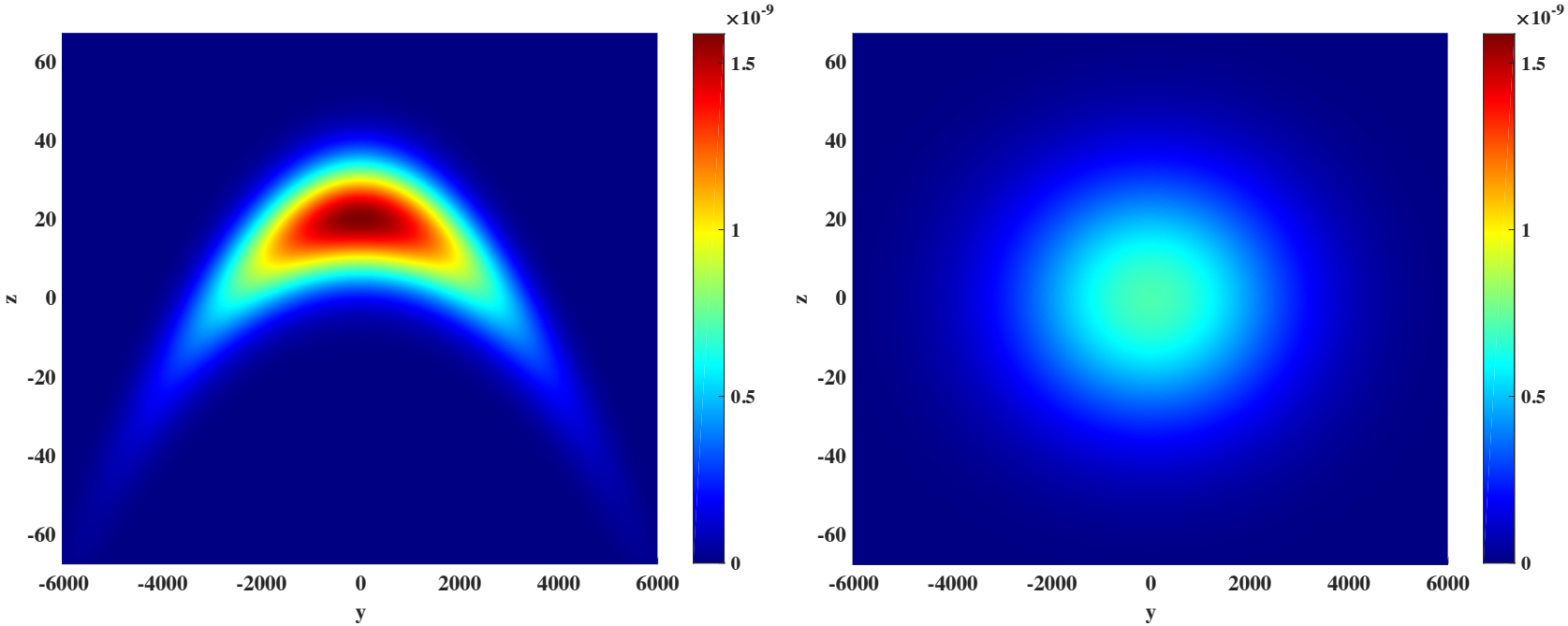
$$p_{\text{conv}}(\mathbf{t}) = |\mathbf{J}|^{-1} p(\mathbf{h}(\mathbf{t})) \tag{82}$$

$$\mathbf{J} \triangleq (\nabla_{\mathbf{z}} \mathbf{t}')' = \left(\nabla_{\mathbf{z}} \mathbf{h}^{-1}(\mathbf{z})'\right)' = \left((\nabla_{\mathbf{t}} \mathbf{h}(\mathbf{t})')'\right)^{-1} \tag{83}$$

  - $\nabla_{\mathbf{z}}$ denotes the gradient operator. $\mathbf{J}$ is the Jacobian matrix.
  - For example, between $r - u - v$ and Cartesian coordinates

$$\mathbf{J} = \left[\frac{\partial}{\partial r}\mathbf{t}, \frac{\partial}{\partial u}\mathbf{t}, \frac{\partial}{\partial v}\mathbf{t}\right] \tag{84}$$

Measurement Conversion



(a) PDF in Cartesian Coordinates

(b) Gaussian Approximation

- Example, The monostatic r-u-v to Cartesian conversion.
- PDF and Gaussian approximation in (a) and (b).
  - Gaussian approximation matches first two moments.
  - Offsets are with respect to mean PDF value.
  - $\mathbf{R} = \text{diag}\left(\left[10, 10^{-3}, 10^{-3}\right]\right)$ and $\mathbf{z}_{\text{true}} = [0, 0, 200 \times 10^3]$.

Measurement Conversion

- For a bijective measurement function $\mathbf{z} = \mathbf{h}(\mathbf{z}_{\text{Cart}})$, $\mathbf{z}_{\text{Cart}} = \mathbf{h}^{-1}(\mathbf{z})$ and additive Gaussian noise, the true mean $\hat{\mathbf{z}}_{\text{Cart}}$ and covariance matrix $\mathbf{R}_{\text{Cart}}$ of a measurement $\mathbf{z}_{\text{meas}}$ converted to Cartesian coordinates is:

$$\mathrm{E}\left\{\mathbf{h}^{-1}(\mathbf{z}) \middle| \mathbf{z}_{\text{meas}}\right\} = \hat{\mathbf{z}}_{\text{Cart}} = \int_{\mathbf{z}\in\mathbb{R}^{d_z}} \mathbf{h}^{-1}(\mathbf{z}) \mathcal{N}\left\{\mathbf{z}; \mathbf{z}_{\text{meas}}, \mathbf{R}\right\} d\mathbf{z} \quad (85)$$

$$\mathrm{E}\left\{\left(\mathbf{h}^{-1}(\mathbf{z}) - \hat{\mathbf{z}}_{\text{Cart}}\right)\left(\mathbf{h}^{-1}(\mathbf{z}) - \hat{\mathbf{z}}_{\text{Cart}}\right)' \middle| \mathbf{z}_{\text{meas}}\right\} = \mathbf{R}_{\text{Cart}} =$$

$$\int_{\mathbf{z}\in\mathbb{R}^{d_z}} \left(\mathbf{h}^{-1}(\mathbf{z}) - \hat{\mathbf{z}}_{\text{Cart}}\right)\left(\mathbf{h}^{-1}(\mathbf{z}) - \hat{\mathbf{z}}_{\text{Cart}}\right)' \mathcal{N}\left\{\mathbf{z}; \mathbf{z}_{\text{meas}}, \mathbf{R}\right\} d\mathbf{z} \quad (86)$$

- Both integrals can be approximated using cubature integration.
- See "Coordinate Systems/Conversions with Covariances/Cubature Conversions" and Atmospheric Models/Standard Exponential Model/Cubature Conversions' in the TCL.

Measurement Conversion

- Cubature-based measurement conversion is the most versatile.
  - Cross-correlation terms are trivially taken into account by including them in $\mathbf{R}$.
  - Non-additive Gaussian noise can be handled by replacing $\mathbf{h}^{-1}(\mathbf{z})$ with $\mathbf{h}^{-1}(\mathbf{z}, \mathbf{w})$.
  - h could include ray-traced atmospheric refractive effects.
- An alternative approach in the literature is based on Taylor series expansions.
  - See "Coordinate Systems/Conversions with Covariances/Taylor-Series Conversions" in the TCL.
  - Given the model

$$\hat{\mathbf{z}} = \mathbf{h}(\mathbf{z}_{\text{Cart}}) - \mathbf{w} \tag{87}$$

  - Invert the equation

$$\mathbf{z}_{\text{Cart}} = \mathbf{h}^{-1}\left(\hat{\mathbf{z}} + \mathbf{w}\right) \tag{88}$$

  - A Taylor series expansion about $\mathbf{w} = \mathbf{0}$ is

$$\mathbf{z}_{\text{Cart}} = \mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right) + \left(\nabla_{\mathbf{z}} \mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right)'\right)' \mathbf{w} + \ldots \tag{89}$$

Measurement Conversion

- Given the Taylor series expansion, some approaches in the literature truncate it at first order and use the expected value and covariance:

$$\begin{aligned}\hat{\mathbf{z}}_{\text{Cart}} =& \mathrm{E}\left\{\mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right) + \left(\nabla_{\mathbf{z}}\mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right)'\right)'\mathbf{w}\right\} \\ =& \mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right) \qquad (90)\end{aligned}$$

$$\mathbf{R}_{\text{Cart}} = \mathrm{E}\left\{\left(\nabla_{\mathbf{z}}\mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right)'\right)'\mathbf{w}\mathbf{w}'\left(\nabla_{\mathbf{z}}\mathbf{h}^{-1}\left(\hat{\mathbf{z}}\right)'\right)\right\} \qquad (91)$$

- For example, for polar measurements $(\hat{r}, \hat{\theta})$:

$$\hat{\mathbf{z}}_{\text{Cart}} = \begin{bmatrix}\hat{r}\cos(\hat{\theta}) \\ \hat{r}\sin(\hat{\theta})\end{bmatrix} \qquad (92)$$

$$\mathbf{R}_{\text{Cart}} = \begin{bmatrix}\sigma_r^2\cos(\hat{\theta})^2 - 2\hat{r}\sigma_{r\theta}\cos(\hat{\theta})\sin(\hat{\theta}) + \hat{r}^2\sigma_\theta^2\sin(\hat{\theta})^2 & (\sigma_r^2 - \hat{r}^2\sigma_\theta^2)\cos(\hat{\theta})\sin(\hat{\theta}) + \hat{r}\sigma_{r\theta}\cos(2\hat{\theta}) \\ (\sigma_r^2 - \hat{r}^2\sigma_\theta^2)\cos(\hat{\theta})\sin(\hat{\theta}) + \hat{r}\sigma_{r\theta}\cos(2\hat{\theta}) & \sigma_r^2\sin(\hat{\theta})^2 + \hat{r}\cos(\hat{\theta})(\hat{r}\sigma_\theta^2\cos(\hat{\theta}) + 2\sigma_{r\theta}\sin(\hat{\theta}))\end{bmatrix} \qquad (93)$$

Measurement Conversion

- However, the estimate from the first-order Taylor series approximation is biased.
- Consider polar measurements with $\sigma_{r\theta} = 0$:

$$\hat{r} = r + w_r \tag{94}$$

$$\hat{\theta} = \theta + w_\theta \tag{95}$$

$$\mathrm{E}\left\{\hat{\mathbf{z}}_{\mathsf{Cart}} \middle| r, \theta\right\} = \begin{bmatrix} \mathrm{E}\left\{\hat{r} \middle| r, \theta\right\} \mathrm{E}\left\{\cos(\hat{\theta}) \middle| r, \theta\right\} \\ \mathrm{E}\left\{\hat{r} \middle| r, \theta\right\} \mathrm{E}\left\{\sin(\hat{\theta}) \middle| r, \theta\right\} \end{bmatrix} \tag{96}$$

- Assuming Gaussian noise,

$$\mathrm{E}\left\{\hat{r} \middle| r, \theta\right\} = r \tag{97}$$

$$\mathrm{E}\left\{\cos(\hat{\theta}) \middle| r, \theta\right\} = \frac{1}{\sqrt{2\pi\sigma_\theta^2}} \int_{-\infty}^{\infty} \cos(\theta + w_\theta) e^{-\frac{w_\theta^2}{2\sigma_\theta^2}} \, dw_\theta = e^{-\frac{\sigma_\theta^2}{2}} \cos(\theta) \tag{98}$$

$$\mathrm{E}\left\{\sin(\hat{\theta}) \middle| r, \theta\right\} = \frac{1}{\sqrt{2\pi\sigma_\theta^2}} \int_{-\infty}^{\infty} \sin(\theta + w_\theta) e^{-\frac{w_\theta^2}{2\sigma_\theta^2}} \, dw_\theta = e^{-\frac{\sigma_\theta^2}{2}} \sin(\theta) \tag{99}$$

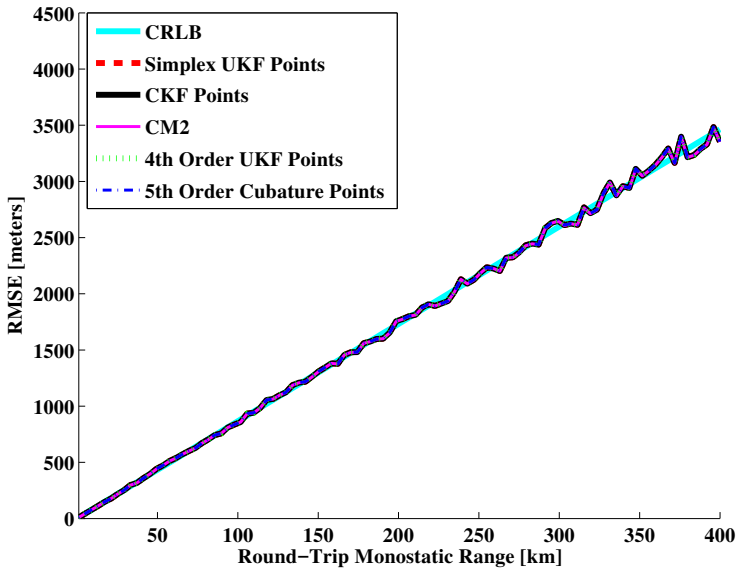
Measurement Conversion

- Thus, the conditional expected value of the converted measurement is scaled by $e^{-\frac{\sigma_\theta^2}{2}}$:

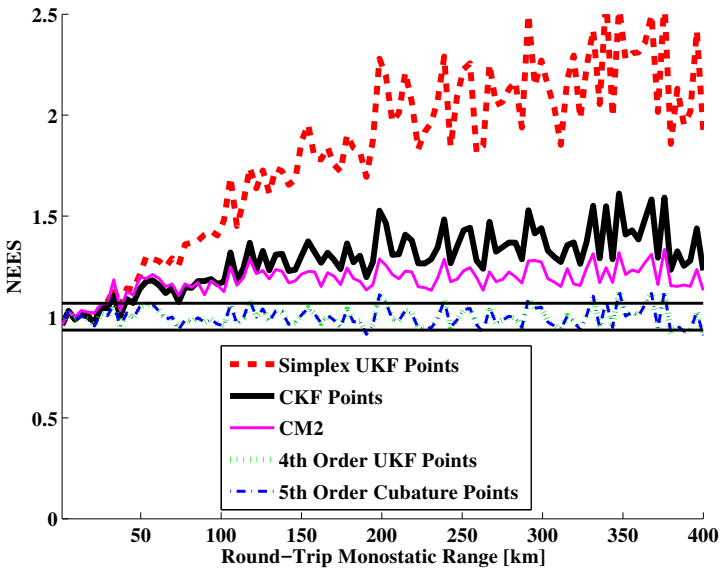
$$\mathrm{E}\left\{\hat{\mathbf{z}}_{\mathsf{Cart}} \middle| r, \theta\right\} = \begin{bmatrix} e^{-\frac{\sigma_\theta^2}{2}} r\cos(\theta) \\ e^{-\frac{\sigma_\theta^2}{2}} r\sin(\theta) \end{bmatrix} \tag{100}$$

- "Debiased" or "unbiased" Taylor-series-based exist in the literature (Here, multiply by $e^{\frac{\sigma_\theta^2}{2}}$).
  - Such techniques attempt to compute better covariance matrices by accounting for noise in the otherwise constant $\hat{r}$, $\hat{\theta}$ terms in the Taylor series expansion.
  - Some methods use a second order Taylor series expansion.
- Taylor expansion-based conversion methods can be computationally simpler but less robust than cubature techniques.

Measurement Conversion Example I



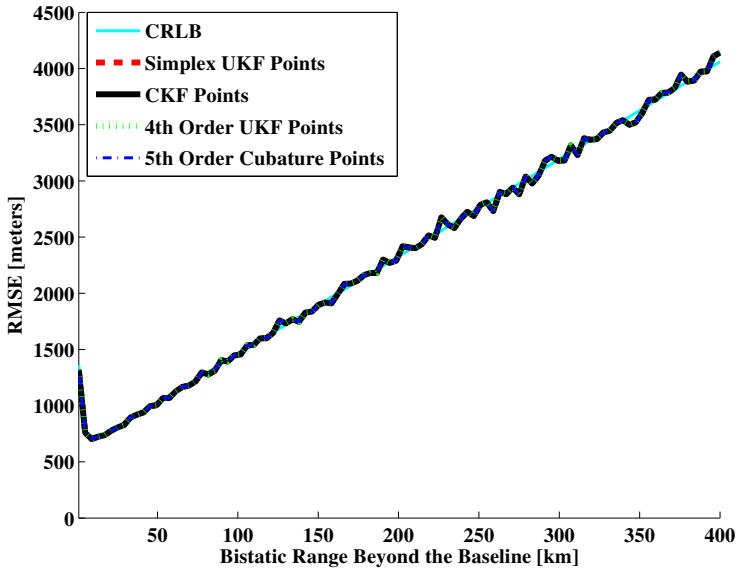
(a) RMSE



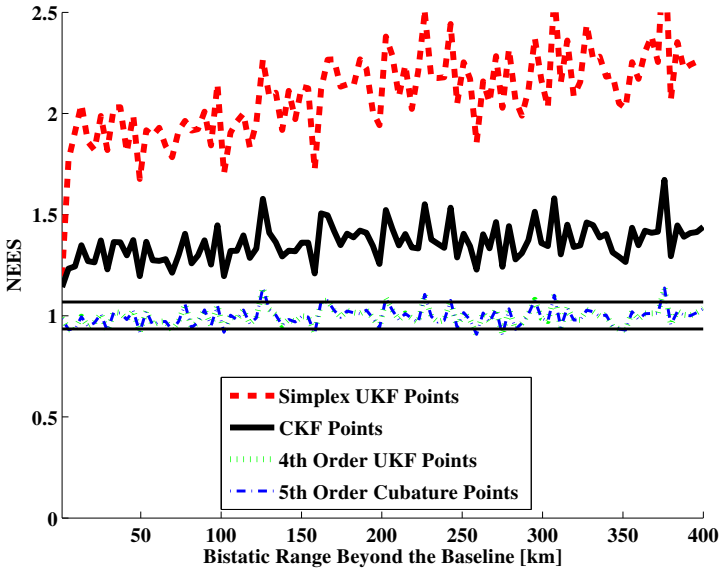
(b) NEES

- Monostatic $r - u - v$ to Cartesian measurement conversion with the target at $u = 0$ and $v = \sin(45°)$; range varied.
- Black NEES lines are the 95% confidence region.
- $\mathbf{R} = \mathrm{diag}([10\,\mathrm{m}, 10^{-2}, 10^{-2}])^2$
- CM2 is a second-order Taylor-series expansion method, simplex UKF=second order cubature, CKF= third-order cubature.

Measurement Conversion Example II



(a) RMSE



(b) NEES

- An example of bistatic $r - u - v$ to Cartesian measurement conversion with the transmitter at $(20\,\text{km}, 20\,\text{km}, 20\,\text{km})$ and target and noise parameters as in monostatic case.
- The difference in algorithms is most obvious in the NEES.

U.S. NAVAL RESEARCH LABORATORY

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

$$\overbrace{p(\mathbf{x}|\mathbf{z})}^{\text{posterior distribution}} = \frac{\overbrace{p(\mathbf{z}|\mathbf{x})}^{\text{measurement distribution}} \overbrace{p(\mathbf{x})}^{\text{prior distribution}}}{\underbrace{p(\mathbf{z})}_{\text{normalizing constant}}} \tag{101}$$

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

$$p(\mathbf{z}) = \int_{\mathbf{x}\in\mathbb{S}} p(\mathbf{z}|\mathbf{x})p(\mathbf{x})\, d\mathbf{x} \tag{102}$$

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

Bayes' Theorem: Monte Hall Example

- Bayes' theorem underlies all rigorous measurement update algorithms in tracking.
- A simple example of Bayes theorem is the Monte Hall problem:
  - You are given a choice of three doors. Behind one door is a car and goats are behind the other two. You pick a door and the host opens a different door behind which there is a goat. What are your odds of finding a car if you stay with the originally chosen door versus picking the other remaining door?

Bayes' Theorem: Monte Hall Example

- x is the door behind which there is a car. The initial set of probabilities $p(x)$ for each door is uniform:

	d_1	d_2	d_3
$p(x)$	$1/3$	$1/3$	$1/3$

- Without loss of generality, assume you choose door 1. The measurement likelihood function $p(z|x)$ is:

x	d_1	d_2	d_3
1	0	$1/2$	$1/2$
2	0	0	1
3	0	1	0

- Without loss of generality, assume that the host opens door number 2. Applying Bayes' theorem one gets $p(x|z)$ to be

	d_1	d_2	d_3
$p(x|z)$	$1/3$	0	$2/3$

- The best choice is to switch doors.
- Most people think it doesn't matter. Bayes' theorem can outperform one's instincts.

Bayes' Theorem and Joint Distributions

- Suppose the joint distribution $p(\mathbf{x}, \mathbf{z})$ is known.
- Using the definition of conditional probability

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \tag{103}$$

- This allows Bayes' theorem to be rewritten

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z})} \tag{104}$$

Bayes' Theorem and Joint Gaussian Distirbutions

- Assume that the state $\mathbf{x}$ and measurement $\mathbf{z}$ are jointly Gaussian.

$$p(\mathbf{x},\mathbf{z}) = \mathcal{N}\left\{ \overbrace{\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}}^{\mathbf{y}}; \overbrace{\begin{bmatrix}\hat{\mathbf{x}}_{\text{prior}}\\ \hat{\mathbf{z}}_{\text{prior}}\end{bmatrix}}^{\hat{\mathbf{y}}}, \overbrace{\begin{bmatrix}\mathbf{P}_{\text{prior}} & \mathbf{P}^{xz}_{\text{prior}}\\ \mathbf{P}^{zx}_{\text{prior}} & \mathbf{P}^{zz}_{\text{prior}}\end{bmatrix}}^{\mathbf{P}^{yy}} \right\} \tag{105}$$

- Using Bayes' rule for the update one gets

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{z})} \tag{106}$$

$$= \frac{\left|\mathbf{P}^{yy}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{y}-\hat{\mathbf{y}})'\mathbf{P}^{yy}(\mathbf{y}-\hat{\mathbf{y}})}}{\left|\mathbf{P}^{zz}_{\text{prior}}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{z}-\hat{\mathbf{z}}_{\text{prior}})'\mathbf{P}^{zz}_{\text{prior}}(\mathbf{z}-\hat{\mathbf{z}}_{\text{prior}})}} \tag{107}$$

Bayes' Theorem: Linear Gaussian Distributions

- After considerable simplification, one finds that the posterior $p(\mathbf{x}|\mathbf{z})$ is Gaussian with mean and covariance matrix

$$\hat{\mathbf{x}}_{\text{posterior}} = \hat{\mathbf{x}}_{\text{prior}} + \mathbf{P}^{xz}_{\text{prior}} \left(\mathbf{P}^{zz}_{\text{prior}}\right)^{-1} \left(\mathbf{z} - \hat{\mathbf{z}}_{\text{prior}}\right) \tag{108}$$

$$\mathbf{P}_{\text{posterior}} = \mathbf{P}_{\text{prior}} - \mathbf{P}^{xz}_{\text{prior}} \left(\mathbf{P}^{zz}_{\text{prior}}\right)^{-1} \mathbf{P}^{zx}_{\text{prior}} \tag{109}$$

- The joint Gaussian assumption holds for the linear Gaussian model $p(\mathbf{x}) \sim \mathcal{N}\left\{\hat{\mathbf{x}}_{\text{prior}}, \mathbf{P}_{\text{prior}}\right\}$ and

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{110}$$

  where $\mathbf{w} \sim \mathcal{N}\left\{\mathbf{0}, \mathbf{R}\right\}$.
- The conditional measurement distribution is Gaussian $p(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}\left\{\mathbf{H}\mathbf{x}, \mathbf{R}\right\}$.
- Note that

$$\hat{\mathbf{z}}_{\text{prior}} = \mathrm{E}\left\{\mathbf{z}\right\} = \mathrm{E}\left\{\mathbf{H}\mathbf{x}\right\} = \mathbf{H}\hat{\mathbf{x}}_{\text{prior}} \tag{111}$$

- It can be shown that $p(\mathbf{z}) \sim \mathcal{N}\left\{\hat{\mathbf{z}}_{\text{prior}}, \mathbf{P}_{zz}\right\}$.

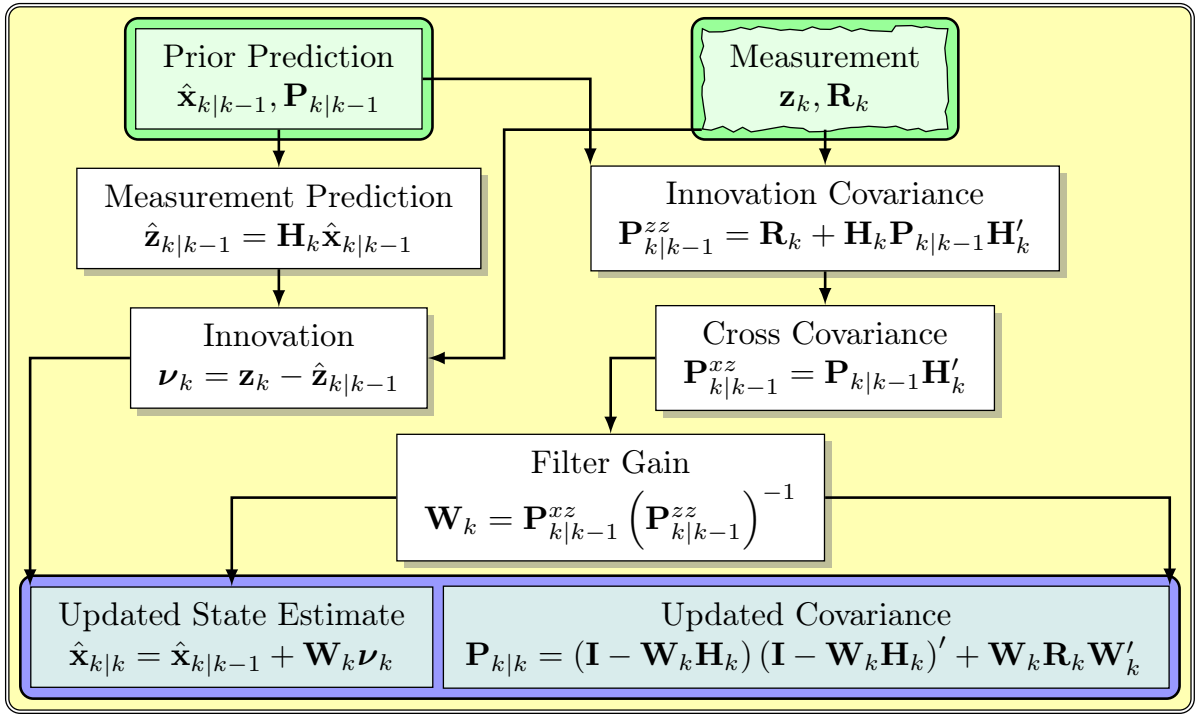
Bayes' Theorem: Linear Gaussian Distributions

- The covariance terms are

$$\begin{aligned}\mathbf{P}_{\text{prior}}^{zz} &= \mathrm{E}\left\{(\mathbf{z}-\hat{\mathbf{z}}_{\text{prior}})(\mathbf{z}-\hat{\mathbf{z}}_{\text{prior}})'\right\} \\ &= \mathbf{R}+\mathbf{H}\mathbf{P}_{\text{prior}}\mathbf{H}' \qquad (112)\\ \mathbf{P}_{\text{prior}}^{xz} &= \mathrm{E}\left\{(\mathbf{x}-\hat{\mathbf{x}}_{\text{prior}})(\mathbf{z}-\hat{\mathbf{z}}_{\text{prior}})'\right\} \\ &= \mathbf{P}_{\text{prior}}\mathbf{H}' \qquad (113)\end{aligned}$$

- Substituting everything back for the linear model one gets the update step for the Kalman filter.
- 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: Joseph's Form

- One would expect the covariance update to be

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{W}_k \left(\mathbf{P}^{xz}_{k|k-1} \right)' \tag{114}$$

- However, finite precision errors can possibly cause the $\mathbf{P}_{k|k}$ to have a negative eigenvalue even if $\mathbf{P}_{k|k-1}$ is positive definite.
- The subtraction is the problem.
- Solution: Replace the subtraction with a quadratic expression.
- The quadratic expression

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{W}_k \mathbf{H}_k) (\mathbf{I} - \mathbf{W}_k \mathbf{H}_k)' + \mathbf{W}_k \mathbf{R}_k \mathbf{W}'_k \tag{115}$$

is algebraically equivalent to (114) and does not suffer a risk of negative eigenvalues.

Bayes' Theorem: Why Use Approximations?

- The Kalman filter update is optimal for measurements that are linear combinations of the target state.
- One approach to handling nonlinear measurements (i.e. anything from a radar) for a Cartesian state is the previously discusses measurement conversion and Gaussian approximation.
- However, why not just apply Bayes' theorem more precisely?
- Bayes' theorem is again:

$$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \tag{116}$$

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

Bayes' Theorem: Why Use Approximations?

- The Gaussian distribution is a conjugate prior distribution to itself.
  - If $p(\mathbf{x})$ is Gaussian and $p(\mathbf{z}|\mathbf{x})$ is Gaussian, then $p(\mathbf{x}|\mathbf{z})$ must be Gaussian.
- See "Mathematical Functions/Statistics/Conjugate Prior Updates."
- Some other examples of conjugate prior distributions are:

$\mathbf{x}$	$p(\mathbf{x})$	$p(\mathbf{z}\|\mathbf{x})$
λ	Poisson	gamma
P_{FA}	binomial	beta
σ^2	scalar normal	inverse gamma
$\mathbf{\Sigma}$	multivariate normal	inverse Wishart

- If the measurement distribution is not conjugate to the prior distribution, the result of Bayes' rule will be increasingly complicated.
  - Not suitable for recursive estimation.

U.S. NAVAL RESEARCH LABORATORY

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?

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

Bayes' Theorem: Can't it Always be Linear?

- Why not always make the state in the local coordinate system of the receiver?
  - For example, $[r, u, v, \dot{r}, \dot{u}, \dot{v}]$ instead of position and velocity.
- The linear Kalman filter measurement update is optimal for additive Gaussian measurement noise.
- This approach is widely used in the open literature for missile control systems.
- The use of range rate becomes much simpler.
- Drawbacks:
  - There is no common "local" coordinate system for multiple sensors/multiple measurement types. E.g. monostatic plus bistatic measurements.
  - Non-maneuvering targets follow curved trajectories in local coordinates.
  - Nonlinear dynamic models are more difficult to handle than nonlinear measurements.

Bayes' Theorem: Can't it Always be Linear?

- Consider a 2D position vector $\mathbf{r}$ in a polar coordinate system:

$$\mathbf{u}_r = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \qquad \mathbf{u}_\theta = \frac{\partial \mathbf{u}_r}{\partial \theta} = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix} \tag{117}$$

$$\mathbf{r} = r\mathbf{u}_r \tag{118}$$

- Note that

$$\dot{\mathbf{u}}_r = \frac{\partial \mathbf{u}_r}{\partial \theta}\dot{\theta} = \dot{\theta}\mathbf{u}_\theta \qquad \dot{\mathbf{u}}_\theta = -\dot{\theta}\mathbf{u}_r \tag{119}$$

- The velocity is

$$\dot{\mathbf{r}} = \dot{r}\mathbf{u}_r + r\dot{\mathbf{u}}_r = \dot{r}\mathbf{u}_r + r\dot{\theta}\mathbf{u}_\theta \tag{120}$$

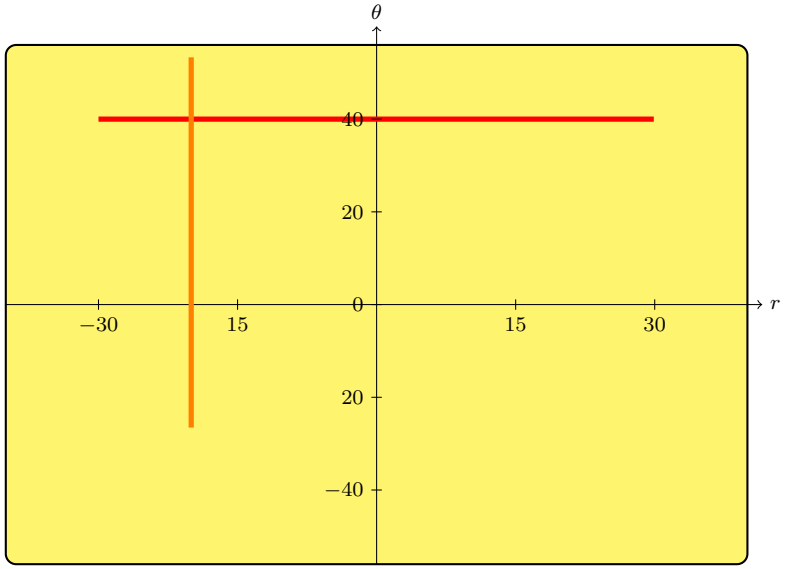
- Thus, the acceleration is

$$\ddot{\mathbf{r}} = \ddot{r}\mathbf{u}_r + \dot{r}\dot{\mathbf{u}}_r + \dot{r}\dot{\theta}\mathbf{u}_\theta + r\ddot{\theta}\mathbf{u}_\theta + r\dot{\theta}\dot{\mathbf{u}}_\theta = \underbrace{\left(\ddot{r} - r\dot{\theta}^2\right)}_{a_r}\mathbf{u}_r + \underbrace{\left(r\ddot{\theta} + 2\dot{r}\dot{\theta}\right)}_{a_\theta}\mathbf{u}_\theta \tag{121}$$

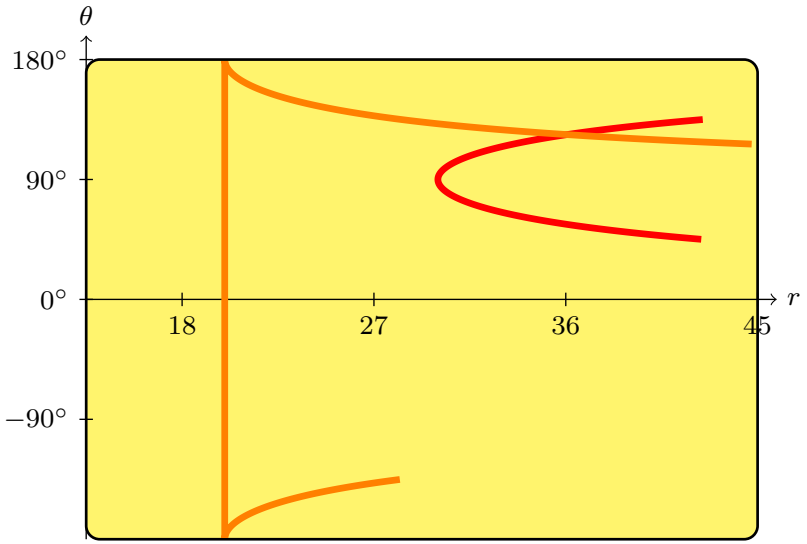
- For constant $a_r = 0$ and $a_\theta = 0$, one gets the *nonlinear* dynamic model:

$$\ddot{r} = r\dot{\theta}^2 \qquad \ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r} \tag{122}$$

Bayes' Theorem: Can't it Always be Linear?



(a) Cartesian Coordinates



(b) Polar Coordinates

- Consider a 2D constant Cartesian velocity dynamic model ($a_r = a_\theta = 0$).
- Constant velocity motion in Cartesian coordinates on the left.
- The same motion in 2D polar coordinates on the right.

Bayes' Theorem: Can't it Always be Linear?

- Similar nonlinearities arise for linear motion translated to other coordinate systems. See "Dynamic Models/Continuous Time/Non-Cartesian Position" in the TCL.
- For range and a single direction cosine:

$$\ddot{r} = \frac{r u \dot{u}^2}{1 - u^2} \tag{123}$$

$$\ddot{u} = \frac{2 \dot{r} \dot{u}}{r} - \frac{u \dot{u}^2}{1 - u^2} \tag{124}$$

- In 3D spherical coordinates:

$$\ddot{r} = r \dot{\phi}^2 + r \dot{\theta}^2 \cos(\phi)^2 \tag{125}$$

$$\ddot{\theta} = \frac{1}{r} \left(-2 \dot{r} \dot{\theta} + 2 r \dot{\theta} \dot{\phi} \tan(\phi) \right) \tag{126}$$

$$\ddot{\phi} = \frac{1}{r} \left(-2 \dot{r} \dot{\phi} - r \dot{\theta}^2 \cos(\phi) \sin(\phi) \right) \tag{127}$$

- One usually chooses to have linear dynamic models and nonlinear measurements.

U.S. NAVAL RESEARCH LABORATORY

STOCHASTIC CALCULUS AND LINEAR DYNAMIC MODELS

Stochastic Calculus: Motivation

- A physicist providing a sophisticated dynamic model (ballistic/ orbital, spiraling, weaving, turning, etc), will probably initially create a deterministic differential equation:

$$\frac{d\mathbf{x}_t}{dt} = \overbrace{\tilde{\mathbf{a}}(\mathbf{x}_t, t)}^{\text{What Physics Tells You}} \tag{128}$$

- . . . but objects under track are not deterministic.
- Physics offers an approach: A stochastic differential equation:

$$d\mathbf{x}_t = \overbrace{\tilde{\mathbf{a}}(\mathbf{x}_t, t)dt}^{\text{What Physics Tells You}} + \overbrace{\mathbf{D}(\mathbf{x}_t, t)d\boldsymbol{\beta}_t}^{\text{Unknown Perturbations}} \tag{129}$$

  - $\tilde{\mathbf{a}}(\mathbf{x}_t, t)$ is the drift function.
  - $\mathbf{D}(\mathbf{x}_t, t)$ is the diffusion matrix.
  - $d\boldsymbol{\beta}_t$ is the differential of a Wiener process.

Stochastic Calculus: Details

- Instead of using the notation

$$d\mathbf{x}_t = \tilde{\mathbf{a}}(\mathbf{x}_t, t)dt + \mathbf{D}(\mathbf{x}_t, t)d\boldsymbol{\beta}_t \tag{130}$$

one typically wants to write a traditional differential equation:

$$\frac{d\mathbf{x}_t}{dt} = \tilde{\mathbf{a}}(\mathbf{x}_t, t) + \mathbf{D}(\mathbf{x}_t, t)\frac{d\boldsymbol{\beta}_t}{dt} \tag{131}$$

- For practical purposes, (130) is equivalent to (131).
- . . . but under strict mathematical definitions, $\frac{d\boldsymbol{\beta}_t}{dt}$ does not exist.
- However integrals over the term are defined.

Stochastic Calculus: Details

- The evolution of the Wiener process vector $\boldsymbol{\beta}_t = [\beta_t(1), \beta_t(2), \ldots, \beta_t(d_w)]'$ is defined as

$$p(\beta_{t_2} - \beta_{t_1}) = \frac{1}{\sqrt{2\pi(t_2 - t_1)}} \exp\left[-\frac{(\beta_{t_2} - \beta_{t_1})^2}{2(t_2 - t_1)}\right] \tag{132}$$

$$\beta_0 = 0 \tag{133}$$

$$\mathrm{E}\left[\beta_{t_2} \middle| \beta_{t_1}\right] = \beta_{t_1} \tag{134}$$

$$\mathrm{E}\left[\beta_t\right] = 0 \tag{135}$$

- This property is important for deriving explicit solutions to certain systems.
- However, one must first define what a stochastic integral is.

Stochastic Calculus: Details

- In Calculus II, definite integrals of real functions are often derived related to Riemann sums:

$$\int_a^b f(t)dt \triangleq \lim_{N\to\infty} \sum_{j=0}^{N-1} f(\tau_j)\left(t_{j+1} - t_j\right) \tag{136}$$

with

$$t_0 = a \qquad t_j = t_0 + \Delta^t j \qquad \Delta^t = \frac{b-a}{N} \tag{137}$$

- $\tau_j \in [t_j, t_{j+1}]$ and it generally does not matter where.
- The more esoteric Riemann-Stieltjes integral adds in another function $g(t)$ as a measure:

$$\int_a^b f(t)dg(t) \triangleq \lim_{N\to\infty} \sum_{j=0}^{N-1} f(\tau_j)\left(g(t_{j+1}) - g(t_j)\right) \tag{138}$$

- Again $\tau_j \in [t_j, t_{j+1}]$ and it generally does not matter where.

Stochastic Calculus: Details

- The strict definition of the Riemann-Stieltjes integral is not suited to measures $g(t)$ that vary infinitely in a given interval.
- A Wiener process β_t varies infinitely in a given interval, but the definition of the integral used is similar to a Riemann-Stieltjes integral:

$$\int_a^b f(t)d\beta_t = \lim_{N\to\infty} \sum_{j=0}^{N-1} f(\tau_j)\left(\beta_{t_{j+1}} - \beta_{t_j}\right) \tag{139}$$

- Unlike with Riemman and Riemann-Stieltjes integrals, choosing different values of $\tau_j \in [t_j, t_{j+1}]$ changes the results.
  - Choosing $\tau_j = t_j$ results in Itô calculus.
  - Choosing $\tau_j = \frac{t_{j+1}+t_j}{2}$ results in Stratonovich calculus.
- Itô calculus is simpler and is usually the only one used.

Stochastic Calculus: Details

- Given a definition of a calculus, an integral over a stochastic differential equation takes a form akin to a Riemann sum:

$$\mathbf{x}_b = \lim_{N\to\infty} \sum_{j=0}^{N-1} \left(\tilde{\mathbf{a}}(\mathbf{x}_{\tau_j}, \tau_j)\Delta^t + \mathbf{D}\left(\mathbf{x}_{\tau_j}, \tau_j\right)\left(\boldsymbol{\beta}_{t_{j+1}} - \boldsymbol{\beta}_{t_j}\right)\right) \quad (140)$$

- The integral result is random, because $\boldsymbol{\beta}_{t_j}$ is random.
- However, the intermediate values of $\mathbf{x}_{\tau_j}$ are required.
- One can consider the integral over an infinitesimal step:

$$\mathbf{x}_{t+\Delta^t} = \mathbf{x}_t + \tilde{\mathbf{a}}(\mathbf{x}_\tau, \tau)\Delta^t + \mathbf{D}\left(\mathbf{x}_\tau, \tau\right)\left(\boldsymbol{\beta}_{t+\Delta^t} - \boldsymbol{\beta}_t\right) \quad (141)$$

  which is akin to Euler's method for deterministic differential equations.
- Stochastic Runge-Kutta methods and other techniques exist for better solutions to stochastic differential equations. See StrongStochRungeKStep and WeakStochRungeKStep in the TCL.

Stochastic Calculus: Linear Systems

- For linear systems, explicit solutions are available.
- For linear systems, Itô and Stratonovich calculus are the same.
  - The calculi only differ if $\mathbf{D}(\mathbf{x}_t, t)$ depends on $\mathbf{x}_t$.
- Consider the linear difference equation:

$$d\mathbf{x}_t = \mathbf{A}\mathbf{x}_t dt + \mathbf{B} d\boldsymbol{\beta}_t \tag{142}$$

  where $\mathbf{A}$ and $\mathbf{B}$ are constant matrices.
- The integral of the stochastic differential equation from t_0 to t is

$$\mathbf{x}_t = \mathbf{x}_{t_0} + \int_{t_0}^{t} \mathbf{A}\mathbf{x}_\tau d\tau + \int_{t_0}^{t} \mathbf{B} d\boldsymbol{\beta}_\tau \tag{143}$$

  where the second integral is stochastic.

Stochastic Calculus: Linear Systems

- Rather than directly solve the integral, we note that the infinitesimal step

$$\mathbf{x}_{t+\Delta^t} = \mathbf{x}_t + \mathbf{A}\mathbf{x}_t\Delta^t + \mathbf{B}\left(\boldsymbol{\beta}_{t+\Delta^t} - \boldsymbol{\beta}_t\right) \tag{144}$$

  is Gaussian distributed if $\mathbf{x}_t$ is deterministic of Gaussian distributed. Thus, the solution to the entire integral is Gaussian distributed.

- As (144) is linear, the solution to the integral must have the form

$$\mathbf{x}_t = \mathbf{F}_{t-t_0}\mathbf{x}_{t_0} + \mathbf{v} \tag{145}$$

  where $\mathbf{v} \sim \mathcal{N}\left\{\mathbf{0}, \mathbf{Q}_{t-t_0}\right\}$ and $\mathbf{w}$ is the same dimensionality as $\boldsymbol{\beta}_t$.
  - $\mathbf{F}_{t-t_0}$ and $\mathbf{Q}_{t-t_0}$ must be determined.

Stochastic Calculus: Linear Systems

- A second infinitesimal step has the form

$$\mathbf{x}_{t+2\Delta^t} = \mathbf{x}_{t+\Delta^t} + \mathbf{A}\mathbf{x}_{t+\Delta^t}\Delta^t + \mathbf{B}\left(\boldsymbol{\beta}_{t+2\Delta^t} - \boldsymbol{\beta}_{t+\Delta^t}\right) \tag{146}$$

$$= \overbrace{\mathbf{x}_t + 2\mathbf{A}\mathbf{x}_t\Delta^t + \mathbf{A}^2\mathbf{x}_t\left(\Delta^t\right)^2}^{\text{Same as two steps for } \frac{d\mathbf{x}_t}{dt} = \mathbf{A}\mathbf{x}_t} + \left(\mathbf{I} + \Delta^t\mathbf{A}\right)\mathbf{B}\left(\boldsymbol{\beta}_{t+\Delta^t} - \boldsymbol{\beta}_t\right) + \mathbf{B}\left(\boldsymbol{\beta}_{t+2\Delta^t} - \boldsymbol{\beta}_{t+\Delta^t}\right) \tag{147}$$

- $\mathbf{F}$ thus comes from integrating the non-stochastic differential equation

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{A}\mathbf{x}_t \tag{148}$$

  given $\mathbf{x}_{t_0}$.
- This differential equation is solved in the control literature.

Stochastic Calculus: Linear Systems

- The integral that must be solved the determine $\mathbf{F}$ is

$$\int_{t_0}^{t} \mathbf{A}\mathbf{x}_t dt \tag{149}$$

- The solution utilizes a property of the exponential function of a matrix.

$$\frac{d}{dt} e^{\mathbf{A}t} = \mathbf{A} e^{\mathbf{A}t} = e^{\mathbf{A}t} \mathbf{A} \tag{150}$$

- The exponential of a matrix is defined as a generalization of the Taylor series expansion of $e^{\lambda t}$:

$$e^{\mathbf{A}t} \triangleq \sum_{k=0}^{\infty} \frac{1}{k!} t^k \mathbf{A}^k \tag{151}$$

Stochastic Calculus: Linear Systems

- The differential equation to solve is

$$\dot{\mathbf{x}}_t = \mathbf{A}\mathbf{x}_t \tag{152}$$

- Pre and post multiply by $e^{-\mathbf{A}t}$

$$e^{-\mathbf{A}t}\dot{\mathbf{x}}_t - e^{-\mathbf{A}t}\mathbf{A}\mathbf{x}_t = \mathbf{0} \tag{153}$$

- Utilizing the property of the matrix exponential

$$\frac{d}{dt}\left(e^{-\mathbf{A}t}\dot{\mathbf{x}}_t\right) = \mathbf{0} \tag{154}$$

- Integrating from t_0 to t yields

$$\mathbf{x}_t = e^{\mathbf{A}(t-t_0)}\mathbf{x}_{t_0} \tag{155}$$

so $\mathbf{F}_{t-t_0} = e^{\mathbf{A}(t-t_0)}$.

Stochastic Calculus: Linear Systems

- We must determine the process noise covariance matrix $\mathbf{Q}$ to finish finding the posterior distribution.
- We propose that the solution to the stochastic differential equation $d\mathbf{x}_t = \mathbf{A}\mathbf{x}_t dt + \mathbf{B}d\boldsymbol{\beta}_t$ is

$$\mathbf{x}_t = \mathbf{F}_t \left(\mathbf{x}_{t_0} + \int_{t_0}^{t} \mathbf{D} d\boldsymbol{\beta}_\tau \right) \tag{156}$$

  - It can be shown that if $t = t_0$, then $\mathbf{x}_t = \mathbf{x}_{t_0}$, satisfying the initial conditions.
- Differentiating (156), one gets

$$d\mathbf{x}_t = \frac{d\mathbf{F}_{t-t_0}}{dt} \left(\mathbf{x}_{t_0} + \int_{t_0}^{t} \mathbf{D} d\boldsymbol{\beta}_\tau \right) + \mathbf{F}_{t-t_0} \mathbf{D} d\boldsymbol{\beta}_t \tag{157}$$

- The proof by induction comes from substituting (157) into $\mathbf{x}_t = \mathbf{x}_0 + \int_{t_0}^{t} d\mathbf{x}_\tau d\tau$ and simplifying to get $\mathbf{x}_t = \mathbf{x}_{t_0} + \int_{t_0}^{t} \mathbf{A}\mathbf{x}_\tau d\tau + \int_{t_0}^{t} \mathbf{B} d\boldsymbol{\beta}_\tau$ again.

Stochastic Calculus: Linear Systems

- From the definition of differences of $\boldsymbol{\beta}_t$ and the Riemann-Stieltjes-like definition of the stochastic integral, one knows that

$$\int_{t_0}^{t} d\boldsymbol{\beta}_\tau \sim \mathcal{N}\left\{\mathbf{0}, (t - t_0)\mathbf{I}\right\} \tag{158}$$

- Given the proposed solution, the covariance of the prediction can be found to be

$$\mathbf{Q}_{t-t_0} = \mathrm{E}\left\{ \left(\mathbf{x}_t - \mathbf{F}_{t-t_0}\mathbf{x}_{t_0}\right)\left(\mathbf{x}_t - \mathbf{F}_{t-t_0}\mathbf{x}_{t_0}\right)' \middle| \mathbf{x}_{t_0} \right\} \tag{159}$$

- The integral is over matrix exponentials ($\mathbf{F}_{t-t_0}$ is a matrix exponential). The solution is the linDynMod2Disc function in the TCL and the algorithm is from

  C. F. Van Loan, "Computing Integrals Involving the Matrix Exponential," IEEE Transactions on Automatic Control, vol. AC-23, no. 3, pp. 395-404, Jun. 1967.

- We now have the tools to convert linear continuous-time models to linear discrete-time models.

Stochastic Calculus: Linear Systems Summary

Summary

- Given a continuous-time stochastic dynamic model of the form

$$d\mathbf{x}_t = \mathbf{A}\mathbf{x}_t dt + \mathbf{B} d\boldsymbol{\beta}_t \tag{160}$$

The value of $\mathbf{x}_t$ predicting from time t_0 to t given a known x_{t_0} can be written as a discrete linear equation

$$\mathbf{x}_t = \mathbf{F}_{t-t_0}\mathbf{x}_0 + \mathbf{v} \tag{161}$$

where

$$\mathbf{v} \sim \mathcal{N}\left\{\mathbf{0}, \mathbf{Q}_{t-t_0}\right\} \tag{162}$$

and the state transition matrix $\mathbf{F}_{t-t_0}$ and process noise covariance matrix $\mathbf{Q}_{t-t_0}$ can be found using matrix exponentials via the linDynMod2Disc function in the TCL.

Stochastic Calculus: Linear Dynamic Models

The following classes of linear dynamic models shall be discussed:

1. The family of nearly constant moment models (velocity, acceleration, jerk, etc.).
   - See FPolyKal and QPolyKal in the TCL.
2. The family of correlated nearly constant moment models (Ornstein-Uhlenbeck, Singer, etc.).
   - See FGaussMarkov and QGaussMarkov in the TCL.

Stochastic Calculus: Linear Dynamic Models

- The nearly constant velocity model (continuous white noise acceleration model) in 1D is

$$\overbrace{\begin{bmatrix} \dot{x}_t \\ \ddot{x}_t \end{bmatrix}}^{d\mathbf{x}_t} = \overbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}^{\mathbf{A}} \mathbf{x}_t dt + \overbrace{\begin{bmatrix} 0 \\ q \end{bmatrix}}^{\mathbf{B}} d\beta_t \tag{163}$$

- Noise is only added to the velocity term.
- q is the process noise intensity. Typical units are $\sqrt{\mathrm{m}^2/\mathrm{s}^3}$.
- The state transition and process noise covariance matrices for predicting $T \triangleq t - t_0$ are

$$\mathbf{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} q^2 \tag{164}$$

Stochastic Calculus: Linear Dynamic Models

- The nearly constant acceleration model in 1D is similarly

$$d\mathbf{x}_t = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}_t dt + \begin{bmatrix} 0 \\ 0 \\ q \end{bmatrix} d\beta_t \tag{165}$$

- Noise is only added to the acceleration term.
- The state transition and process noise covariance matrices for predicting T ahead are (omitting subscripts of $\mathbf{F}$ and $\mathbf{Q}$):

$$\mathbf{F} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix} q^2 \tag{166}$$

Stochastic Calculus: Linear Dynamic Models

- The pattern can be generalized to where the nth derivative has the noise added:

$$\mathbf{A} = \begin{cases} 0 & \text{for } n = 0 \text{ (the scalar case)} \\ \begin{bmatrix} \mathbf{0}_{n,1} & \mathbf{I}_{n,n} \\ 0 & \mathbf{0}_{1,n} \end{bmatrix} & \text{for } n > 0 \end{cases} \tag{167}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{0}_{n,1} \\ 1 \end{bmatrix} q \tag{168}$$

- The values in row r and column c (starting from 0) of the transition matrix and process noise covariance matrix to predict ahead T are:

$$F_{r,c} = \begin{cases} \dfrac{T^{c-r}}{(c-r)!} & \text{if } c - r \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{169}$$

$$Q_{r,c} = \frac{T^{(n-r)+(n-c)+1}}{(n-r)!(n-c)!((n-r)+(n-c)+1)} q^2 \tag{170}$$

Stochastic Calculus: Linear Dynamic Models

- 1D motion models can be generalized to d dimensions as $\mathbf{F}^d$ and $\mathbf{Q}|^d$ via a Kronecker product:

$$\mathbf{F}^d = \begin{bmatrix} \mathbf{I}_{d,d}F_{0,0} & \mathbf{I}_{d,d}F_{0,1} & \dots & \mathbf{I}_{d,d}F_{0,n} \\ \mathbf{I}_{d,d}F_{1,0} & \mathbf{I}_{d,d}F_{1,1} & \dots & \mathbf{I}_{d,d}F_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{d,d}F_{n,0} & \mathbf{I}_{d,d}F_{n,1} & \dots & \mathbf{I}_{d,d}F_{n,n} \end{bmatrix} \tag{171}$$

$$\mathbf{Q}^d = \begin{bmatrix} \mathbf{I}_{d,d}Q_{0,0} & \mathbf{I}_{d,d}Q_{0,1} & \dots & \mathbf{I}_{d,d}Q_{0,n} \\ \mathbf{I}_{d,d}Q_{1,0} & \mathbf{I}_{d,d}Q_{1,1} & \dots & \mathbf{I}_{d,d}Q_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{d,d}Q_{n,0} & \mathbf{I}_{d,d}Q_{n,1} & \dots & \mathbf{I}_{d,d}Q_{n,n} \end{bmatrix} \tag{172}$$

  where $\mathbf{I}_{dd}$ is a $d \times d$ identity matrix.
- In Matlab, use Fd=kron(F,eye(d,d)).

Stochastic Calculus: Linear Dynamic Models

- Observed changes in velocity/acceleration, etc. of real targets are correlated over time.
  - Consider a turning airplane.
- The class of Gauss-Markov dynamic models adds a correlation term to the highest-order moment. In 1D for an nth order model with decorrelation time τ:

$$\mathbf{A} = \begin{cases} -\frac{1}{\tau} & \text{for } n = 0 \text{ (position only)} \\ \begin{bmatrix} \mathbf{0}_{n,1} & \mathbf{I}_{n,n} \\ 0 & \left[\mathbf{0}_{1,n-1}, -\frac{1}{\tau}\right] \end{bmatrix} & \text{for } n > 0 \end{cases} \tag{173}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{0}_{n,1} \\ 1 \end{bmatrix} q \tag{174}$$

Stochastic Calculus: Linear Dynamic Models

- The elements of the state transition matrix and the process noise covariance matrix for the discretized Gauss-Markov model are:

$$F_{r,c} = \begin{cases} \dfrac{T^{c-r}}{(c-r)!} & \text{if } r \leq c \text{ and } c < n \\ (-\tau)^{n-r}\left(e^{-\frac{T}{\tau}} - \displaystyle\sum_{i=0}^{n-r-1} \dfrac{\left(-\frac{T}{\tau}\right)^i}{i!}\right) & \text{if } c = n \\ 0 & \text{otherwise} \end{cases} \tag{175}$$

$$\begin{aligned} Q_{r,c} =& q^2(-\tau)^{2n-r-c+1} \sum_{i=0}^{n-r-1} (-1)^i \left(1 - e^{-\frac{T}{\tau}} \sum_{k=0}^{i} \frac{\left(\frac{T}{\tau}\right)^k}{k!}\right) \\ &+ q^2(-\tau)^{2n-r-c+1} \sum_{i=0}^{n-c-1} (-1)^i \left(1 - e^{-\frac{T}{\tau}} \sum_{k=0}^{i} \frac{\left(\frac{T}{\tau}\right)^k}{k!}\right) \\ &- q^2 \frac{1}{2}(-\tau)^{2n-r-c+1} \left(1 - e^{-2\frac{T}{\tau}}\right) \\ &+ q^2(-\tau)^{2n-r-c} \sum_{i=0}^{n-r-1} \sum_{j=0}^{n-c-1} (-1)^{i+j} \frac{T\left(\frac{T}{\tau}\right)^{i+j}}{i!j!(i+j+1)} \end{aligned} \tag{176}$$

Stochastic Calculus: Linear Dynamic Models

- The most popular Gauss-Markov model is the Singer model, which has $n = 2$ for correlated acceleration:

$$\mathbf{F} = \begin{bmatrix} 1 & T & \tau^2\left(\frac{T}{\tau} - 1 + e^{-\frac{T}{\tau}}\right) \\ 0 & 1 & \tau\left(1 - e^{-\frac{T}{\tau}}\right) \\ 0 & 0 & e^{-\frac{T}{\tau}} \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} Q_{0,0} & Q_{0,1} & Q_{0,2} \\ Q_{0,1} & Q_{1,1} & Q_{1,2} \\ Q_{0,2} & Q_{1,2} & Q_{2,2} \end{bmatrix} \tag{177}$$

$$Q_{0,0} = q^2\frac{\tau^2}{6}\left(2T^3 - 6\tau T^2 + 6T\tau^2\left(1 - 2e^{-\frac{T}{\tau}}\right) + 3\tau^3\left(1 - e^{-\frac{2T}{\tau}}\right)\right) \tag{178}$$

$$Q_{0,1} = q^2\frac{\tau^2}{2}e^{-\frac{2T}{\tau}}\left(\tau + e^{\frac{T}{\tau}}(T - \tau)\right)^2 \tag{179}$$

$$Q_{0,2} = q^2\tau^2 e^{-\frac{T}{\tau}}\left(\tau\sinh\left(\frac{T}{\tau}\right) - T\right) \tag{180}$$

$$Q_{1,1} = q^2\frac{\tau^2}{2}\left(2T + \tau\left(-3 + e^{-\frac{2T}{\tau}}\left(4e^{\frac{T}{\tau}} - 1\right)\right)\right) \tag{181}$$

$$Q_{1,2} = q^2\frac{\tau^2}{2}e^{-\frac{2T}{\tau}}\left(e^{\frac{T}{\tau}} - 1\right)^2 \tag{182}$$

$$Q_{2,2} = q^2\frac{\tau}{2}\left(1 - e^{-\frac{2T}{\tau}}\right) \tag{183}$$

- The model keeps the acceleration near zero.

Stochastic Calculus: Linear Dynamic Models

- The Ornstein-Uhlenbeck model is a Gauss-Markov process with $n = 0$.
- The model keeps the position near zero.
- The Ornstein-Uhlenbeck process tends to be used in the literature for certain simulations where targets should not escape.
- The equations are

$$F = e^{-\frac{T}{\tau}} \tag{184}$$

$$Q = q^2 \frac{\tau}{2} \left(1 - e^{-\frac{2T}{\tau}}\right) t) \tag{185}$$

Stochastic Calculus vs. Direct Discrete Models

- Direct discrete-time models exist and do not require stochastic calculus.
- One example is the discrete white-noise acceleration model. In 1D with discrete time k:

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{\Gamma} v \tag{186}$$

$$\mathbf{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad \mathbf{\Gamma} = \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \tag{187}$$

- The term v is scalar noise with variance q^2.
- The term $\mathbf{\Gamma}$ treats the noise as inputs to velocity and acceleration terms in a simple scalar kinematic model

$$\mathbf{x} = \dot{x}T + \frac{1}{2}\ddot{x}T^2 \tag{188}$$

Stochastic Calculus: Process Noise

- The power spectral density term q^2 of the process noise is a design parameter.
- If one typically predicts over a duration of T and the maximum change in the highest order moment of your state (e.g. velocity, acceleration, etc) is v_{MAX}, a rule-of-thumb for most discretized dynamic models is

$$q^2 \approx \frac{\left(\frac{3}{4} v_{\text{MAX}}\right)^2}{T} \tag{189}$$

- Multiple other rules of thumb exist (including for direct discrete models) and are programmed into the processNoiseSuggest function in the TCL.

Stochastic Calculus vs. Direct Discrete Models

- Though simple to derive, direct discrete models should usually be avoided:
  - Predictions are temporally inconsistent when used with varying revisit rates T.
  - This is explained in more detail when considering the Kalman filter prediction.
  - The process noise covariance matrix is singular, which can sometimes cause issues.
- Functions for some linear direct discrete models in the TCL include FVanKeuk. QPolyKalDirectDisc, QPolyKalDirectAlt and QVanKeuk.

U.S. NAVAL RESEARCH LABORATORY

THE LINEAR KALMAN FILTER PREDICTION

The Linear Kalman Filter Prediction

- The stochastic dynamic models describe prediction when the initial state $\mathbf{x}$ is deterministic.
- What if the initial state is uncertain?
- Assume a Gaussian prior at discrete step $k-1$ with mean $\hat{\mathbf{x}}_{k-1|k-1}$ and covariance matrix $\mathbf{P}_{k-1|k-1}$.
- The goal is to predict forward to discrete step k, resulting in a predicted distribution with mean and covariance matrix $\hat{\mathbf{x}}_{k|k-1}$ and covariance matrix $\mathbf{P}_{k|k-1}$.
- The discretized or discrete dynamic model

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{190}$$

where the subscript of $\mathbf{F}$ refers to the discrete time, not the duration of the prediction and $\mathbf{w}_{k-1} \sim \mathcal{N}\{\mathbf{0}, \mathbf{Q}_{k-1}\}$.

- $\mathbf{F}_{k-1}$ and $\mathbf{Q}_{k-1}$ can be for any linear model; they depend on the prediction interval T.

The Linear Kalman Filter Prediction

- The dynamic models provides the conditional prediction distribution:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) \sim \mathcal{N}\left\{\mathbf{F}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}\right\} \tag{191}$$

- However, $\mathbf{x}_{k-1}$ is not deterministic. The Law of Total Probability must be used

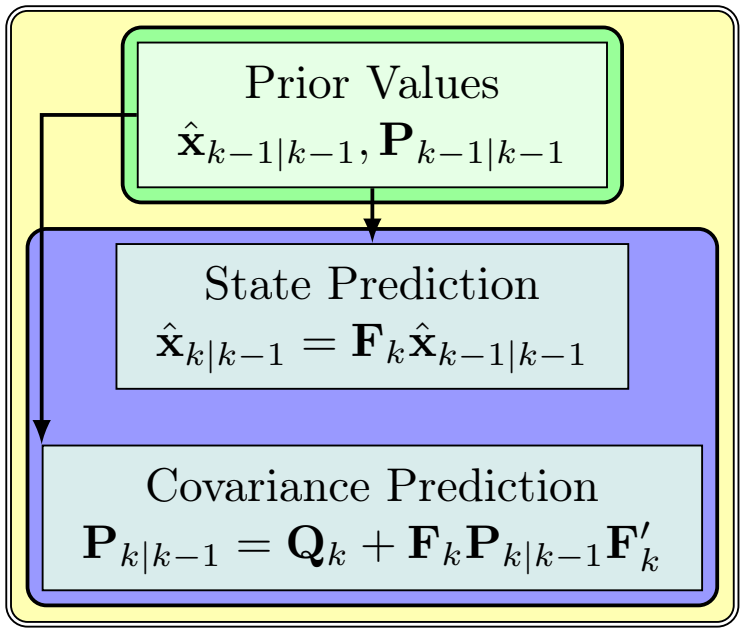
$$p(\mathbf{x}_k) = \int_{\mathbf{x}_{k-1}\in\mathbb{R}^{d_x}} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1})d\mathbf{x}_{k-1} \tag{192}$$

- The prior is $p(\mathbf{x}_{k-1}) \sim \mathcal{N}\left\{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}\right\}$.
- The result is Gaussian with mean and covariance matrix:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{193}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}'_{k-1} \tag{194}$$

The Linear Kalman Filter Prediction Summary



- The final two equations are the prediction step of the standard Kalman filter.
- See the discKalPred function in the TCL.
- We now have tools to approximate measurements as Gaussian, predict states forward and update states with converted measurements.

U.S. NAVAL RESEARCH LABORATORY

Discretized Versus Direct Discrete and the Kalman Filter

- Consider two scenarios:
  - Predicting a target state forward by T using the Kalman filter and then predicting it forward by another T.
  - Predicting a target state forward by $2T$.
- The results of both predictions should be the same.
- If using a discretized dynamic model, they are the same.
- If using a direct discrete model, they are not the same.
- Direct discrete models are problematic with possibly missed detection and variable revisit rates.

U.S. NAVAL RESEARCH LABORATORY

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.

Linear Initial State Estimation

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

$$\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} \hat{\mathbf{z}}_{\text{Cart}} \\ \mathbf{0}_{d_x - d_z} \end{bmatrix} \tag{195}$$

$$\hat{P}_{0|0} = \begin{bmatrix} \mathbf{R}_{\text{Cart}} & \mathbf{0}_{d_z, d_x - d_z} \\ \mathbf{0}_{d_x - d_z, d_z} & \text{diag}([\sigma_1^2, \sigma_2^2, \ldots, \sigma_{d_x - d_z}^2]) \end{bmatrix} \tag{196}$$

    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.

Linear Initial State Estimation

- An unknown state component would have an ∞ value in the corresponding diagonal of its covariance matrix.
- One does not want to use infinite quantities, so variants of the information filter exist.
- Instead of estimating $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$, one estimates $\mathbf{P}_{k|k}^{-1}$ and the information state

$$\hat{\mathbf{y}}_{k|k} = \mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k} \tag{197}$$

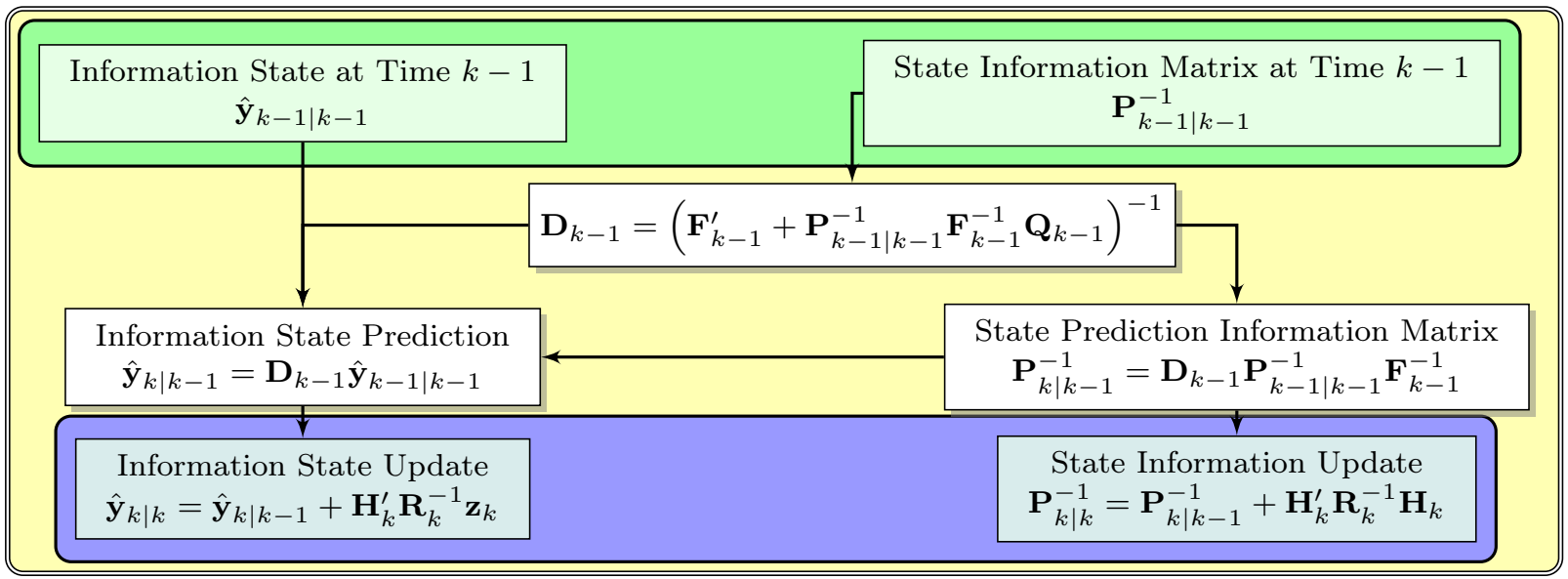
starting with an uninformative state

$$\hat{\mathbf{y}}_{0|-1} = \mathbf{0} \tag{198}$$

$$\mathbf{P}_{0|-1} = \mathbf{0} \tag{199}$$

- The information filter is algebraically equivalent to the Kalman filter.

Linear Initial State Estimation



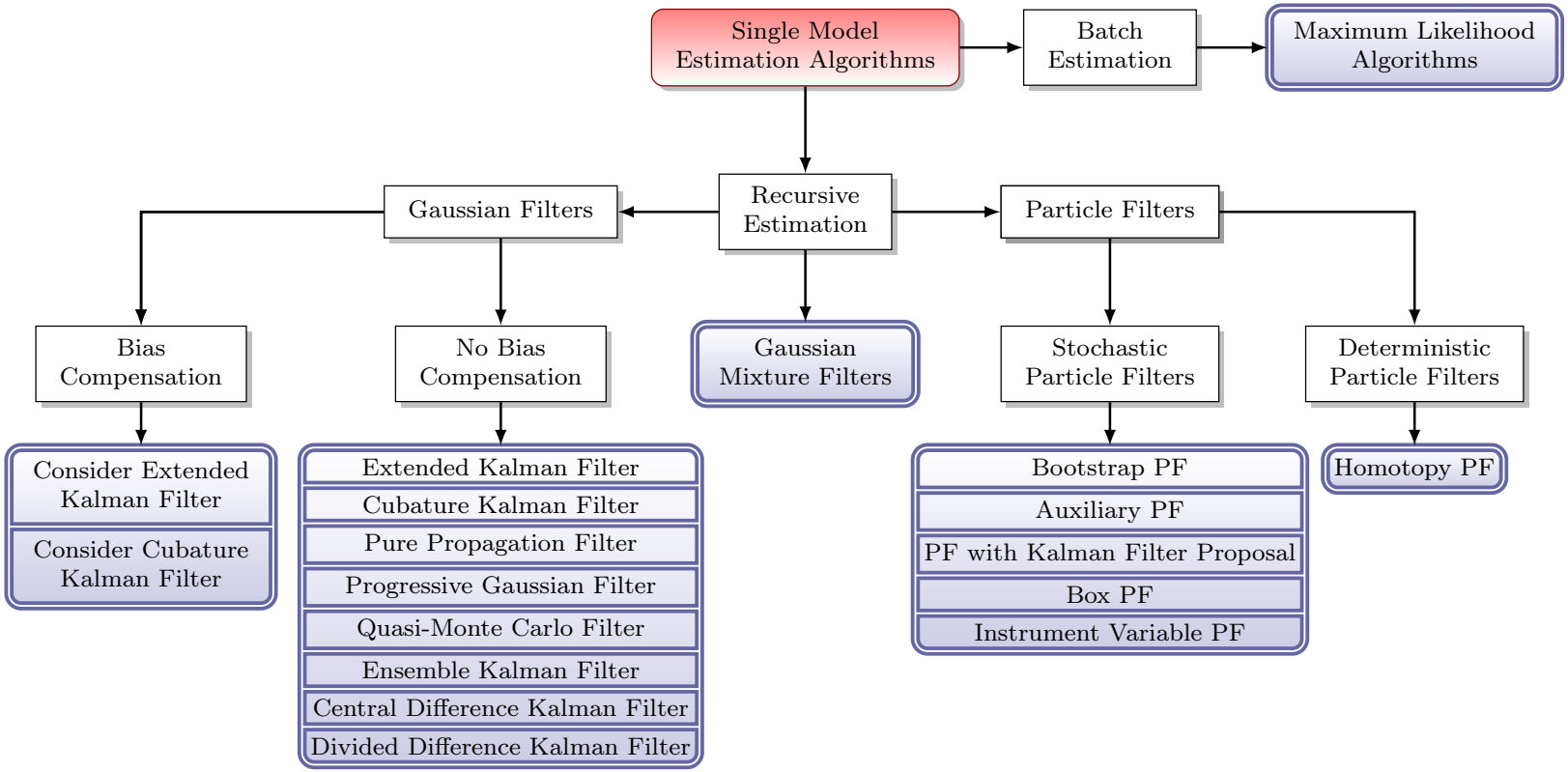
- Multiple variants of the information filter exist; one is shown above, combining prediction and update steps.
- Once enough measurements for state observability have been filtered, one can recover the state:

$$\hat{\mathbf{x}}_{k|k} = \mathbf{P}_{k|k}\hat{\mathbf{y}}_{k|k} \tag{200}$$

U.S. NAVAL RESEARCH LABORATORY

NONLINEAR MEASUREMENT UPDATES

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.

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

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{w} \tag{201}$$

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

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}^{xz}_{k|k-1} \left(\mathbf{P}^{zz}_{k|k-1} \right)^{-1} \left(\mathbf{z} - \hat{\mathbf{z}}_{k|k-1} \right) \tag{202}$$

$$\mathbf{P}_{\downarrow k} = \mathbf{P}_{k|k-1} - \mathbf{P}^{xz}_{k|k-1} \left(\mathbf{P}^{zz}_{k|k-1} \right)^{-1} \mathbf{P}^{zx}_{k|k-1} \tag{203}$$

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

Nonlinear Measurement Updates

The integrals are:

$$\hat{\mathbf{z}}_{k|k-1} = \mathrm{E}\left[\mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k\right] \tag{204}$$

$$= \int_{\mathbb{R}^{d^x}} \mathbf{h}(\mathbf{x}_k) \mathcal{N}\left\{\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right\} d\mathbf{x}_k \tag{205}$$

$$\mathbf{P}^{zz}_{k|k-1} = \mathrm{E}\left[\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right)\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right)'\right] \tag{206}$$

$$= \mathrm{E}\left[\left(\mathbf{w}_k + \mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1}\right)\left(\mathbf{w}_k + \mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1}\right)'\right] \tag{207}$$

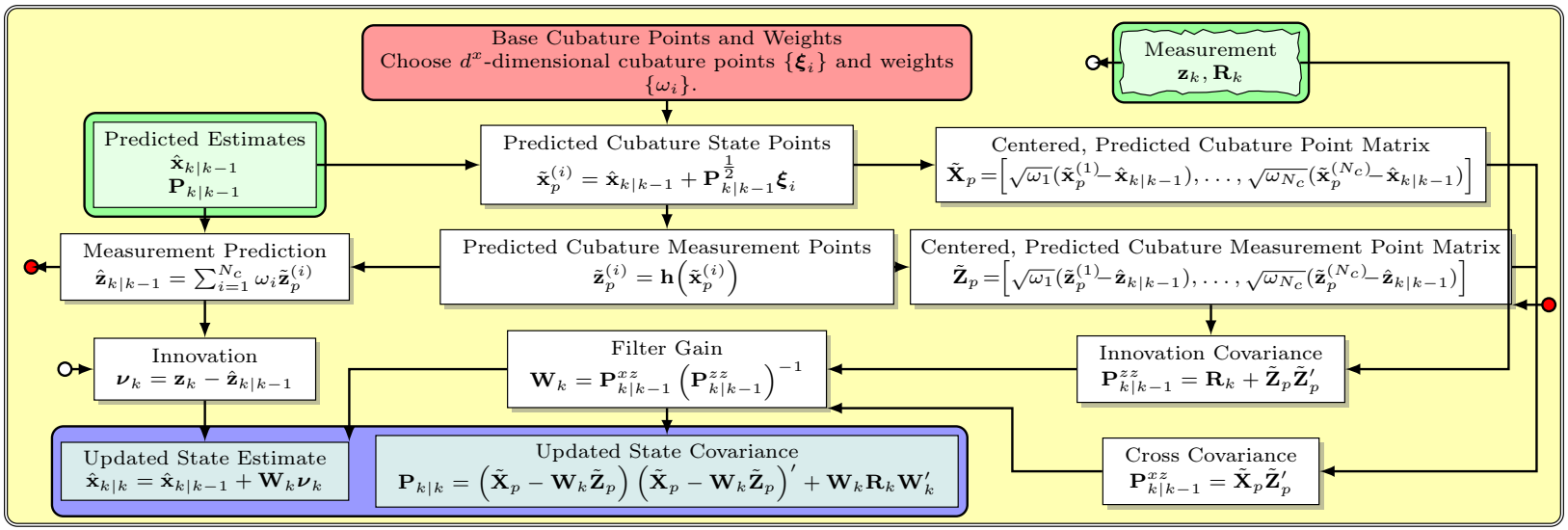
$$= \mathbf{R}_k + \int_{\mathbb{R}^{d^x}} \left(\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1}\right)\left(\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1}\right)' \mathcal{N}\left\{\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right\} d\mathbf{x}_k \tag{208}$$

$$\mathbf{P}^{xz}_{k|k-1} = \mathrm{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right)'\right] \tag{209}$$

$$= \mathrm{E}\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k - \hat{\mathbf{z}}_{k|k-1}\right)'\right] \tag{210}$$

$$= \int_{\mathbb{R}^{d^x}} \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathbf{h}(\mathbf{x}_k) + \hat{\mathbf{z}}_{k|k-1}\right)' \mathcal{N}\left\{\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right\} d\mathbf{x}_k \tag{211}$$

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.

Nonlinear Measurement Updates: EKF

- The Quasi Monte-Carlo Kalman filter is the same as the CKF, but uses Monte Carlo integration for the integrals.
- An alternative approach is to use a Taylor series expansion of the nonlinear function.
- A first-order Taylor series expansion of $\mathbf{h}(\mathbf{x}_k)$ about the point $\hat{\mathbf{x}}_{k|k-1}$ is:

$$\mathbf{h}\left(\mathbf{x}_k\right) \approx \mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right) + \left(\nabla_{\mathbf{x}} \mathbf{h}\left(\mathbf{x}_{k|k-1}\right)'\right)' \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right) \quad (212)$$

- The expected values are thus:

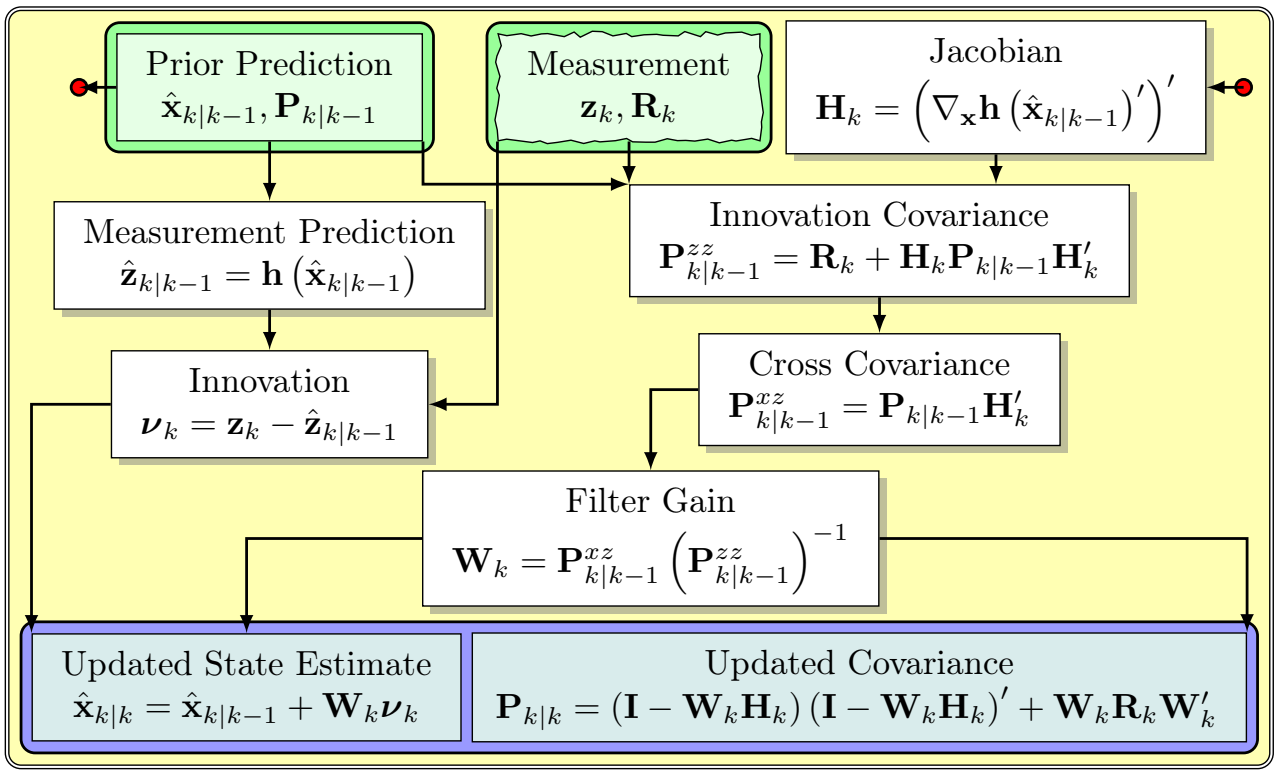
$$\hat{\mathbf{z}}_{k|k-1} = \mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}\right) \quad (213)$$

$$\mathbf{H}_k = \left(\nabla_{\mathbf{x}} \mathbf{h}\left(\mathbf{x}_{k|k-1}\right)'\right)' \quad (214)$$

$$\mathbf{P}^{zz}_{k|k-1} = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k' \quad (215)$$

$$\mathbf{P}^{xz}_{k|k-1} = \mathbf{P}_{k|k-1} \mathbf{H}_k' \quad (216)$$

Nonlinear Measurement Updates: EKF



- The first-order EKF measurement update is shown above.
- A second-order variant is well known but seldom used.

U.S. NAVAL RESEARCH LABORATORY

Nonlinear Measurement Updates: EKF

- The Taylor series expansion of the EKF is about the (unknown) true value $\mathbf{x}_k$.
- Iterative formulations:
  - Perform an update to get an approximate $\hat{\mathbf{x}}_{k|k}$.
  - Redo the Taylor series expansion using the approximate $\hat{\mathbf{x}}_{k|k}$ instead of $\hat{\mathbf{x}}_{k|k-1}$.
  - Redo the update using the new expansion (and the old prior values).
  - Repeat if desired.
- Benefits of second order and iterative formulations tend to be small.
- The EKF is more likely to diverge than a CKF of an adequate order.

U.S. NAVAL RESEARCH LABORATORY

Nonlinear Measurement Updates

- In the EKF and CKF, measurements need not be Cartesian-convertible.
- Once initiated, measurements that cannot be Cartesian converted can often be used.
- Range rate can be used, though benefits tend to only be during initiation and for maneuvering targets.

U.S. NAVAL RESEARCH LABORATORY

SQUARE ROOT FILTERS

U.S. NAVAL RESEARCH LABORATORY

Square Root Filters

- Square root filters propagate a type of square root of the covariance matrix instead of the matrix itself.
- Square root propagation can
  - Reduce finite precision errors.
  - Provide a method to guarantee covariance symmetry and positive (semi-)definiteness.
    - This is a reason to prefer such techniques.
- Two main types of square root filters exist:
  1. Those using a Cholesky-style decomposition.
  2. Those using a LDL' decomposition.
- We consider one form of the first type here.

Square Root Filters

- Instead of $\mathbf{P}_{k|k}$, Cholesky-style filters compute $\mathbf{S}_{k|k}$ such that

$$\mathbf{P}_{k|k} = \mathbf{S}_{k|k}\mathbf{S}'_{k|k} \tag{217}$$

- Such filters also take $\mathbf{S}_k^R$ and $\mathbf{S}_k^Q$ such that

$$\mathbf{R}_k = \mathbf{S}_k^R \left(\mathbf{S}_k^R\right)' \tag{218}$$

$$\mathbf{Q}_k = \mathbf{S}_k^Q \left(\mathbf{S}_k^Q\right)' \tag{219}$$

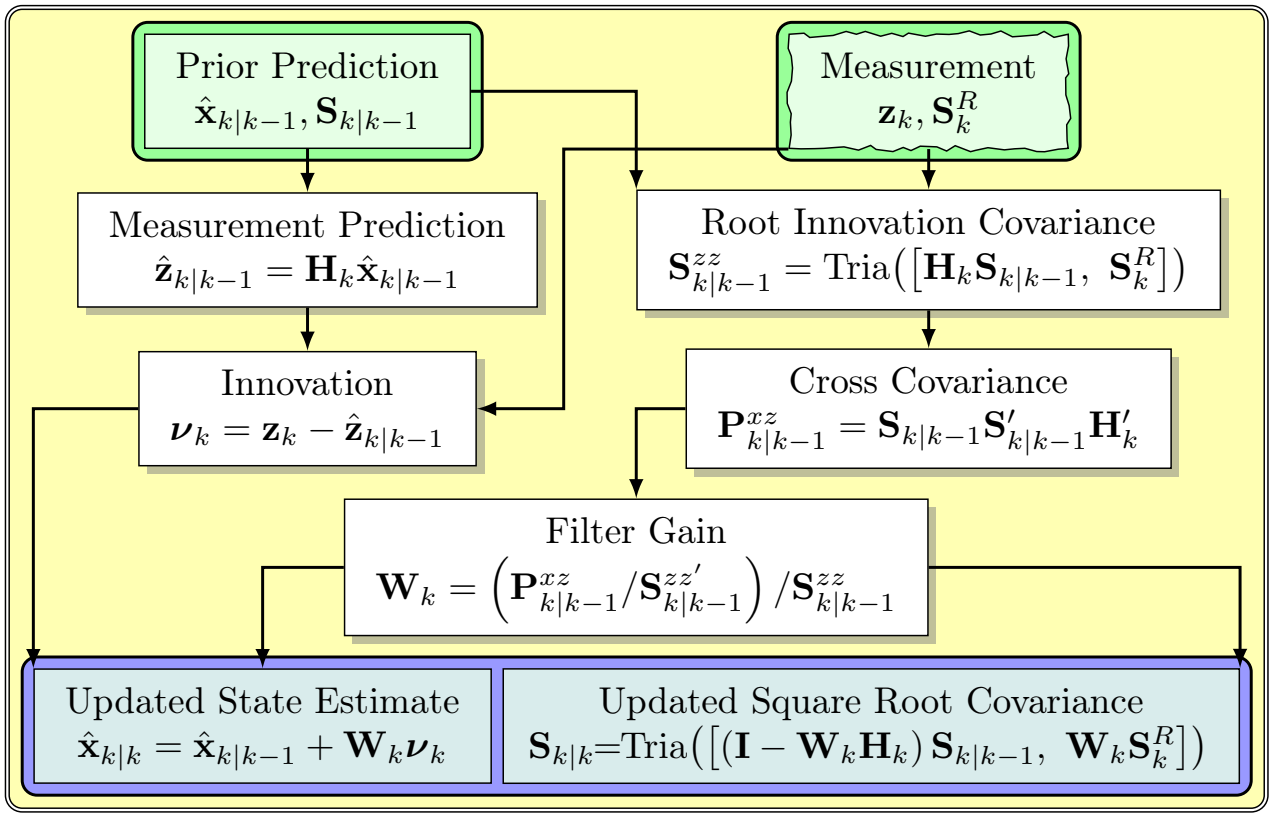
- The chol command with the "lower" option can be used in Matlab.
- If a matrix might be positive semidefinite the cholSemiDef function in the TCL can handle it.
- The conversion to square root form just involves algebraic manipulations.

Square Root Filters

- Expressions for the following filters shall be given in square root form:
  1. Kalman Filter Update
     - See sqrtKalmanUpdate in the TCL.
  2. Kalman Filter Prediction
     - See sqrtDiscKalPred in the TCL.
  3. Cubature Kalman Filter Update
     - See sqrtCubKalUpdate in the TCL.
  4. Extended Kalman Filter Update
     - See sqrtEKFUpdate in the TCL.
- Further (complicated) optimizations might be possible. See Chapter 7 of:

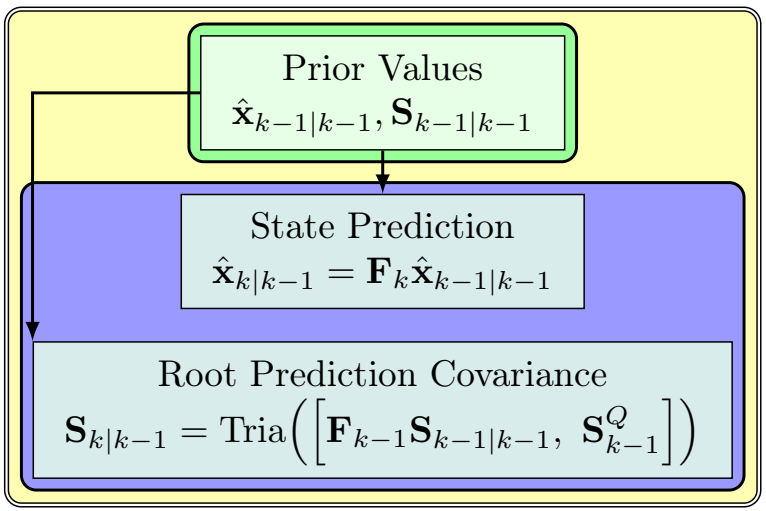
  P. S. Maybeck, Stochastic Models, Estimation, and Control. New York: Academic Press, 1979, vol. 1.

Square Root Filters: Square Root Kalman Filter Update



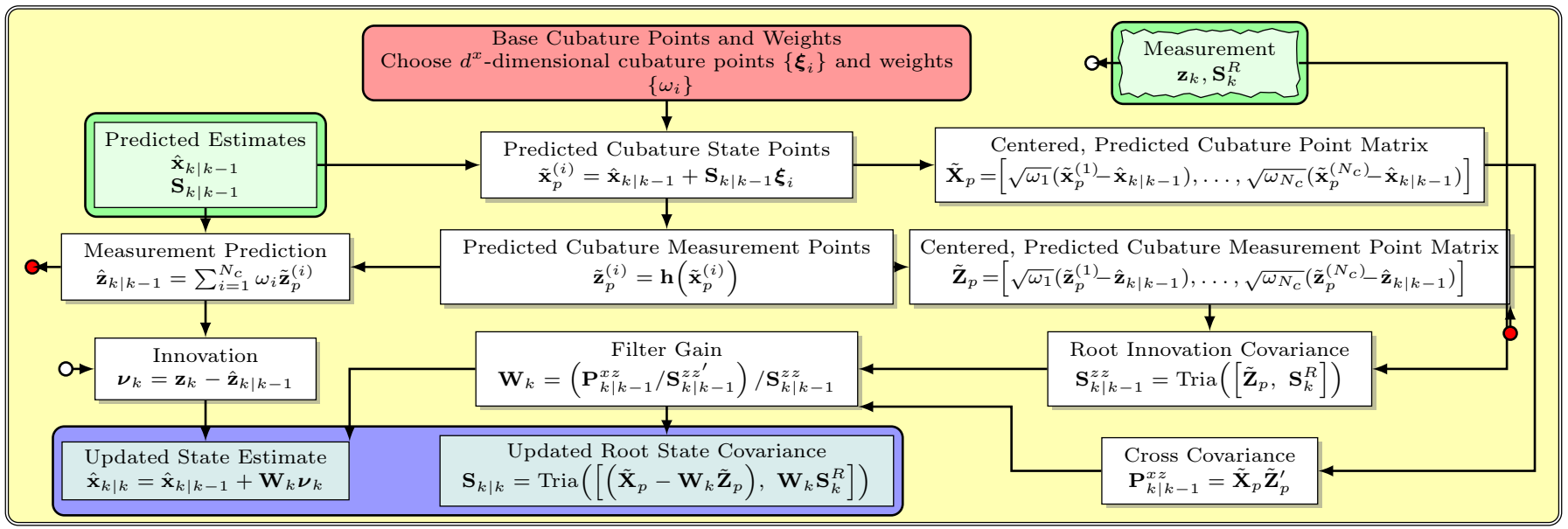
- Square root Kalman filter update.
- $S = \text{Tria}(A)$ uses a Q-R decomposition to obtain a square lower-triangular matrix from the rectangular matrix argument A and can be implemented using [~,R]=qr(A',0);S=R'; It is in the TCL as the tria function.

Square Root Filters: Square Root Kalman Filter Prediction



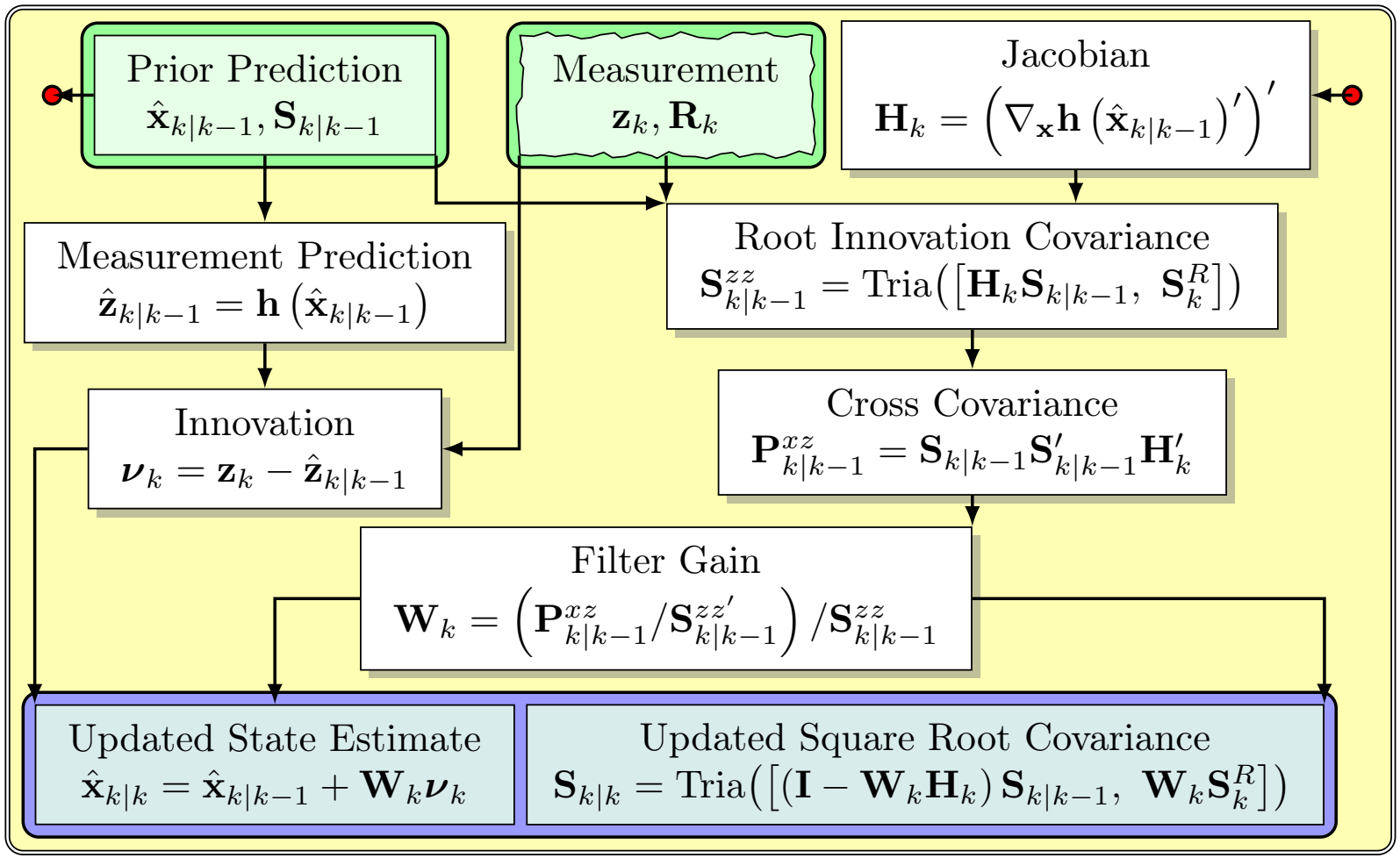
- The discrete square root Kalman filter prediction step.

Square Root Filters: Square Root CKF Update



- The square root CKF update is similar to the CKF update utilizing Joseph's form.

Square Root Filters: Square Root EKF Update



- The square root EKF update.

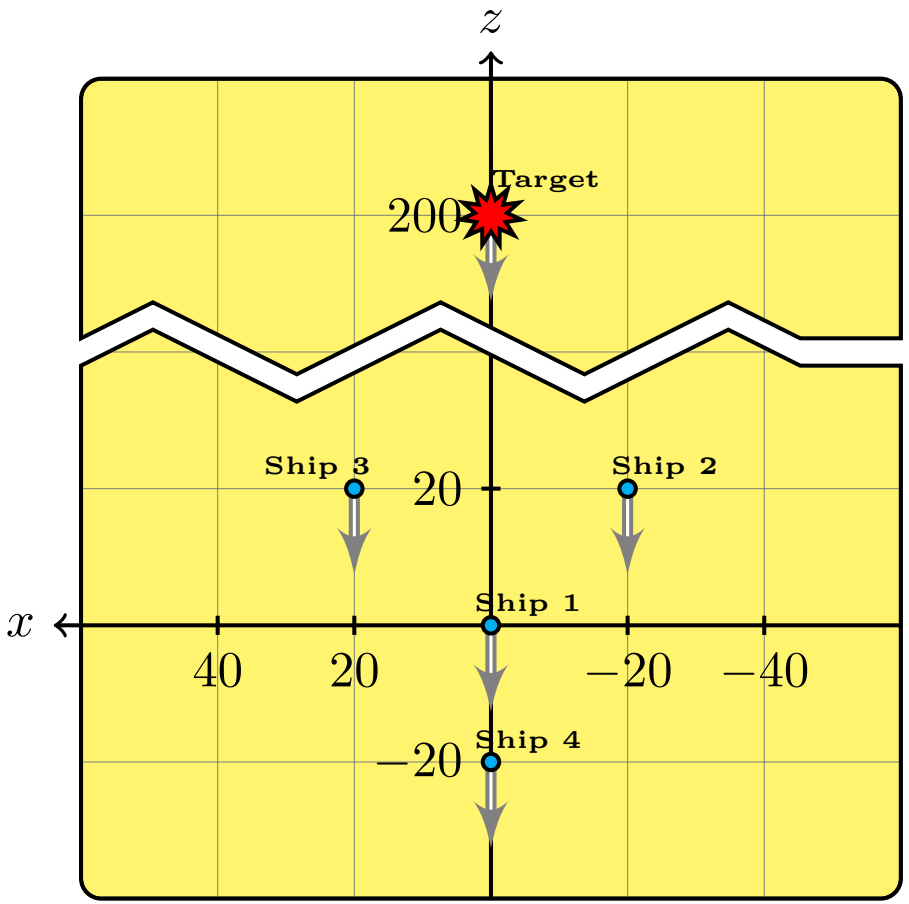
U.S. NAVAL RESEARCH LABORATORY

DIRECT FILTERING VERSUS MEASUREMENT CONVERSION

Filtering Versus Measurement Conversion

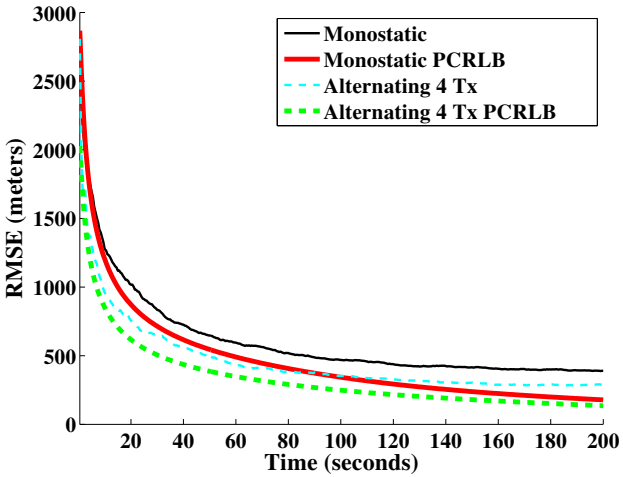
- From what has been covered, two common approaches for basic tracking exist:
  1. Cartesian converting measurements (and covariances) and using a linear filter.
  2. Directly using measurements in a nonlinear filter.
- These shall be compared in a simple example.

Filtering Versus Measurement Conversion

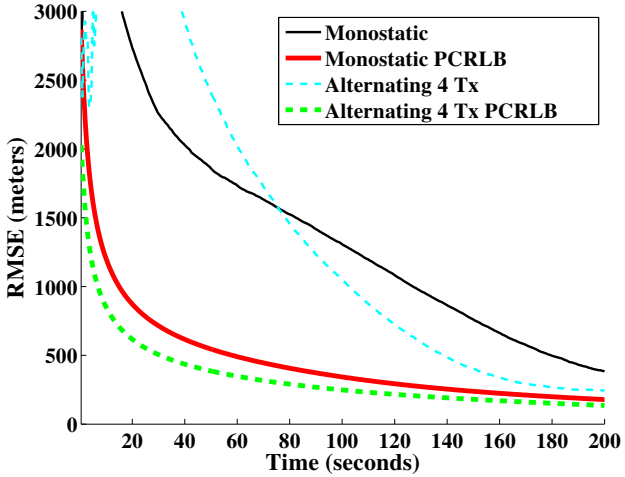


- A flat Earth.
- All ships on the surface traveling $-10\,\mathrm{m/s}$ in the negative z direction.
- The target initially at an altitude of $7\,\mathrm{km}$ going $100\,\mathrm{m/s}$.
- Radars on ships pointed 15° up from the horizontal.
- $\tilde{q} = 0.4802\,\mathrm{m}^2/\mathrm{s}^3$
- Measurements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

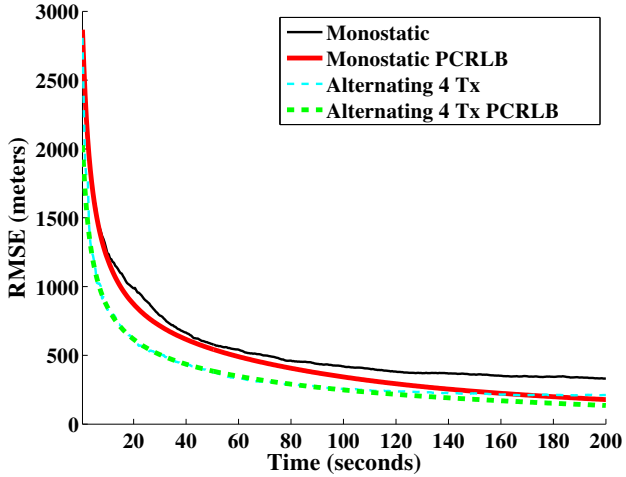
Filtering Versus Measurement Conversion: RMSE



(a) Converted Measurements, RMSE



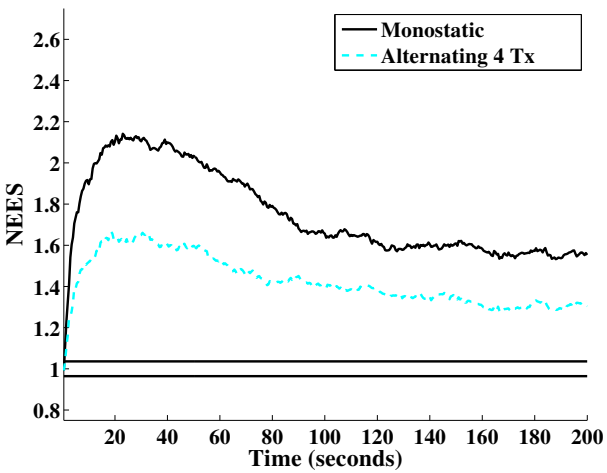
(b) EKF, RMSE



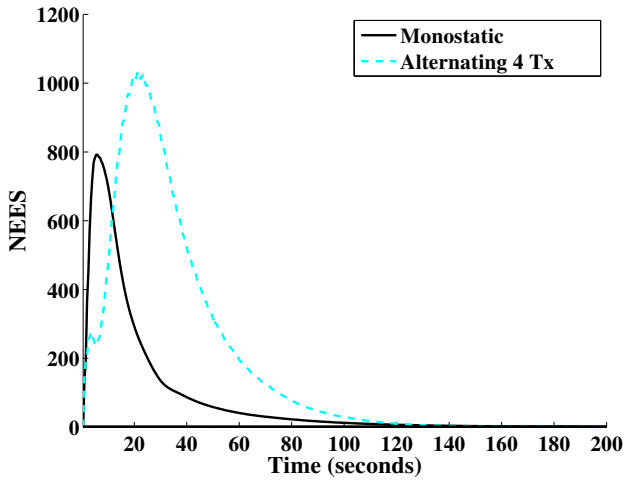
(c) CKF, RMSE

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

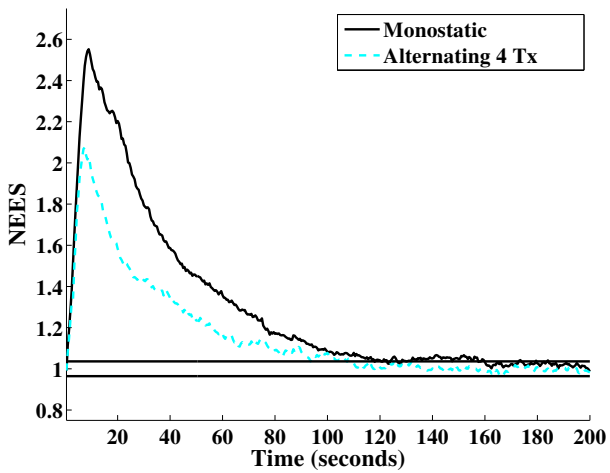
Filtering Versus Measurement Conversion: NEES



(a) Converted Measurements, NEES



(b) EKF, NEES



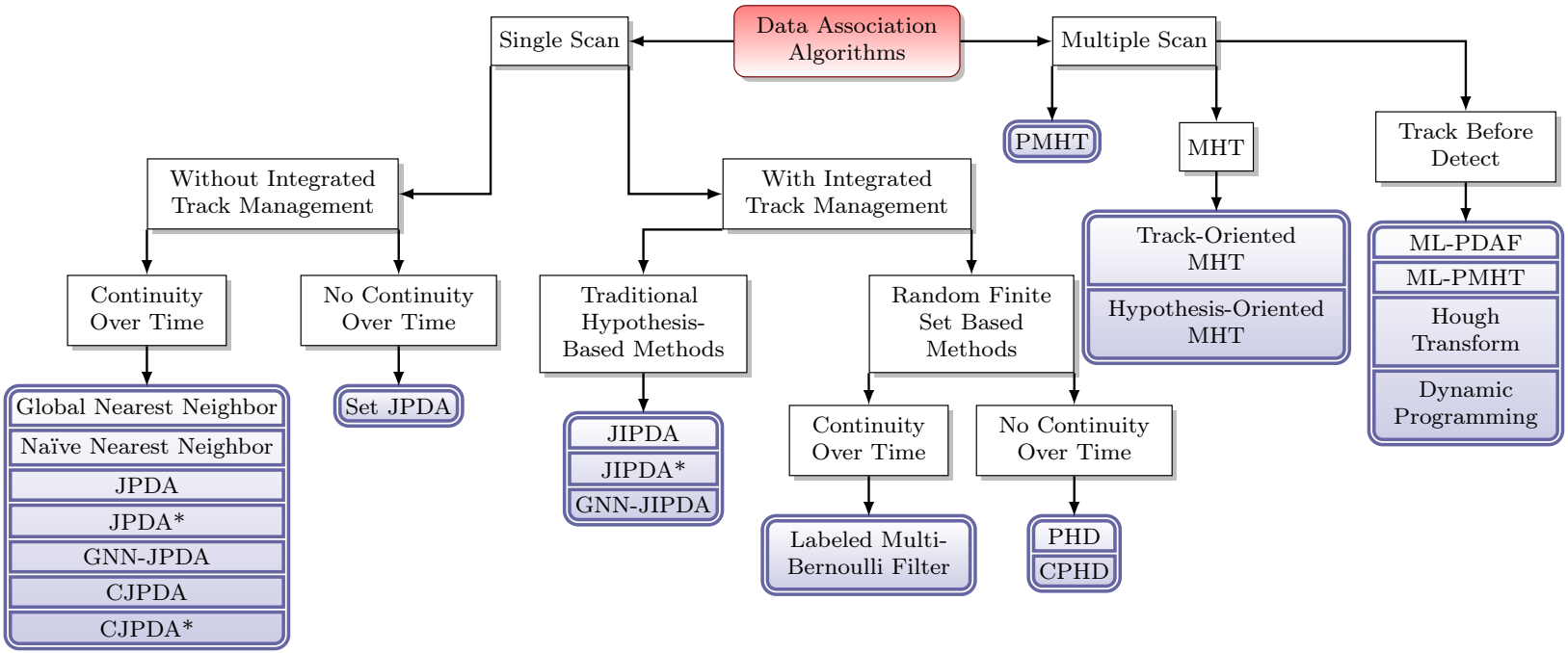
(c) CKF, NEES

- 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

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:

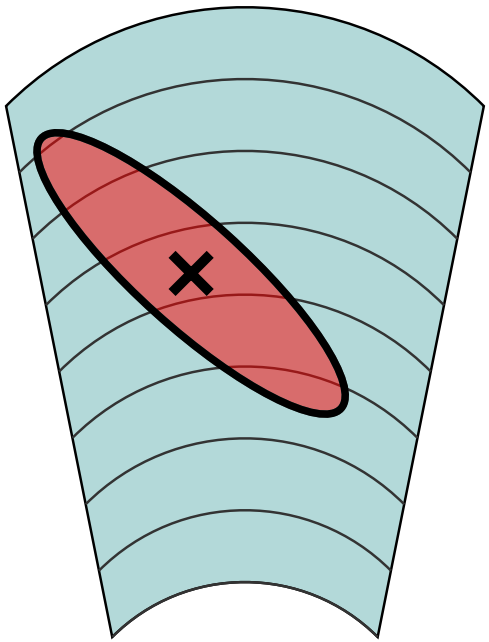
1. The Likelihood Function.
2. Naïve Nearest Neighbor, the Score Function, and Global Nearest Neighbor (GNN)
3. Probabilistic Data Association (PDA) and Joint Probabilistic Data Association (JPDA) variants

U.S. NAVAL RESEARCH LABORATORY

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, which could consist of:
  - Measurements from one sensor at multiple times.
  - Measurements from multiple sensors at one time.
  - Measurements from multiple sensors at multiple times.

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

$$i = \arg\min_i \left\| \mathbf{H}^p\mathbf{x} - \mathbf{z}_i^{\text{Cart}} \right\|^2 \tag{220}$$

- 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

- From the Law of Total Probability:

$$p(\mathbf{z}|\mathbf{Z}^{k-1}, \boldsymbol{\Theta}^{k-1}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x}|\mathbf{Z}^{k-1}, \boldsymbol{\Theta}^{k-1}) \tag{221}$$

- In all of the Kalman filter variants covered thus far

$$p(\mathbf{x}|\mathbf{Z}^{k-1}, \boldsymbol{\Theta}^{k-1}) = \mathcal{N}\left\{\mathbf{x}; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right\} \tag{222}$$

- In the linear Kalman filter, the parameters for the product distribution have already been computed. It can be shown that:

$$p(\mathbf{z}|\mathbf{x})p(\mathbf{x}|\mathbf{Z}^{k-1}, \boldsymbol{\Theta}^{k-1}) = \mathcal{N}\left\{\mathbf{z}; \hat{\mathbf{z}}, \mathbf{P}^{zz}_{k|k-1}\right\} \tag{223}$$

- In the CKF and EKF, the same expression is almost always used to approximate the likelihood.

The Likelihood Function

- Written out, the likelihood of the ith measurement:

$$p(\mathbf{z}_i|\mathbf{Z}^{k-1},\boldsymbol{\Theta}^{k-1}) \triangleq \tilde{\Lambda}\left(\theta^i\right) = \left|2\pi\mathbf{P}^{zz,i}_{k|k-1}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right)'\mathbf{P}^{zz,i}_{k|k-1}\left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right)} \quad (224)$$

- $\mathbf{P}^{zz,i}_{k|k-1}$ 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*:

$$-\log\left(\tilde{\Lambda}\left(\theta^i\right)\right) \propto \left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right)'\mathbf{P}^{zz,i}_{k|k-1}\left(\mathbf{z}-\hat{\mathbf{z}}_{k|k-1}\right) \quad (225)$$

- 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.

The Likelihood Function

- If one target is present and $\mathbf{R}$ is the same for all measurements, the maximum likelihood (ML) assignment has the minimum Mahalanobis distance.
- If $\mathbf{R}$ varies between measurements, this is not the case!
- For a counter example:
  - Consider two scalar Gaussian distributions: $\mathcal{N}\left\{z_1; 0, \sigma_1^2\right\}$ and $\mathcal{N}\left\{z_2; 0, \sigma_2^2\right\}$.
  - Choose $z_1 = 1$, $\sigma_1 = 1$, $\sigma_2^2 = e^{-10}$, $z_2^2 = 8e^{-10}$.
  - One observes that

$$\frac{z_1^2}{\sigma_1^2} < \frac{z_2^2}{\sigma_2^2} \iff 1 < 8 \tag{226}$$

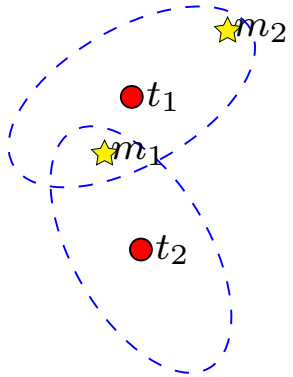
$$\left(2\pi\sigma_1^2\right)^{-\frac{1}{2}} e^{-\frac{z_1^2}{2\sigma_1^2}} < \left(2\pi\sigma_2^2\right)^{-\frac{1}{2}} e^{-\frac{z_2^2}{2\sigma_2^2}} \iff \frac{1}{\sqrt{2\pi e}} < \frac{e}{\sqrt{2\pi}} \tag{227}$$

  - For a predicted measurement of zero, the Mahalanobis distance choice disagrees with the ML choice.

The Likelihood Function

- To choose the ML measurement for assignment:
  - Use the likelihood function if measurement covariance matrices vary.
  - Use the Mahalanobis distance when all measurements have the same covariance matrix.
- The Mahalanobis distance can be used for gating regardless of whether the accuracies vary.
- What if multiple targets are present?

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 here is computed in makeStandardLRMatHyps and makeStandardCartOnlyLRMatHyps in the TCL.
- Define
  - θ as an assignment of measurements to targets at a given time.
  - $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ is the current set of observations.
  - m is the cardinality of $\mathbf{Z}$.
  - I_p is all prior information.
- One can write:

$$\Pr\{\theta, I_p|\mathbf{Z}\} = \Pr\{\theta, I_p|\mathbf{Z}, m\} \quad m \text{ adds no new information} \quad (228)$$

$$= \frac{1}{c_1} p(\mathbf{Z}|m, \theta, I_p) \Pr\{\theta, I_p|m\} \quad \text{Bayes' Theorem} \quad (229)$$

$$= \frac{1}{c_1} p(\mathbf{Z}|m, \theta, I_p) \Pr\{\theta|m, I_p\} \Pr\{I_p|m\} \quad \text{Conditional Probability} \quad (230)$$

The Score Function

- The final simplification is

$$\Pr\{\theta, I_p|\mathbf{Z}\} = \frac{1}{c_1} p(\mathbf{Z}|m, \theta, I_p) \Pr\{\theta|m, I_p\} \Pr\{I_p\} \quad m \text{ adds no info} \tag{231}$$

- Specifications:
  - The prior information I_p informs about N_T targets.
  - The function $\xi_t(\theta)$ return the observations associated with target t under θ or the empty set $\emptyset$ if nothing is assigned.
  - The function $\bar{\xi}(\theta)$ returns all false alarms and $\bar{\xi}_i(\theta)$ the ith one.
- The PDF of the first term of $\Pr\{\theta, I_p|\mathbf{Z}\}$ can be written

$$p(\mathbf{Z}|m, \theta, I_p) = p_c\left(\bar{\xi}(\theta)|m, \theta, I_p\right) \prod_{i=1}^{N_T} p_t\left(\xi_t(\theta)|m, \theta, I_p\right) \tag{232}$$

- p_c is a false alarm (clutter) PDF, and p_t is a target PDF.

The Score Function

- The target PDF is

$$p_t\left(\xi_t(\theta)|m,\theta,I_p\right) = \begin{cases} 1 & \text{if } \xi_t(\theta) = \oslash \\ \tilde{p}_t\left(\xi_t(\theta)|m,\theta,I_p\right) & \text{otherwise} \end{cases} \tag{233}$$

  where $\tilde{p}_t\left(\xi_t(\theta)|m,\theta,I_p\right)$ equals the previously developed likelihood of a measurement assigned to a target given the predicted state.

- If all false alarms are independent,

$$p_c\left(\bar{\xi}(\theta)|m,\theta,I_p\right) = \prod_{i=1}^{\phi(\theta)} p_c\left(\bar{\xi}_i(\theta)|m,\theta,I_p\right). \tag{234}$$

  where $\phi(\theta)$ is the number of false alarms in hypothesis θ.

The Score Function

- Define $\delta(\theta)$ as a set of Boolean detection indicators for the association hypothesis θ such that $\delta_t(\theta) = 1$ if target t was observed.
- The PDF of the second term of $\Pr\{\theta, I_p|\mathbf{Z}\}$ can be written

$$\Pr\{\theta|m, I_p\} = \Pr\{\theta|m, I_p, \delta(\theta), \phi(\theta)\} \Pr\{\delta(\theta), \phi(\theta)|m, I_p\} \tag{235}$$

  using the Law of Total Probability.
- The first term of (235) is the probability of an association of measurements to targets given one knows which targets have been observed and how many measurements are present.

The Score Function

- Without the actual measurements given and with I_p presumably uninformative, all possibilities are equiprobable:

$$\Pr\{\theta|m, I_p, \delta(\theta), \phi(\theta)\} = \left(\overbrace{\binom{m}{m-\phi(\theta)}}^{\text{Choose measurements originating from targets}} \overbrace{(m-\phi(\theta))!}^{\text{Assign chosen measurements to targets}} \right)^{-1} \tag{236}$$

$$= \frac{\phi(\theta)!}{m!} \tag{237}$$

where it was assumed that $m - \phi(\theta) \geq 0$, i.e. that the hypothesis in question is valid.

The Score Function

- Suppressing the argument (θ) in δ and ϕ, and letting P_D^t be the detection probability of target t, the second term of $\Pr\{\theta|m, I_p\}$ is

$$\Pr\{\delta, \phi|m, I_p\} = \Pr\{\delta|m, \phi, I_p\} \Pr\{\phi|m, I_p\} \tag{238}$$

$$= \left(\frac{\Pr\{\delta \bigcap (m-\phi) \text{ targets seen}|\phi, I_p\}}{\Pr\{(m(k)-\phi) \text{ targets seen}|\phi, I_p\}} \right) \cdot \left(\frac{\Pr\{m|\phi, I_p\} \Pr\{\phi\}}{\Pr\{m\}} \right) \tag{239}$$

$$= \left(\frac{\prod_{t=1}^{N_T} \left(P_D^t\right)^{\delta_t} \left(1-P_D^t\right)^{1-\delta_t}}{\Pr\{(m(k)-\phi) \text{ targets seen}|\phi, I_p\}} \right) \cdot \left(\frac{\Pr\{(m(k)-\phi) \text{ targets seen}|\phi, I_p\} \Pr\{\phi\}}{\Pr\{m\}} \right) \tag{240}$$

$$= \frac{\Pr\{\phi\}}{\Pr\{m\}} \prod_{t=1}^{N_T} \left(P_D^t\right)^{\delta_t} \left(1-P_D^t\right)^{1-\delta_t} \tag{241}$$

- In(240), the prior information I_p informs on P_D^t.

The Score Function: P_D^t

- P_D^t can be set as a nominal design parameter.
  - Never set P_D^t to 1.
- Often, P_D^t is approximated based on observed target amplitudes over time.
- P_D^t can also be estimated by fitting to a stochastic radar cross section model and using the radar range equation.
- The Swerling models in "Mathematical Functions/Statistics/Detection Statistics/" in the TCL can help determine P_D^t from stochastic target power models.

The Score Function

- Combining expressions for the terms of $\Pr\{\theta|m, I_p\}$, one gets

$$\Pr\{\theta|m, I_p\} = \frac{\phi!}{m!}\frac{\Pr\{\phi\}}{\Pr\{m\}}\prod_{t=1}^{N_T}\left(P_D^t\right)^{\delta_t}\left(1-P_D^t\right)^{1-\delta_t} \quad (242)$$

- Substituting that and the other terms back into the expression for $\Pr\{\theta, I_p|Z\}$ leads to

$$\Pr\{\theta, I_p|Z\} = \frac{\phi!\Pr\{\phi\}}{c_2}\prod_{i=1}^{\phi} p_c\left(\bar{\xi}_i|m, \theta, I_p\right)$$

$$\cdot\prod_{t=1}^{N_T}\left(P_D^t p_t\left(\xi_t|m, \theta, I_p\right)\right)^{\delta_t}\left(1-P_D^t\right)^{1-\delta_t}\Pr\{I_p\} \quad (243)$$

where missing normalization terms are in c_2.

The Score Function

- Dividing out the clutter PDF for measurements assigned to targets:

$$\Pr\{\theta, I_p|\mathbf{Z}\} = \frac{\phi!\Pr\{\phi\}}{c_3}\Pr\{I_p\} \cdot \prod_{t=1}^{N_T}\left(P_D^t \frac{p_t\left(\xi_t|m,\theta,I_p\right)}{p_c\left(\xi_t|m,\theta,I_p\right)}\right)^{\delta_t}\left(1-P_D^t\right)^{1-\delta_t} \tag{244}$$

- We want to get rid of the $\phi!\Pr\{\phi\}$ term.
- Assume a constant false alarm density λ over the observation volume V, with false alarms from a Poisson process. Then

$$\Pr\{\phi\} = e^{-\tilde{\lambda}}\frac{\tilde{\lambda}^{\phi}}{\phi!}. \tag{245}$$

with $\tilde{\lambda} = \lambda V$.

The Score Function

- Substituting the false alarm model:

$$\Pr\{\theta, I_p|Z\} = \frac{1}{c_4}\Pr\{I_p\}\prod_{t=1}^{N_T}\left(P_D^t\frac{p_t\left(\xi_t|m,\theta,I_p\right)}{(\lambda V)p_c\left(\xi_t|m,\theta,I_p\right)}\right)^{\delta_t}(1-P_D^t)^{1-\delta_t} \tag{246}$$

- It is generally undesirable to have to compute the volume V. The volume can be eliminated assuming that false alarms are uniformly distributed in the observation region:

$$p_c\left(\xi_t|m,\theta,I_p\right) = \frac{1}{V} \tag{247}$$

The Score Function

- The result of a uniform false alarm model is:

$$\Pr\{\theta, I_p|Z\} = \frac{1}{c_4}\Pr\{I_p\}\prod_{t=1}^{N_T}\left(P_D^t\frac{p_t\left(\xi_t|m,\theta,I_p\right)}{\lambda}\right)^{\delta_t}(1-P_D^t)^{1-\delta_t} \tag{248}$$

- Taking the logarithm of $\Pr\{\theta, I_p|Z\}$ and discarding constant terms, one gets

$$\ln\{\Pr\{\theta, I_p|Z\}\} \propto \Lambda(\theta) = \sum_{t=1}^{N_T}\delta_t \ln\left(P_D^t\frac{p_t\left(\xi_t|m,\theta,I_p\right)}{\lambda}\right) + (1-\delta_t)\ln(1-P_D^t) \tag{249}$$

The Score Function

- The marginal change in the log-likelihood for assigning target t to measurement i ($i = 0$ is false alarm) is

$$\Delta\Lambda_{t,i} = \begin{cases} \ln\left(P_D^t \dfrac{p_t\left(\mathbf{z}_i | I_p\right)}{\lambda} \right) & \text{if } i \neq 0 \\ \ln(1 - P_D^t) & \text{if } i = 0 \end{cases} \tag{250}$$

- The previous information I_p is taken to be a set of assignments of past measurements to targets.
  - $p_t\left(\xi_t | m, \theta, I_p\right)$ is the previously developed likelihood function.

The Score Function

- Assuming a Gaussian prior, the marginal change in the log-likelihood is

$$\Delta\Lambda_{t,i} = \begin{cases} \ln\left(P_D^t \dfrac{\mathcal{N}\left\{ \mathbf{z}_i, \hat{\mathbf{z}}_{k|k-1}^t, \mathbf{P}_{k|k-1}^{zz,i,t} \right\}}{\lambda} \right) & \text{if } i \neq 0 \\ \ln(1 - P_D^t) & \text{if } i = 0 \end{cases} \tag{251}$$

  - $\hat{\mathbf{z}}_{k|k-1}^t$ is the predicted measurement from the tth target,
  - $\mathbf{P}_{k|k-1}^{zz,i,t}$ 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.
- We want to derive what the Cartesian version of λ given λ in the measurement coordinate system.

The Score Function: Converting λ

- Say $\mathbf{x} = \mathbf{f}_x(\mathbf{y})$.
- Volume integrals in each system are related by

$$\int_{\mathbf{x}\in\mathbb{S}_x} d\mathbf{x} = \int_{\mathbf{y}\in\mathbb{S}_y} |\mathbf{J}(\mathbf{y})| d\mathbf{y} \tag{252}$$

where $\mathbb{S}_x$ and $\mathbb{S}_y$ are the same regions, but in different coordinate systems and $\mathbf{J}$ is the Jacobian:

$$\mathbf{J}(\mathbf{y}) = \left[\frac{\partial}{\partial y_1} f_x(\mathbf{y}), \frac{\partial}{\partial y_2} f_x(\mathbf{y}), \ldots, \frac{\partial}{\partial y_{d_y}} f_x(\mathbf{y})\right] \tag{253}$$

The Score Function: Converting λ

- Over a miniscule volume,
$$V_x = |\mathbf{J}(\mathbf{y})|\, V_y \tag{254}$$
- Thus, given λ in local (y) coordinates, the value in Cartesian (x) coordinates is:
$$\lambda_x = \frac{1}{|\mathbf{J}(\mathbf{y})|}\lambda_y \tag{255}$$
- In the TCL, the functions makeStandardLRMatHyps and makeStandardCartOnlyLRMatHyps can compute all marginal likelihood values, including conversion effects.
- In the TCL, necessary Jacobians are in "Coordinate Systems/Jacobians/Converted Jacobians" and include calcRuvConvJacob and calcPolarConvJacob, among others.
- For unconverted range rate components, the TCL has functions including calcRuvRRConvJacob and calcPolarRRConvJacob.

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?

$$N_{\text{hyp}} = \underbrace{\sum_{l=0}^{\min(m,N_T)}}_{\substack{\text{Sum over the number} \\ \text{of targets observed}}} \overbrace{\binom{N_T}{l}}^{\substack{\text{Choose which targets} \\ \text{are observed}}} \underbrace{\binom{m}{l}}_{\substack{\text{Choose which measurements} \\ \text{are not false alarms}}} \overbrace{l!}^{\substack{\text{Assign the measurements} \\ \text{to the targets}}} \tag{256}$$

- 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.

U.S. NAVAL RESEARCH LABORATORY

GNN Assignment

- The most power supercomputer (Tianhe-2, China) is 33.85 petaflops.
- A lower bound of execution time is one floating point operation per hypothesis.
- It would take over 3.88×10^{9106} years to visit each hypothesis.
- However, one can optimally solve the assignment problem in under 4 seconds (worst case) on an old laptop.
- In much of tracking, one cannot formulate an optimal solution and tell a first semester programmer to make it fast using a GPU or FPGA.
- Techniques from theoretic computer science are required for computational feasibility.

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:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \sum_{i=1}^{N_R} \sum_{j=1}^{N_C} \Delta\Lambda_{i,j} x_{i,j} \tag{257}$$

$$\text{subject to} \sum_{j=1}^{N_C} x_{i,j} = 1 \quad \forall i \qquad \text{Every target is assigned to an event.} \tag{258}$$

$$\sum_{i=1}^{N_R} x_{i,j} \leq 1 \quad \forall j \qquad \text{Not every event is assigned to a target.} \tag{259}$$

$$x_{i,j} \in \{0,1\} \quad \forall x_{i,j} \qquad \text{Equivalent to } x_{i,j} \geq 0 \quad \forall x_{i,j} \tag{260}$$

- $N_R = N_T$ 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

$$\mathbf{C}_l \triangleq \overbrace{\begin{bmatrix} \Delta\Lambda_{1,1} & \dots & \Delta\Lambda_{1,m} \\ \Delta\Lambda_{2,1} & \dots & \Delta\Lambda_{2,m} \\ \vdots & \ddots & \vdots \\ \Delta\Lambda_{N_T,1} & \dots & \Delta\Lambda_{N_T,m} \end{bmatrix}}^{\text{Assignment Costs}} \overbrace{\begin{bmatrix} \Delta\Lambda_{1,0} & -\infty & \dots & -\infty \\ -\infty & \Delta\Lambda_{2,0} & \dots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ -\infty & -\infty & \dots & \Delta\Lambda_{N_T,0} \end{bmatrix}}^{\text{Missed Detection Costs}}. \tag{261}$$

- 2D assignment is a binary integer programming problem.
- Integer programming problems often cannot be solved in polynomial time (NP-complete).
- The 2D assignment problem can be solved in polynomial time.

GNN Assignment

- Consider assignments of n targets to measurements without false alarms and missed detections. The complexities of popular algorithms are:
  - The Munkres algorithm $O(n^4)$.
  - The Jonker-Volgenant (JV) algorithm $O(n^3)$.
  - The Auction Algorithm $O(Cn^3)$.
    - Weakly polynomial. C can make it arbitrarily bad, or produce suboptimal results.
  - The ϵ-scaled Auction Algorithm $O(n^3)$.
    - Not as fast as the JV algorithm.
- The auction algorithm is easy to understand.
- The auction algorithm appears in many textbooks.
- Do not use the auction algorithm.
- A modified JV algorithm is implemented as assign2D and kBest2DAssign and is an option in singleScanUpdate in the TCL.

U.S. NAVAL RESEARCH LABORATORY

GNN Assignment

- The Jonker-Volgenant-Castañon (JVC) algorithm is popular in the tracking literature.
  - Castañon published a rectangular (possible missed detections) JV modification.
  - The JVC algorithm is popular because code is provided.
  - The algorithm initializes a rectangular JV algorithm with an auction algorithm.
  - The ϵ term in the auction algorithm can cause the JV algorithm to sometimes produce suboptimal results.
  - The auction algorithm initialization is not necessary; don't use the JVC algorithm.
- A rectangular modification to the JV algorithm, is derived in
  D. F. Crouse, "On Implementing 2D Rectangular Assignment Algorithms," IEEE Transactions on Aerospace and Electronic Systems, vol. 52, no. 4, pp. 1679-1696, Aug. 2016.
- The derivation is lengthy, involves Lagrangian relaxation and dual-primal optimization and is omitted here.

U.S. NAVAL RESEARCH LABORATORY

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

$$\mathbf{x}_{k|k}^{t} = \mathrm{E}\left\{\mathbf{x}_{k}^{t} \,|\mathbf{Z}, I_p\right\} = \sum_{i=0}^{m} \beta^{i,t} \hat{\mathbf{x}}_{k|k}^{t,i} \tag{262}$$

$$\mathbf{P}_{k|k}^{t} = \mathrm{E}\left\{ \left(\mathbf{x}_{k}^{t} - \hat{\mathbf{x}}_{k|k}^{t}\right)\left(\mathbf{x}_{k}^{t} - \hat{\mathbf{x}}_{k|k}^{t}\right)' \middle| \mathbf{Z}, I_p \right\} \tag{263}$$

$$= \sum_{i=0}^{m} \beta^{i,t} \left(\mathbf{P}_{k|k}^{t,i} + \left(\mathbf{x}_{k}^{t,i} - \hat{\mathbf{x}}_{k|k}^{t}\right)\left(\mathbf{x}_{k}^{t,i} - \hat{\mathbf{x}}_{k|k}^{t}\right)' \right) \tag{264}$$

  - $\beta_{i,t}$ 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 $\mathbf{P}_{k|k}^{t}$ that is not quadratic in form and subject to finite precision errors.

U.S. NAVAL RESEARCH LABORATORY

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.
  - JPDA variants including cross terms begin with a C for coupled.
  - Including the cross terms worsens the performance (worsens track coalescence).
- The hardest part of the PDA/JPDA is the computation of the β values.

The PDA and JPDA Algorithms

- TIt was shown that the probability going into the score function can be expanded:

$$\Pr\{\theta, I_p|\mathbf{Z}\} = \frac{1}{c_1} p(\mathbf{Z}|m, \theta, I_p) \Pr\{\theta|m, I_p\} \Pr\{I_p\} \quad (265)$$

- In the JPDA, we assume that the prior information is fixed (Gaussian distributions, not multiple hypotheses). Thus, $\Pr\{I_p\} = 1$, so

$$\Pr\{\theta, I_p|\mathbf{Z}\} = \frac{1}{c_1} p(\mathbf{Z}|m, \theta, I_p) \Pr\{\theta|m, I_p\} \quad (266)$$

$$= \Pr\{\theta|\mathbf{Z}, I_p\} \quad (267)$$

- This means that the previously derived $\Pr\{\theta, I_p|\mathbf{Z}\}$ can be used to compute the β terms.

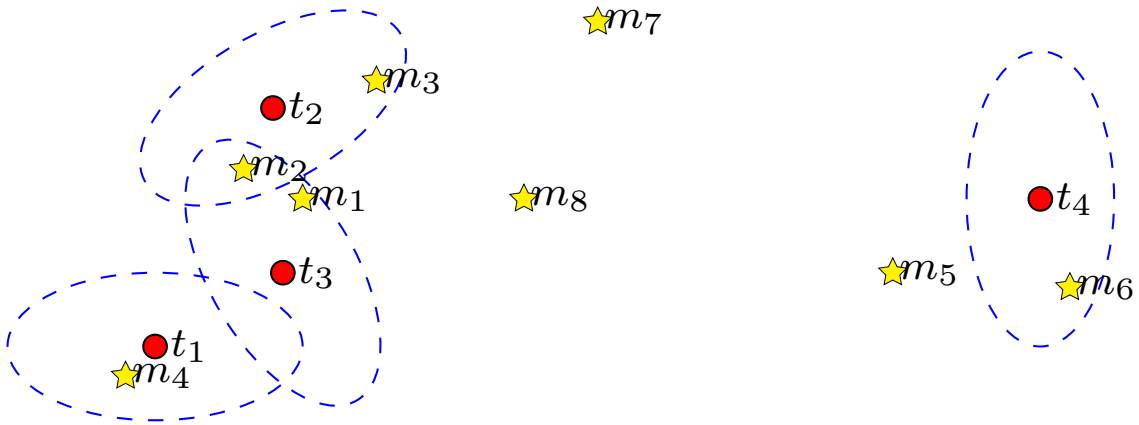
The PDA and JPDA Algorithms

- The probabilities $\Pr\{\theta|\mathbf{Z}, I_p\}$ are called joint association event probabilities.
- The target-measurement association probabilities are sums of joint association event probabilities:

$$\beta^{i,t} = \sum_{\theta:\xi_t(\theta)=i} \Pr\{\theta|\mathbf{Z}, I_p\} \tag{268}$$

- These can only be computed exactly for small number of targets in clusters.
- Approximations must be used if cluster sizes are too large.

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.

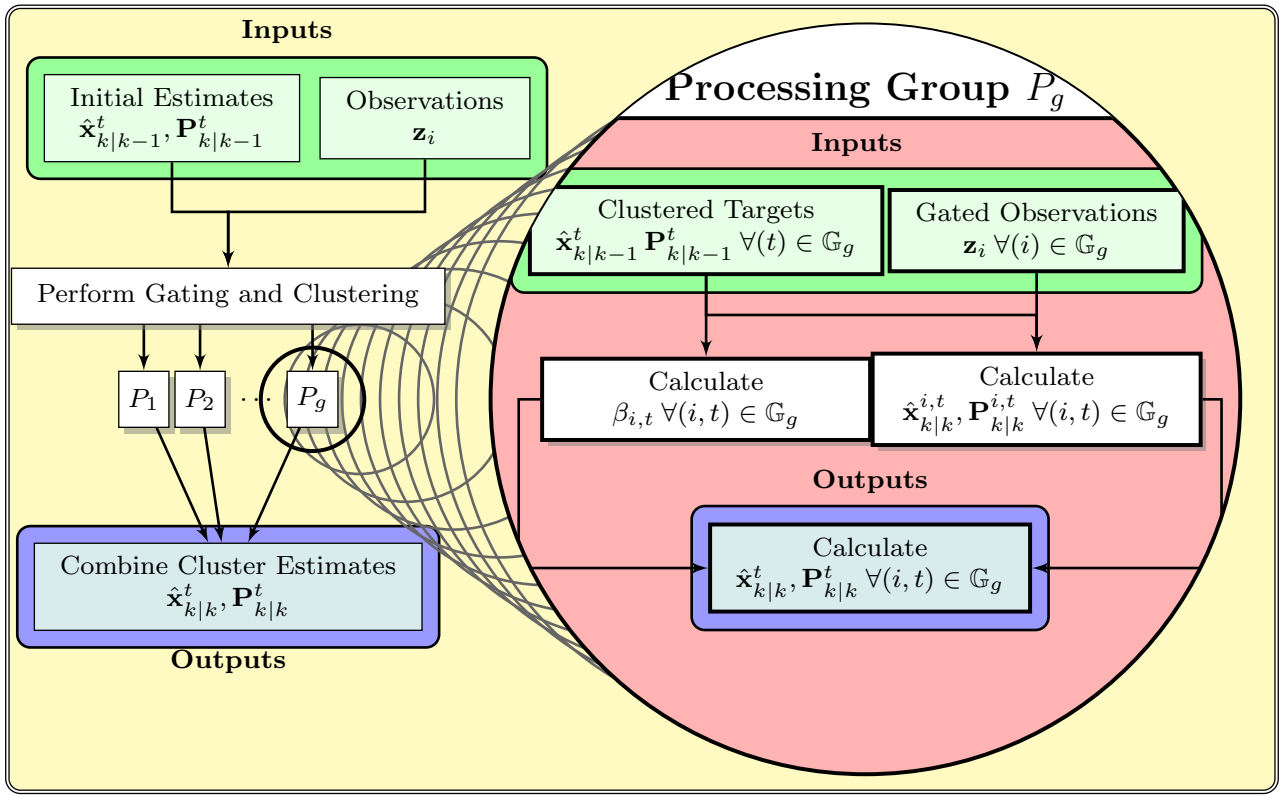
The PDA and JPDA Algorithms

- Targets are in a common cluster if they contest a measurement.
- Measurements are associated with clusters.
- Separate clusters of targets/ measurements can be processed by separate JPDAs.
- The β values for small clusters can be determined exactly.
- The β values of large clusters must be approximated.

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

- Brute-force evaluation adds all joint association events θ and adds the (usually unnormalized) probabilities $\Pr\{\theta|\mathbf{Z}, I_p\}$ to the appropriate $\beta_{i,t}$ terms.
- Normalization can be done afterwards. If $\tilde{\beta}_{i,t}$ are the unnormalized association probabilities, then

$$\beta_{i,t} = \frac{\tilde{\beta}_{i,t}}{\sum_{j=1}^{m} \tilde{\beta}_{j,t}} \tag{269}$$

- The brute force method can be implemented as four nested 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 PDA and JPDA Algorithms: Computing β

- To compute the necessary combinations and permutations for the brute-force method, many of the functions in the folder "Mathematical Functions/Combinatorics" of the TCL can be used. These include genNextCombo and unrankCombination, and unrankPermutation, among many others.
- The determinant of a square matrix is often taught as

$$|\mathbf{A}| = \sum_{\sigma \in \mathbb{S}^n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i} \tag{270}$$

where $\mathbb{S}^n$ is the set of all permutations of the values 1 to n and $\operatorname{sign}(\sigma)$ is the sign (a.k.a. signature or parity) of the permutation.

The PDA and JPDA Algorithms: Computing β

- A permutation can be expressed as a series of swaps (transpositions, inversions) of adjacent elements. The sign of the permutation is related to the number of inversion as:

$$\operatorname{sgn}(\sigma) = (-1)^{N_{\text{inversions}}} \tag{271}$$

  - In the TCL, the function permutationCycles can compute the sign of a permutation.

- The permanent of a square matrix is defined to be

$$\operatorname{perm}(\mathbf{A}) \triangleq \sum_{\sigma \in \mathbb{S}^n} \prod_{i=1}^{n} a_{i,\sigma_i} \tag{272}$$

with $\operatorname{perm}(\emptyset) = 1$.

The PDA and JPDA Algorithms: Computing β

- The permanent of a square matrix is the determinant without the $-$ signs.
- The permanent of a square matrix is the sum of the products of all combinations selecting one element from each row and column.
- The permanent of a rectangular $m \times n$ matrix with $m \leq n$ is also defined as

$$\operatorname{perm}(\mathbf{A}) \triangleq \sum_{\sigma \in \mathbb{P}^{n,m}} \prod_{i=1}^{m} a_{i,\sigma_i} \tag{273}$$

  where $\mathbb{P}^{n,m}$ is the set of all length m permutations of n items (known as arrangements).
- The permanent of a rectangular $m \times n$ matrix with $m \leq n$ is the sum of the product of all arrangements selecting one item from each row and at most one item from each column.

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:

$$\mathbf{C} \triangleq \overbrace{\begin{bmatrix} \Delta\tilde{\Lambda}_{1,1} & \dots & \Delta\tilde{\Lambda}_{1,m} \\ \Delta\tilde{\Lambda}_{2,1} & \dots & \Delta\tilde{\Lambda}_{2,m} \\ \vdots & \ddots & \vdots \\ \Delta\tilde{\Lambda}_{N_T,1} & \dots & \Delta\tilde{\Lambda}_{N_T,m} \end{bmatrix}}^{\tilde{\mathbf{C}}: \text{ Assignment Likelihoods}} \overbrace{\begin{bmatrix} \Delta\tilde{\Lambda}_{1,0} & 0 & \dots & 0 \\ 0 & \Delta\tilde{\Lambda}_{2,0} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Delta\tilde{\Lambda}_{N_T,0} \end{bmatrix}}^{\text{Missed Detection Likelihoods}} \tag{274}$$

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

$$\beta_{j,k} = \Delta\tilde{\Lambda}_{j,k} \frac{\sum_{\sigma \in \mathbb{P}^{N_T-1,N_T-1+m}} \prod_{\substack{n=1 \\ n\neq j}}^{N_T} c_{n,\sigma_n}}{\sum_{\sigma \in \mathbb{P}^{N_T,N_T+m}} \prod_{n=1}^{N_T} c_{n,\sigma_n}} \tag{275}$$

The PDA and JPDA Algorithms: Computing β

- The expression simplifies to

$$\beta_{j,k} = \Delta\tilde{\Lambda}_{j,k}\frac{\text{perm}\left(\bar{\mathbf{C}}_{j,k}\right)}{\text{perm}\left(\mathbf{C}\right)} \tag{276}$$

  where $\bar{\mathbf{C}}_{j,k}$ is the matrix $\mathbf{C}$ after removing row j and column k.

- The matrix determinant can be evaluated in polynomial time using Gaussian elimination.
- The matrix permanent cannot be evaluated in polynomial time unless P=NP.
  - The relation between P and NP complexity classes is a major unsolved problem in theoretical computer science.
- 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.

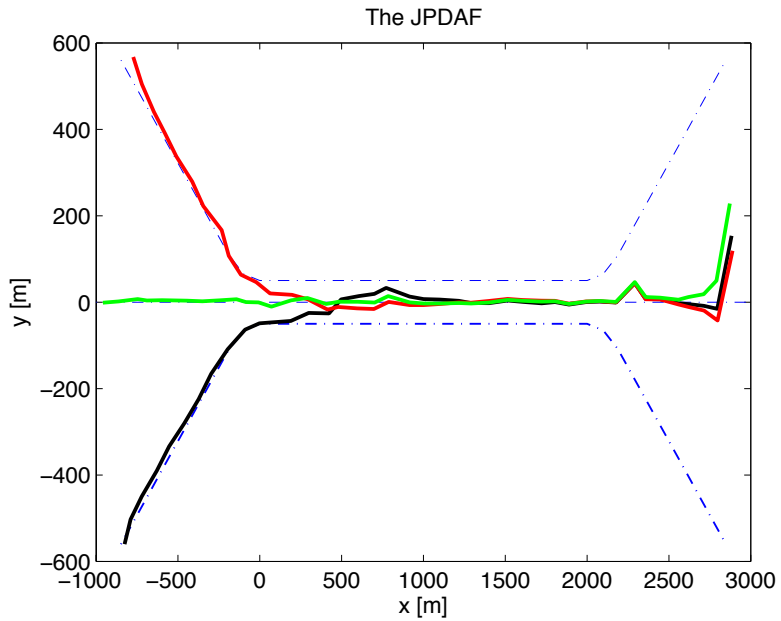
The JPDA Algorithm: Coalescence

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

$$p(\mathbf{x}) = \frac{1}{2}\delta\left(\mathbf{x} - \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) + \frac{1}{2}\delta\left(\mathbf{x} - \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) \tag{277}$$

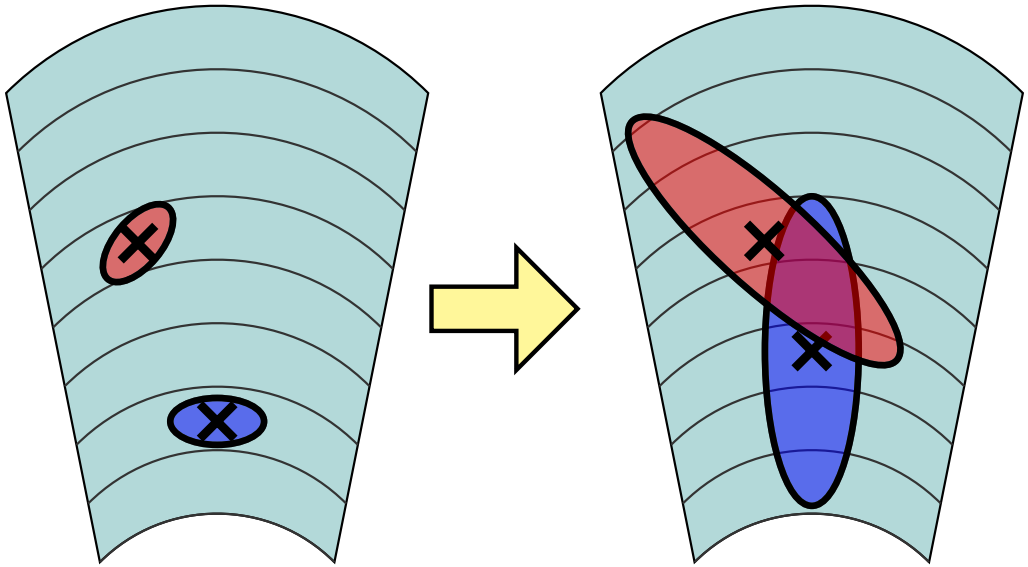
- One target is located at $+1$ and one target is located at -1, but we do not know which.
- $\mathrm{E}\{\mathbf{x}\} = \mathbf{0}$, where no target is located.
  - Identity uncertainty causes track coalescence!

The JPDA Algorithm: Coalescence



- Target identity is important, but we really want to know where things are.
- The above plot is typical of targets remaining close for many scans in the JPDA filter.

The JPDA Algorithm: Coalescence



(a) Prior to Prediction

(b) After Prediction

- Targets can be "too close" in the JPDAF even when well resolved by the sensor.
- Closeness is related to predicted target uncertainty.
- Well-separated targets becomes "too close" if prediction interval is too large.

U.S. NAVAL RESEARCH LABORATORY

The JPDA Algorithm: Coalescence

- Coalescence is not a "bias".
- Coalescence is the result of using the expected value given uncertain identity.
- The expected value is a minimum mean squared error (MMSE) estimate.
- A form of the minimum mean optimal subpattern assignment (MMOSPA) method:
  - Avoids the coalescence of the expected value.
  - Still has the same smoothing properties as the expected value versus ML.

The JPDA Algorithm: Coalescence

- The most common form of the optimal subpattern assignment metric used for tracking is

$$d^{OE}(\hat{\mathbf{x}}, \mathbf{x}) = \frac{1}{N_T} \min_{\mathbf{a}} \|\hat{\mathbf{x}}_{\mathbf{a}} - \mathbf{x}\|^2 \tag{278}$$

  - $\hat{\mathbf{x}}$ and $\mathbf{x}$ are vectors of the stacked state estimates and the true state vectors for N_T targets:

$$\hat{\mathbf{x}} = \left[\left(\hat{\mathbf{x}}^1\right)', \left(\hat{\mathbf{x}}^2\right)', \ldots, \left(\hat{\mathbf{x}}^{N_T}\right)' \right]' \tag{279}$$

$$\mathbf{x} = \left[\left(\mathbf{x}^1\right)', \left(\mathbf{x}^2\right)', \ldots, \left(\mathbf{x}^{N_T}\right)' \right]' \tag{280}$$

  - $\mathbf{a}$ is a permutation vector determining the order of the states in the stacked vector:

$$\mathbf{a} = [a_1, a_2, \ldots, a_{N_T}]' \tag{281}$$

$$\hat{\mathbf{x}}_{\mathbf{a}} = \left[\left(\hat{\mathbf{x}}_{a_1}\right)', \left(\hat{\mathbf{x}}_{a_2}\right)', \ldots, \left(\hat{\mathbf{x}}_{a_{N_T}}\right)' \right]' \tag{282}$$

U.S. NAVAL RESEARCH LABORATORY

The JPDA Algorithm: Coalescence

- MMOSPA estimation is essentially an expected value with target identity removed.
- Typically, the ordering for the MMOSPA estimate is computed only using position components of the state.
  - The strict OSPA definition mixes unlike units (e.g. position and velocity).
- The Set JPDAF is an approximate MMOSPA estimator.
- In the TCL, use calcSetJPDAUpdate to use the Set JPDAF. It is also an option in singleScanUpdate.
- An approximate MMOSPA estimator is given in MMOSPAApprox in the TCL.

U.S. NAVAL RESEARCH LABORATORY

The JPDA Algorithm: Coalescence

- In practice, the Set JPDAF is not used
  - It discards too much identity information.
  - It is slower than alternatives.
- Two common alternatives to resist coalescence are
  1. The GNN-JPDA
  2. The JPDA*

The JPDA Algorithm: Coalescence

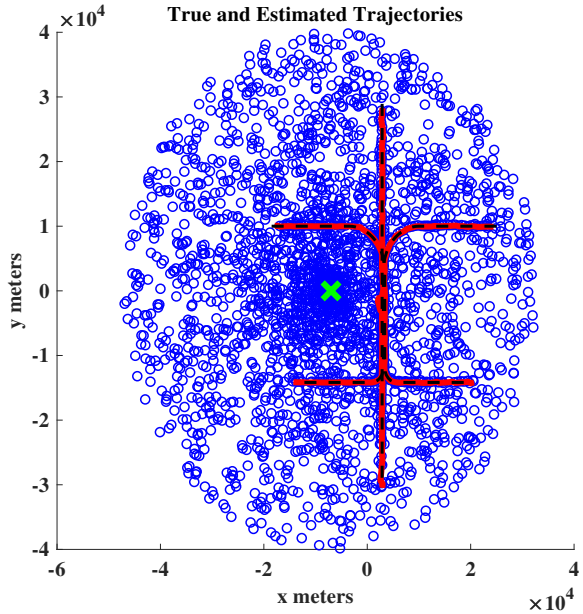
- The GNN-JPDA is simple:
  1. Determine the measurement to use with a GNN filter, giving $\hat{\mathbf{x}}_{k|k}$.
  2. Compute $\mathbf{P}_{k|k}$ as in the JPDA, using the GNN estimate as the mean $\hat{\mathbf{x}}_{k|k}$.
- The hard assignment avoids coalescence.
- Computing $\mathbf{P}_{k|k}$ as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

U.S. NAVAL RESEARCH LABORATORY

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.

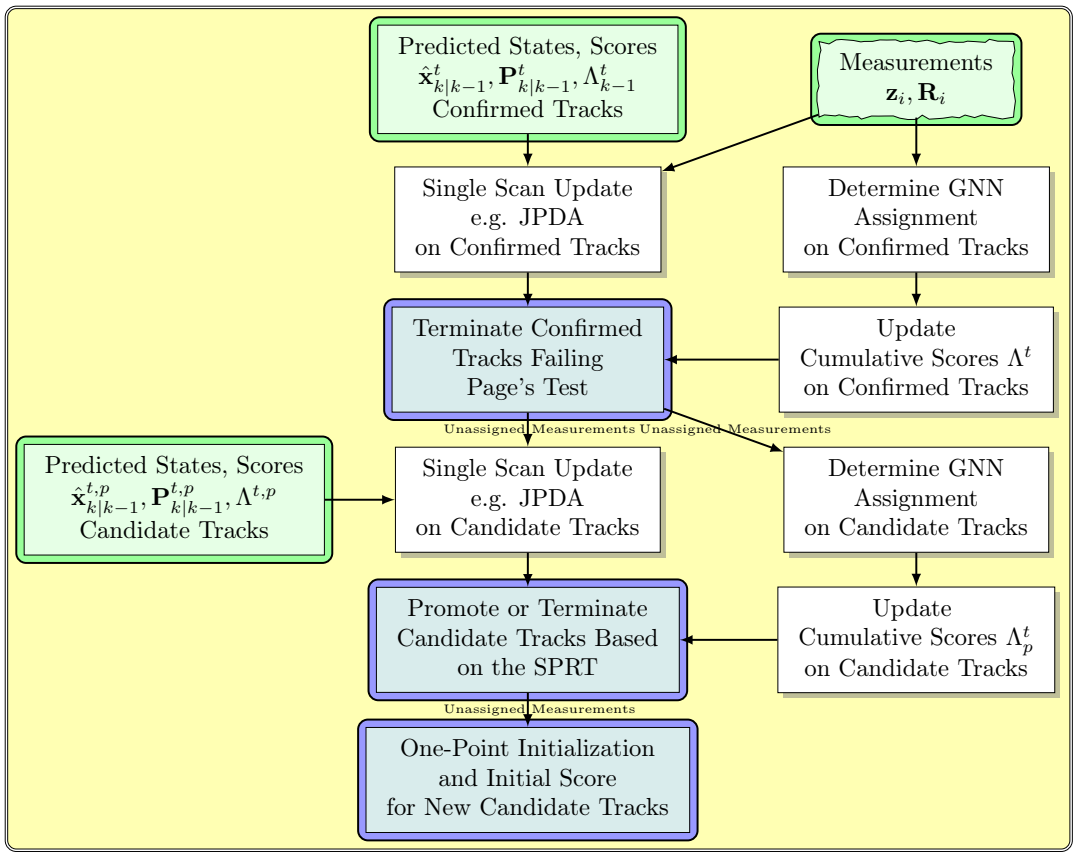
U.S. NAVAL RESEARCH LABORATORY

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.

U.S. NAVAL RESEARCH LABORATORY

A Cascaded Logic Tracker

- Measurements not assigned to confirmed tracks or candidate tracks become new candidate tracks via single point initialization.
- New candidate tracks assigned an initial track score Λ_p^t.
- As candidate tracks are updated with measurements, the likelihood increment $\Delta\Lambda_t$ of the GNN hypothesis for that track is added to Λ_p^t.
- A sequential probability ratio test (SPRT) determines whether to declare the track confirmed or false.

A Cascaded Logic Tracker: The SPRT

- Let $p_m(\mathbf{Z}_m|\theta)$ be the conditional PDF of a track given m measurements in $\mathbf{Z}_m$ under the hypothesis θ that the track exists.
- $p_m(\mathbf{Z}_m|\emptyset)$ be the conditional PDF of false alarms given m measurements in $\mathbf{Z}_m$ under the hypothesis that no track exists.
- A conditional probability ratio is

$$l_m(\mathbf{Z}_m) = \frac{p_m(\mathbf{Z}_m|\theta)}{p_m(\mathbf{Z}_m|\emptyset)} \tag{283}$$

- We choose two positive constants a and b such that at step m:
  - If $a > l_m(\mathbf{Z}_m) > b$ the test continue for another step.
  - If $a \leq l_m(\mathbf{Z}_m)$, it is declared there is a track.
  - If $b \geq l_m(\mathbf{Z}_m)$, it is declared there is no track.
- This test is the SPRT for likelihood ratios.

A Cascaded Logic Tracker: The SPRT

- The acceptance and rejection criteria can be rewritten:
  - Declare no track if $ap_m(\mathbf{Z}_m|\emptyset) \leq p_m(\mathbf{Z}_m|\theta)$
  - Declare a track if $bp_m(\mathbf{Z}_m|\emptyset) \geq p_m(\mathbf{Z}_m|\theta)$
- Define the region of values of $\mathbf{Z}_m$ where a track is declared as $\mathbb{L}_1$.
- Define the region of values of $\mathbf{Z}_m$ where a no track is declared as $\mathbb{L}_2$.
- Integrating the acceptance criterion over $\mathbb{L}_1$. and the rejection criterion over $\mathbb{L}_2$, one gets

$$b(1-\alpha) \geq \tilde{\beta} \tag{284}$$

$$1-\tilde{\beta} \geq a\alpha \tag{285}$$

where

- α is the probability of falsely declaring a target.
- $\tilde{\beta}$ is the probability of falsely declaring no target.

A Cascaded Logic Tracker: The SPRT

- From the previous two equations, taking the inequalities as equalities (which assumes the boundaries are not exceeded by much), one gets

$$\ln(a) = \ln\left(\frac{1-\tilde{\beta}}{\alpha}\right) \tag{286}$$

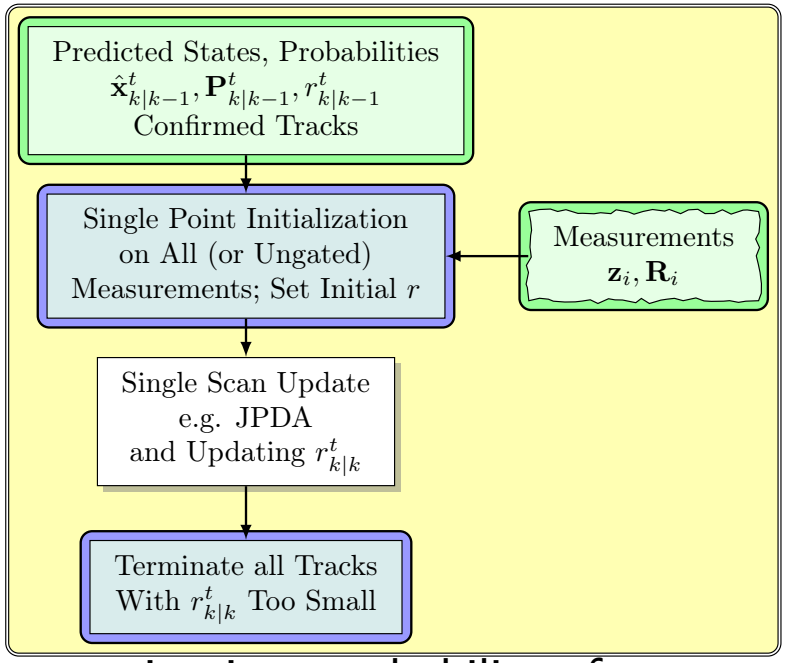
$$\ln(b) = \ln\left(\frac{\beta}{1-\alpha}\right) \tag{287}$$

- When using the cumulative dimensionless score function Λ_p^t, a candidate tracks is rejected when $\Lambda_p^t \leq \ln(b)$ and accepted when $\Lambda_p^t \geq \ln(a)$.
- The SPRT was extensively studied by Abraham Wald in the 1940s.
- Once accepted, a different test must be used to terminate a confirmed track.

A Cascaded Logic Tracker: The SPRT

- If the SPRT were continued after confirmation, the cumulative score would generally keep increasing.
- When a track ends, the score would need to come back down to terminate if one tried to continue the SPRT.
- Page's test is as follows:
  - When confirmed, set the cumulative track score to a bound value b_p.
  - If the cumulative track score ever exceeds b_p, reset it to b_p.
  - If the cumulative track score ever dips below a lower bound a_p, then the track is terminated.
- The determination of a_p can be difficult.
- A simple approximation is to reuse the bounds $\ln(a)$ and $\ln(b)$ from the SPRT.

An Integrated Tracker



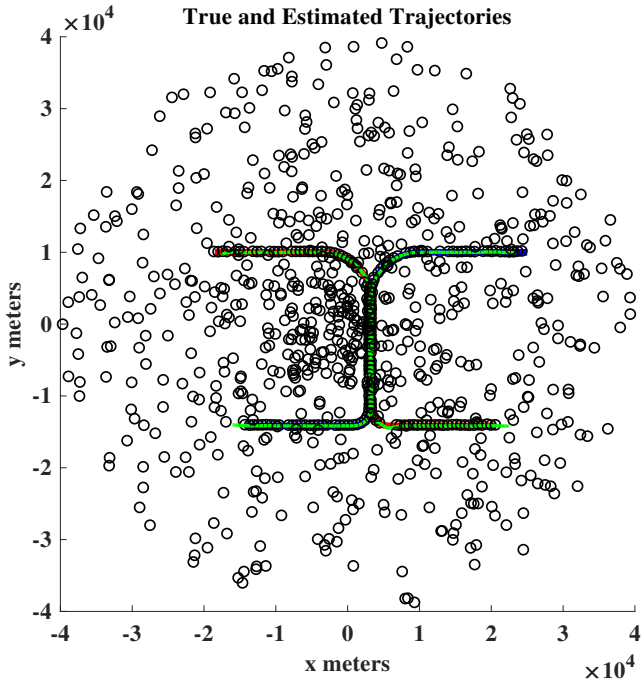
- Integrated trackers maintain a probability of target existence with each possible target.
- Usually, a track is not considered firm until its existence probability exceeds a threshold.
- A track is not terminated until its existence probability goes below a lower threshold.
- Measurement update implemented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

U.S. NAVAL RESEARCH LABORATORY

An Integrated Tracker

- Unlike the standard JPDA, has r terms: target existence probabilities.
- The algorithm names are the same as JPDA variants, but with an I for integrated after the J.
  - For example: JIPDA, GNN-JIPDA, JIPDA*, etc.
- Direct derivations of the JIPDA tend to be hard to follow.
- The Track-Oriented Marginal Multiple Target Multi-Bernoulli-Poisson (TOMB/P) is essentially the same filter with a different model for starting tracks.
- The TOMB/P uses a Poisson prior for the number of undetected targets born each scan. The JIPDA does not. All measurements are just assigned an initial probability of being a track.
  - The TOMB/P's Poisson birth model is mathematically appealing, but has no physical basis.

An Integrated Tracker



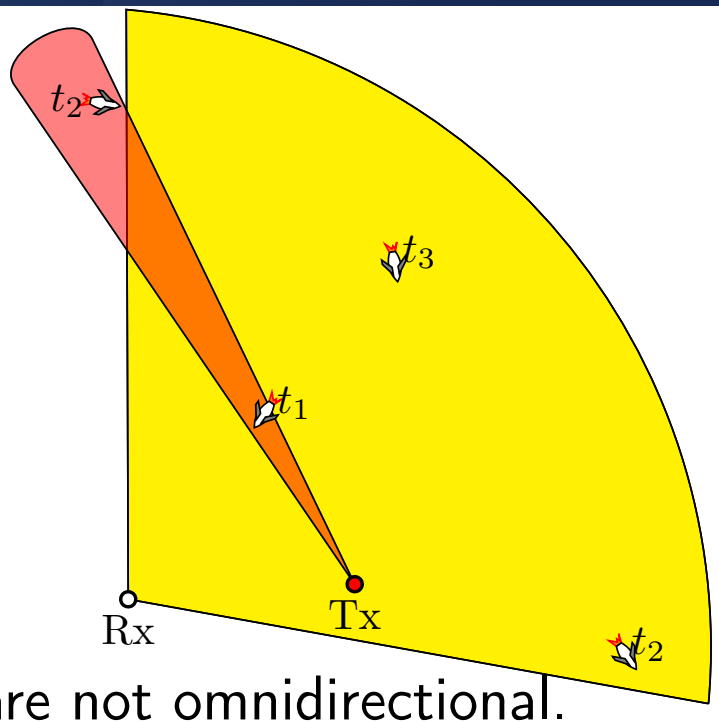
- The most straightforward derivation of the JIPDA class of filters uses finite set statistics.
- A proper coverage of finite set statistics is beyond the scope of this presentation.

- 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

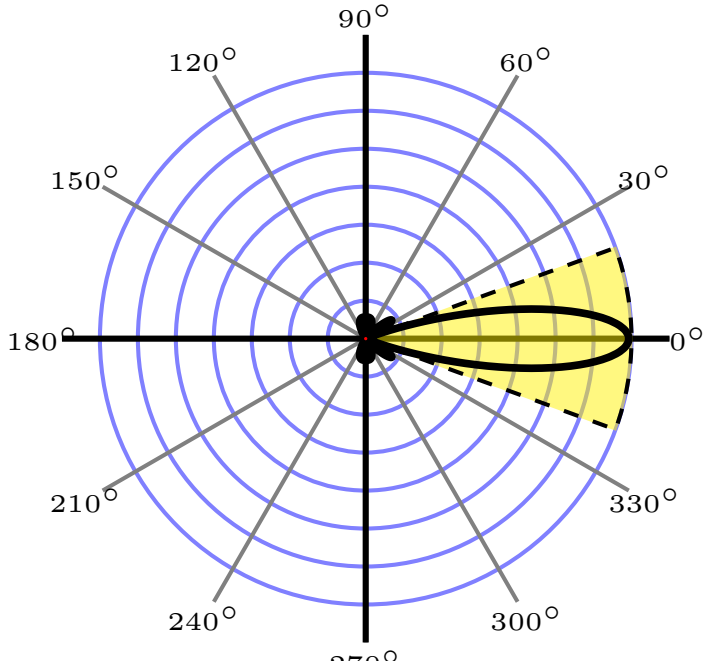
DEALING WITH BEAMS

Dealing with Beams



- Most sensors are not omnidirectional.
- During a dwell, the score functions of targets not illuminated or not in the viewing region should not change.
- Only target T_1 could be expected to be observed by the receiver above.
- A maximum time without detections should be a design criterion to help terminate tracks that will not be revisited.

Dealing with Beams



- If a target is determined to be in the transmit and receive beams, then one will often adjust P_D based on the predicted positions in the beams and the beamshapes.
- This can be more difficult than one might expect. The Swerling models in "Mathematical Functions/Statistics/Detection Statistics/" in the TCL might help.

U.S. NAVAL RESEARCH LABORATORY

SUMMARY

U.S. NAVAL RESEARCH LABORATORY

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.

U.S. NAVAL RESEARCH LABORATORY

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

U.S. NAVAL RESEARCH LABORATORY

QUESTIONS?