Multiple Target Tracking

Tracking and Data Fusion with a Requirements Perspective

James K Beard, Ph.D. EE
jkbeard@ieee.org
http://jameskbeard.com

Philadelphia IEEE Section Night
November 16, 2021

Topics

- Mission and Requirements
- Single Object Tracking Fundamentals
- Fundamentals of Tracking in Dense Environments
- Care and Feeding of Kalman Filters
- Care and Feeding of Data Fusion Methods
- Summary

Supplements to Today's Presentation

- Recent IEEE Presentations
  - IEEE AESS Virtual Distinguished Lecturer Webinar Series
    - http://ieee-aess.org/webinar-series-october
    - October 12, 2021, Tracking and Sensor Data Fusion, by Wolfgang Koch
    - October 19, 2021, Data Association and Target Tracking by Peter Willett
    - October 28, 2021, Systematic Filter Design for Tracking Maneuvering Targets, by Dale Blair
  - IEEE AES Tutorials
    - https://ieee-aess.org/aess-systems-magazine-tutorials-list
    - Blackman, S. on the MHT (2004)
    - Bar-Shalom et. al. On Probabalistic Data Association (2005)
- There is a Separate Appendix to This Presentation with Math and Equations

Mission and Requirements

- Scenarios are anywhere we have a requirement for tracking objects
- Mission and Requirements of Situation Awareness Flows Down to Tracker Configurations
- A Theater of Operations may be comprised of Multiple Scenarios
- Computer Requirements Flow Down from Performance Requirements
  - Processor performance has improved more than three orders of magnitude since 1985
  - Focus on performance first, then computer requirements

Common Scenarios

- Surveillance Area or Volume, Battlefield, Potential and Active
  - Air
  - Space
  - Surface
- Maritime
  - Littoral area surveillance
  - Civilian and commercial traffic monitoring
  - Search and rescue
  - DHS requirements
  - Open Ocean
  - Subsurface

Stated and Implied Requirements

- Stakeholders with stated or implied requirements include
  - Government and commercial funding organizations
  - Program management and supervising organizations
  - Laws and regulations, including DFARS
  - Users of the system and data, including ILS and support organizations
  - Future users, disposal and environmental concerns
- Key requirements are on data and interfaces
  - Information inputs such as ADS-B and SSR, AIS, C2, GMTI, etc.
  - Data requirements
    - Object position and velocity, with accuracies
    - Update rate, descriptive data content
  - Classification; what is the object under track
  - Object Situation: underway, adrift, distress, attack, cruise, boost,…

Mahalanobis Distance and Localization Ellipsoid

- Mahalanobis Distance s, for Filter Error

$$\vec{e} = \vec{y} - \vec{h}(\vec{x}_{EXT}), \text{ measurements minus expected measurements}$$

$$E = \text{Cov}\{\vec{e}\} = H \cdot P_{EXT} \cdot H^T + R, \quad H = \frac{\partial \vec{h}(\vec{x}_{EXT})}{\partial \vec{x}_{EXT}}$$

$$s = \sqrt{\vec{e}^T \cdot E^{-1} \cdot \vec{e}}$$

- Quantity s^2 is chi-square distributed with k degrees of freedom, where k is the number of measurements
- Localization Ellipsoid
  - Interior of surface defined as the locus of $\vec{v}$ in the quadratic form

$$\vec{v}^T \cdot E^{-1} \cdot \vec{v} = 1$$

Propagation of States and Covariance

- Motion

$$\frac{d\vec{x}}{dt} = \vec{f}(\vec{x})\cdot\vec{x}+G\cdot\vec{w}(t),\quad \vec{x}_0=\vec{x}_{OLD},\quad \mathrm{Cov}\{\vec{w}\}=Q$$

$$\Phi(t,t_0) = \frac{\partial\vec{x}}{\partial\vec{x}_0},\quad \frac{d\Phi}{dt}=\frac{\partial\vec{f}(\vec{x})}{\partial\vec{x}}=F(t),\quad \Phi(t_0)=I$$

$$\vec{x}(t) \approx \Phi(t,t_0)\cdot\vec{x}_0+G\cdot\vec{w}$$

$$P = \Phi\cdot P_0\cdot\Phi^t+G\cdot Q\cdot G^T$$

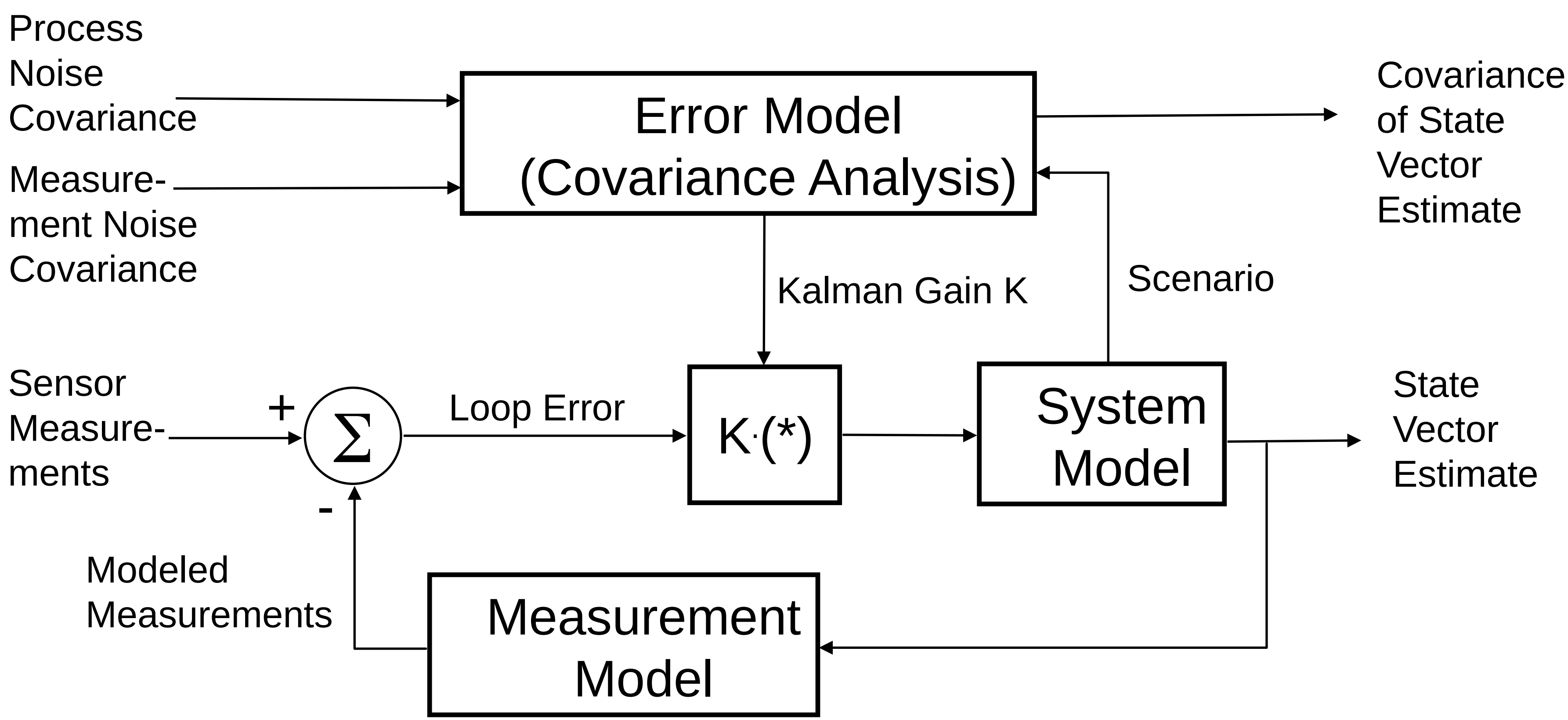
- Propagation of covariance through a transformation

$$\vec{y} = \vec{h}(\vec{x})+\vec{v},\quad \mathrm{Cov}\{\vec{v}\}=R$$

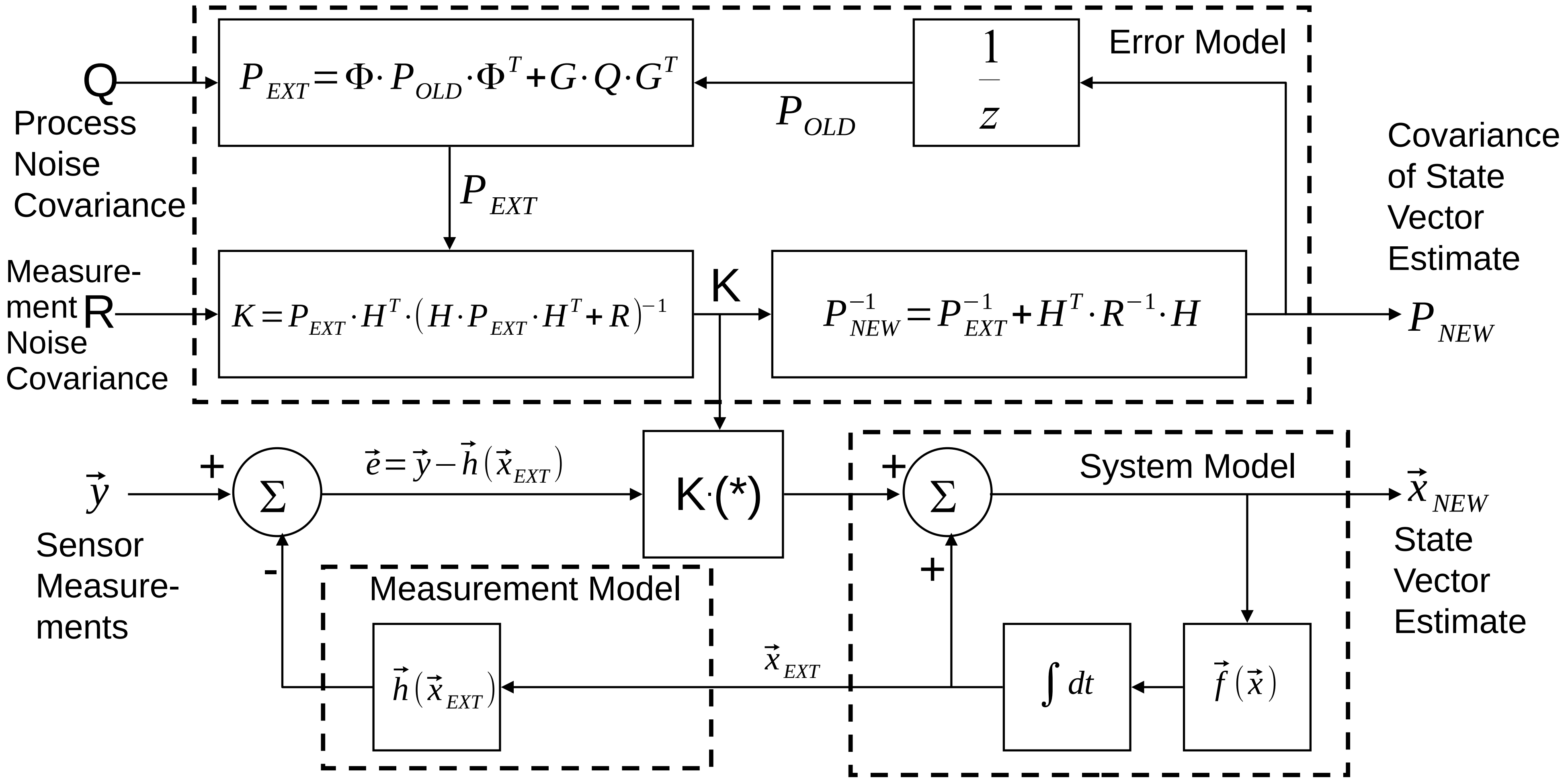
$$H = \frac{\partial\vec{h}(\vec{x})}{\partial\vec{x}}$$

$$\mathrm{Cov}\{\vec{y}\} = H\cdot P\cdot H^T+R$$

Classical Kalman Filter Concept



Classical Kalman Filter Operations



Realities of Kalman Filters

- The Extended Kalman Filter Works with Non-Gaussian Noises
  - Computed measurements such as monopulse
  - Nonlinear transformations of noisy quantities
- Measurements are Often Correlated
  - Sensor fusion is often with data from Kalman filters, which produces highly correlated errors
  - Any filtering of measurements produces correlation of errors
- Redemption
  - Classical Kalman EKF error analysis is variance analysis
  - Kalman filter updates are minimum variance estimates
  - Resulting biases are nearly always much smaller than random errors
  - Filter configuration can be modified to account for correlated measurements
  - Batch estimators are tolerant of correlated noise

Estimated Covariances in Kalman Filters

- In most practical Kalman filters
  - Estimated variances are often about five times that of the actual variances of the estimated states
  - Forcing variance estimates smaller in the Kalman filter results in slowed following of changes in behavior of the tracked object
  - Actual variances of filter outputs is best obtained by separate covariance estimators
- More Accurate Covariances
  - Supports better association of detection data to object tracks
  - Improves reliability of correct association in dense environments
- Batch estimators
  - Asymptotically statistically efficient
  - Provide accurate estimates of covariances of the states
  - Provide an opportunity to improve association history
  - Can be used to re-initialize Kalman filter with more accurate states, covariances

Other Tracker Types

- Chapman-Kolmogorov Expected States
  - See next slide for the Chapman-Kolmogorov equation
  - Back to basics, general probability distribution functions, recursively estimates Bayesian probability density of estimated states
- Particle, Unscented Filters use Mapping
  - Multiple Markov processes using specific “noise” values
  - Use importance sampling or unscented transformations
  - Estimate states, covariances from samples of Markov process results
- Must solve the same measurements-to-track association as the Kalman filter trackers to define measurement data

Chapman-Kolmogorov Equation

- Probability Density of the State Vector

$$p(x_k|\boldsymbol{Z}_1^k)=\frac{p(z_k|x_k)\int p(x_k|x_{k-1})\cdot p(x_{k-1}|\boldsymbol{Z}_1^{k-1})\cdot dx_{k-1}}{p(z_k|\boldsymbol{Z}_1^{k-1})}$$

- Each Term:

$p(x_k\|\boldsymbol{Z}_1^k)$	The desired pdf of current states x_k given database $\boldsymbol{Z}_1^k$
$p(x_k\|x_{k-1})$	Provided by the state transision (object motion) model
$p(x_{k-1}\|\boldsymbol{Z}_1^{k-1})$	This is the result of the previous update
$p(z_k\|x_k)$	Density function of the measurements
$p(z_k\|\boldsymbol{Z}_1^{k-1})$	Normalizes RHS pdf, independent of state vector

From AESS Distinguished Lecturer Series, October 19, 2021, by Peter Willett (U. Conn.), slide 6

Care and Feeding of Kalman Filters

- Tracker Requirements can Conflict
  - Association reliability, responsiveness, accuracy
  - Begin design with a three-tier tracker architecture to address each requirement separately
  - Combine tracker later if performance in a tier meets two requirements
- The Covariance Estimation Process
  - Lower estimated covariance gives better accuracy, but is also a trade with responsiveness
  - First priority is to avoid covariance collapse
  - Avoid process noise where it isn't needed
  - Adaptive, not-full-rank process noise can support tracking high performance objects with minimum accuracy impact

Make the Kalman Update Statistically Efficient

- The Kalman update equation

$$\vec{x}_{NEW} = \vec{x}_{EXT} + K \cdot \left(\vec{y} - \vec{h}\left(\vec{x}_{EXT} \right) \right)$$

- Statistically efficient when the states are linear functions of the measurements

$$\vec{h}\left(\vec{x}_{EXT} \right) = H \cdot \vec{x}_{EXT}$$

- Simple, obvious general policy:

$$H = I$$

- At the Kalman update, make the states the same as the measurements
- Aircraft trackers using $\ln(R/R_0)$ and direction cosines for aircraft position
  - Very well conditioned numerically, all quantities unitless and small in magnitude
  - Simple in formulation; isometric relation between quaternions and complex variables supports simple algebraic equations
  - Doppler measurements map well to (range rate)/(range)
- Compute object position from states in Cartesian coordinates for C2

Tracker Architectures

- Develop Design with Three Tiers of Trackers
  - Simple one-measurement adaptive estimators of each individual measurement type
    - Separate simple EKFs for range, direction cosines, Doppler (or other local sensor outputs)
    - States are estimates of measurements and derivative(s)
    - Running estimate of process noise amplitude to adapt to object maneuvers
    - Serves to support association
  - Accuracy-driven main tracker
    - Kalman filter design driven by accuracy requirements
    - Uses IMM, supports MHT, supports real-time tracking, alerting
  - Batch estimator
    - Achieves essentially the Cramer-Rao Bound
    - Supports decisions
  - Advantages
    - Separates functions of association, tracking, decisions
    - Does better job of all three
- Combine Tracker Tiers Later if Performance Robustly Meets Requirements

Covariance Estimation

- Covariance Collapse: Matrix P Becomes Nearly Singular
  - Filter loses responsiveness to tracked object motion
  - Always happens
    - When you compute $P_{NEW} = (I - K \cdot H) \cdot P_{EXT}$ (causes true filter instability!)
    - When you use zero process noise in acceleration states in NCA filters, velocity states in NCV filters
  - Exhibited with “easy” tracks (high SNR, straight-line constant velocity objects)
- Steps to avoid covariance collapse
  - Use nonzero process noise in acceleration states of NCA filters, velocity states in NCV filters
  - Update covariance matrix using one of
    - Classical: Joseph Stabilized Form, $P_{NEW} = (I - K \cdot H) \cdot P_{EXT} \cdot (I - K \cdot H)^T + H^T \cdot R \cdot H$
    - Better: Inverse form, $P_{NEW}^{-1} = P_{EXT}^{-1} + H^T \cdot R^{-1} \cdot H$
    - Best: A square root filter

Square Root Filters

- All Carry Covariance Matrix as Matrix Square Root
  - A Cholesky factorization of the covariance matrix, its inverse, or a matrix equal to a similarity transform of a Cholesky factorization
  - A Square Root filter cuts word length requirements for the covariance matrix P in half
- Best-Known Types of Square Root Filters
  - Square Root Information Filter (SRIF)
  - UDUT Filter
  - Potter square root filter
- All Square Root Filters
  - Are algebraically identical to EKF
  - Represent covariance as necessarily positive definite
  - Allow bypassing of EKF state extrapolation with better object motion models

Minimizing Process Noise

- Use Adaptive Process Noise
  - Estimate a scale factor on the process noise covariance
  - Use squared Mahalanobis distance of error signal as a measurement of process noise magnitude of variance
- Minimize use of process noise
  - Models of motion of exoatmospheric objects usually don't need process noise
  - Use accurate motion models to characterize complex object motion (maneuver, re-entry, boost, etc.)
  - Ground objects have little vertical process noise
  - Use "smart" process noise when applicable

Tracking High Performance Objects

- Aircraft motion uncertainties
  - Unalerted: uncertainties are mainly along velocity vector
  - Alerted and engaged: larger uncertainties mostly normal to velocity vector
  - Use IMM to model and adjust to abrupt changes in object motion
  - Use adaptive process noise to adjust to changes in object dynamics
- Boosting, Re-Entering Object Uncertainties are Mostly Along Acceleration Vector
- Use lower rank process noise, non-square G
  - Smart process noise
  - Minimizes use of process noise, does use what is required

Example of Smart Process Noise

- Use matrix G to define aircraft coordinates front, left, up
  - Object velocity, normalized, is toward front
  - Object acceleration, normalized, is toward up
- Use process noise covariance Q to control relative magnitudes in aircraft coordinates

$$\vec{x} = \begin{bmatrix} \vec{p} \\ \vec{v} \\ \vec{a} \end{bmatrix}, \quad G = \begin{bmatrix} \vec{0} & \vec{0} & \vec{0} \\ \vec{0} & \vec{0} & \vec{0} \\ \overrightarrow{uv}_{NOSE} & \overrightarrow{uv}_{LEFT} & \overrightarrow{uv}_{UP} \end{bmatrix}, \quad Q = \begin{bmatrix} \sigma^2_{NOSE} & 0 & 0 \\ 0 & \sigma^2_{LEFT} & 0 \\ 0 & 0 & \sigma^2_{UP} \end{bmatrix}$$

Dense Environment Considerations

- When aircraft tracks interact, couple the trackers
  - Example: Aircraft moving together in a formation
  - Carry two or more objects in the same Kalman filter
  - Augment the state vector to include states for all objects tracked here
  - Augmented covariance matrix includes cross-correlations between tracks
- When return-to-track associations become ambiguous
  - Use Track Before Detect methods
  - Multiple Hypothesis Tracking (MHT) is the best right now
  - Target-oriented, Hypothesis-oriented (TO-MHT, HO-MHT)
  - Probabilistic MHT

Track Before Detect

- Methods
  - Multiple Estimator – Multiple Hypothesis Tracking (MHT)
  - Interactive Multiple Models (IMM) compares results of multiple object behavior models within a single Kalman update
  - Range-Doppler Map Warp/Averaging
  - Others
- Practical Considerations
  - MHT is the “gold standard” but is an art and a science for each application
  - IMM can be used alone or combined with MHT
  - Some methods, like range-Doppler map averaging, don’t perform detection in themselves

Warped Range-Doppler Map

- Performs Multiple Simultaneous Estimation of Returned Signal Amplitude for Each Cell in the Range-Doppler Map
- Each Cell is a Simple NCV Kalman Filter State or Measurement
  - Estimates the target signal strength in each range-Doppler map cell
  - Warping is by interpolating each Doppler row by a delta range found as the range rate for that Doppler, multiplied by the time between range-Doppler maps $\Delta R = [dR/dt] \cdot \Delta t$
  - Update is by averaging warped previous map with current map
  - Noise and clutter average down, objects average up
- Detection is Done on the Averaged Range-Doppler Map
  - Optimal for constant range rate objects
  - Accelerating objects will have a signature that is a “trail” that can be searched for using image processing methods

Care and Feeding of Data Fusion Methods

- Why Data Fusion?
- Realities of Data Fusion
- Dealing With Data Fusion Issues

Principle and Motivation for Sensor Fusion

- Given: two or more state vectors with their covariance matrices
- Combining them (see Appendix)

$$P_{NEW}^{-1} = P_1^{-1} + P_2^{-1} + \ldots = \sum_i P_i^{-1}$$

$$\vec{x}_{NEW} = P_{NEW} \cdot \left(P_1^{-1} \cdot \vec{x}_1 + P_2^{-1} \cdot \vec{x}_2 \ldots \right) = P_{NEW} \cdot \sum_i P_i^{-1} \cdot \vec{x}_i$$

- Merged localization ellipsoid is wholly contained within every input localization ellipsoid
- Problem: Data From Different Sensors
  - Will have biases that almost always drift with time
  - Object Positions are usually provided in different coordinate systems
  - C2 and other sensor data may have different state vectors

Dirty Secrets in Data Fusion

- There is no substitute for a good sensor
- Downstream processing cannot absolve the sins of upstream processing.
- The fused answer may be worse than the estimate from the best sensor.
- There are no magic algorithms.
- There will never be enough training data.
- It is difficult to quantify the value of data fusion.
- Fusion is not a static process.
- From Ligins, Hall & Llinas, pp 11-12

Data Fusion Issues

- Coordinate and Time Offsets
  - Object positions, velocities from C2 will differ from local coordinates of the same objects
  - Biases will drift, may be different for objects in different positions
  - Systems sending data via C2 will have clock offsets, data may not be current
- Methods to Deal With Offsets
  - Use training data; including real data reduces surprises later
  - When biases are small and variable, model them as noises
  - When biases are large or drift with time, estimate them as states
  - When data is old, extrapolate it to current time and fuse
- Correlated Measurement Errors
  - Model correlated data as a Markov process, adjust tracker configuration accordingly
  - Change the system model when needed

Summary

- Begin With Requirements
  - Stakeholder requirements
  - User requirements
  - Interoperability, ILS, environmental, and disposal requirements
- Observe limitations
  - System implementation resources
    - Facilities and personnel
    - Technology
    - Real data for development and testing
  - Schedule and Budget
- Minimize the Use of Process Noise

Keep a Systems Engineering Perspective

- Employ Risk Management
  - Defense Acquisition University Summary: https://www.dau.edu/tools/Lists/DAUTools/Attachments/140/RIO-Guide-January2017.pdf
  - Problems revealed in I&T can ripple back to starting requirements
  - Have a plan for everything predictable
- Use a Statistically Efficient Kalman Update
  - Tolerating a nonlinear measurement equation increases the covariance of the updated state vector
  - The “hit” on performance accrues every update

Use Real Data in Development

- Simulation Data Incorporates
  - What you predict will happen, exactly
  - Well-behaved noise
  - Great for early development
- Real Data Incorporates
  - What happens out there that you don't know about
  - All signals including overlapping returns and interference
  - Noise as seen by the sensor outputs, C2, etc.
  - Surprises in real data are a big plus for the developer

> The world will do what it wants to do, not what you think it will do. (John Nash)

Coordinate Systems are Critical

- Different Coordinate Systems
  - Measurements and Kalman update – sensor coordinates
  - Object motion – object coordinates, NED, ECIC, platform coordinates, as appropriate for accurate object motion modeling
  - C2 data – Battlefield coordinates specified by BMC4I
- Hazards in Coordinate Systems
  - Use of inertial over long periods must account for Earth's rotation
  - Use of Earth-rotating NED or radar coordinates must include Coriolis in object motion models (or, better, model the object motion in Earth-rotating sensor or object coordinates)
  - For exoatmospheric objects, use ECIC coordinates
  - Coordinate origins, orientations, physical units, effective time epoch, all must agree in data hand-offs

Realities of Statistical Estimation

- There is a Trade Between Complexity and Accuracy
  - Modeling tracked object behavior as process noise allows very simple object motion models but allows unnecessary process noise to map into tracked position errors
  - Accurate models of object behavior minimize required process noise for more robust operation
  - Less process noise produces more accurate tracks, better data-to-track association
- Adding States Reduces Accuracy
  - Information provided by sensors is bandwidth or data rate times dynamic range
  - Information in all states cannot exceed the input information
  - Adding states dilutes information available to estimate each state
- Add states Cautiously
  - When necessary to estimate drifting biases
  - Temporarily, when useful in development
  - On operator command or automatically when appropriate

Conclusion

- A More Accurate Tracker Supports Better Association
- Use Statistically Efficient Kalman Update
- Keep Process Noise to an Absolute Minimum
- Consider All Appropriate Tracker Types
- When Object Density Impacts Tracker Performance, Look at an MHT

Appendix and Bibliography

Multiple Target Tracking: APPENDIX

Tracking and Data Fusion with a Requirements Perspective

Table of Contents