Robust Cooperative Localization

Steven Dourmashkin
Graduate Research Assistant

4/5/17

Cooperative Human-Robot Intelligence Laboratory
Ann and H.J. Smead Aerospace Engineering Sciences
University of Colorado at Boulder

COHRINT

Research Overview

- **Vision:** teams of humans and robots performing opportunistic, scalable cooperative localization.
- **Important** in uncertain and/or hostile environments, with…
  - Obstructed / limited measurements
  - Minimal communication
- **Approach:**
  - Fuse state estimates among human, robot, and/or vehicle agents tracking and communicating with one another in an ad-hoc, scalable fashion.
  - Use radio ranging as primary opportunistic measurement to maintain precise state estimation when particular agents loose reliable GPS.

Squad X Concept, source: DARPA

Technical Problem Statement

- Develop and analyze partial state DDF techniques to allow cooperative localization and tracking in a scalable, ad-hoc network of soldiers and vehicles in the presence of obstructed GPS measurements.
  - **Phase I** (Fall 2016 – Spring 2017)**:** Build a simulation to verify successful cooperative localization given Army's available sensors and corresponding characteristics.
  - **Phase II** (Summer 2017 – Spring 2018): Implement framework on hardware using realistic sensors and environments.

Problem Setup

- "Multi-robot SLAM with moving landmarks that communicate minimally"

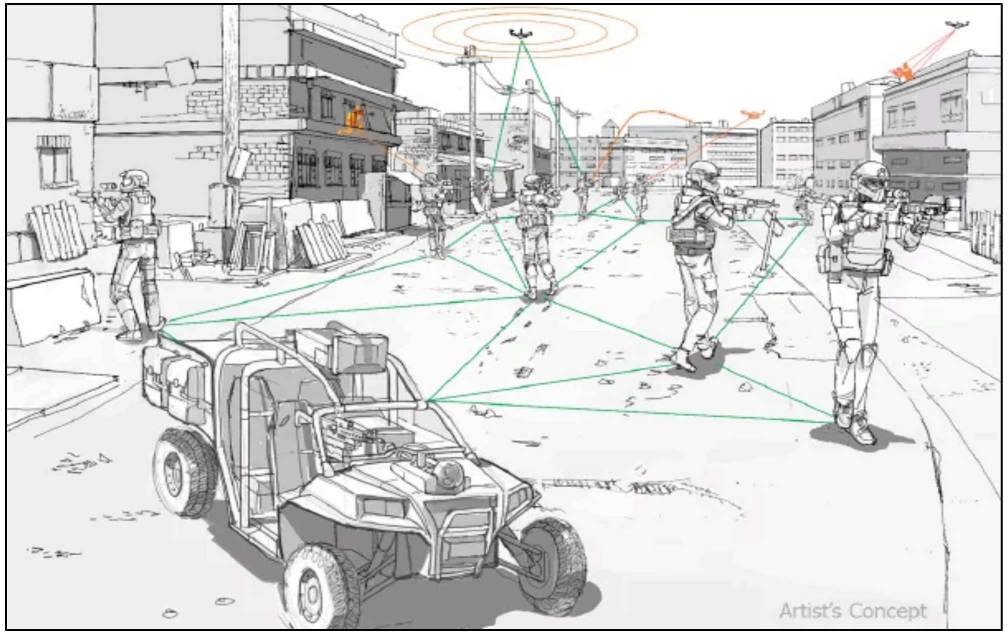
- **Agents:** Human, robot, and/or vehicle agents

$$A = \{a_i\}_{i=1}^{N_A}$$

- **Tracked Processes:** human, robot, and/or vehicles tracked by Agent a_i

$$P = \{p_i^j\}_{j=1}^{N_p}$$

- **State Space**: combination of local navigation filter + tracking filter states for Agent a_i

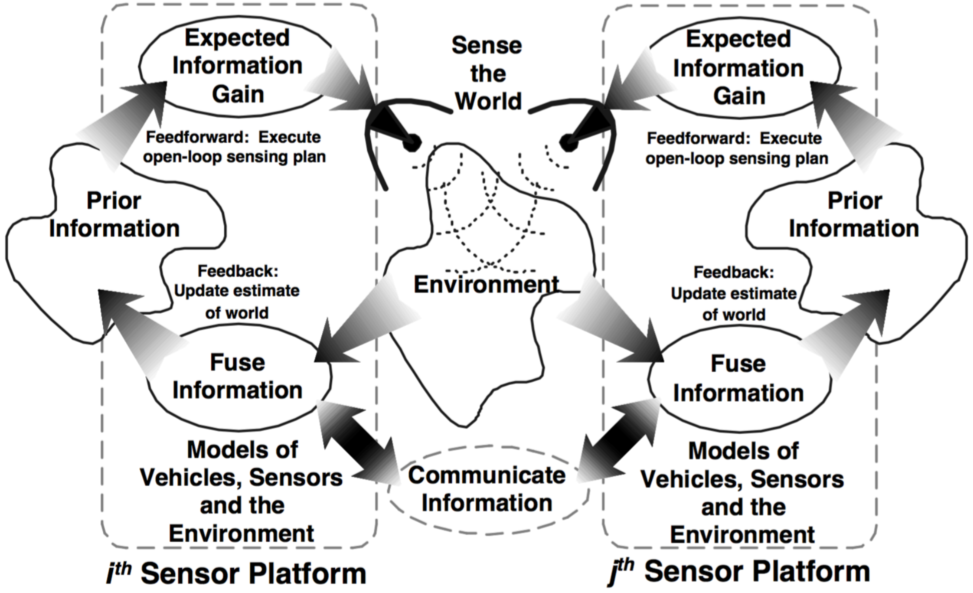


Squad X Concept, source: DARPA

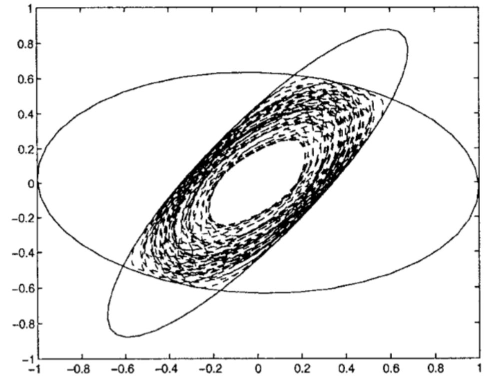
$$\mathbf{x}_i = \left[\delta\mathbf{x}_i \;\vdots\; \mathbf{s}_i^j \;\; \ldots\right]^T = \left[\underbrace{\delta\mathbf{p}_i \;\; \delta\mathbf{v}_i \;\; \rho_i \;\; \delta\mathbf{x}_i^a \;\; \delta\mathbf{x}_i^g}_{\text{local nav. states}} \;\vdots\; \underbrace{\mathbf{s}_i^j \;\; \dot{\mathbf{s}}_i^j \;\; \ldots}_{\text{tracking states of each } \{p_i^j\}_{j=1}^{N_p}}\right]^T$$

Previous Work

- **Vanilla Distributed Data Fusion (DDF) (1,2,3)**
  - Assumes uncorrelated state uncertainties, tree network topology, identical states across agents
- **SLAM (4)**
  - Involves nonlinear, iterative least-squares optimization → not scalable because of required uncertainty bookkeeping
- **GPS-denied solutions (5,6)**
  - Computationally expensive, requires high quality sensors, and does not take advantage of sensor data available to other agents (i.e., information sharing) (e.g., LASOIS)
- **Covariance Intersection / Factorized DDF (7,8)**
  - Generally conservative and applied to agents with consistent tracking models

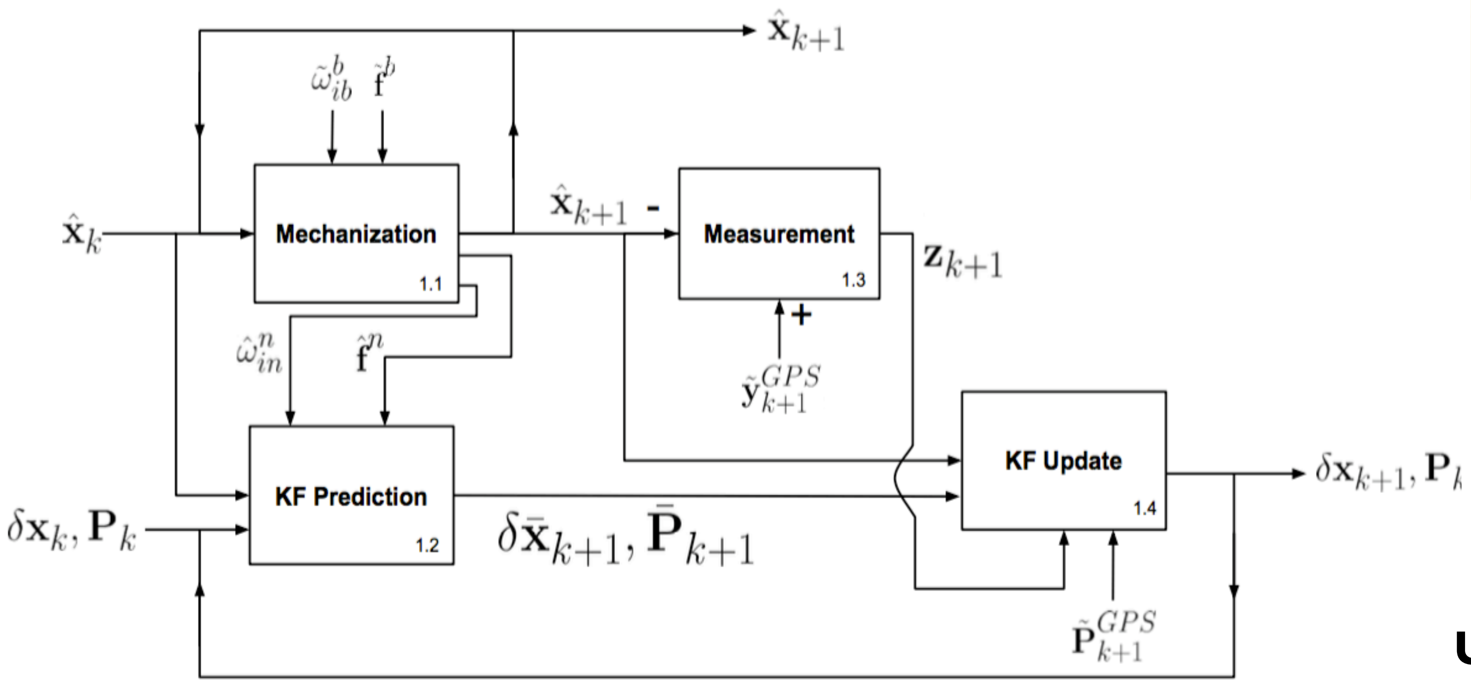


B. Grocholsky et. al.[3]



S. Julier et. Al.[7]

Previous Talk: Nav. Filter



Prediction:

$$\begin{bmatrix} \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{v}} \\ \dot{\rho} \\ \delta\dot{\mathbf{x}}_a \\ \delta\dot{\mathbf{x}}_g \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{pp} & \mathbf{F}_{pv} & \mathbf{F}_{p\rho} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{vp} & \mathbf{F}_{vv} & \mathbf{F}_{v\rho} & -\hat{\mathbf{R}}^n_b\mathbf{F}_{va} & \mathbf{0} \\ \mathbf{F}_{\rho p} & \mathbf{F}_{\rho v} & \mathbf{F}_{\rho\rho} & \mathbf{0} & \hat{\mathbf{R}}^n_b\mathbf{F}_{\rho g} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{gg} \end{bmatrix} \begin{bmatrix} \delta\mathbf{p} \\ \delta\mathbf{v} \\ \rho \\ \delta\mathbf{x}_a \\ \delta\mathbf{x}_g \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\hat{\mathbf{R}}^n_b & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}^n_b & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \nu_a \\ \nu_g \\ \omega_a \\ \omega_g \end{bmatrix}$$

Update:

$$\begin{aligned} \mathbf{z}_{k+1} &= \delta\mathbf{p}^e = [\delta x, \delta y, \delta z] \\ &= \tilde{\mathbf{y}}^{GPS}_{k+1} - (\hat{\mathbf{p}} + \delta\mathbf{p})^e \end{aligned}$$

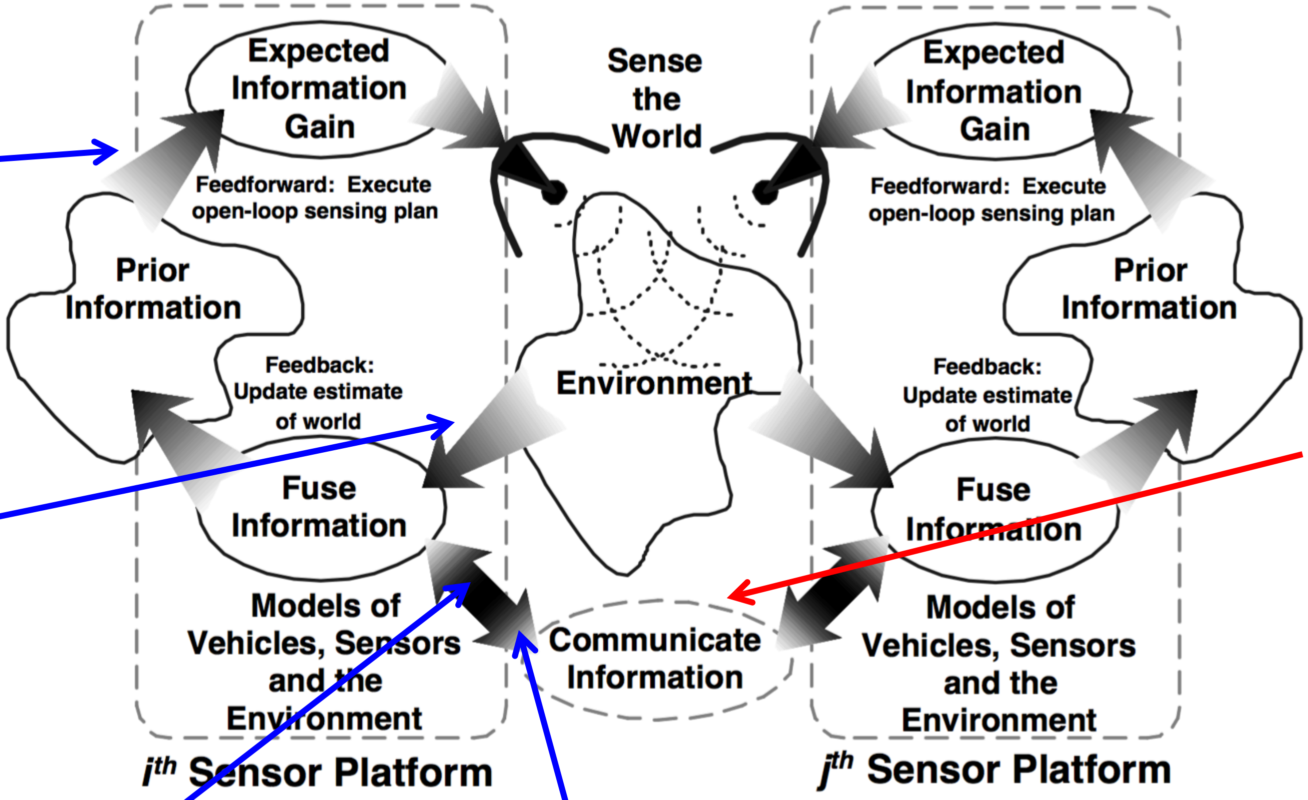
Write-up: https://tinyurl.com/nav-filter

$$\delta\mathbf{p}^e = \mathbf{R}^e_n\mathbf{D}\delta\mathbf{p} \longrightarrow \mathbf{z} = [\mathbf{R}^e_n\mathbf{D}, \mathbf{0}, \ldots, \mathbf{0}]\delta\mathbf{x} + \omega_{gps}$$

Distributed Data Fusion (DDF) Overview

1. Agent *i* runs filter **prediction** locally

2. Agent *i* runs filter **update** locally



1. **Channel Filter (CF)** between agents *i* and *j* **predicts** propagation of common information

2. CF receives local estimates from agents *i* and *j*

3. CF **updates** common information

4. Agent *i* fuses *new* information from agent *j* through **DDF**

3. Agent *i* communicates local estimate relevant to agent *j*

B. Grocholsky et. al.[3]

DDF Equations

- Most DDF solutions replace local Kalman Filter with **Information Filter (IF)** – better when more sensors than local states, which is typically the case with distributed estimation.[2]
- However...
  - Navigation filter equations are linearized about nominal state → need actual states at each iteration rather than information form.
  - Need nominal states to recover shared tracking estimations
- **IF states / covariance:**

$$\mathbf{y} = P^{-1}\mathbf{x}, Y = P^{-1}$$

$$\mathbf{i}^i_{k+1} = (H^i_{k+1})^T (R^i_{k+1})^{-1} \mathbf{z}^i_{k+1},$$
$$I^i_{k+1} = (H^i_{k+1})^T (R^i_{k+1})^{-1} H^i_{k+1},$$

- **IF prediction:**

$$M_k = F_k^{-T} Y_{k|k} F_k^{-1},$$
$$L_k = \left[I_n - M_k \left(M_k + Q_k^{-1} \right)^{-1} \right]$$
$$\mathbf{y}_{k+1|k} = L_k F_k^{-T} \mathbf{y}_{k|k},$$
$$Y_{k+1|k} = L_k M_k.$$

IF update:

$$\tilde{\mathbf{y}}_{k+1|k+1} = \mathbf{y}_{k+1|k} + \sum_{i=1}^{N} \mathbf{i}^i_{k+1},$$
$$\tilde{Y}_{k+1|k+1} = Y_{k+1|k} + \sum_{i=1}^{N} I^i_{k+1},$$

DDF Equations

- **Channel Filter (CF)** between communicating neighbors *i* and *j* to estimate common information to avoid double-counting.

CF states / covariance: $\mathbf{y}^{ij}, Y^{ij}$

<u>CF prediction</u>

(same as IF)

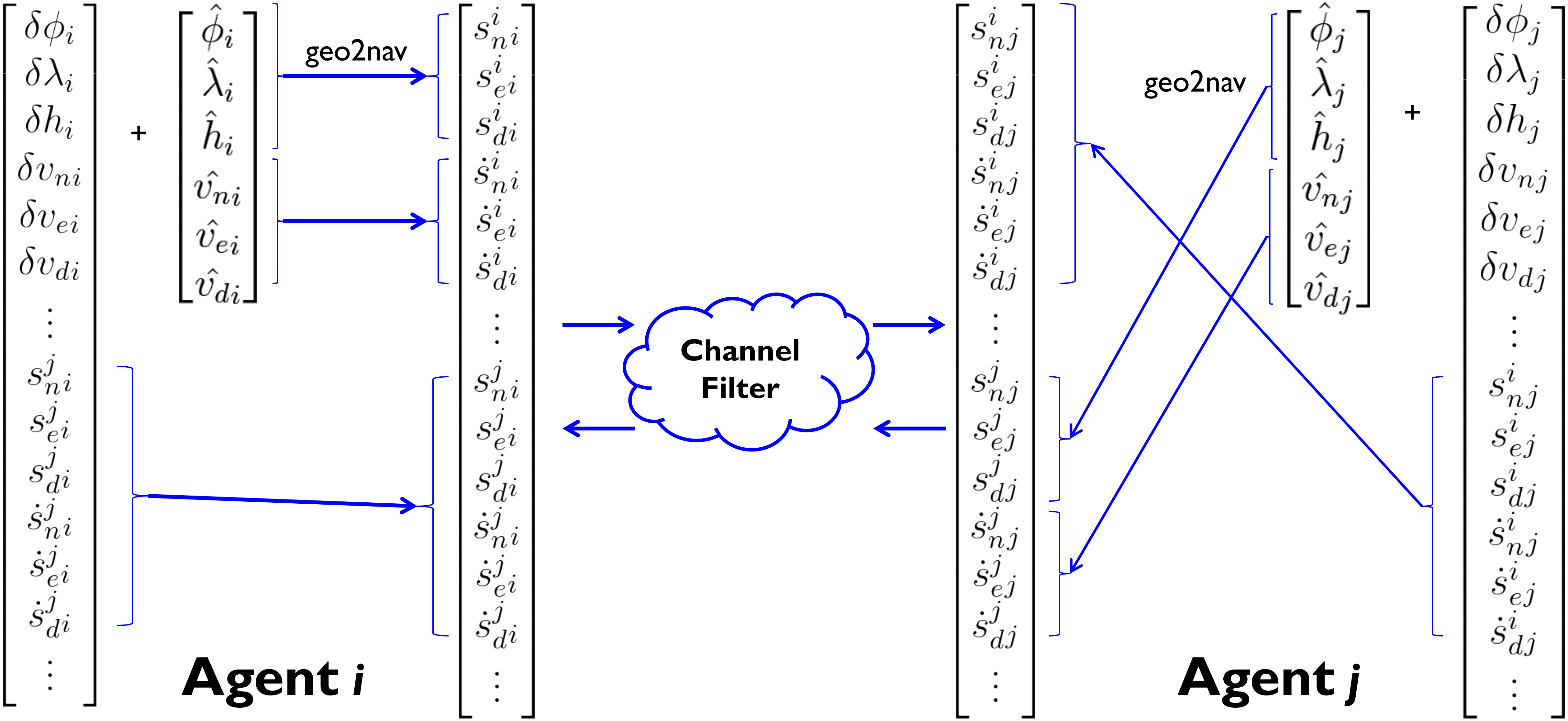
<u>CF update</u>

$$\mathbf{y}^{ij}_{k+1|k+1} = -\mathbf{y}^{ij}_{k+1|k} + \tilde{\mathbf{y}}^{i}_{k+1|k+1} + \tilde{\mathbf{y}}^{j}_{k+1|k+1},$$
$$Y^{ij}_{k+1|k+1} = -Y^{ij}_{k+1|k} + \tilde{Y}^{i}_{k+1|k+1} + \tilde{Y}^{j}_{k+1|k+1},$$

<u>DDF update</u>

$$\mathbf{y}^{i}_{k+1|k+1} = \tilde{\mathbf{y}}^{i}_{k+1|k+1} + \sum_{j \in N_i} \left[\tilde{\mathbf{y}}^{j}_{k+1|k+1} - \mathbf{y}^{ij}_{k+1|k} \right],$$
$$Y^{i}_{k+1|k+1} = \tilde{Y}^{i}_{k+1|k+1} + \sum_{j \in N_i} \left[\tilde{Y}^{j}_{k+1|k+1} - Y^{ij}_{k+1|k} \right],$$

DDF Implementation

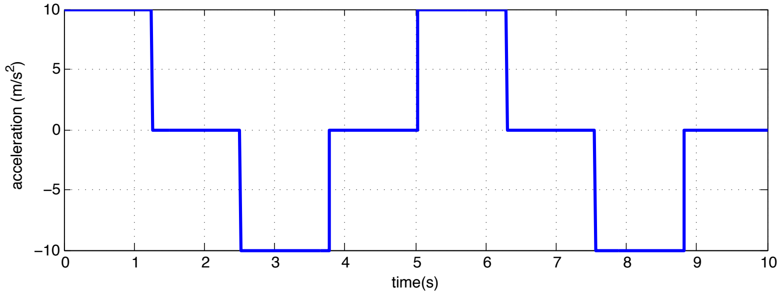


Motion of Tracked States

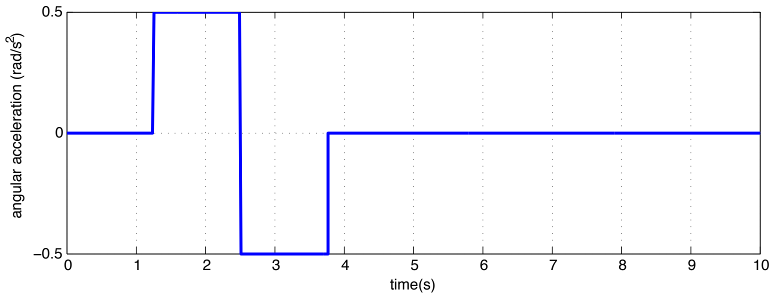
Truth: Dubin's (Unicycle) Model

$$\dot{v}_n = V cos(\psi), \dot{v}_e = V sin(\psi), \dot{\psi} = \omega$$

$\dot{V}$:



$\dot{\omega}$:



Local Nav. Filter: Complex, linearized dynamics

$$\begin{bmatrix} \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{v}} \\ \dot{\rho} \\ \delta\dot{\mathbf{x}}_a \\ \delta\dot{\mathbf{x}}_g \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{pp} & \mathbf{F}_{pv} & \mathbf{F}_{p\rho} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{vp} & \mathbf{F}_{vv} & \mathbf{F}_{v\rho} & -\hat{\mathbf{R}}_b^n \mathbf{F}_{va} & \mathbf{0} \\ \mathbf{F}_{\rho p} & \mathbf{F}_{\rho v} & \mathbf{F}_{\rho\rho} & \mathbf{0} & \hat{\mathbf{R}}_b^n \mathbf{F}_{\rho g} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{gg} \end{bmatrix} \begin{bmatrix} \delta\mathbf{p} \\ \delta\mathbf{v} \\ \rho \\ \delta\mathbf{x}_a \\ \delta\mathbf{x}_g \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\hat{\mathbf{R}}_b^n & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_b^n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \nu_a \\ \nu_g \\ \omega_a \\ \omega_g \end{bmatrix}$$

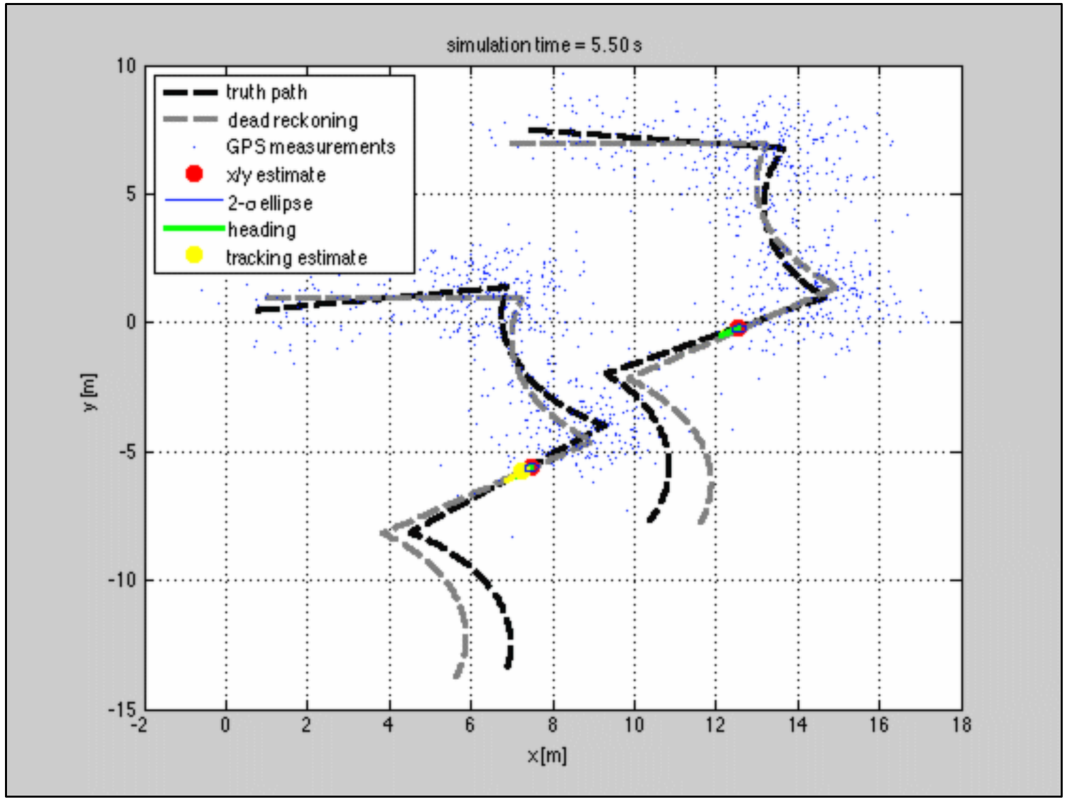
Modeled: Nearly-Constant Acceleration (NCA)

$$\begin{bmatrix} \dot{s}_n \\ \dot{s}_e \\ \dot{s}_d \\ \ddot{s}_n \\ \ddot{s}_e \\ \ddot{s}_d \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s_n \\ s_e \\ s_d \\ \dot{s}_n \\ \dot{s}_e \\ \dot{s}_d \end{bmatrix} + \begin{bmatrix} \omega_n \\ \omega_e \\ \omega_d \\ \nu_n \\ \nu_e \\ \nu_d \end{bmatrix}$$

Toy Problem

- Two agents moving in unicycle motion, communicating and tracking each other with Cartesian range measurements.

$$\mathbf{x}_i = \begin{bmatrix} \left.\begin{matrix} x_i \\ y_i \\ \theta_i \end{matrix}\right\} \text{local "nav." states} \\ \hline \left.\begin{matrix} s^j_{xi} \\ s^j_{yi} \\ \vdots \end{matrix}\right\} \text{tracked position states} \end{bmatrix}$$



NCV Tracking Setup

NCV Model: $\mathbf{x}^j_{k+1} = \mathbf{x}^j_k + \omega^j_k, E[\omega^j_k \omega^{jT}_l] = Q\delta_{kl}$

$$Q = \begin{bmatrix} q_x & 0 \\ 0 & q_y \end{bmatrix}, q_x = q_y = q$$

NCV Tuning Parameter

Truth (unicycle):

$|v_c|_{max} = 5 \text{ m/s}$ $\quad E[\mu_v \mu_v^T] = (0.1)^2$

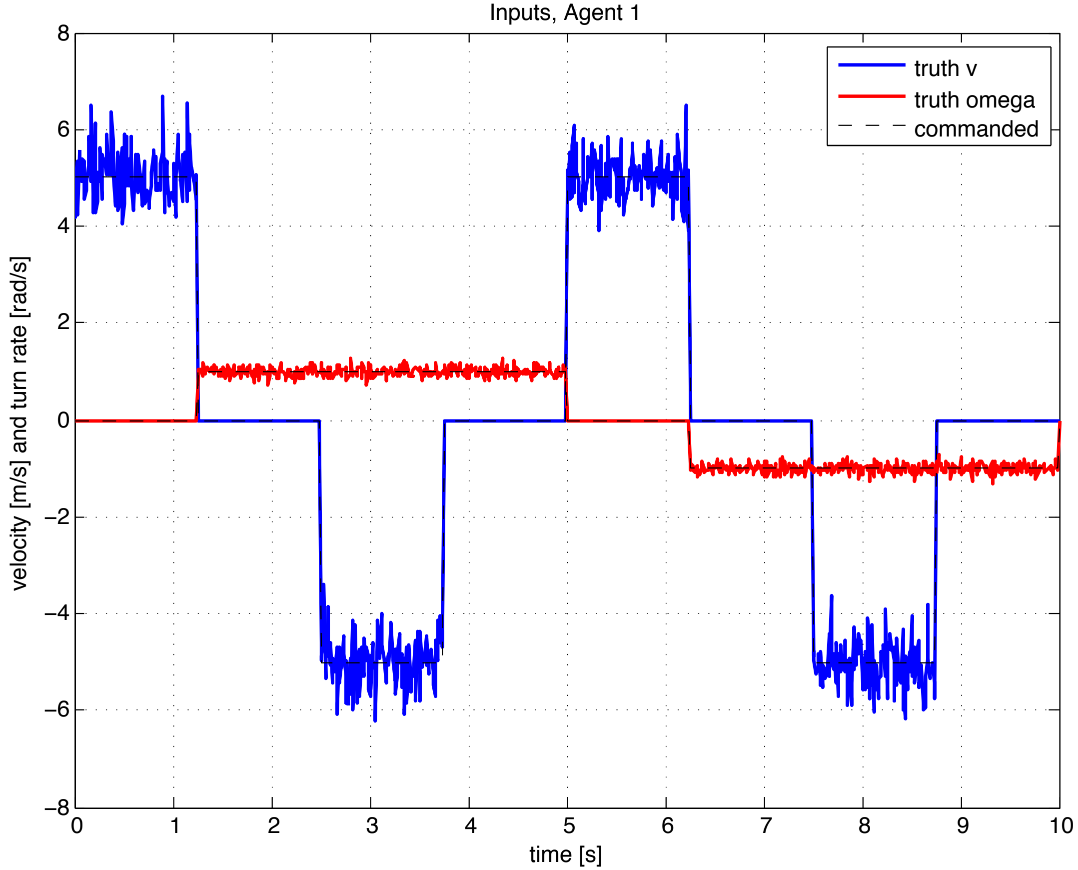
$$\dot{x} = v_c(1 + \mu_v)\cos\theta$$

$$\dot{y} = v_c(1 + \mu_v)\sin\theta$$

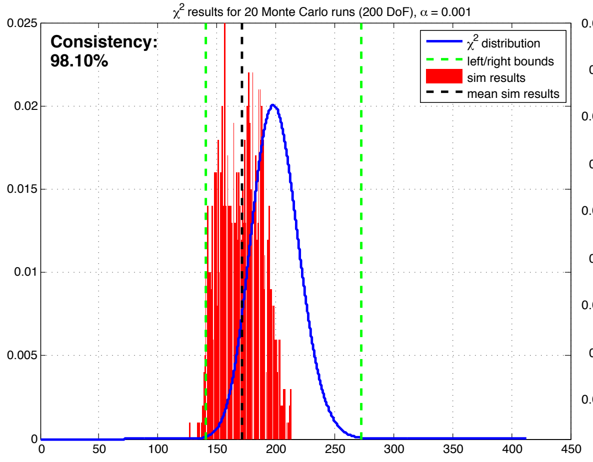
$$\dot{\theta} = \omega_c(1 + \mu_\omega)$$

$E[\mu_\omega \mu_\omega^T] = (0.1)^2$

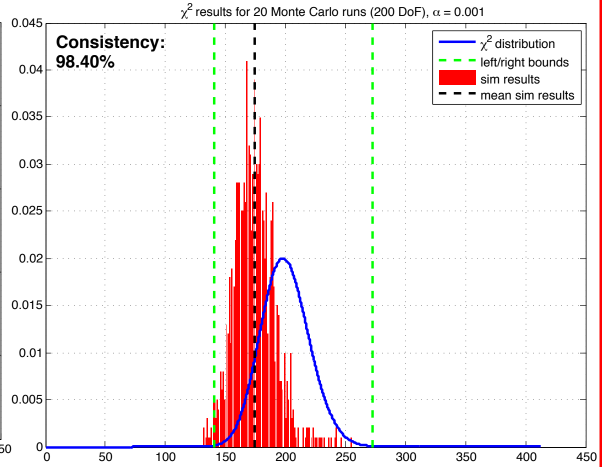
$|\omega_c|_{max} = 1 \text{ rad/s}$



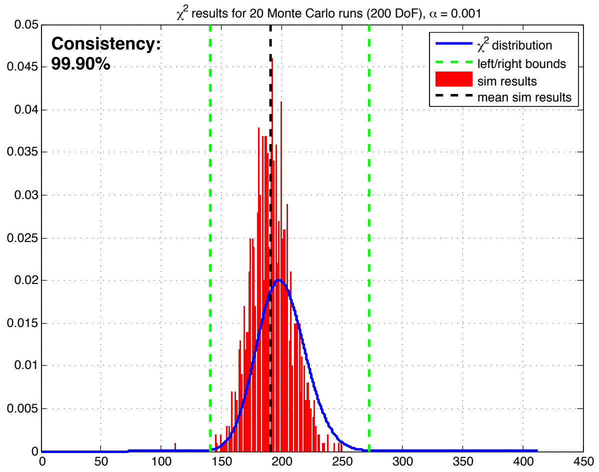
Chi-squared Results for different NCV q values, 20 sims



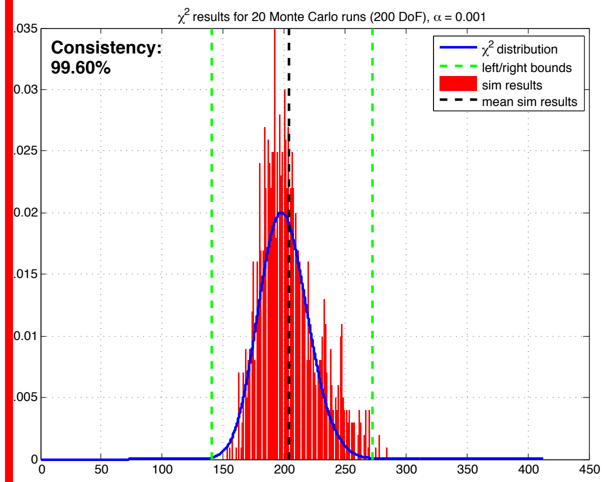
$$q = (0.5\ \mathrm{m/s})^2$$



$$q = (1\ \mathrm{m/s})^2$$

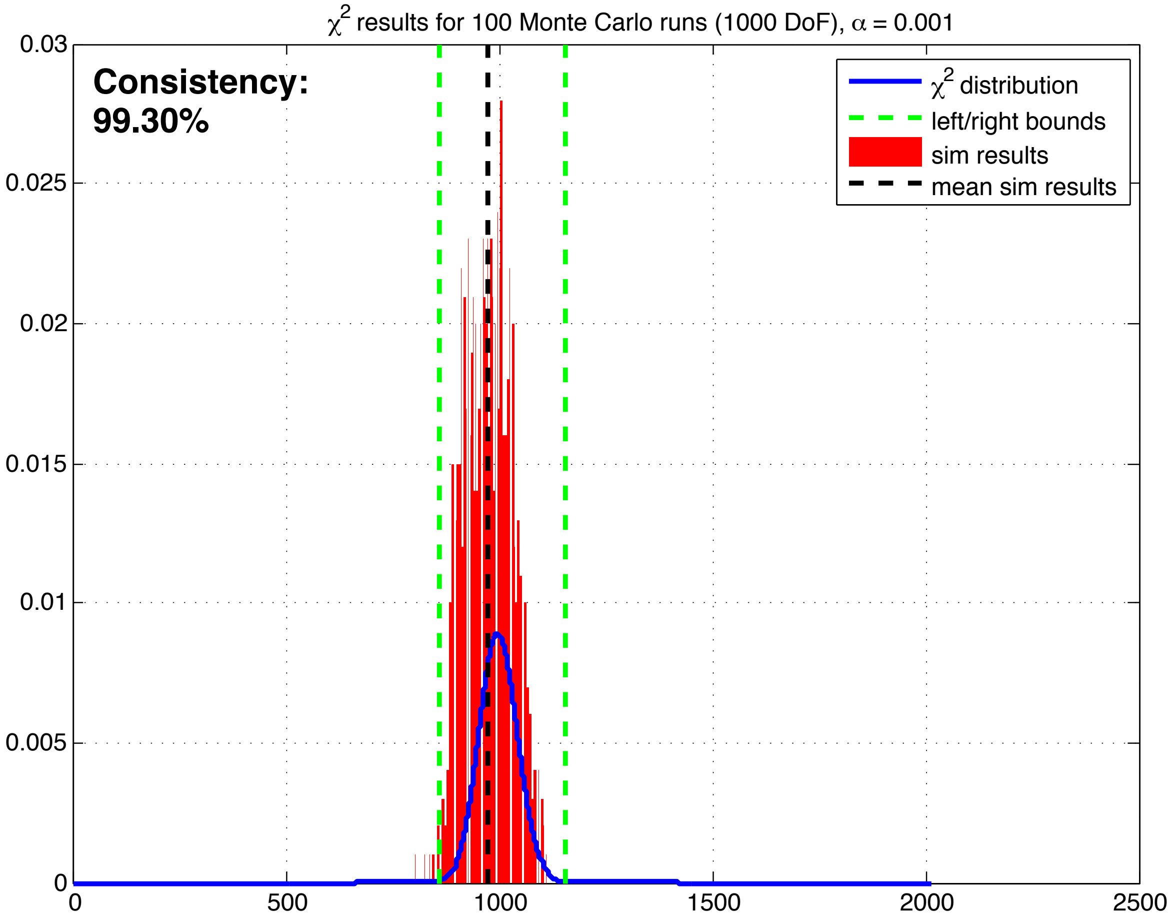


$$q = (2\ \mathrm{m/s})^2$$



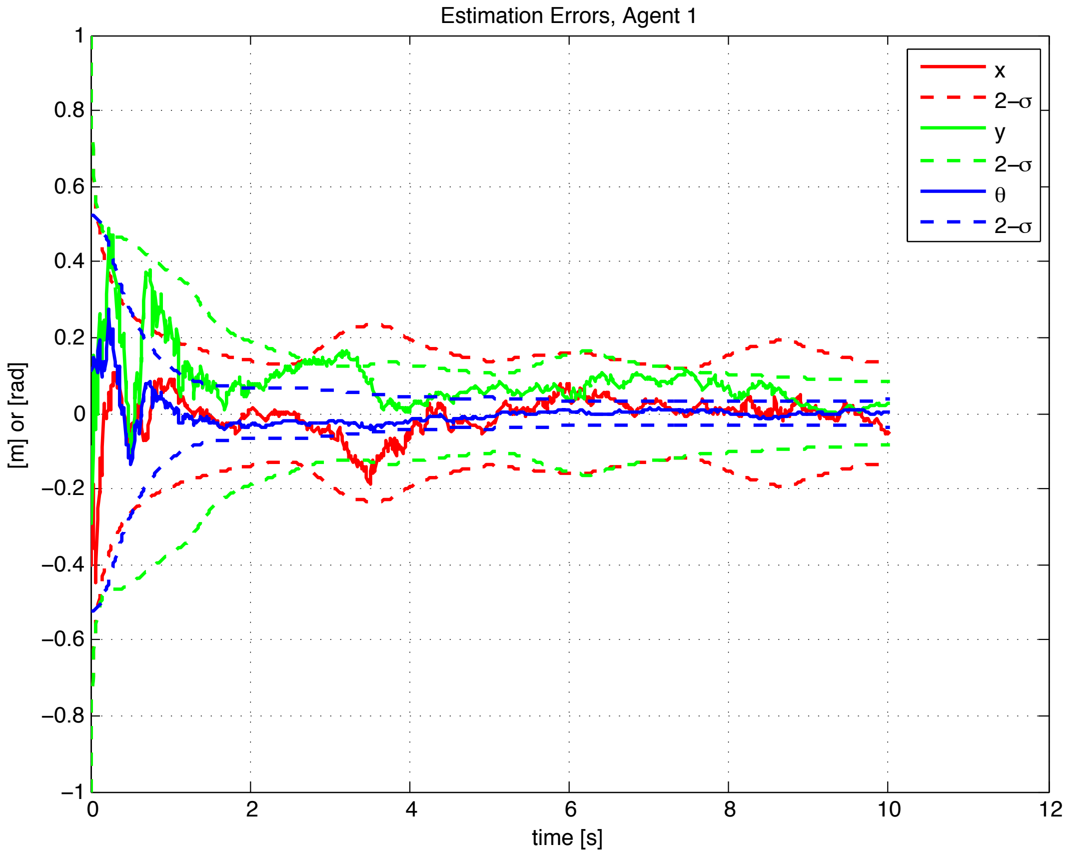
$$q = (5\ \mathrm{m/s})^2$$

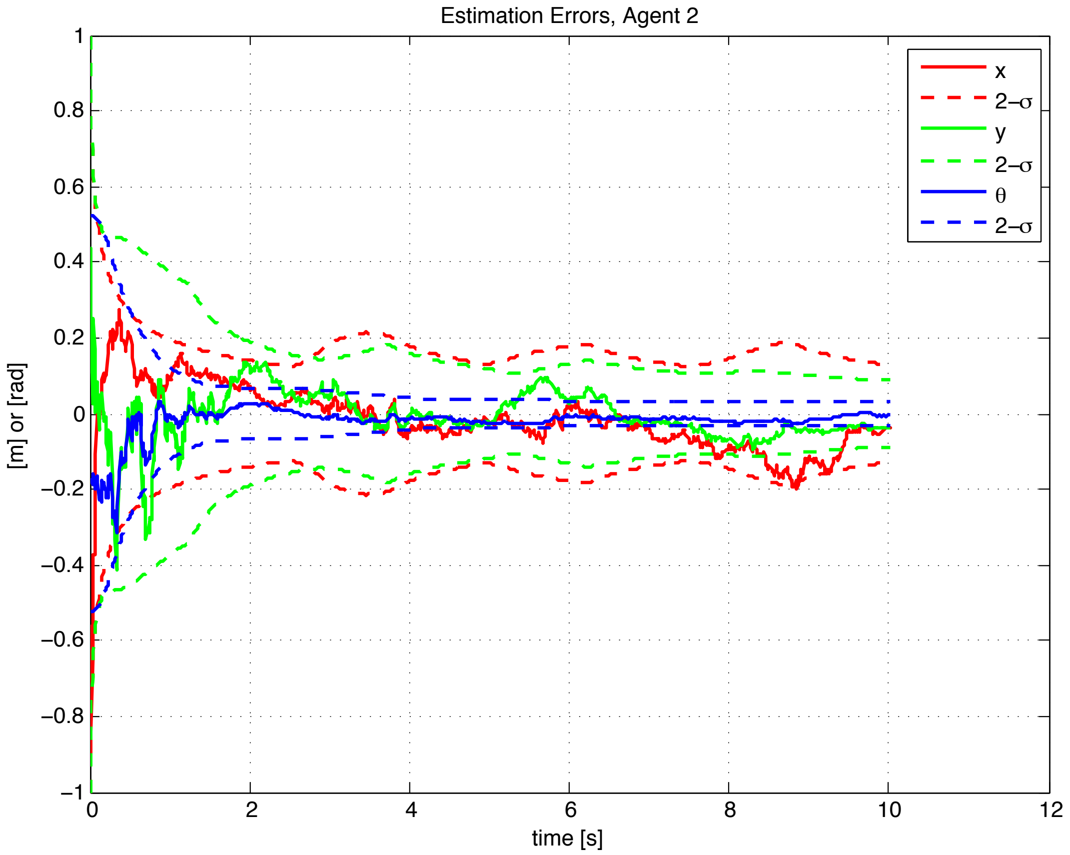
Chi-squared NCV, 100 sims



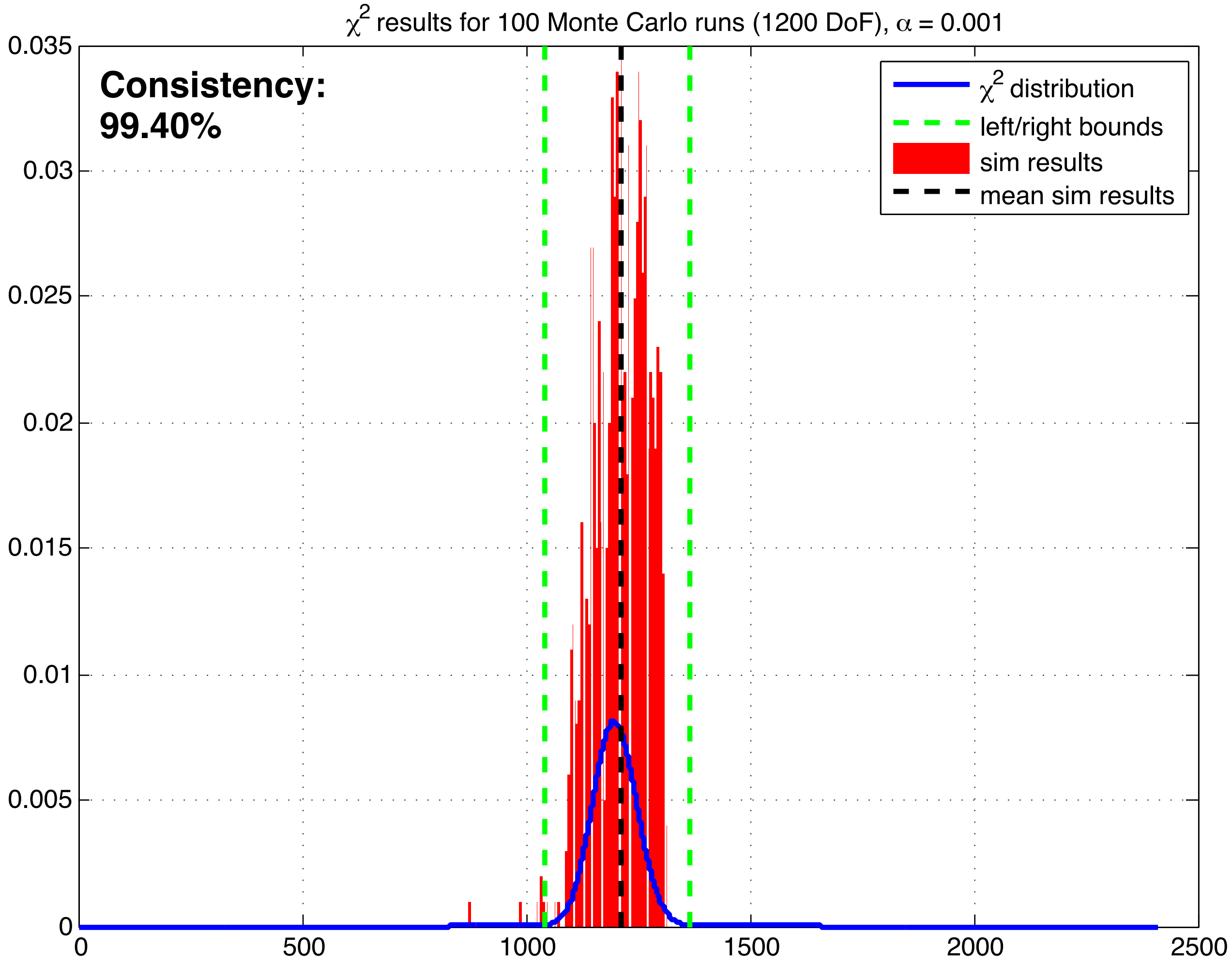
$$q = (2\ \mathrm{m/s})^2$$

2-agent EKF with **unicycle + shared input tracking**, no DDF – example estimation errors for both agents

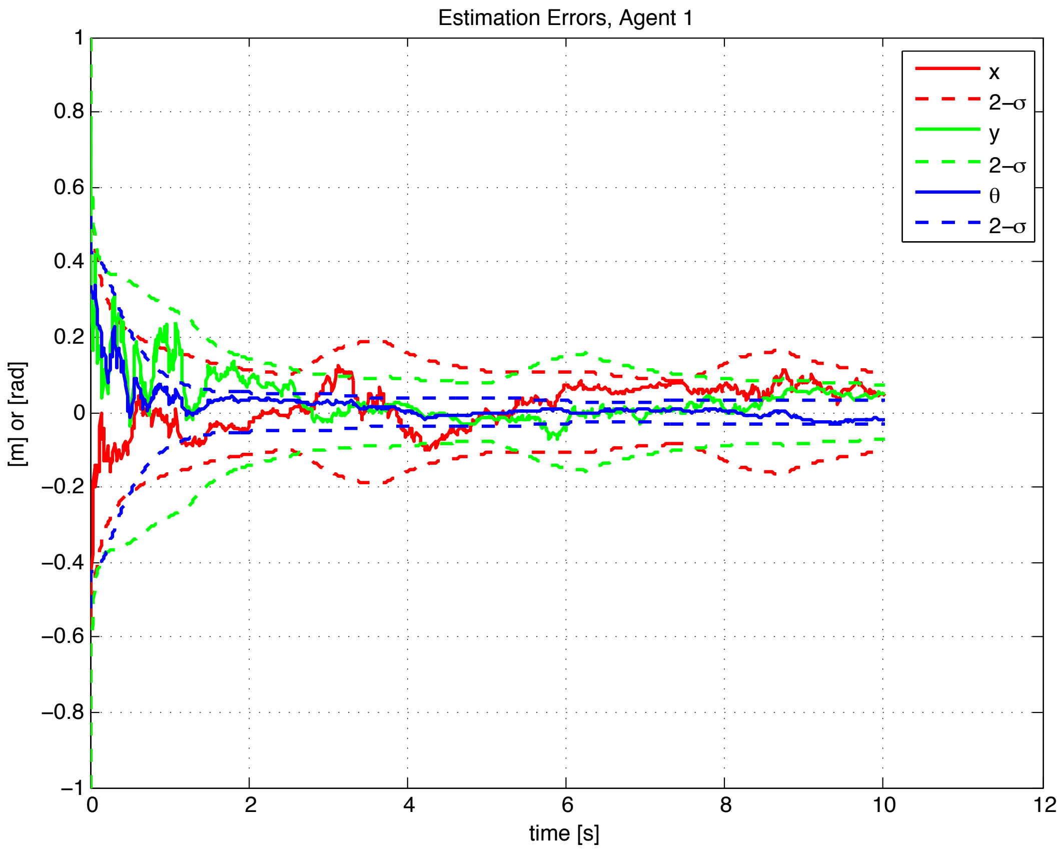


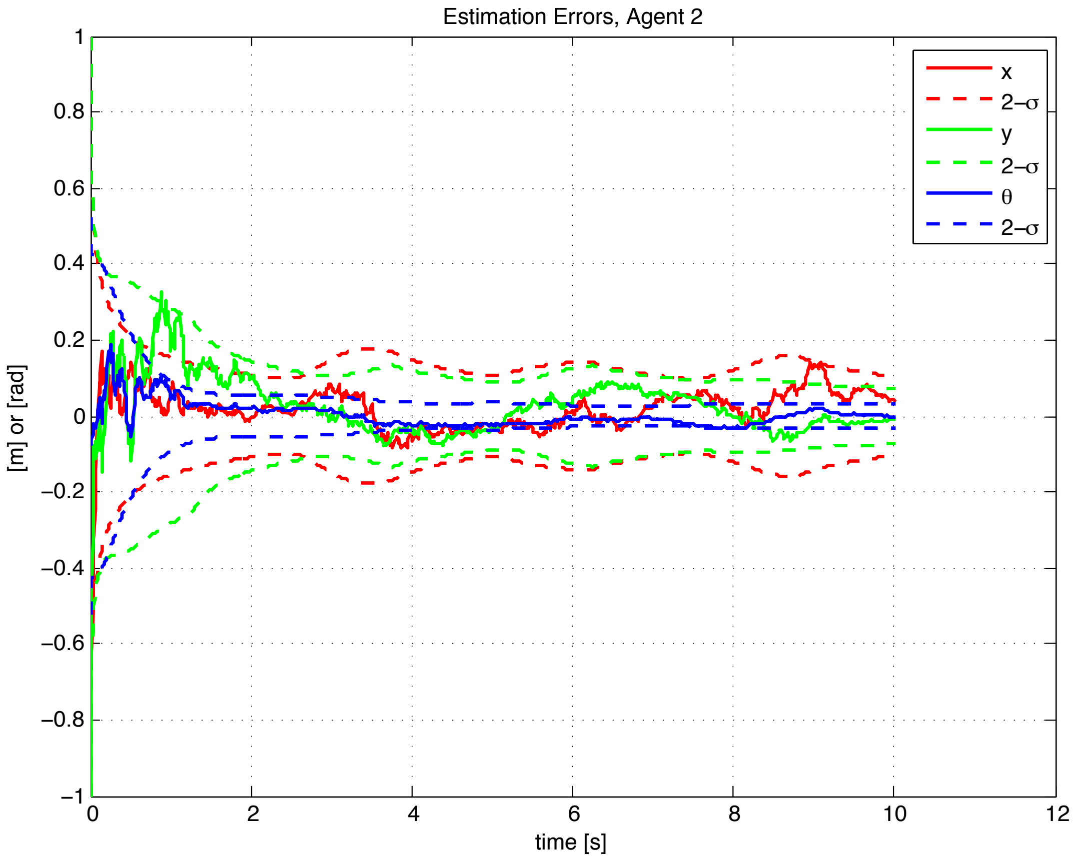


2-agent EKF with **unicycle + shared input tracking**, no DDF – chi-squared results for 100 sims

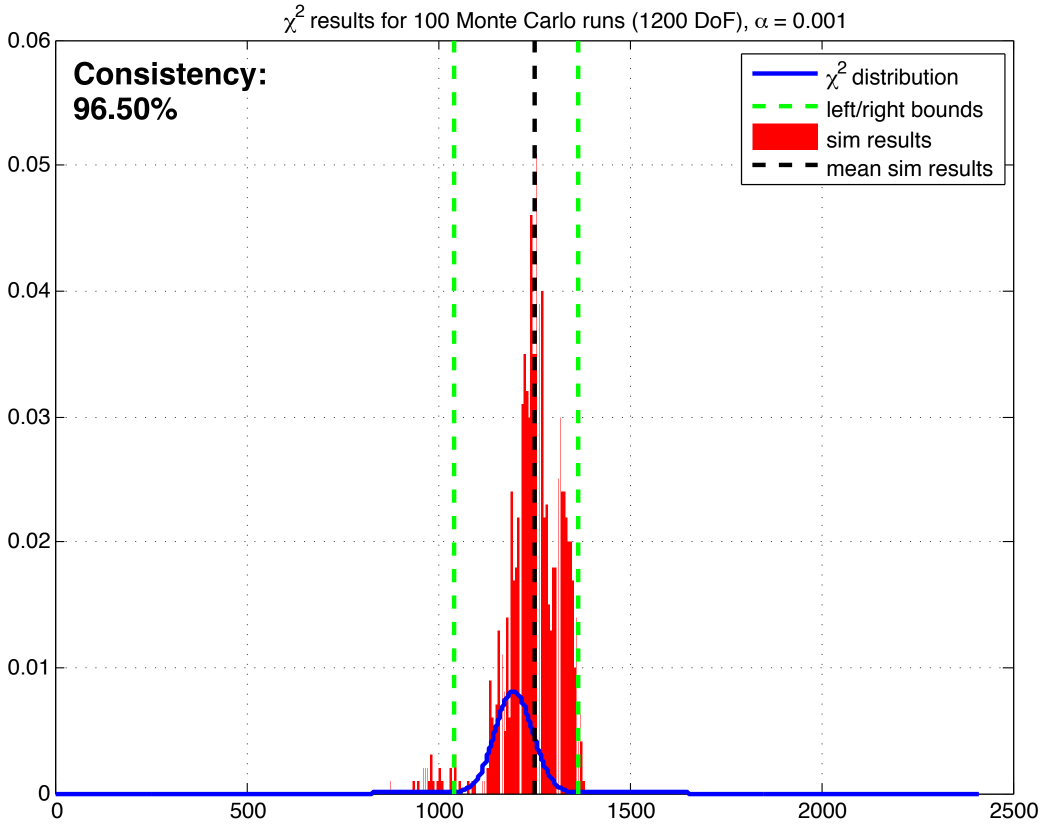


2-agent EKF with **unicycle + shared input tracking** **<u>& DDF</u>** – example estimation errors for both agents

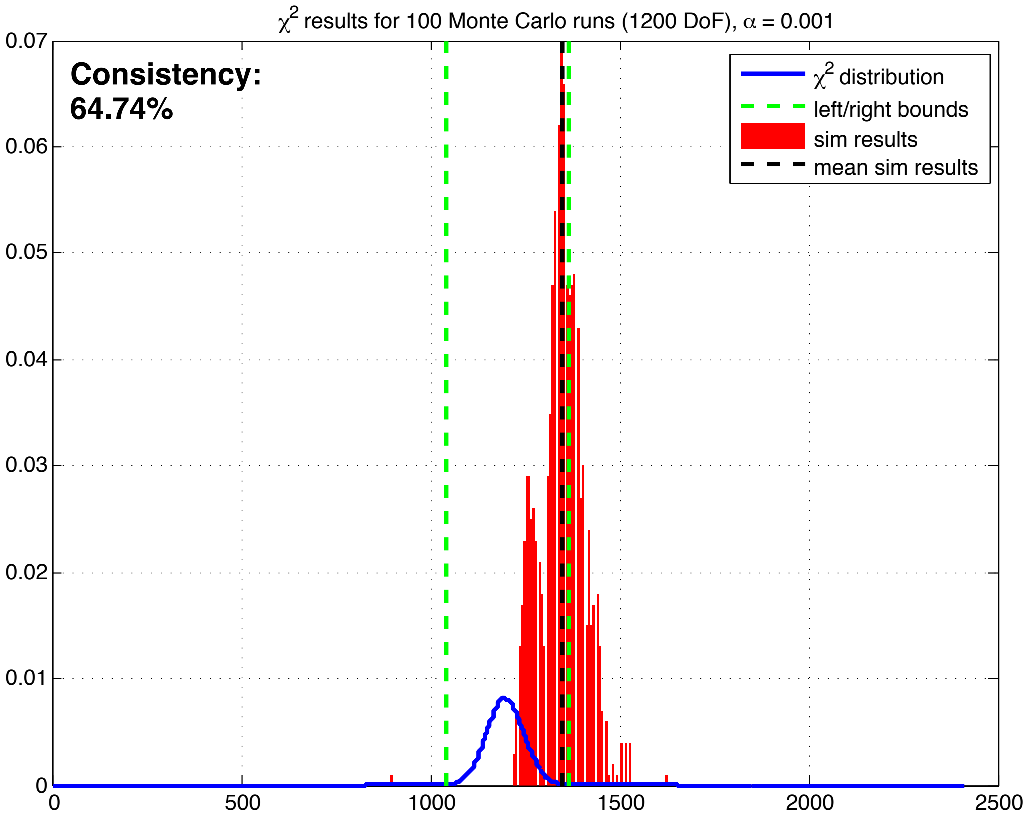




2-agent EKF with **unicycle + shared input tracking <u>& DDF</u>** – chi-squared results



30 runs



100 runs

Next Steps

1. Achieve consistency of channel filter DDF implementation of unicycle tracking model.
2. Test channel filter implementation with alternate tracking models
   - NCV model
   - Unicycle without shared inputs
3. Implement DDF algorithm + tracking model in nav. filter and verify consistency.
4. Simulate more realistic range measurements / characteristics (e.g., through radio-ranging)

References

1. S. Grime, H. F. Durrant-Whyte. *Data Fusion in Decentralized Sensor Networks*. Control Engineering Practice vol. 2 no. 5 pp. 849-863, 1994.
2. M. Cambell, N. Ahmed. *Distributed Data Fusion*. IEEE Control Systems Magazine, p.83. August 2016.
3. B. Grocholsky, A. Makarenko, T. Kaupp, and H. F. Durrant-Whyte. *Scalable Control of Decentralised Sensor Platforms*. April 2003.
4. V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert. *Incremental Distributed Inference from Arbitrary Poses and Unkown Data Association*. IEEE Control Systems Magazine, pp. 41-74, 2016.
5. M. McClelland, M. Campbell, T. Estlin. *Qualitative Relational Mapping for Planetary Rovers*. Proceedings of Workshop on Intelligent Robotic Systems AAAI. Washington, USA, pp. 110-113, 2013.
6. R. Li, S. He, B. Skopljak, et al. *A Multisensor Integration Approach toward Astronaut Navigation for Landed Lunar Missions*. Journal of Field Robotics, 31(2):245-262, 2014.
7. J. Farrell, *Aided Navigation: GPS with High Rate Sensors*, McGraw-Hill Companies, 2008.
8. S. Julier, J. Uhlmann. *A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations*. Americal Control Covference, 1997.
9. N. Ahmed. *Conditionally Factorized DDF for General Distributed Bayesian Estimation*, 1997.

Papers To Read

1. Will 2011 (Decentralized Geolocation and Bias Estimation for Uninhabited Aerial Vehicles with Articulating Cameras)
2. Uhlmann 1999 (application of covariance intersection and DDF to a Mars rover)
3. Hurley 2002 (which talks about generalizing the covariance intersection idea)
4. Noack 2014 (distributed fusion of KFs with unequal state vectors)
5. Nisar 2014 MFI paper (my original paper on the idea of FDDF)
6. Nisar FUSION 2016 paper (multiple agents tracking a common target with local sensor biases that they don't want to fuse, but need to maintain estimates for)
7. Carillo-Arce IROS 2013 (the one I sent you earlier on cooperative localization)
8. Sijs 2010 (ellipsoidal intersection)
9. Noack 2017 (which talks about fixing consistency issues with ellipsoidal intersection)
10. Martin and Chang FUSION 2005 (Data Fusion trees)

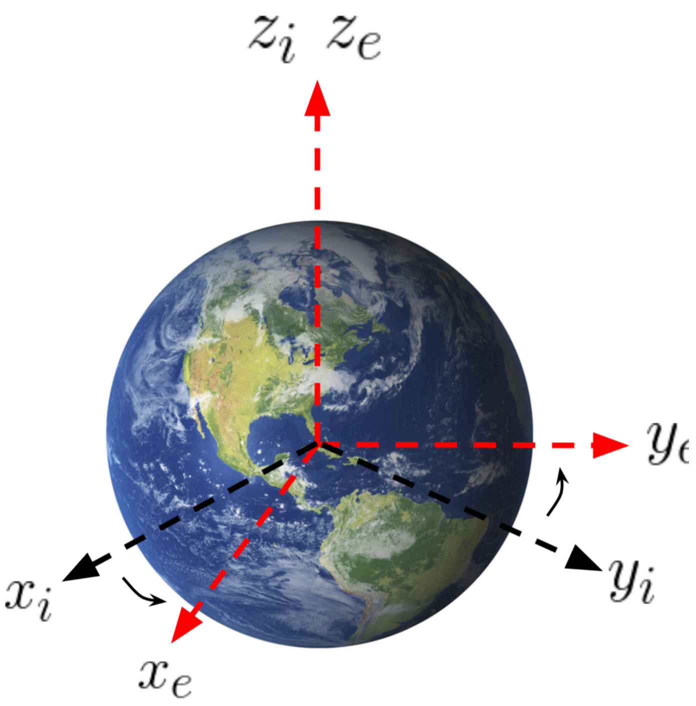
Appendix

COHRINT

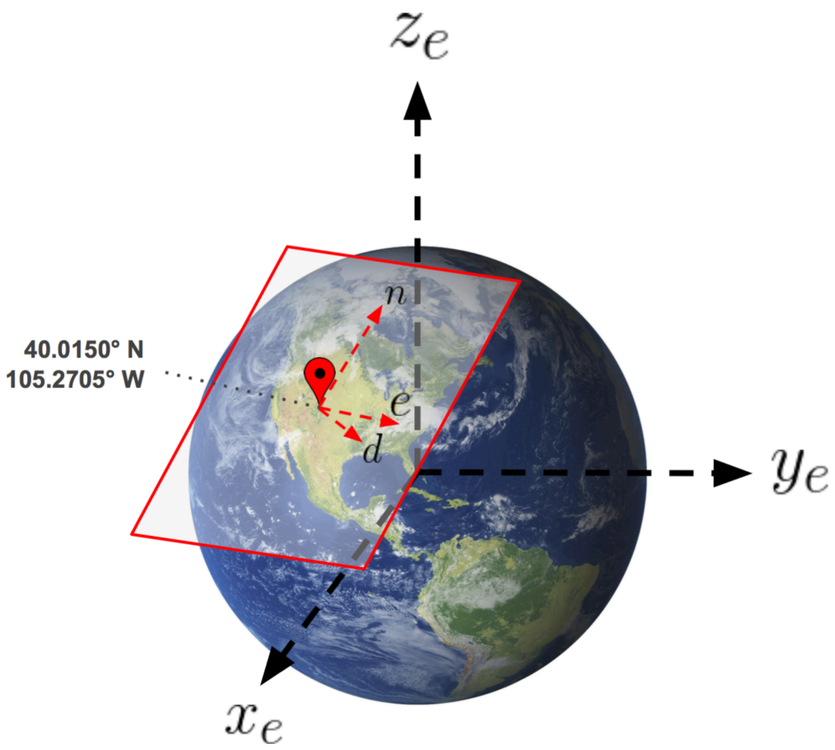
Coordinate Systems

Navigation frame

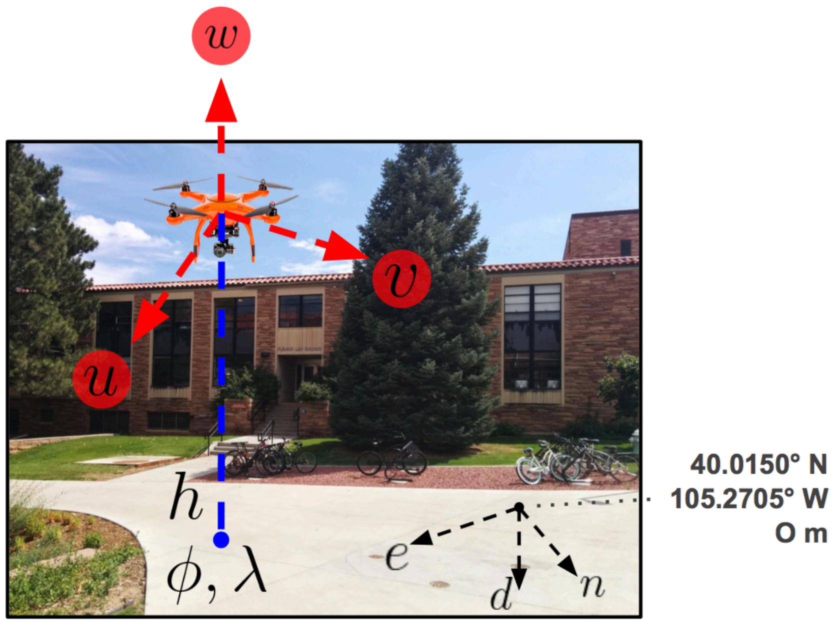
ECI & ECEF



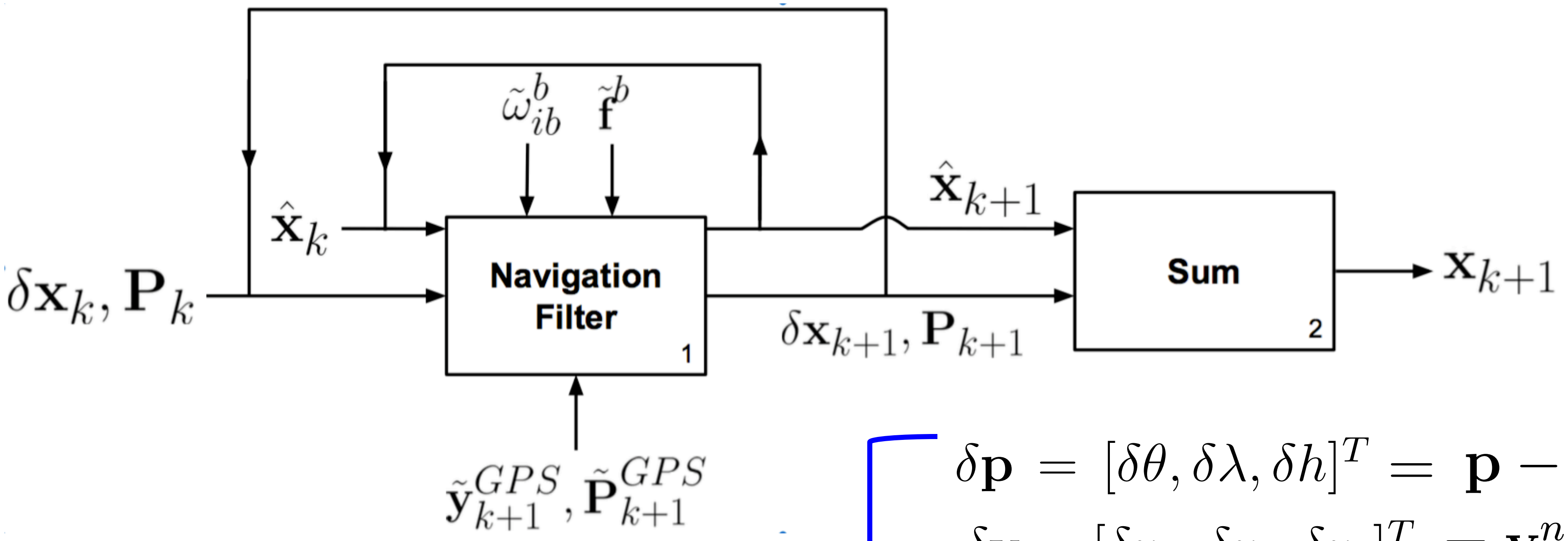
ECEF & Tangential frame (North-East-Down)



Navigation, Body, & Geographic frame



Navigation Filter Block Diagram – High Level



33 states

$$\delta\mathbf{x} = [\delta\mathbf{p}, \delta\mathbf{v}, \rho, \delta\mathbf{x}_a, \delta\mathbf{x}_g]^T$$

$$\delta\mathbf{p} = [\delta\theta, \delta\lambda, \delta h]^T = \mathbf{p} - \hat{\mathbf{p}}$$

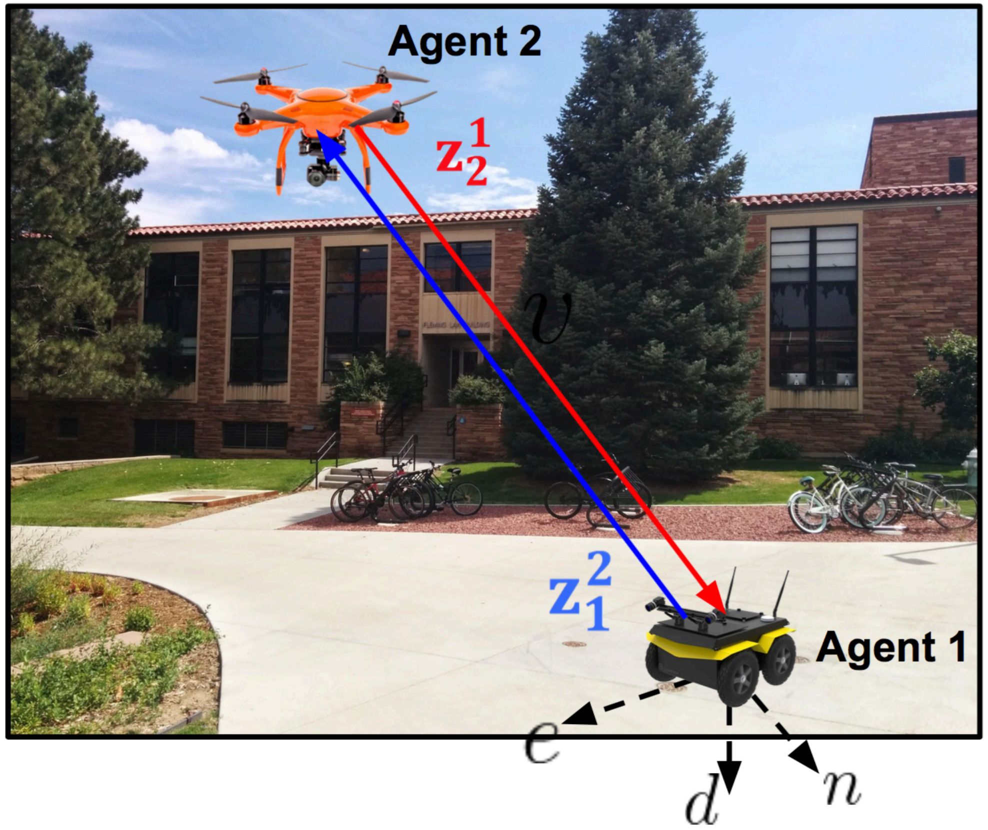
$$\delta\mathbf{v} = [\delta v_n, \delta v_e, \delta v_d]^T = \mathbf{v}^n_e - \hat{\mathbf{v}}^n_e$$

$$\rho = [\epsilon_N, \epsilon_E, \epsilon_D], \; \hat{\mathbf{R}}^n_b = (\mathbf{I} - [\rho\times])\mathbf{R}^n_b$$

$$\delta\mathbf{x}_a = [\mathbf{x}^T_{b_a}, \mathbf{x}^T_{A_a}, \cancel{\mathbf{x}^T_{ka}}], \; \delta\mathbf{f}^b = \mathbf{F}_{va}\delta\mathbf{x}_a + \nu_a$$

$$\delta\mathbf{x}_g = [\mathbf{x}^T_{b_g}, \mathbf{x}^T_{A_g}, \cancel{\mathbf{x}^T_{kg}}], \; \delta\omega^b_{ib} = \mathbf{F}_{\rho g}\delta\mathbf{x}_g + \nu_g$$

Tracking Measurement Equation

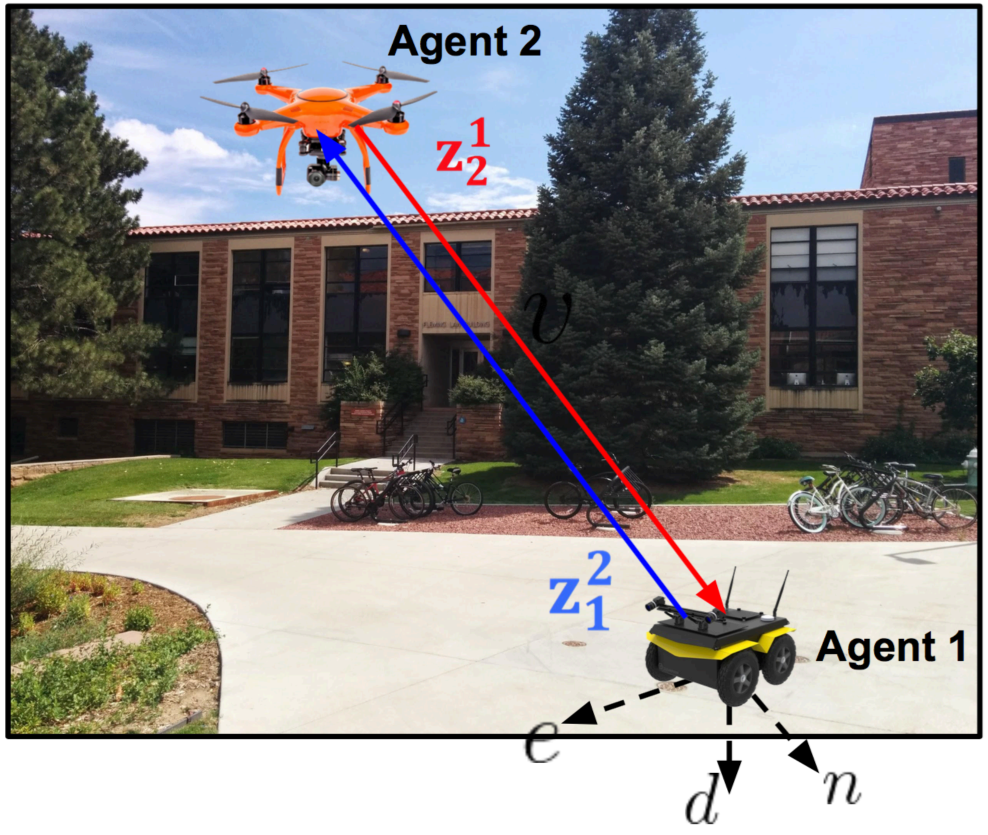


$$\mathbf{z}_1^2 = \overbrace{\begin{bmatrix} \Delta p_{1,n}^2 \\ \Delta p_{1,e}^2 \\ \Delta p_{1,d}^2 \end{bmatrix}}^{\text{Range Meas.}} = \begin{bmatrix} \overbrace{s_n^2}^{\text{Tracking States}} - \overbrace{(\hat{p}_{n,1} + \delta p_{n,1})}^{\text{Corrected Nav. States}} + \overbrace{\omega_n}^{\text{Range Sensor Noise}} \\ s_e^2 - (\hat{p}_{e,1} + \delta p_{e,1}) + \omega_e \\ s_d^2 - (\hat{p}_{d,1} + \delta p_{d,1}) + \omega_d \end{bmatrix}$$

$$\mathbf{z}' = \mathbf{H}\delta\mathbf{x} + \omega$$

$$\begin{bmatrix} \Delta p_{1,n}^2 + \hat{p}_{n,1} \\ \Delta p_{1,d}^2 + \hat{p}_{e,1} \\ \Delta p_{1,d}^2 + \hat{p}_{d,1} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & & 1 & 0 & 0 & \\ 0 & -1 & 0 & \dots & 0 & 1 & 0 & \dots \\ 0 & 0 & -1 & & 0 & 0 & 1 & \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \overbrace{\begin{bmatrix} \delta\phi \\ \delta\lambda \\ \delta h \\ \vdots \\ s_n^2 \\ s_e^2 \\ s_d^2 \\ \vdots \end{bmatrix}}^{\delta\mathbf{x}'} + \begin{bmatrix} \omega_n \\ \omega_e \\ \omega_d \end{bmatrix}$$

Meas. Sharing Measurement Equations



$$\textbf{Agent 2} \xrightarrow{\text{Tracking, GPS: } \mathbf{z}_2^1,\, \tilde{\mathbf{y}}_2^{GPS}} \textbf{Agent 1}$$

1. Tracking:

$$\mathbf{z}_2^1 = -\mathbf{z}_1^2 \rightarrow$$ same tracking meas. eq.

1. GPS:

$$\begin{bmatrix} \tilde{\mathbf{y}}_{2,n}^{GPS} \\ \tilde{\mathbf{y}}_{2,e}^{GPS} \\ \tilde{\mathbf{y}}_{2,d}^{GPS} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 1 & 0 & 0 & \dots \\ 0 & \dots & 0 & 1 & 0 & \dots \\ 0 & \dots & 0 & 0 & 1 & \dots \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \vdots \\ s_n^2 \\ s_e^2 \\ s_d^2 \\ \vdots \end{bmatrix}$$

DDF Equations

- Most DDF solutions replace local Kalman Filter with **Information Filter (IF)** – better when more sensors than local states, which is typically the case with distributed estimation.[2]
- However...
  - Navigation filter equations are linearized about nominal state → need actual states at each iteration rather than information form.
  - Need nominal states to recover shared tracking estimations
- **IF states / covariance:**

$$\mathbf{y} = P^{-1}\mathbf{x}, Y = P^{-1}$$

$$\mathbf{i}^i_{k+1} = (H^i_{k+1})^T (R^i_{k+1})^{-1} \mathbf{z}^i_{k+1},$$
$$I^i_{k+1} = (H^i_{k+1})^T (R^i_{k+1})^{-1} H^i_{k+1},$$

- **IF prediction:**

$$M_k = F_k^{-T} Y_{k|k} F_k^{-1},$$
$$L_k = \left[I_n - M_k\left(M_k + Q_k^{-1}\right)^{-1}\right]$$
$$\mathbf{y}_{k+1|k} = L_k F_k^{-T} \mathbf{y}_{k|k},$$
$$Y_{k+1|k} = L_k M_k.$$

IF update:

$$\tilde{\mathbf{y}}_{k+1|k+1} = \mathbf{y}_{k+1|k} + \sum_{i=1}^{N} \mathbf{i}^i_{k+1},$$
$$\tilde{Y}_{k+1|k+1} = Y_{k+1|k} + \sum_{i=1}^{N} I^i_{k+1},$$

DDF Equations

- **Channel Filter (CF)** between communicating neighbors *i* and *j* to estimate common information to avoid double-counting.

CF states / covariance: $\mathbf{y}^{ij}, Y^{ij}$

CF prediction

(same as IF)

CF update

$$\mathbf{y}^{ij}_{k+1|k+1} = -\mathbf{y}^{ij}_{k+1|k} + \tilde{\mathbf{y}}^{i}_{k+1|k+1} + \tilde{\mathbf{y}}^{j}_{k+1|k+1},$$
$$Y^{ij}_{k+1|k+1} = -Y^{ij}_{k+1|k} + \tilde{Y}^{i}_{k+1|k+1} + \tilde{Y}^{j}_{k+1|k+1},$$

DDF update

$$\mathbf{y}^{i}_{k+1|k+1} = \tilde{\mathbf{y}}^{i}_{k+1|k+1} + \sum_{j\in N_i} \left[\tilde{\mathbf{y}}^{j}_{k+1|k+1} - \mathbf{y}^{ij}_{k+1|k}\right],$$
$$Y^{i}_{k+1|k+1} = \tilde{Y}^{i}_{k+1|k+1} + \sum_{j\in N_i} \left[\tilde{Y}^{j}_{k+1|k+1} - Y^{ij}_{k+1|k}\right],$$