Sigma-Point Kalman Filter based Integrated Navigation Systems (Overview)

Eric A. Wan, Rudolph van der Merwe, Alexander Bogdanov, Geoff Harvey

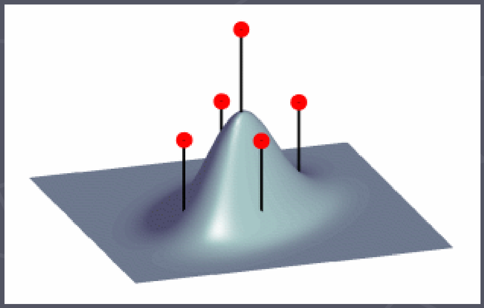
OGI School of Science and Engineering at OHSU

SPKF Integrated Navigation

- Integration of IMU, GPS, and additional sensors (e.g., altimeter) to provide accurate vehicle state estimation.
  - States (position, attitude, velocity, etc.) used for feedback control, fault detection, tracking, trajectory planning, etc.
- Core to all current Integrated Navigation systems is the Extended Kalman Filter (EKF).
- New Sigma-Point Kalman Filter (SPKF) is theoretically superior to the EKF
- Software Solution
- Provides significant performance/cost benefits.
- Applicable to Manned or Unmanned vehicles.
- Patent Pending

Sigma Point Kalman Filtering

- New approach to nonlinear recursive Bayesian Inference.
- Consistently outperform the Extended Kalman Filter (EKF).
- Same order computational complexity as the EKF.
- Accurate to at least the 2nd order (3rd for Gaussian inputs).
- Efficient "sampling" approach using only functional evaluations (no analytic derivatives).
- *See separate presentation or references for details.*



SPKF Integrated Navigation Solution

- Kinematic System Model
  - 16 state 6DOF model of vehicle dynamics used for state estimation
    - 3D position and velocity (inertial frame : North, East, Down)
    - 4D attitude (body frame : quaternion representation)
    - IMU gyro rate and linear acceleration biases (6D)
  - IMU used to capture and summarize forces and moments operating on vehicle
    - "Vehicle Independent"
  - Accounts for coordinate transformation between body and inertial frames.
  - Accounts for sensor coupling due to geometry (e.g., GPS offset from center of gravity)

- Sensor updates
  - GPS - Measures lagged position and velocity in inertial frame
  - Altimeter - Barometric measurement of absolute altitude
  - Nonlinear function of the system state.
  - Used for discrete measurement update (prediction correction)

- Efficient SPKF implementation
  - Provides accurate state estimates accounting for nonlinearities, asynchronous, and lagged measurement update.

Details: Kinematic Model

- State :

$$\mathbf{x} = [\mathbf{p}\ \ \mathbf{v}\ \ \mathbf{e}\ \ \mathbf{b}_{\boldsymbol{\omega}}\ \ \mathbf{b}_{\mathbf{a}}]^T = [x_N\ x_E\ x_D\ \ v_N\ v_E\ v_D\ \ e_0\ e_1\ e_2\ e_3\ \ b_p\ b_q\ b_r\ \ b_{ax}\ b_{ay}\ b_{az}]^T$$

- IMU data first corrected using estimated biases:

$$\bar{\boldsymbol{\omega}} = \boldsymbol{\omega} - \mathbf{b}_{\boldsymbol{\omega}} = [\bar{p}\ \bar{q}\ \bar{r}]^T = [p\ q\ r]^T - [b_p\ b_q\ b_r]^T$$

$$\bar{\mathbf{a}} = \mathbf{a} - \mathbf{b}_{\mathbf{a}} = [\bar{a}_x\ \bar{a}_y\ \bar{a}_z]^T = [a_x\ a_y\ a_z]^T - [b_{ax}\ b_{ay}\ b_{az}]^T$$

- Time-update of state at IMU rate using nonlinear “kinematic” function

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_k, \bar{\boldsymbol{\omega}}_k, \bar{\mathbf{a}}_k) \quad \begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\mathbf{v}} = \overset{b\to i}{\mathbf{T}}\left(\bar{\mathbf{a}} - \dot{\bar{\boldsymbol{\omega}}} \times \mathbf{r}_{IMU} - \bar{\boldsymbol{\omega}} \times (\bar{\boldsymbol{\omega}} \times \mathbf{r}_{IMU})\right) + [0\ 0\ g]^T \\ \dot{\mathbf{e}} = -\tfrac{1}{2}\tilde{\boldsymbol{\Omega}} + l\left(1 - \|\mathbf{e}\|^2\right)\mathbf{e} \\ \dot{\mathbf{b}}_{\boldsymbol{\omega}} = 0 \\ \dot{\mathbf{b}}_{\mathbf{a}} = 0 \end{cases}$$

IMU offset from CG

Lagrange multiplier

Details: Kinematic Model

- Body-to-inertial frame transformation (DCM) matrix:

$$\overset{b\to i}{\mathbf{T}} = \left(\overset{i\to 2}{\mathbf{T}}\right)^T = \frac{1}{2}\begin{bmatrix} 1-2(e_2^2+e_3^2) & 2(e_1e_2+e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & 1-2(e_1^2+e_3^2) & 2(e_2e_3+e_0e_1) \\ 2(e_1e_3+e_0e_2) & 2(e_2e_3+e_0e_1) & 1-2(e_1^2+e_2^2) \end{bmatrix}$$

- Quaternion update matrix:

$$\tilde{\mathbf{\Omega}} = \begin{bmatrix} 0 & \overline{p} & \overline{q} & \overline{r} \\ -\overline{p} & 0 & -\overline{r} & \overline{q} \\ -\overline{q} & \overline{r} & 0 & -\overline{p} \\ -\overline{r} & -\overline{q} & \overline{p} & 0 \end{bmatrix}$$

- Euler solution:

$$\mathbf{p}_{k+1} = \mathbf{p}_{k+1} + \dot{\mathbf{p}}_k \cdot dt$$

$$\mathbf{v}_{k+1} = \mathbf{v}_{k+1} + \dot{\mathbf{v}}_k \cdot dt$$

$$\mathbf{e}_{k+1} = \exp\left(-\tfrac{1}{2}\tilde{\mathbf{\Omega}}\cdot dt\right)\mathbf{e}_k = \left[\mathbf{I}\left(\cos(s) + l\cdot\left(1-\|\mathbf{e}\|^2\right)\cdot dt\right) - \tfrac{1}{2}\tilde{\mathbf{\Omega}}\cdot dt\cdot\frac{\sin(s)}{s}\right]\mathbf{e}_k$$

$$s = \tfrac{1}{2}\sqrt{(\overline{p}_k\cdot dt)^2 + (\overline{q}_k\cdot dt)^2 + (\overline{r}_k\cdot dt)^2}$$

Details: Kinematic Model

- Kinematic process model comments
  - IMU data first corrected using estimated biases.
  - Time-update of state at IMU rate using nonlinear “kinematic” function
    - Accounts for sensor coupling due to geometry (e.g., GPS offset from center of gravity)
    - Quaternion updated with exact nonlinear solution
    - Accounts for coordinate transformation between body and inertial frames.

- Sensor observations (GPS, Altimeter)
  - Nonlinear function of the system state.
  - Nonlinear effect modeled: sensor latency, quantization.
  - Used for discrete measurement update (prediction correction)

Details: Observation Model - GPS

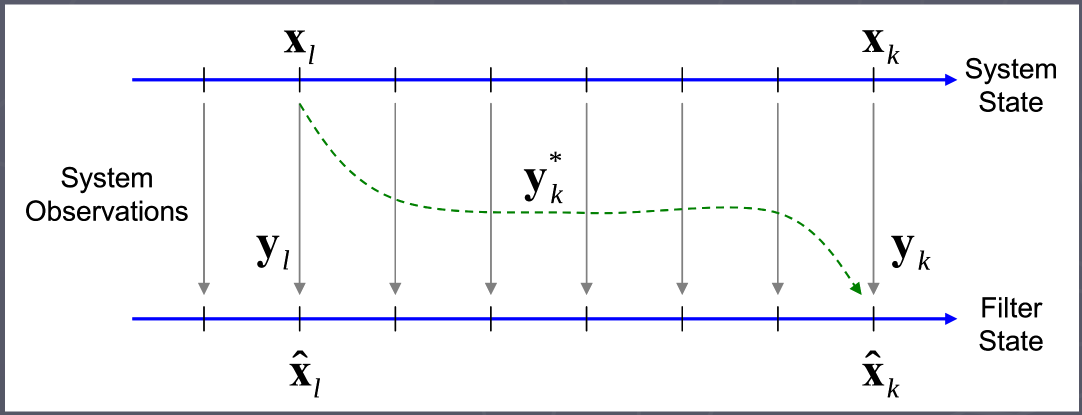
- Measure lagged position and velocity in inertial frame

$$\mathbf{p}_k^* = \mathbf{p}_l + \overset{b\to i}{\underset{l}{\mathbf{T}}} \cdot \mathbf{r}_{GPS} + \mathbf{n}_{\mathbf{p}_l}$$

$$\mathbf{v}_k^* = \mathbf{v}_l + \overset{b\to i}{\underset{l}{\mathbf{T}}} \cdot \bar{\boldsymbol{\omega}}_l \times \mathbf{r}_{GPS} + \mathbf{n}_{\mathbf{v}_l}$$

$$l = k - N_{lat}$$

$$N_{lat} = \{\text{GPS latency}\} / dt$$



- Practical issues that were addressed:
  - antenna not at CG
  - measurement latency → current reading corresponds with vehicle state in the past. Complicates measurement updates.
  - SPKF formulation allows for an elegant way to deal with latencies.

Details: Observation Model - Altimeter

- Barometric measurement of absolute altitude.

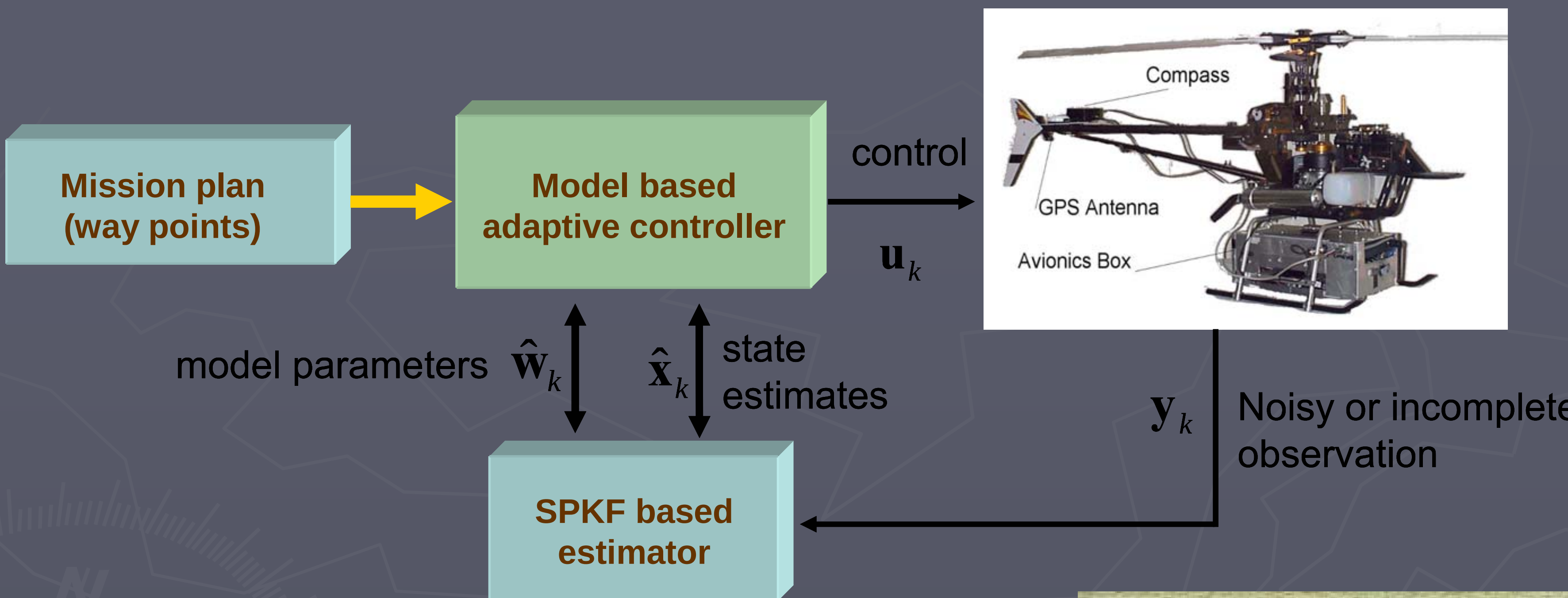
$$z_k^{ALT} = -(1/\gamma)\log\left(\rho_{quant}\mathrm{floor}\left((\rho_0 \exp(\gamma z_k) + n_k^{ALT})/\rho_{quant}\right)/\rho_0\right)$$

γ : atmospheric presure decay rate

ρ_0 : sea level preasure ρ_{quant} : quantization preasure

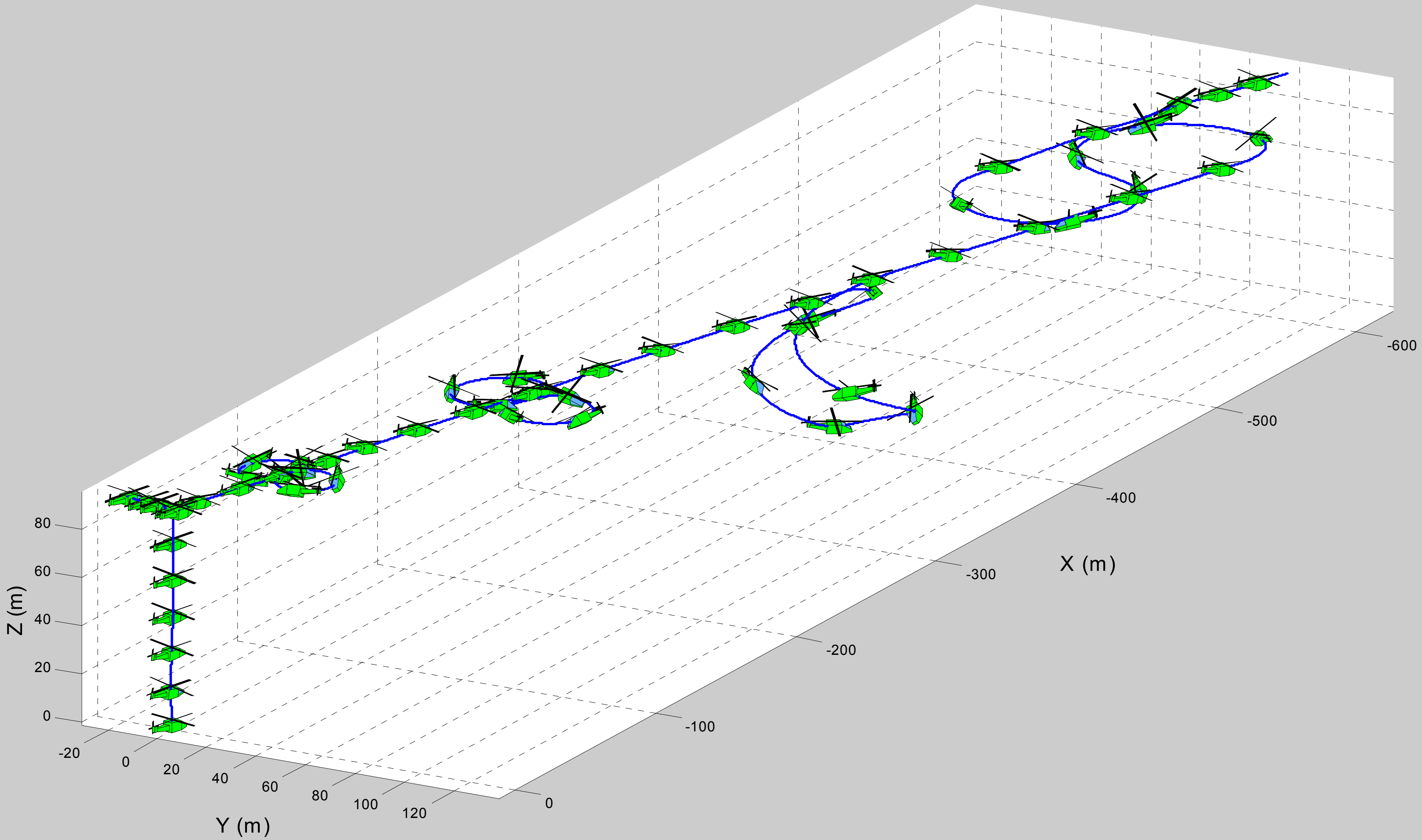
- Issues:
  - Nonlinear due to quantization effects : makes SPKF use attractive
  - Low resolution
  - Inaccurate close to ground due to rotor-downwash
  - Can be affected by atmospheric conditions
  - Ideally combined with other relative altitude sensors such as sonar, radar or lidar.

Test Platform: X-Cell-90 Helicopter

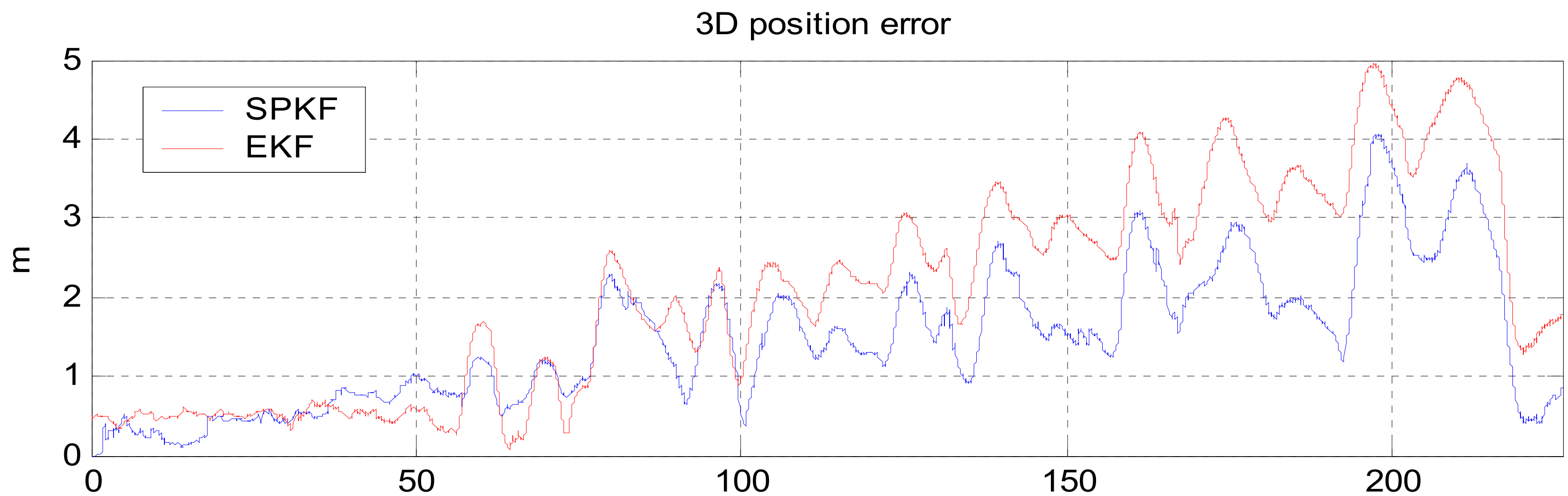


- Complete avionics suite
  - Flight computer (300MHz DSP400), IMU, GPS, Barometric altimeter, Three-Axis Magnetic compass, Wireless Ethernet link, Flash memory, Custom servo board, R/C transmitter, Hardware-in-the loop system for testing.
- MIT-Draper X-Cell-90 Dynamic Model
  - High fidelity quaternion based nonlinear model
  - 26 states and ~70 parameters (rotor forces, torque, and thrust, flapping dynamics, horizontal stabilizer and vertical tail forces and moments, fuselage drag, and actuator states)

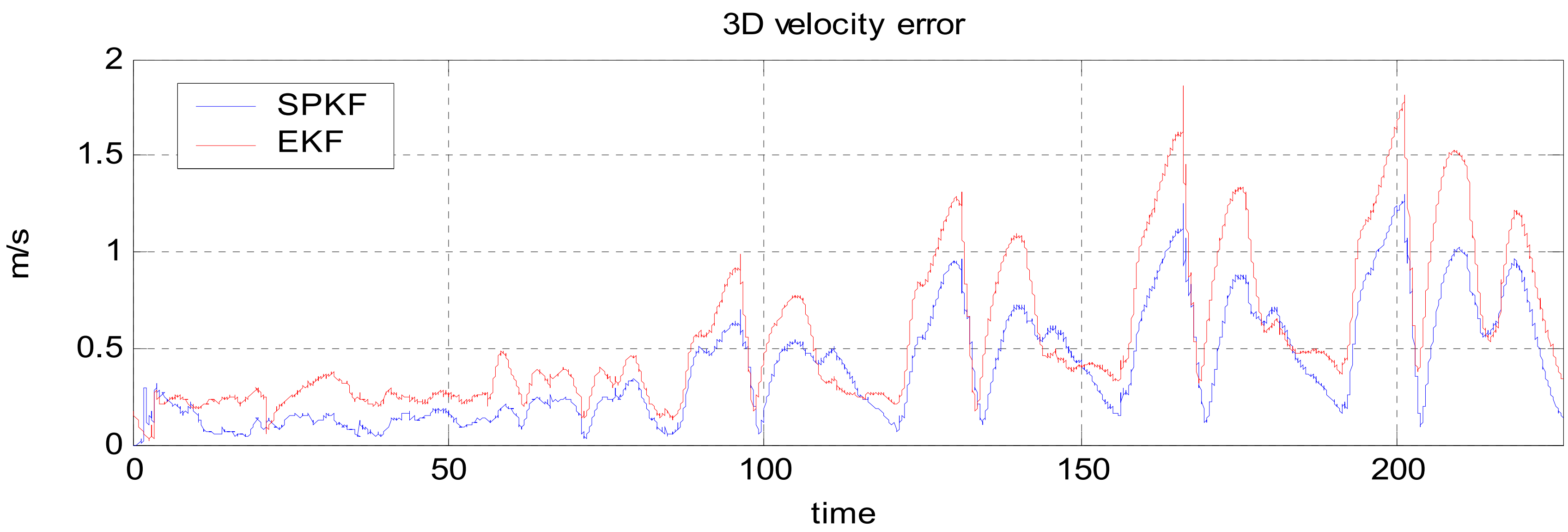
State Estimation Experiment: Test Trajectory



State Estimation Experimental Results (Position & Velocity)



32% improvement



34% improvement

State Estimation Experimental Results (Euler Angles)

34% improvement

32% improvement

65% improvement

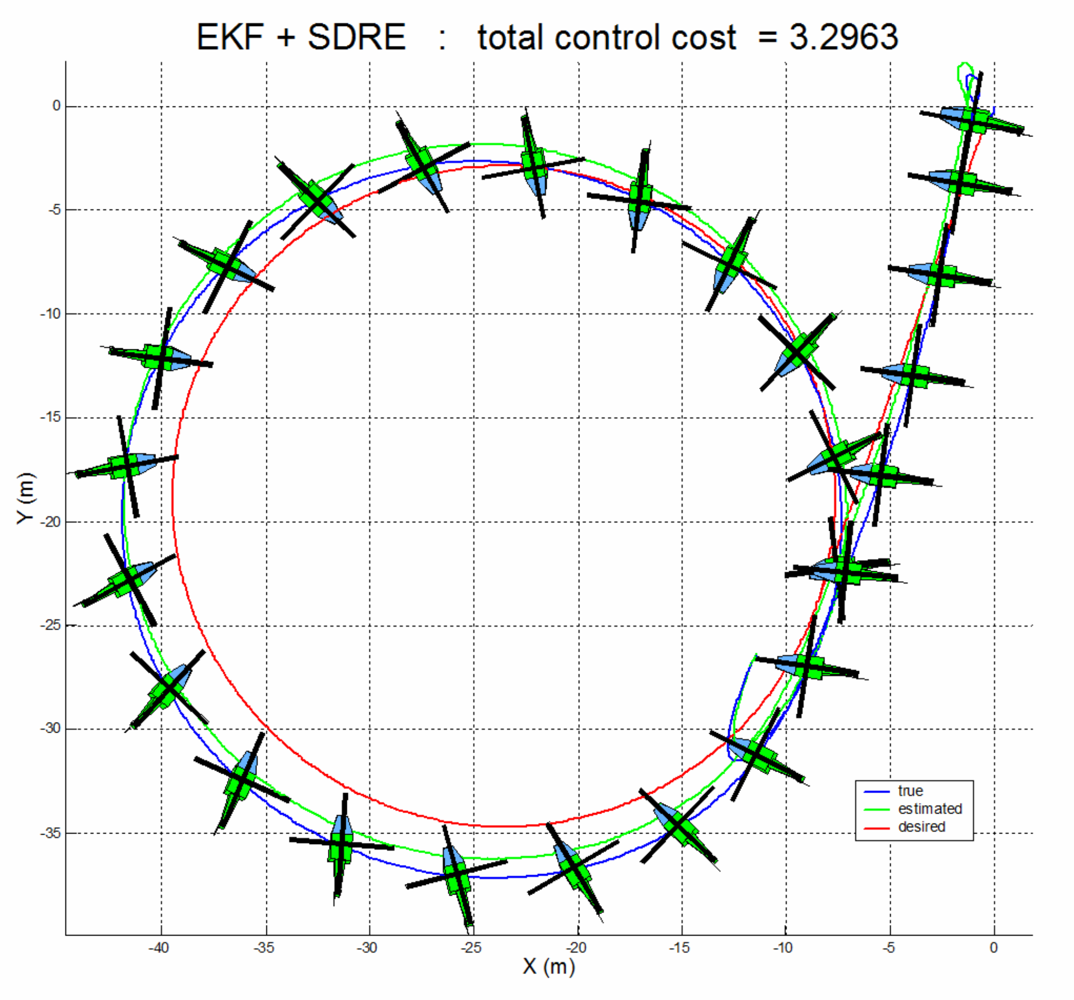
Performance Summary

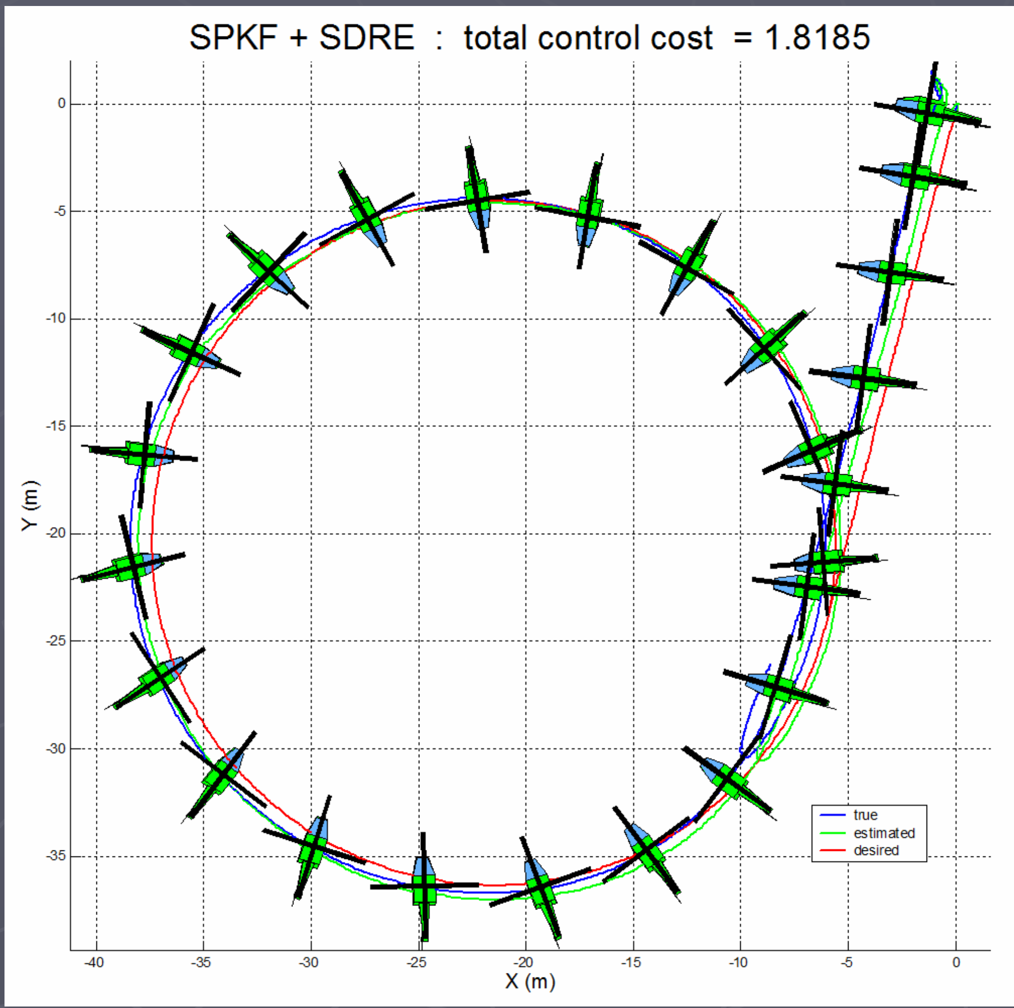
- EKF vs. SPKF with and without GPS Latency Compensation

Algorithm	Average RMS Error				
	position (m)	velocity (m/s)	Euler angles (degrees) roll	pitch	yaw
EKF	2.1	0.57	0.25	0.32	2.29
SPKF - without LC	1.9 (10%)	0.52 (9%)	0.20 (20%)	0.26 (19%)	1.03 (55%)
SPKF - with LC	1.4 (32%)	0.38 (34%)	0.17 (32%)	0.21 (34%)	0.80 (65%)

Closed Loop State Estimation & Control

- Use of SPKF estimator reduces control cost by 45%

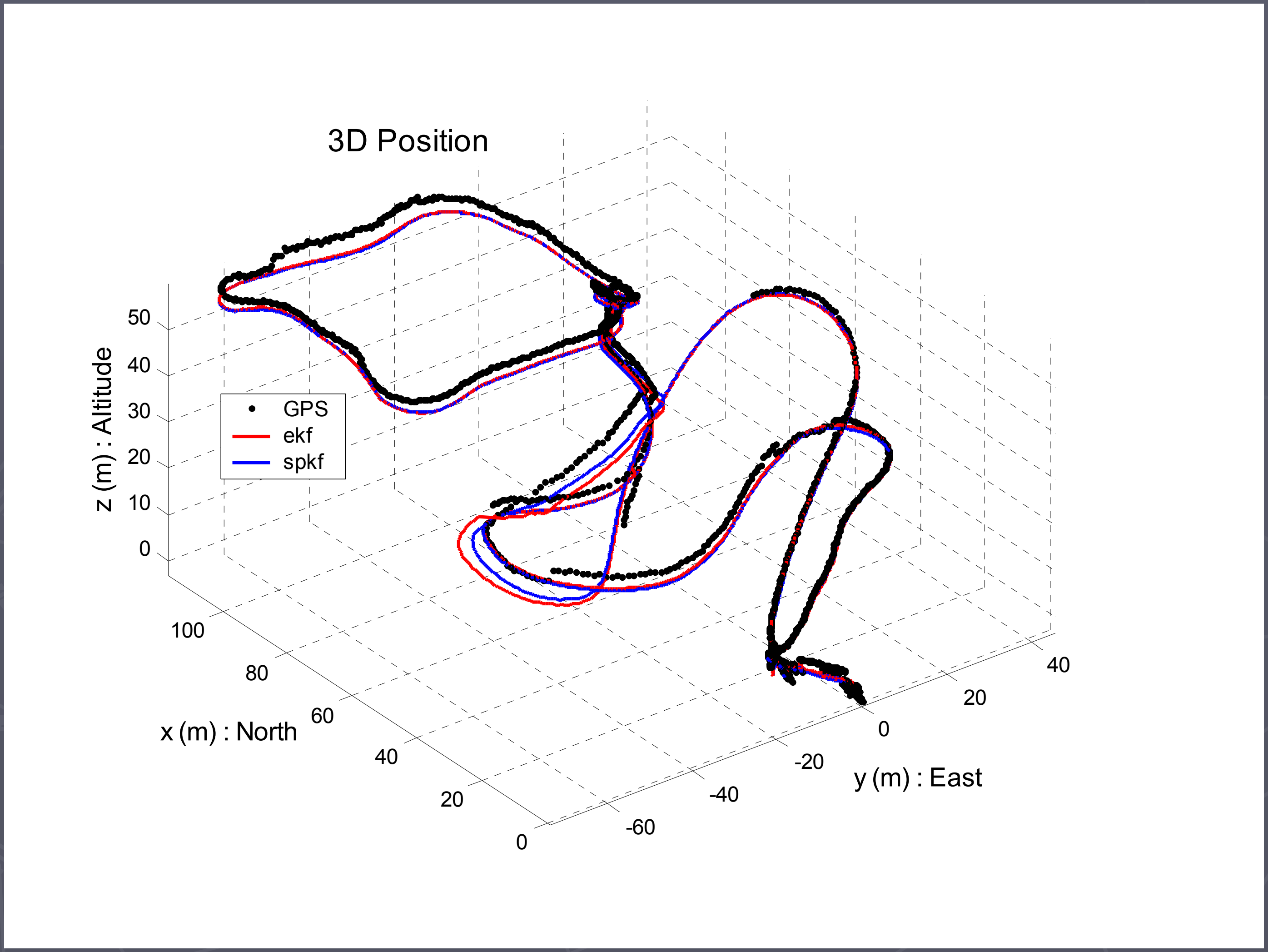




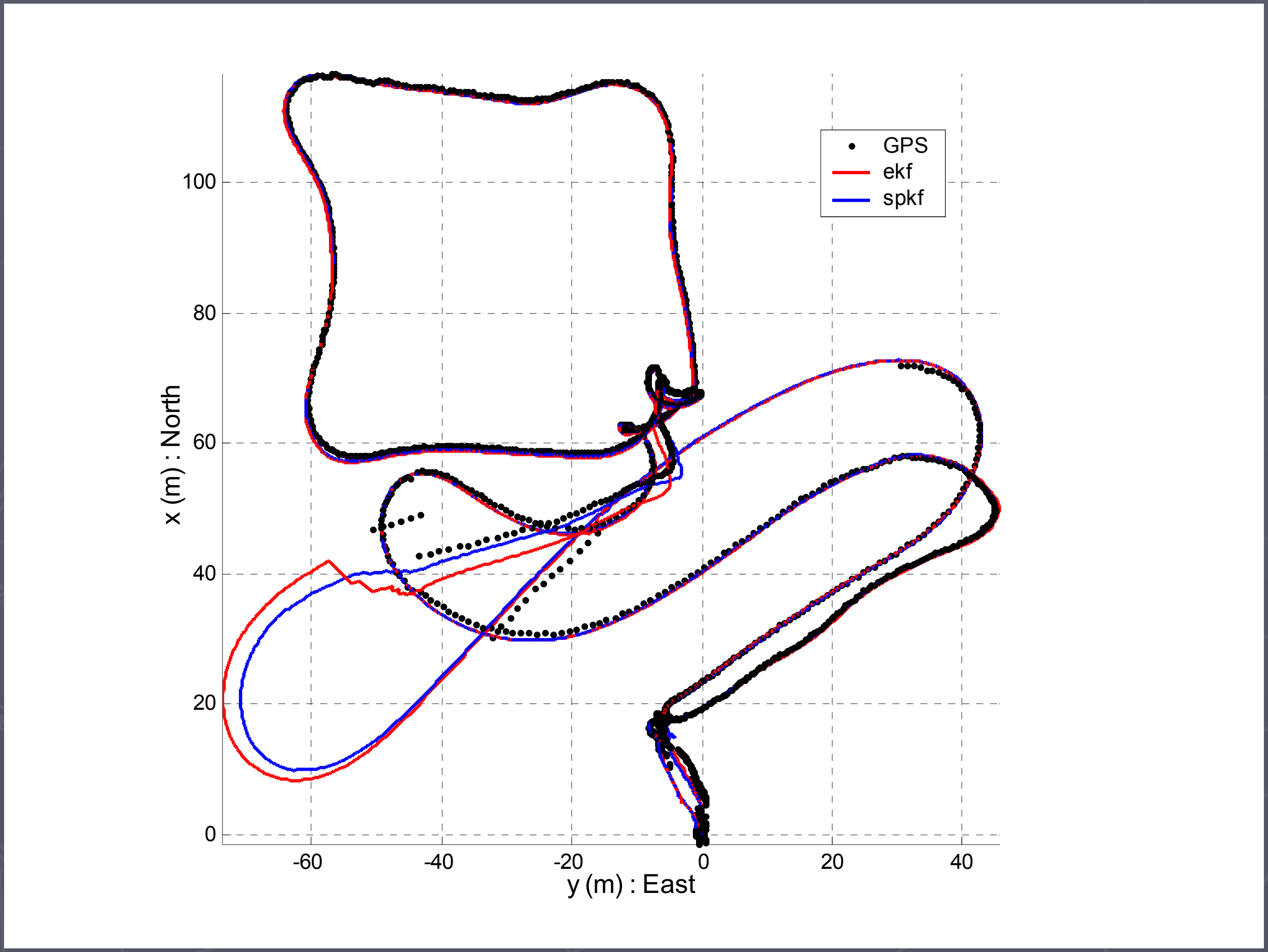
EKF
Control cost : 3.30

SPKF
Control cost : 1.82

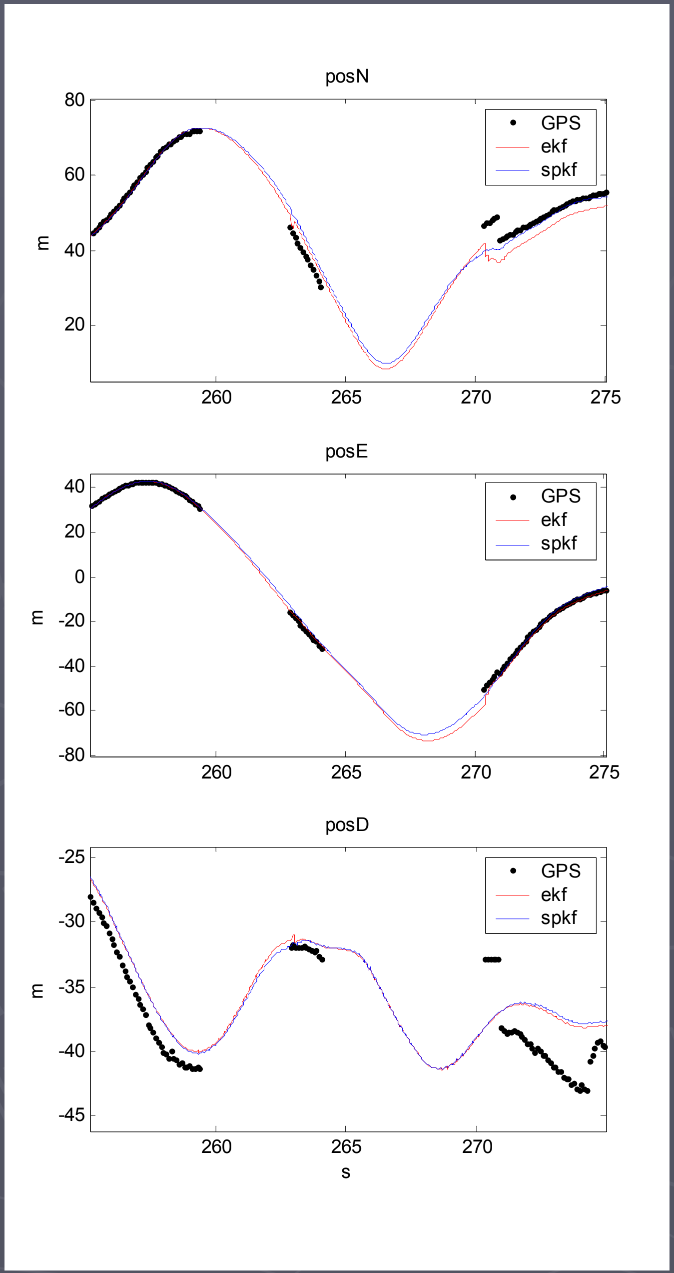
Results on Actual Flight Data

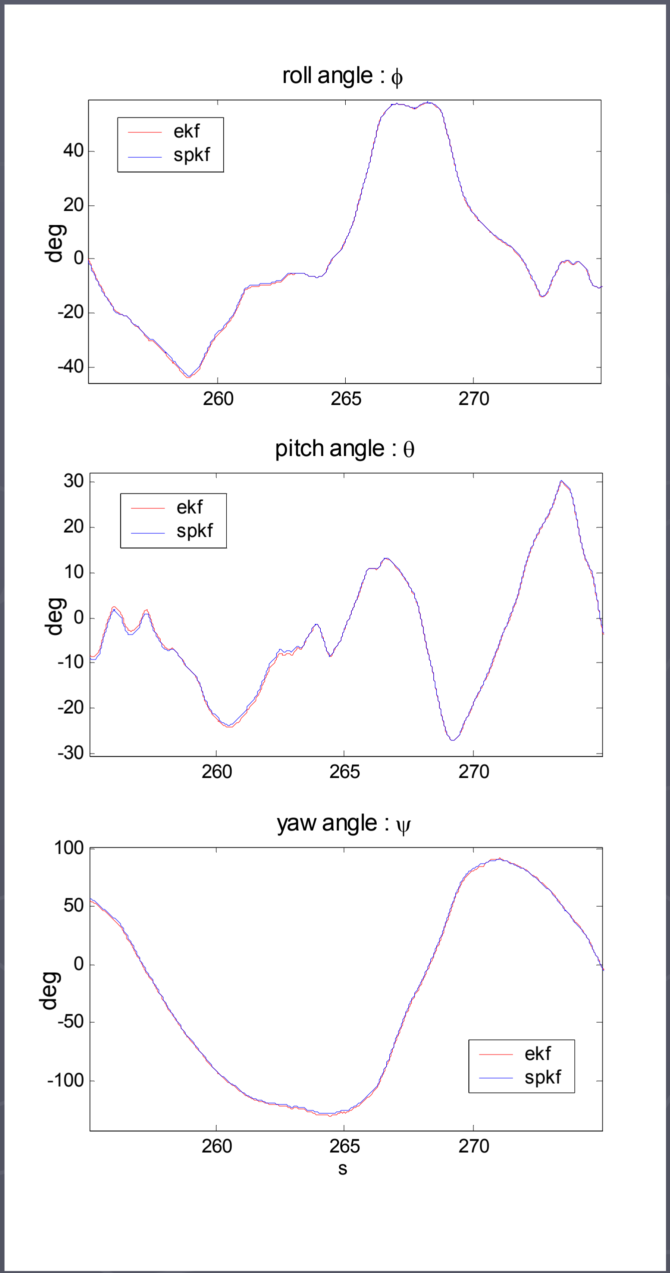


Results on Actual Flight Data

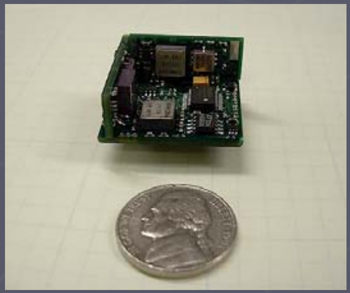


Results on Actual Flight Data



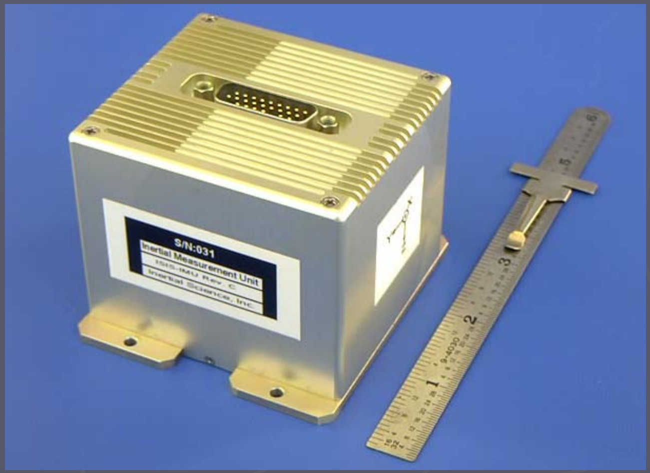


SPKF Cost Savings

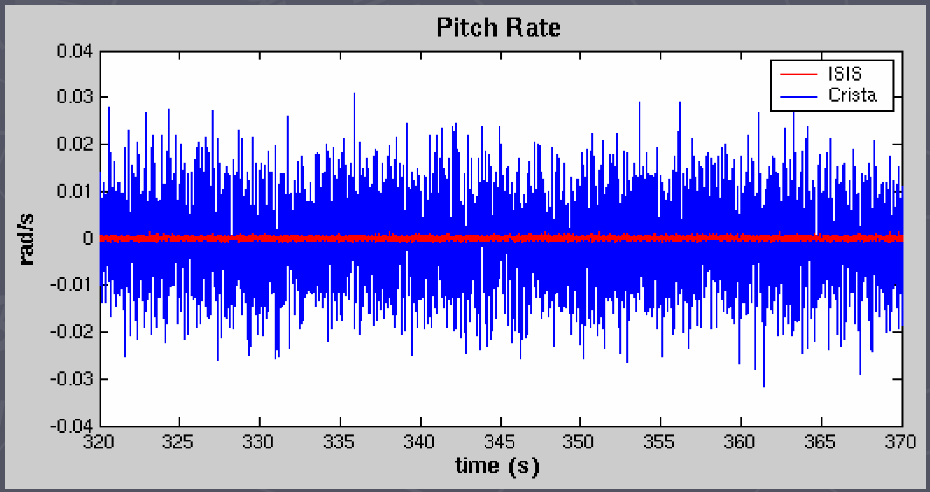


Cloud Cap Crista IMU
.35" x 1.50" x 0.80, 19 grams
< $2k

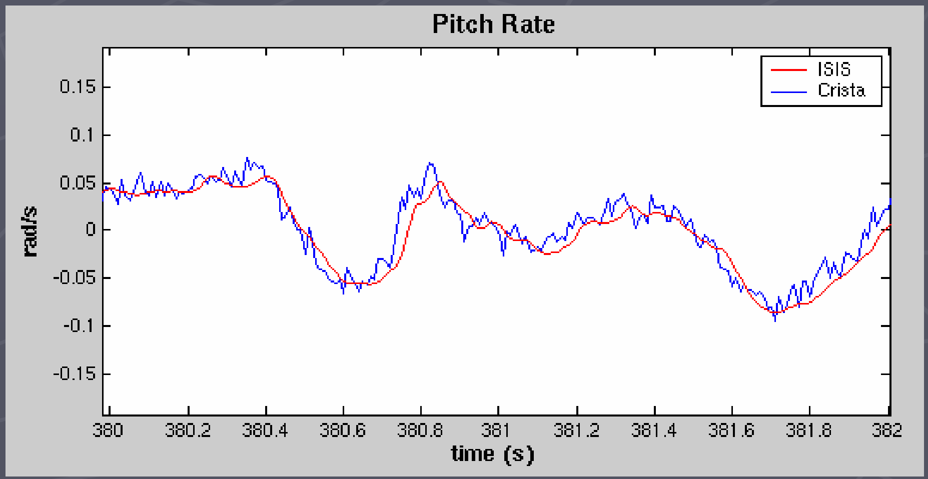
?
=



Inertial Sciences ISIS-IMU
3.30" x 2.5 " x 1.83 , 250 grams
> $10k



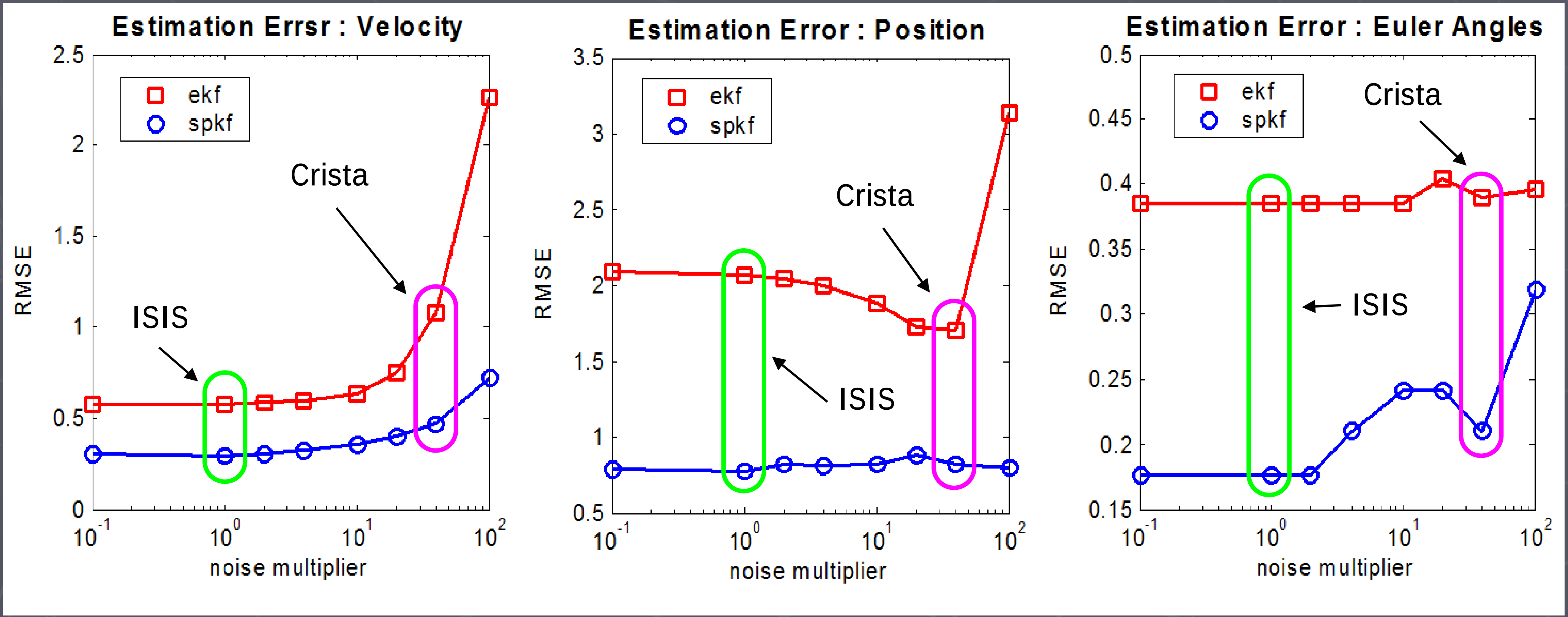
Noise floor std: (gyros=30x, accel=10x)



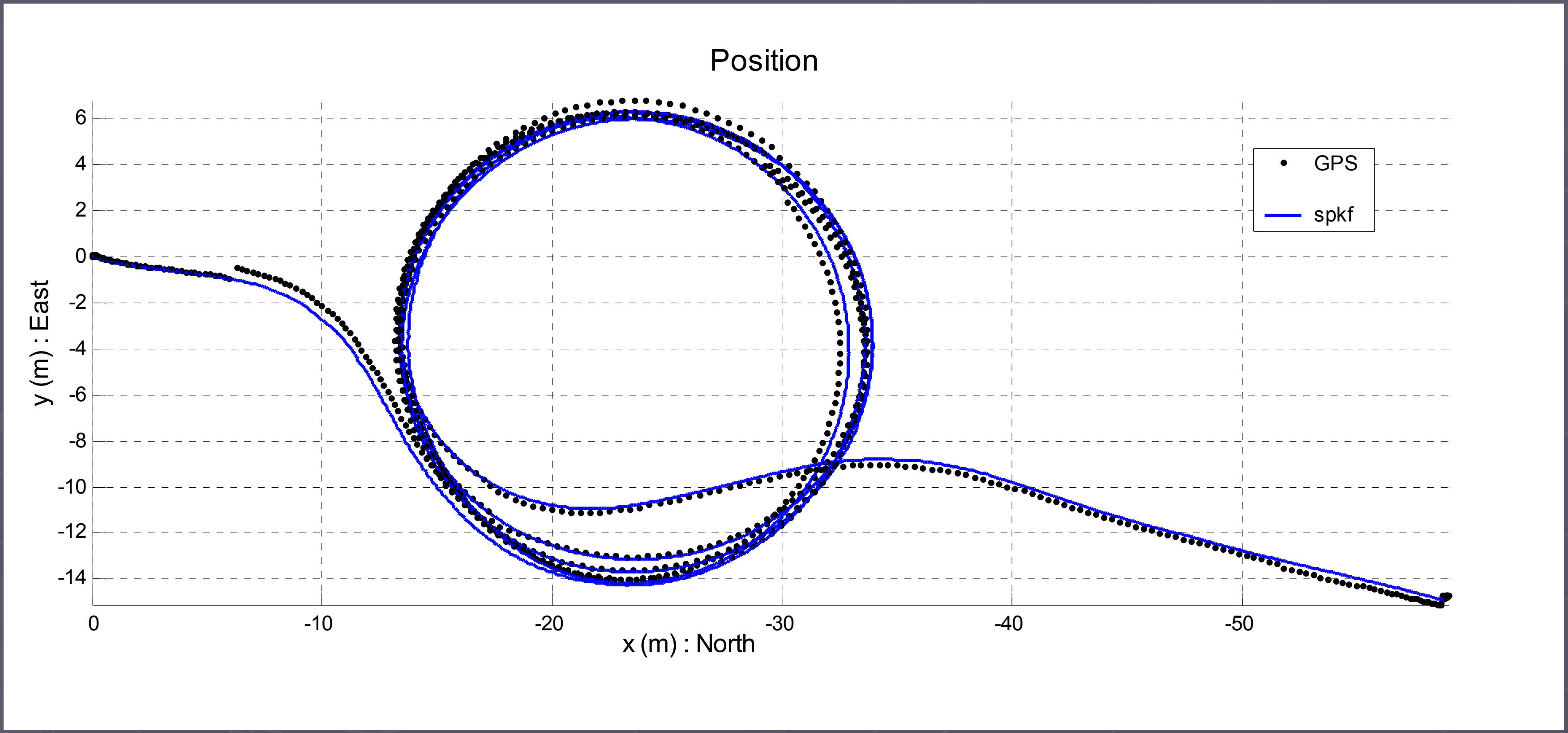
Signal Dynamics

ISIS vs. Crista IMU: Estimation Performance

- Simulation Results (worst case : 40x difference in noise level std.)

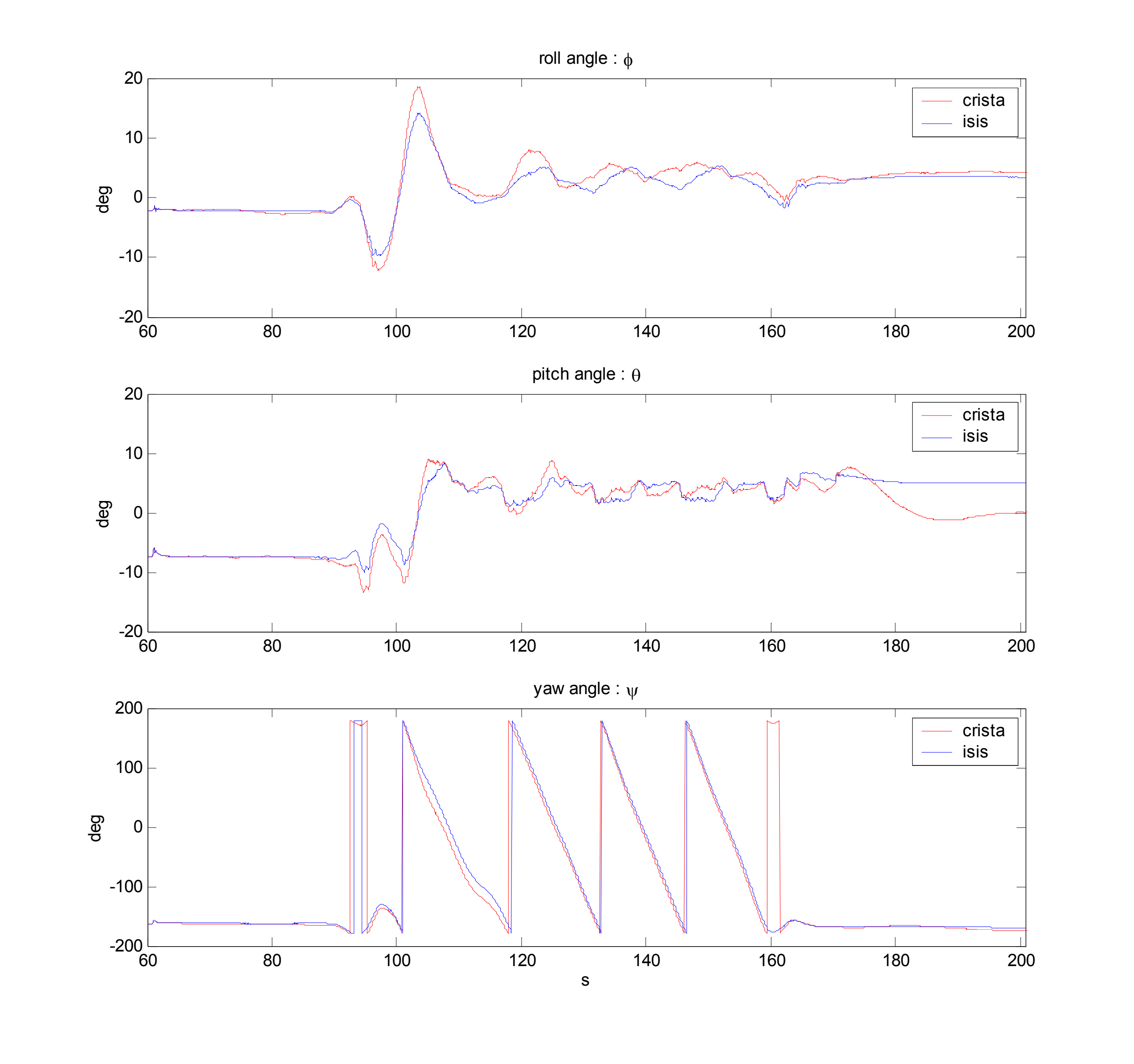


ISIS vs. Crista – Real Data Experiments



2-d position (driving in a car)

ISIS vs. Crista – Real Data Experiments



Attitude estimates

Summary

- Improved integrated navigation solution.

- Superior performance over traditional EKF based systems
  - More accurately accounts for nonlinearities.
  - Exact modeling of asynchronous and lagged sensor updates.
  - Computational load is equivalent to EKF.

- Vehicle Independent: Easily ported to other UAVs, UGVs, UUVs, etc.

- Cost Savings: Can use less expensive/less accurate sensors.

Extensions/Enhancements

- Equation Error Formulation
- IMU Noise Modeling
- Adaptive noise covariance estimation
- System initialization and calibration
- Other Sensors:
  - Compass, sonar, radar, laser, lidar, video, etc.
- Tightly-coupled GPS integration
  - Integrate individual satellite signals using SPKF.
- Distributed and Multiple IMU/GPS integration

Extras

SPKF Based Latency Compensation

- Combine current prediction of state with lagged innovation

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \boldsymbol{\kappa}_k \tilde{\mathbf{y}}_{k-n} \qquad \boldsymbol{\kappa}_k = \mathbf{P}_{\mathbf{x}_k \tilde{\mathbf{y}}_{k-n}} \left(\mathbf{P}_{\tilde{\mathbf{y}}_{k-n}}\right)^{-1} \qquad n = N_{lat}$$

- Key insight: Maintain lagged state and covariance:

$$\hat{\mathbf{x}}_{k-n}^- \qquad \mathbf{P}_{\mathbf{x}_k \mathbf{x}_{k-n}}^- = E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right)\left(\mathbf{x}_{k-n} - \hat{\mathbf{x}}_{k-n}^-\right)^T\right]$$

  - ***Sigma-point approach*** implicitly calculates: $\mathbf{P}_{\mathbf{x}_k \tilde{\mathbf{y}}_{k-n}}$

$$\left.\begin{matrix} \hat{\mathbf{x}}_{k-n}^- \\ \mathbf{P}_{\mathbf{x}_k \mathbf{x}_{k-n}}^- \end{matrix}\right\} \rightarrow \boxed{\text{sigma-points: } \chi_i} \rightarrow \boxed{\mathbf{h}(\bullet)} \rightarrow \boxed{\text{SPKF calculated covariance}} \rightarrow \mathbf{P}_{\mathbf{x}_k \tilde{\mathbf{y}}_{k-n}}$$

SPKF Based Latency Compensation

- Augment state and redefine process & observation model

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_l \end{bmatrix} \qquad \mathbf{x}_{k+1}^a = \begin{bmatrix} \mathbf{f}\left(\mathbf{x}_k\right) \\ \mathbf{x}_l \end{bmatrix} \qquad \mathbf{y}_k = \breve{\mathbf{h}}\left(\mathbf{x}_k^a\right) = \begin{cases} \mathbf{h}_1\left(\mathbf{x}_k\right) & k \neq l + N_{lat} \\ \mathbf{h}_2\left(\mathbf{x}_l\right) & k \neq l + N_{lat} \end{cases}$$

- When lagged sensor measurement arrives at $k = l + N_{lat}$
  - Result of time-update:

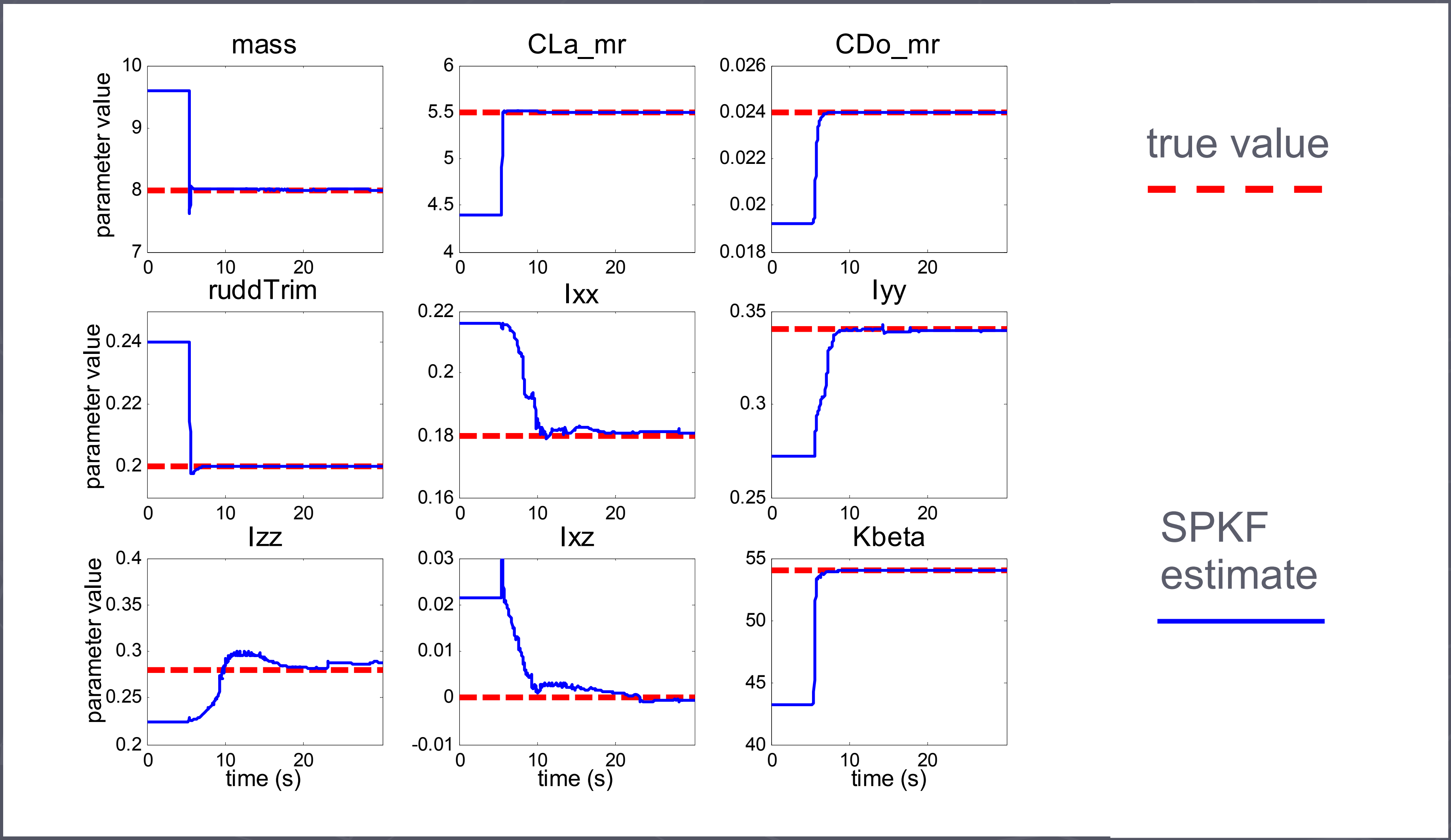
$$\hat{\mathbf{x}}_k^{a-} = \begin{bmatrix} \hat{\mathbf{x}}_k^- \\ \hat{\mathbf{x}}_l^* \end{bmatrix} \qquad \mathbf{P}_{\mathbf{x}_k^a}^- = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_k\mathbf{x}_k}^- & \mathbf{P}_{\mathbf{x}_k\mathbf{x}_l}^- \\ \mathbf{P}_{\mathbf{x}_l\mathbf{x}_k}^- & \mathbf{P}_{\mathbf{x}_l\mathbf{x}_l}^- \end{bmatrix} \qquad l = k - N_{lat}$$

  - Measurement update (Kalman gain):

$$\mathbf{P}_{\mathbf{x}_k^a\tilde{\mathbf{y}}_l} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_k\tilde{\mathbf{y}}_l} \\ \mathbf{P}_{\mathbf{x}_l\tilde{\mathbf{y}}_l} \end{bmatrix} \qquad \boldsymbol{\kappa}_k = \mathbf{P}_{\mathbf{x}_k^a\tilde{\mathbf{y}}_l}\left(\mathbf{P}_{\tilde{\mathbf{y}}_l}\right)^{-1} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_k\tilde{\mathbf{y}}_l}\mathbf{P}_{\tilde{\mathbf{y}}_l}^{-1} \\ \mathbf{P}_{\mathbf{x}_l\tilde{\mathbf{y}}_l}\mathbf{P}_{\tilde{\mathbf{y}}_l}^{-1} \end{bmatrix}$$

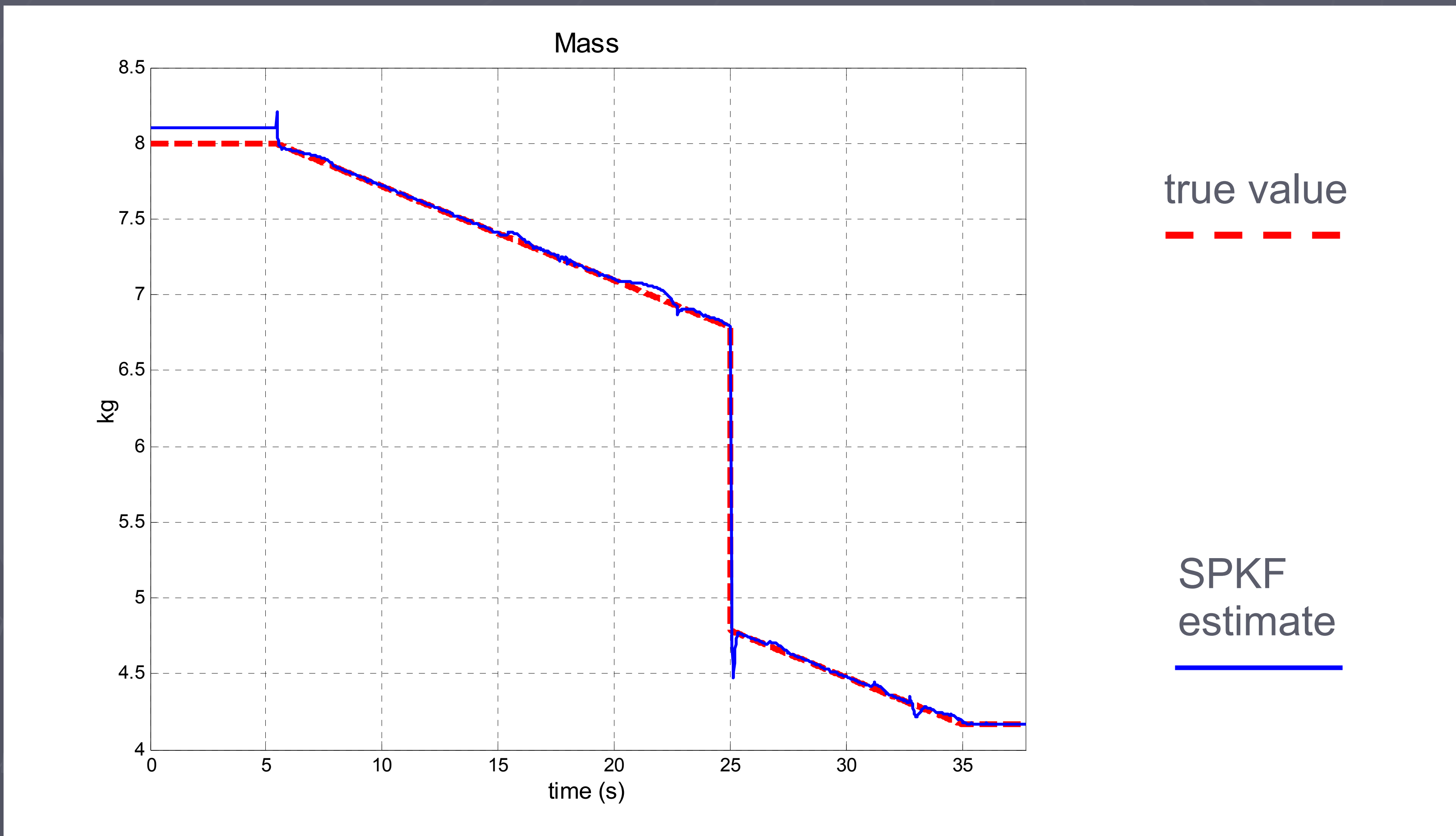
Experimental Results

- Parameter Estimation



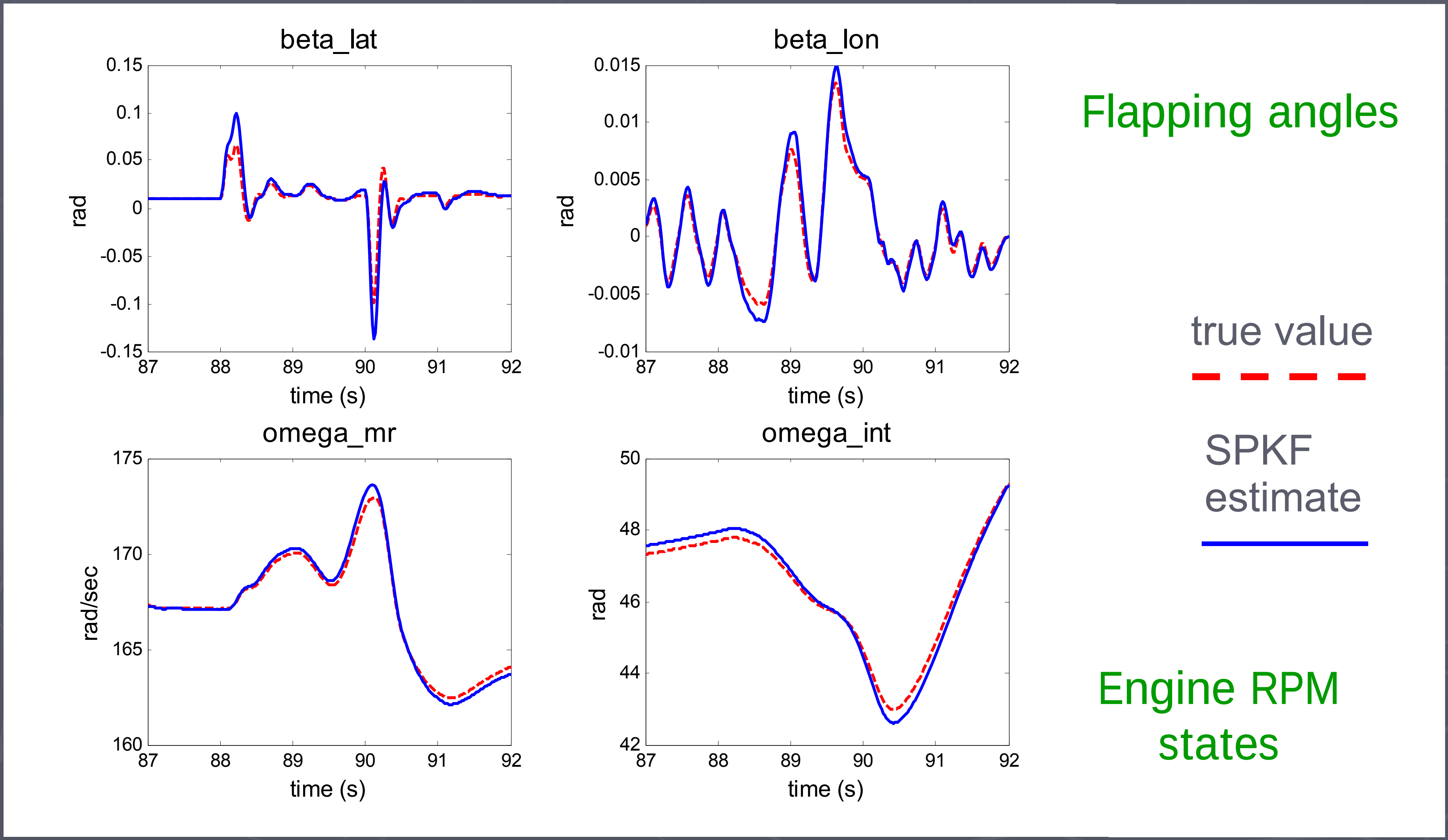
Experimental Results

- Parameter Estimation



Experimental Results

- Dual Estimation: Auxiliary States & Parameters



Experimental Results

- Dual Estimation: Dynamic Mass Tracking