Factor Graph Based Incremental Smoothing in Inertial Navigation Systems

Vadim Indelman[1], Stephen Williams[1], Michael Kaess[2] and Frank Dellaert[1]

1) College of Computing, Georgia Institute of Technology

2) CSAIL, Massachusetts Institute of Technology (MIT)

July 2012

Introduction

- Modern navigation systems rely on different sensors:
  - IMU, GPS, Vision, step sensor, etc.

Big Dog [Boston Dynamics]

AR Drone [Parrot]

Sting [Georgia Tech]

Introduction

- Modern navigation systems rely on different sensors
  - IMU, GPS, Vision, step sensor, etc.
- These sensors can potentially be asynchronous and operating at multiple frequencies

- Common approach for information fusion in navigation systems: extended Kalman filter (EKF)
- Incorporating measurements from different sources: typically involves maintaining an augmented state vector
  - The whole augmented state vector is updated each time
    - **Expensive**!
    - In practice, only part of the variables are affected
  - Handling delayed measurements is not trivial [Zhang and Bar-Shalom, 2011]

Introduction (Cont.)

In this work: An adaptive fixed-lag smoother is proposed

- A non-linear optimization over **<u>all</u>** states (current and past) using all the available measurements
  - Maximum a posteriori (MAP) estimate
  - Often referred to as full SLAM and bundle adjustment in robotics
- Efficient incremental optimization is possible using a factor graph formulation:
  - Exploit sparsity
  - **Only part of the variables are updated** – variables that are expected to benefit from the new measurement
- Based on incremental smoothing technique developed in SLAM community:
  - [Dellaert and Kaess, 2006], [Kaess, et al., 2012]

Related Work

- Bundle Adjustment (BA) [Thrun, 2005]
  - Commonly used in robotics to solve the full SLAM problem
  - Real time?

- BA was recently suggested for information fusion in inertial navigation systems:
  - [Mourikis and Roumeliotis 2008]:
    - Augmented-state EKF for incorporating IMU and vision measurements
    - Batch BA for loop closures
  - [Bryson, et al. 2009]:
    - Batch non-linear optimization formulation for fusing IMU, GPS and visual measurements
    - Designed for off-line terrain reconstruction

- Incremental Smoothing and Mapping [Dellaert and Kaess, 2006], [Kaess, et al., 2012]
  - Real time - using factor graph, Bayes net and Bayes tree representations

Factor Graph Formulation

- The maximum a posteriori (MAP) estimate is given by

$$\hat{\mathcal{X}} = \arg\max_{\mathcal{X}} \left(p(\mathcal{X}) \right)$$

  - $\mathcal{X}$: all the navigation states over time
  - $p(\mathcal{X})$: joint probability given all measurements up to current time

- $p(\mathcal{X})$ can be explicitly written in terms of individual probabilities representing process and measurement models
  - For example:

$$p(\mathcal{X}) = p(x_0) \prod_j p(x_j | x_{j-1}) \prod_k p\left(z_k | x_{j_k}\right)$$

- Factor graph formulation

$$p(\mathcal{X}) \propto \prod_i f_i\left(\mathcal{X}_i\right)$$

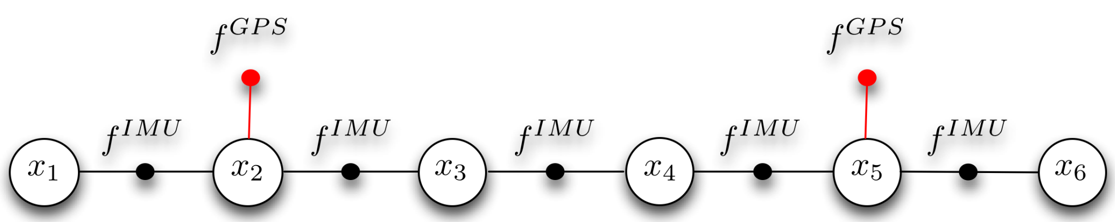
  - $\mathcal{X}_i$ is a subset of states related by the *i*th measurement\process model

Factor Graph Formulation (Cont.)

$$p(\mathcal{X}) \propto \prod_i f_i(\mathcal{X}_i)$$

- Factor graph $G = (\mathcal{F}, \mathcal{X}, \mathcal{E})$
  - Two type of nodes:
    - Variable nodes $x_j \in \mathcal{X}_i \subset \mathcal{X}$ are associated with system states
    - Factor nodes $f_i \in \mathcal{F}$ are associated with measurements
  - Edges always connect between variable and factor nodes

- For example:
  - A small factor graph with IMU and GPS measurements and basic navigation states



Factor Graph Formulation (Cont.)

$$p(\mathcal{X}) \propto \prod_i f_i\left(\mathcal{X}_i\right)$$

- Factor graph $G = (\mathcal{F}, \mathcal{X}, \mathcal{E})$
  - Two type of nodes:
    - Variable nodes $x_j \in \mathcal{X}_i \subset \mathcal{X}$ are associated with system states
    - Factor nodes $f_i \in \mathcal{F}$ are associated with measurements
  - Edges always connect between variable and factor nodes

- Assuming a Gaussian distribution, MAP estimate corresponds to a non-linear least-squares optimization
  - For example: $z_i = h_i(\mathcal{X}_i) + n$ $\qquad f_i\left(\mathcal{X}_i\right) \doteq \exp\left(\left\|h_i\left(\mathcal{X}_i\right) - z_i\right\|_\Sigma^2\right)$

$$\arg\max_{\mathcal{X}} \prod_i f_i(\mathcal{X}_i) = \arg\min_{\mathcal{X}} J\left(\mathcal{X}\right)$$

    - with the cost function:

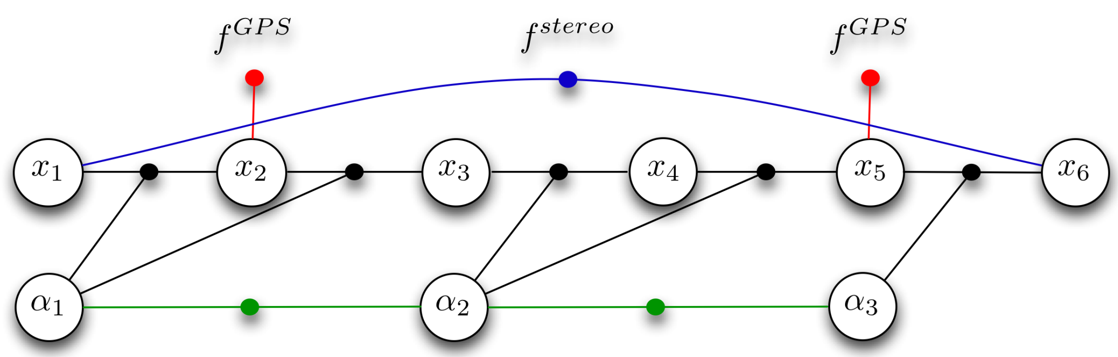
$$J\left(\mathcal{X}\right) \doteq \sum_i \left\|h_i\left(\mathcal{X}_i\right) - z_i\right\|_\Sigma^2$$

Factor Graph Formulation (Cont.)

- Factor graph framework
  - Allows handling different possibly asynchronous sensors at varying frequencies
  - Provides plug and play capability:
    - New sensors are additional sources of factors that get added to the graph
    - If a sensor becomes unavailable: do not add any factors from this sensor
      - No special procedure or coordination is required

Basic navigation states:

IMU errors parameterization:
(to be discussed next)



Inertial Navigation - Factor Graph Formulation

- Inertial navigation process model: $x_{k+1} = h\left(x_k, \alpha_k, z_k\right)$
  - x_k: navigation state at time t_k
  - $z_k \doteq \left[\begin{array}{cc} a_m^T & \omega_m^T \end{array}\right]^T$: IMU measurements (acc and gyro)
  - α_k : calculated model of IMU errors - used for correcting IMU measurements

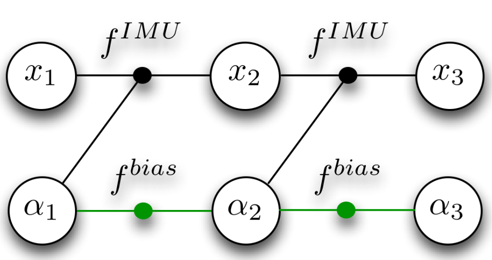
    In this work - we will refer to α_k as "bias" vector (can be general model in practice)

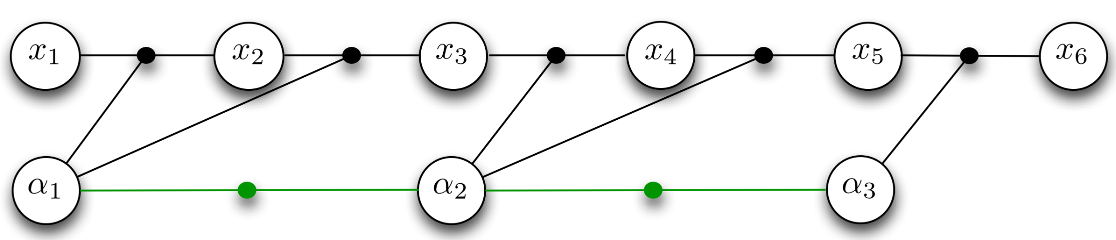
- Time propagation of α_k: $\alpha_{k+1} = g\left(\alpha_k\right)$

- Factor formulations:

$$f^{IMU}\left(x_{k+1}, x_k, \alpha_k\right) \doteq \exp\left(\left\|x_{k+1} - h\left(x_k, \alpha_k, z_k\right)\right\|^2_{\Sigma_x}\right)$$

$$f^{bias}\left(\alpha_{k+1}, \alpha_k\right) \doteq \exp\left(\left\|\alpha_{k+1} - g\left(\alpha_k\right)\right\|^2_{\Sigma_\alpha}\right)$$



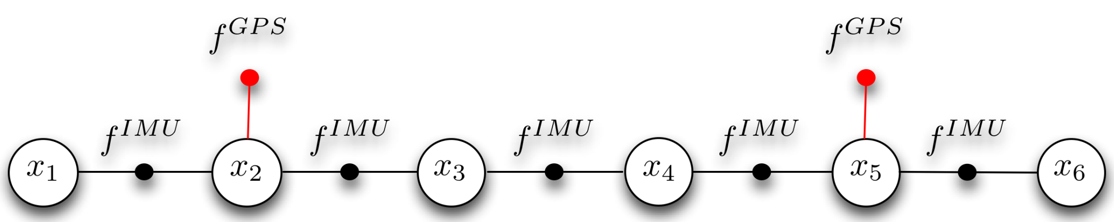


Factor Graph Formulation for Additional Sensors

- GPS:
  - Can be treated as unary factor
  - Time delayed-measurements are easily accommodated

$$z_k^{GPS} = h^{GPS}(x_l) + n_{GPS} \qquad\qquad t_k > t_l$$

$$\Rightarrow \quad f^{GPS}(x_l) \doteq \exp\left(\left\| z_k^{GPS} - h^{GPS}(x_l) \right\|^2_{\Sigma_{GPS}} \right)$$

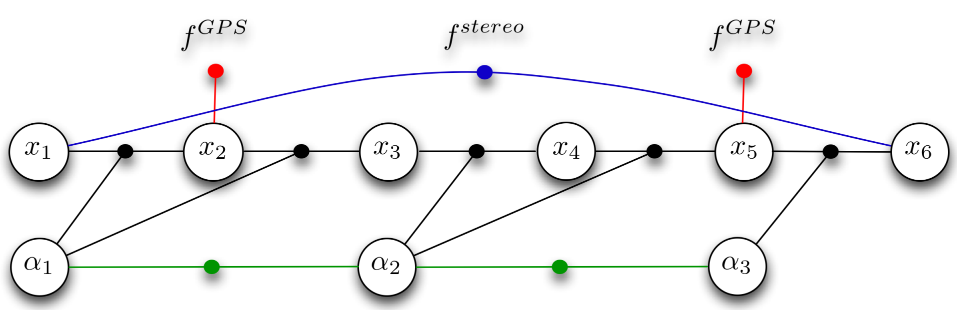


Factor Graph Formulation for Additional Sensors (Cont.)

- Monocular camera measurements
  - Assuming known landmarks and camera calibration – define unary factors
  - Unknown landmarks (SLAM):
    - Landmarks are added as variable nodes to the factor graph
    - Binary factor connecting between appropriate navigation and landmark nodes

- Stereo vision measurements
  - The relative transformation T_Δ between two stereo frames T_{k_1}, T_{k_2} can be estimated (assuming a known baseline)
  - Binary factor:

$$f^{stereo}\left(x_{k_1}, x_{k_2}\right) \doteq \exp\left(\|T_\Delta - (T_{k_1} - T_{k_2})\|^2_{\Sigma_{T_\Delta}}\right)$$



Incremental Batch Optimization

- Goal: $\hat{\mathcal{X}} = \arg\min_{\mathcal{X}} J(\mathcal{X})$
- The optimization involves repeated linearization within a standard non-linear optimizer
- Assuming some linearization point $\mathcal{X}_0$, look for an update $\boldsymbol{\Delta}$ such that:

$$\arg\min_{\boldsymbol{\Delta}} \left(\| A(\mathcal{X}_0)\boldsymbol{\Delta} - \mathbf{b}(\mathcal{X}_0) \|_{\Sigma}^2 \right)$$

  – $A(\mathcal{X}_0)$: (sparse) Jacobian matrix

  – $\mathbf{b}(\mathcal{X}_0)$: right-hand-side (rhs, residual)

- The linearization point is then updated ($\mathcal{X}_0 + \boldsymbol{\Delta}$)

Can we do an efficient incremental optimization?

Inference Using Factor Graphs

$$\arg\min_{\mathbf{\Delta}} \left(\| A(\mathcal{X}_0)\mathbf{\Delta} - \mathbf{b}(\mathcal{X}_0) \|^2_{\Sigma} \right)$$

- Solving for $\mathbf{\Delta}$ typically requires factoring the Jacobian A into a triangular form (e.g., QR)
  - For example:

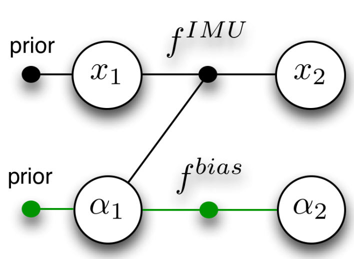
Jacobian matrix

$$A = \begin{bmatrix} \times & & & \\ & \times & & \\ \times & \times & \times & \\ & \times & & \times \end{bmatrix} \quad \begin{matrix} x_1 & \alpha_1 & x_2 & \alpha_2 \end{matrix}$$

Factorized Jacobian matrix

$$R = \begin{bmatrix} \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \quad \begin{matrix} x_1 & \alpha_1 & x_2 & \alpha_2 \end{matrix}$$

Factor graph



?

Inference Using Factor Graphs (Cont.)

Jacobian matrix

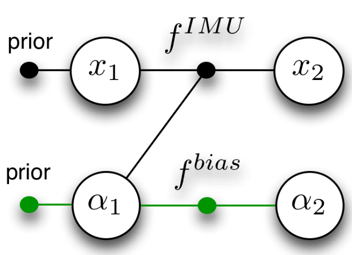
$$A = \begin{bmatrix} \times & & & \\ & \times & & \\ \times & \times & \times & \\ & \times & & \times \end{bmatrix}$$
$$\quad x_1 \quad \alpha_1 \quad x_2 \quad \alpha_2$$

Factorized Jacobian matrix

$$R = \begin{bmatrix} \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix}$$
$$\quad x_1 \quad \alpha_1 \quad x_2 \quad \alpha_2$$

- This is equivalent to converting the factor graph into a Bayes net:
  - A variable ordering is selected (e.g. $x_1, \alpha_1, x_2, \alpha_2$)
  - Each node in the factor graph is eliminated from the graph, forming a node in a Bayes net
  - The Bayes net is equivalent to the matrix R
    - Used to obtain the update by back-substitution
  - Elimination order affects the structure of the Bayes net and the corresponding amount of computation

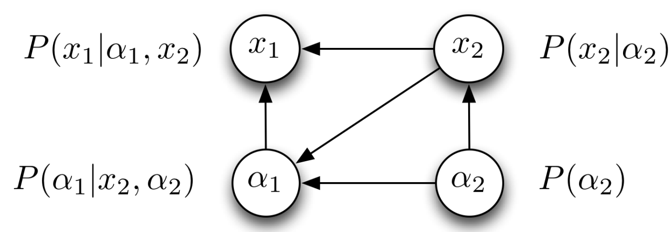
Factor graph



Elimination order

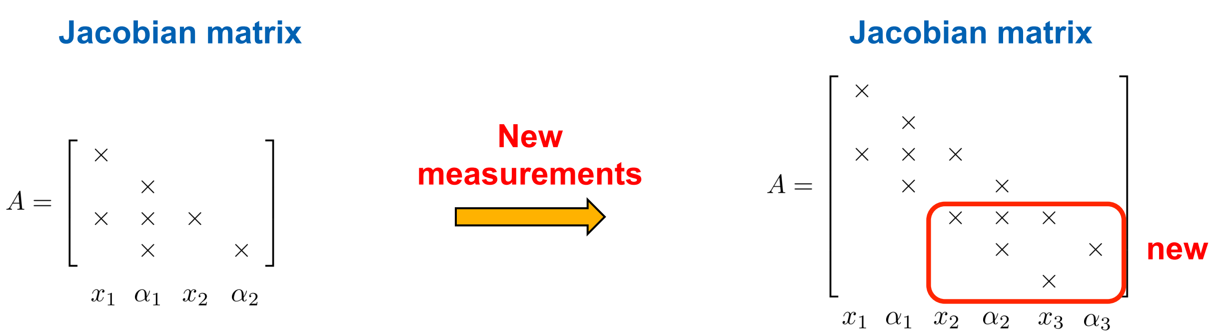
$x_1, \alpha_1, x_2, \alpha_2$

Bayes net



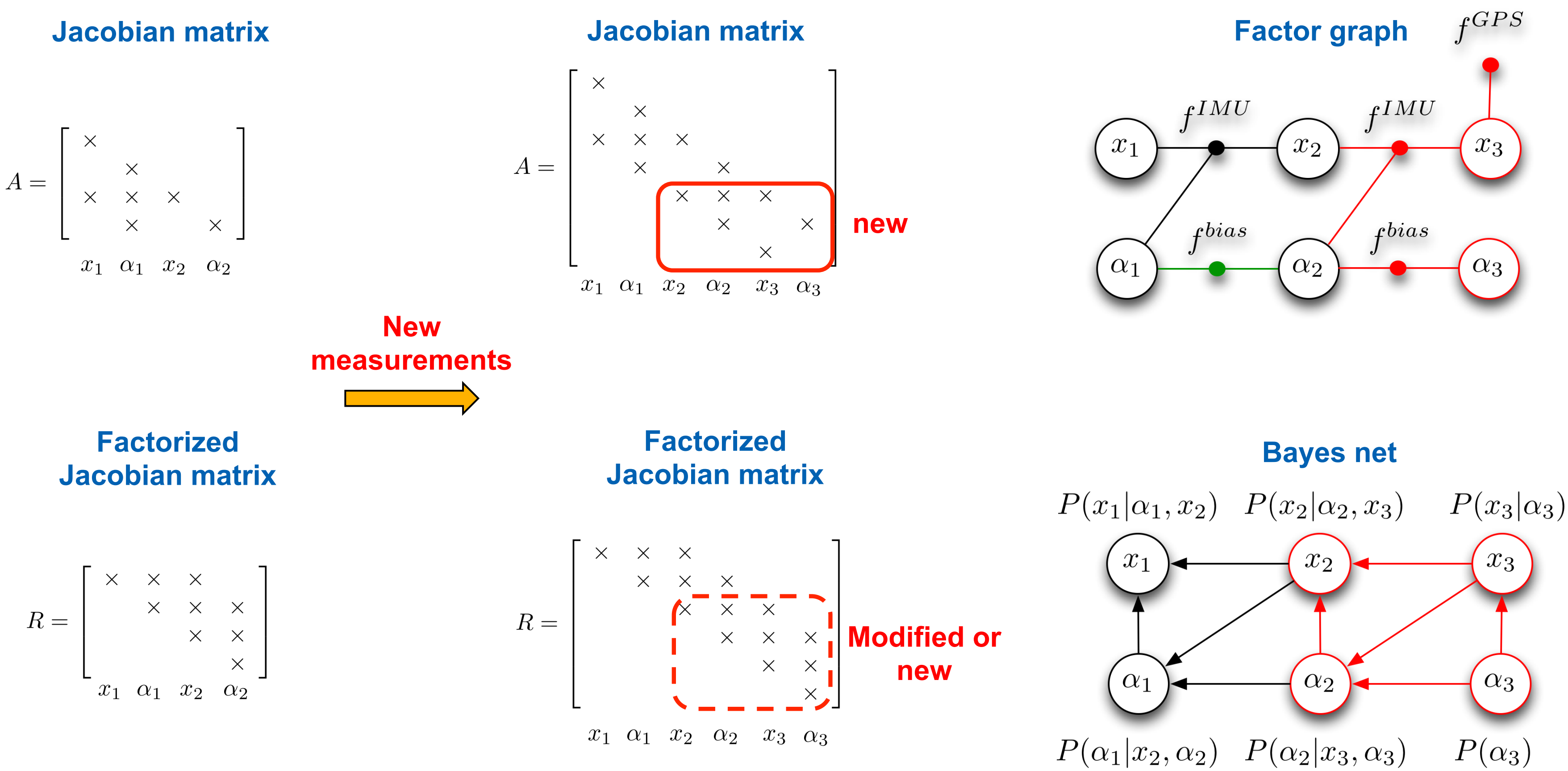
Incremental Inference Using Factor Graphs

- Adding **new** measurements
  - Each new measurement will generate a new factor in the graph
  - Equivalent to adding a new block-row to the Jacobian matrix A
- Optimization can proceed incrementally
  - Many of the calculations are the same as in the previous step - can be **reused**
  - **Only part** of the Bayes net is modified
- For example:



Incremental Inference Using Factor Graphs (Cont.)

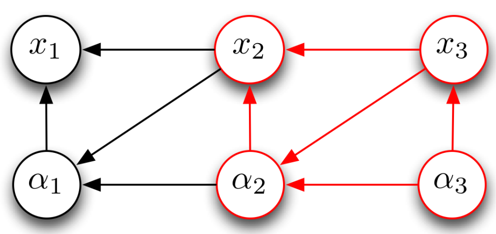
- Adding new measurements



Incremental Inference Using Factor Graphs

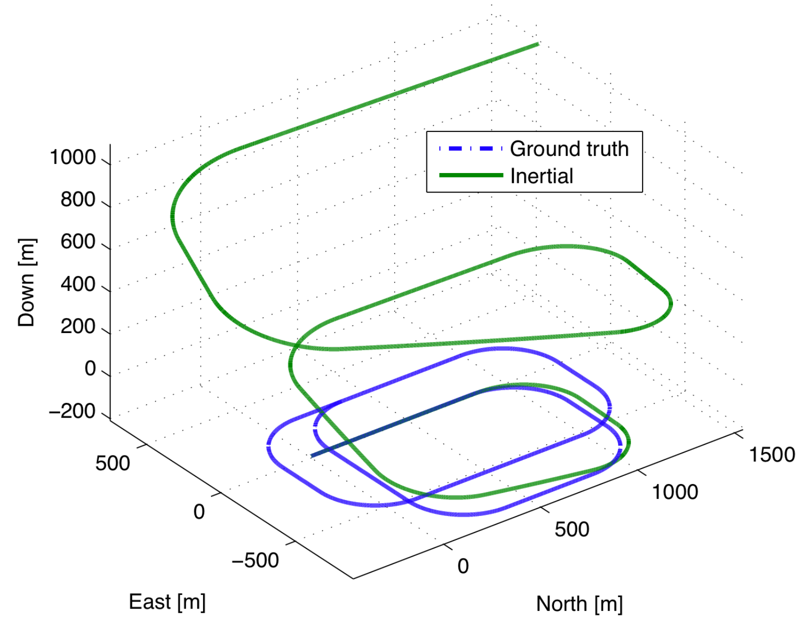
- Back-substitution - Solving for Δ given a Bayes net:
  - Bayes net is an efficient representation of the (sparse) triangular matrix R
  - Δ can be recovered fast [Kaess, et al., 2012]
    - Calculated only for some of the variables
    - Variables with a negligible Δ are identified and skipped

- Adaptive fixed-lag smoother
  - Processing IMU measurements:
    - Involves updating only a small (~4) number of nodes in Bayes net
  - Other (lower-frequency) measurements – appropriate parts of the Bayes net are modified



Results

- Simulated flight of an aerial vehicle
  - Velocity: 40 m/s velocity
  - Constant height: 200 m above mean ground level
  - Ground elevation: $\pm$ 50 m



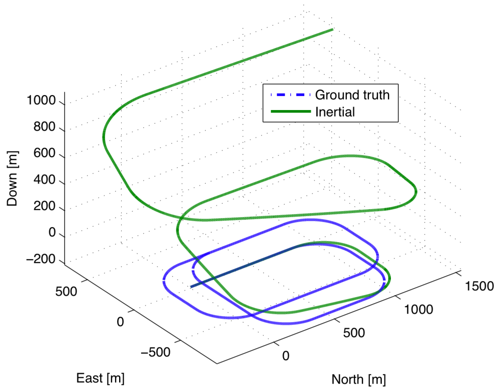
- Synthetic measurements of different sensors

Sensor	Accuracy (1σ)	Frequency
IMU	Acc. bias : 10 mg Gyro. bias : 10 deg/hr	100 Hz
GPS	Accuracy : 10 m	1 Hz
Stereo Camera	Image noise : 0.5 pix	0.5 / 0.1 Hz

  - Stereo camera produces relative pose measurements @ 0.5 Hz
  - Observations of short-track known landmarks @ 0.1 Hz:
    - Each landmark is observed for 3-4 frames
    - Each landmark is known within 10 m accuracy (1σ)

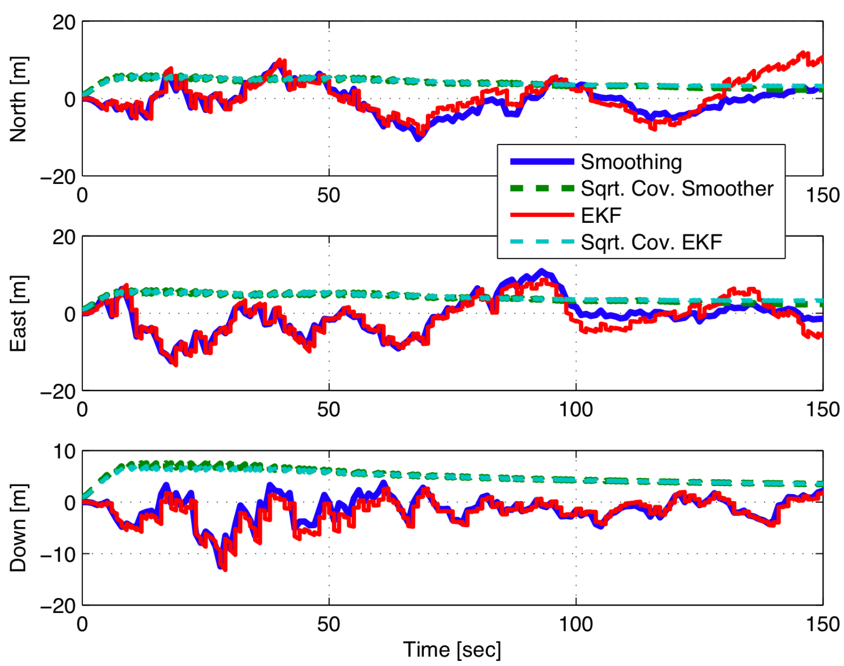
Results

- Incremental Smoothing vs. EKF
  - IMU @ 100 Hz
  - GPS @ 1 Hz: 10 m accuracy (1σ values)



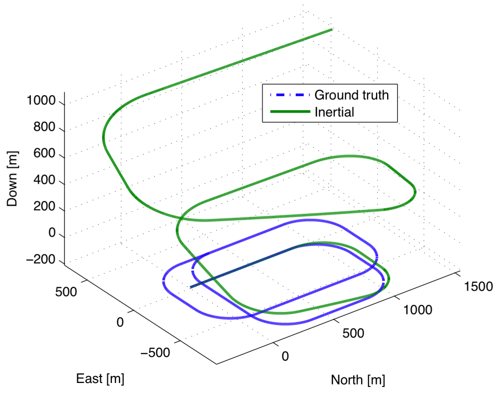
- Smoother timing performance: 4 ms (mean) with a standard deviation of 2.7 ms

Position estimation errors

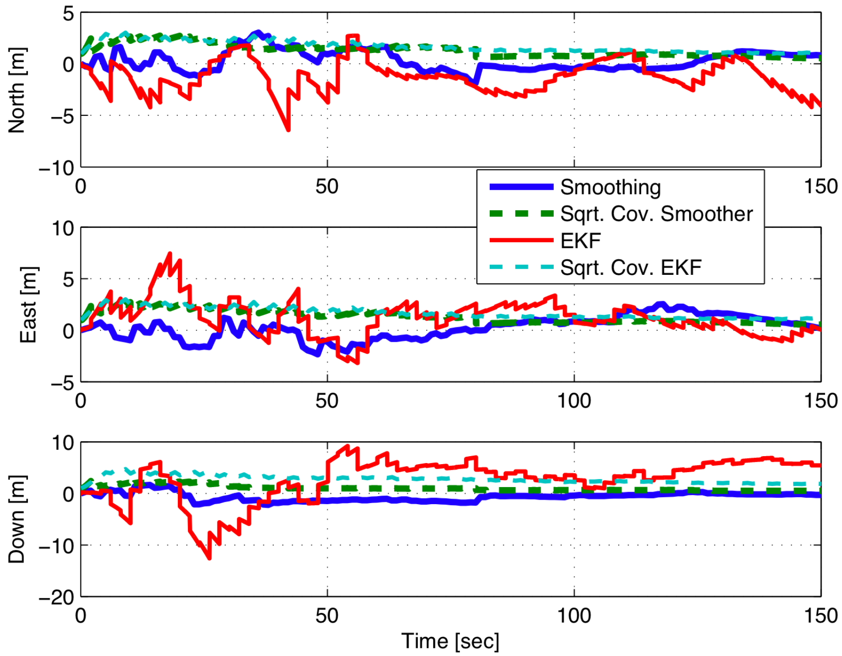


Results

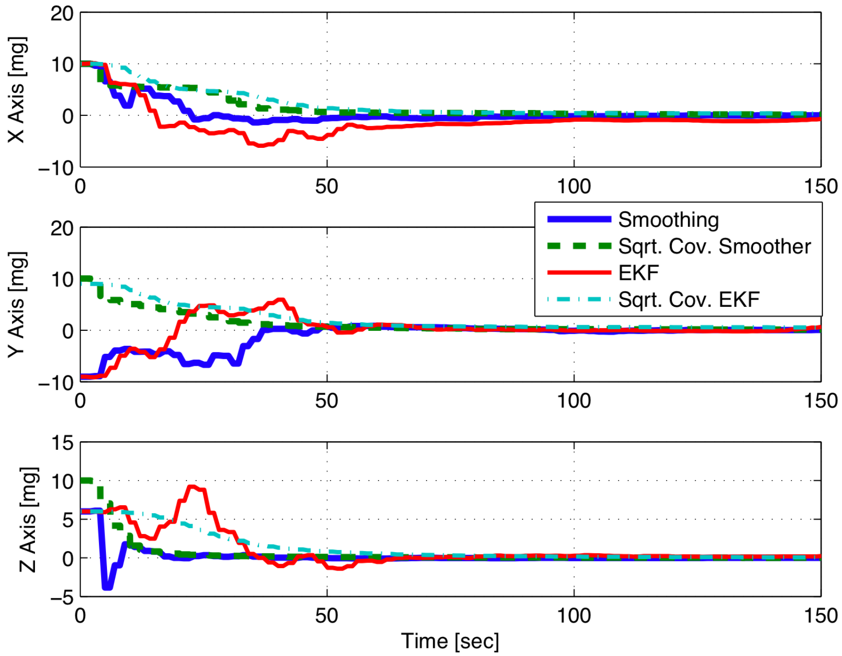
- Incremental Smoothing vs. EKF
  - IMU @ 100 Hz
  - Visual observations of short-track known landmarks @ 0.5 Hz



Position estimation errors

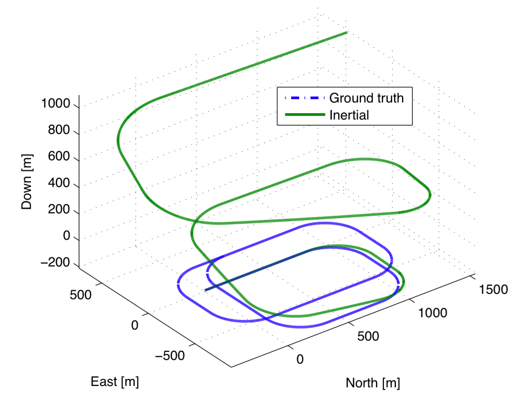


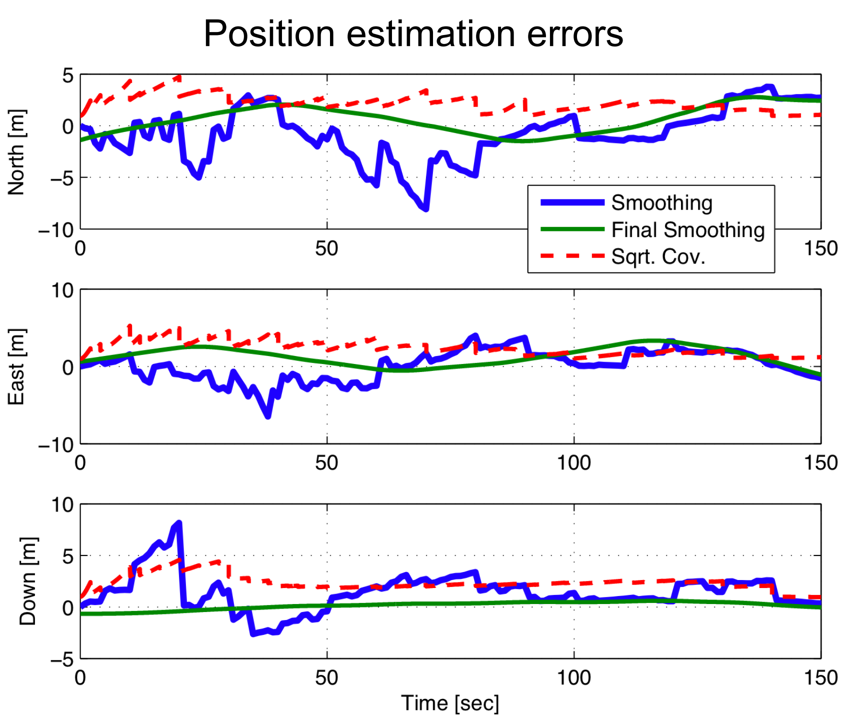
Accelerometer bias estimation errors

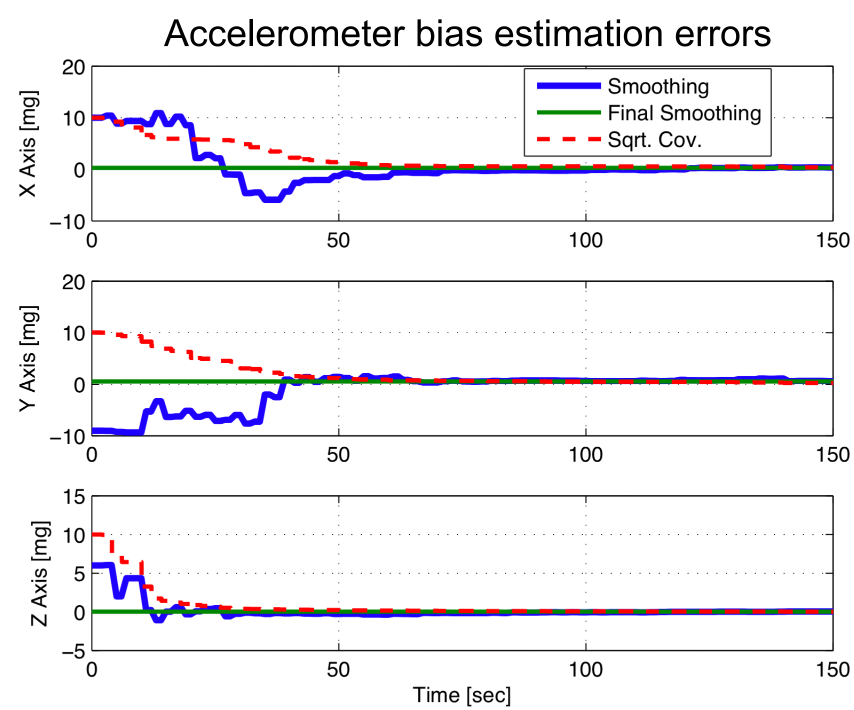


Results

- Incremental Smoothing in a Multi-Sensor Scenario
  - IMU @ 100 Hz
  - Relative pose measurements (from stereo camera) @ 0.5 Hz
  - Visual observations of short-track known landmarks @ 0.1 Hz







Conclusions

- We presented an incremental smoothing approach for inertial navigation
  - Flexible:
    - Allows to incorporate multi-rate and delayed measurements
    - Plug-and-play capabilities
  - Adaptive fixed-lag smoother:
    - Only a small number of variables are updated
    - Capable of operating at high frequency

- Loop closure measurements can also be incorporated in a factor graph framework:

"Concurrent Filtering and Smoothing"

M. Kaess, S. Williams, V. Indelman, R. Roberts, J. Leonard, F Dellaert

Fusion 2012