Concurrent Filtering and Smoothing

Michael Kaess[1], Stephen Williams[2],
Vadim Indelman[2], Richard Roberts[2],
John Leonard[1], Frank Dellaert[2]

1) CSAIL, Massachusetts Institute of Technology (MIT)
2) College of Computing, Georgia Institute of Technology

Navigation

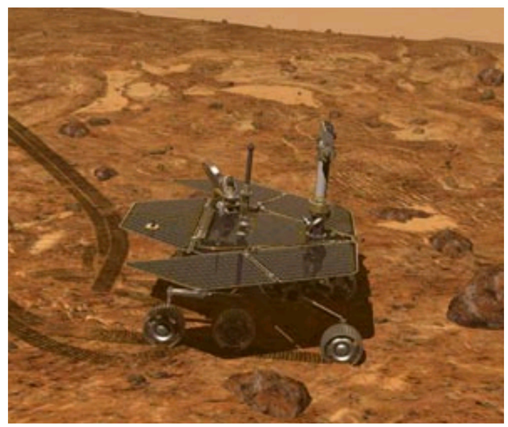
- Tracking position, velocity and orientation of an observer using IMU, GPS, LBL (acoustic), wind speed, camera…

Aircraft

[wikipedia]

Mars Rover

[JPL]



Underwater

[MIT]

- How to solve? Depends on who you ask!

Navigation Community: Filtering

- Established, well-tested solution in Aerospace etc.
- Estimate current state
- Objective:

$$\hat{x}_t = \arg\max_{x_t} p(x_t|Z)$$

- Update:

$$p(x_{t+1}|Z) = \int_{x_t} p(x_{t+1}, x_t|Z)$$

[wikipedia]

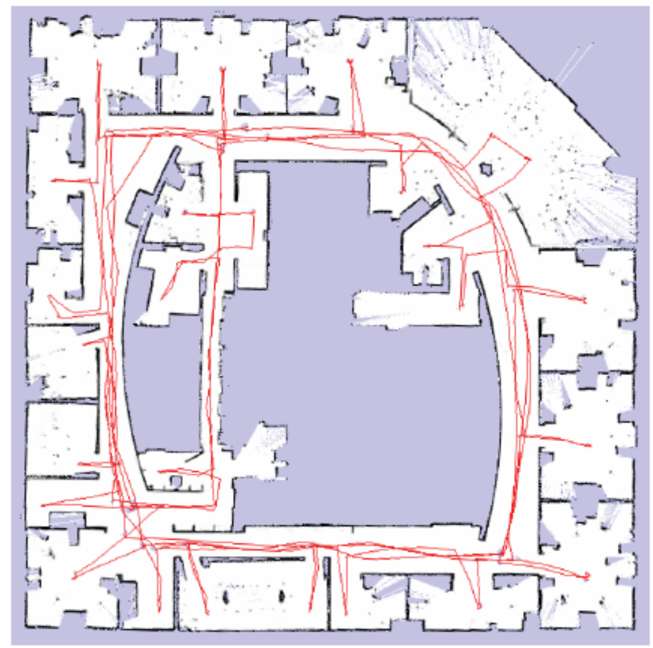
Robotics Community: Smoothing

- Full SLAM (Simultaneous Localization and Mapping)
- Estimate all states, current and past
- Objective:

$$\hat{X}_t = \arg\max_{X_t} p(X_t|Z)$$

- Update:

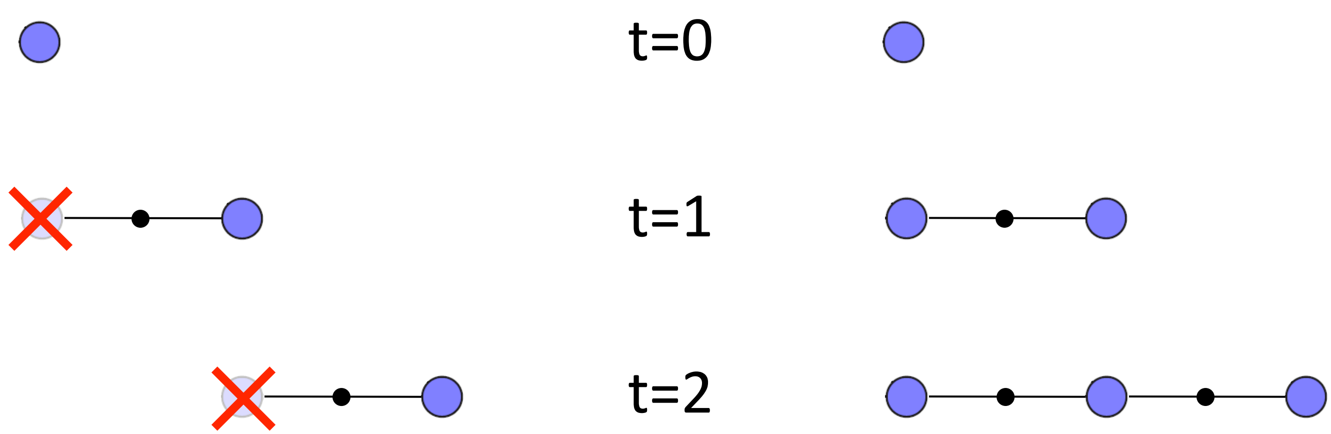
$$p(X_{t+1}|Z) = p(x_{t+1}, X_t|Z)$$



Map of Intel Labs

Filtering vs. Smoothing

Filtering | Smoothing



- Constant high frame rate
- Only current state is tracked

- Allows loop closure
- No constant time guarantee

Can we combine the advantages of both methods?

Filtering and Smoothing

- Can we combine the advantages of both methods?

- Goals:
  - Perform smoothing in a separate, asynchronous process
  - Maintain real-time performance of the filtering process
    - Minimize calculations of any required synchronization
  - Produce the optimal Bayesian solution

Related Work

- Combining filtering and loop closing [Eustice, Singh, and Leonard 2006]
  - Uses an augmented state filter to allow loop closures
  - Maintains real-time performance
  - Produces only an approximate solution

- Parallel tracking and mapping [Klein and Murray 2007] [Newcombe et al. 2011]
  - Performs Bundle Adjustment (BA) in a separate process
  - Relocalizes after BA instead of fusing results
  - Does not incorporate additional navigation sensors

- Dual-layer estimator [Mourikis and Roumeliotis 2008]
  - Combines an EKF with BA
  - EKF must be rolled back to incorporate the BA update in a consistent manner

Filtering and Smoothing (Cont.)

- In this work:
  - We perform smoothing and filtering in parallel
    - High-rate measurements are processed by the filter
    - Loop closures are added directly to the smoother
  - Smoothing and filtering are considered two components of a single optimization problem
    - Ensures the optimal Bayesian estimate is obtained
  - The problem is represented using a Bayes tree
    - Intuitive graphical model
    - Exploits sparsity
    - Allows incremental inference

Bayes Tree

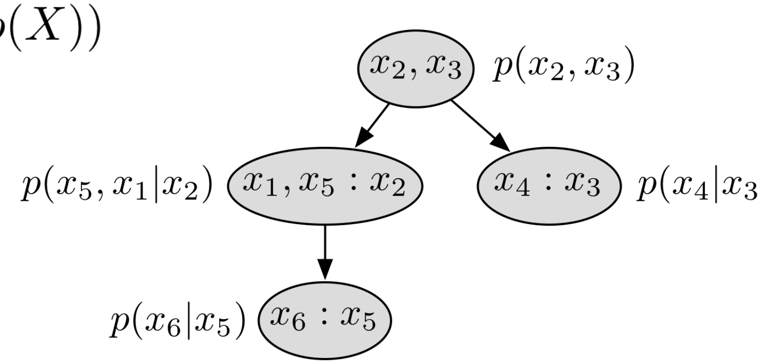
- The maximum a posteriori (MAP) estimate is given by

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

- Applying the chain rule yields a factorization:

$$p\left(X\right) = \prod_i p\left(X_i | S_i\right)$$

  – where $X_i, S_i \subset X$

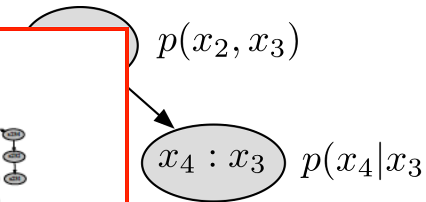


- Different factorizations exist, **depending on the order in which variables are chosen**
- Given a particular factorization, a unique Bayes tree can be constructed
  – Each node represents a conditional distribution
  – Each node is conditioned only on its ancestors in the tree
  – Solving for $\hat{X}$ involves applying the chain rule, starting from the root

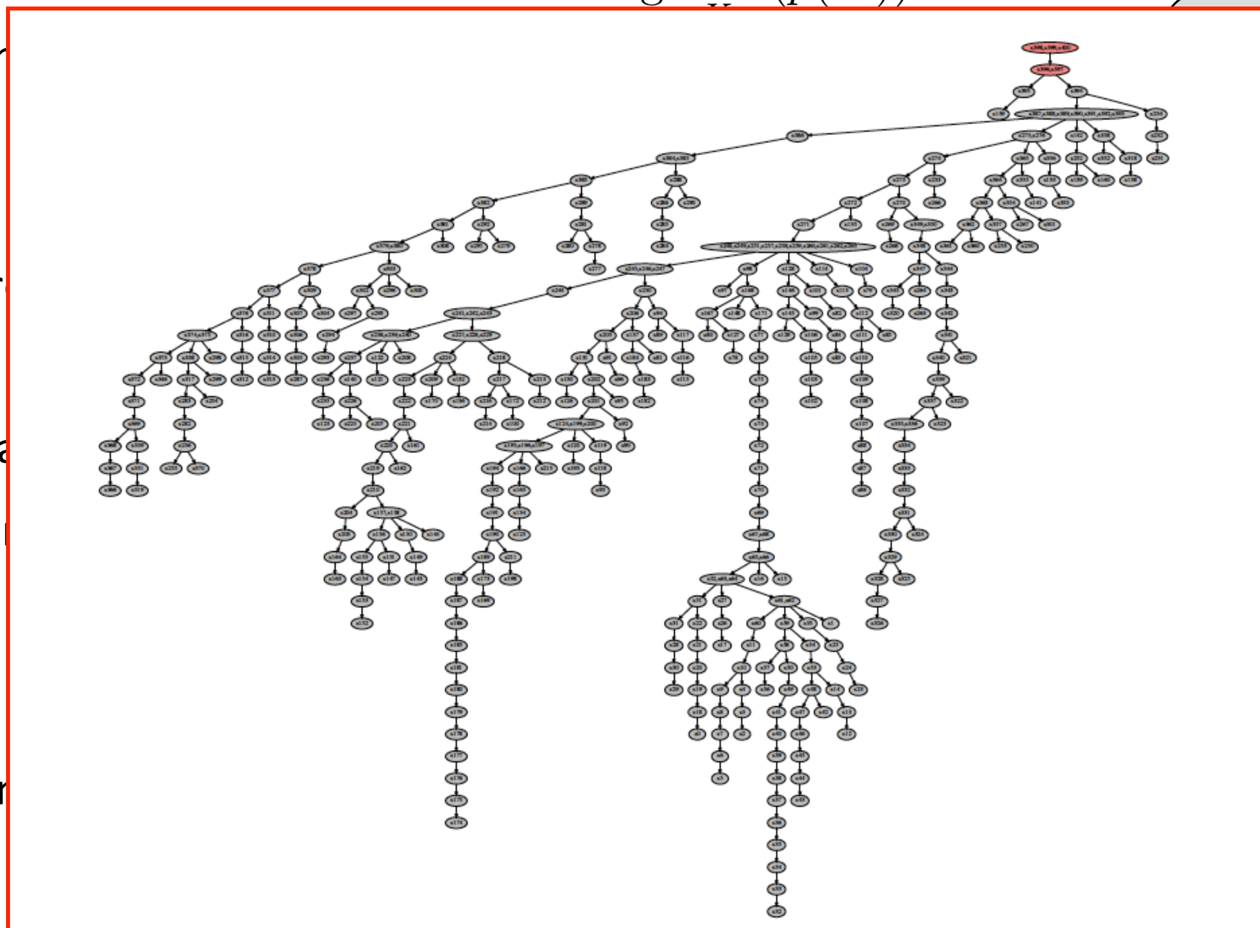
Bayes Tree

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

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



- Applying th
  - wher

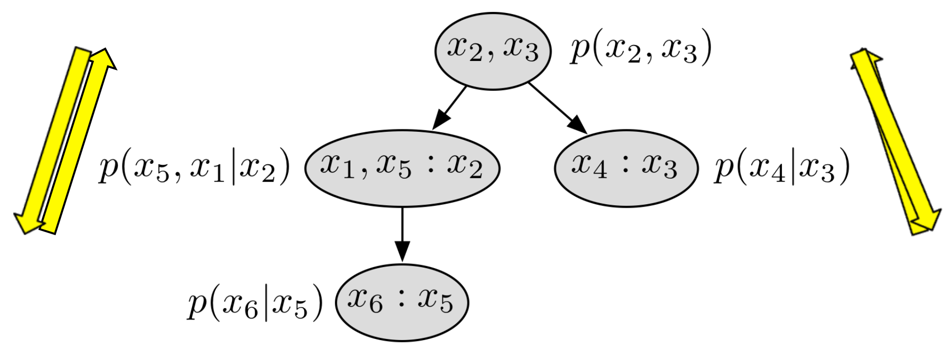


- Different fa **e chosen**
- Given a pa
  - Each
  - Each
  - Solvi

Bayes Tree – Gaussian Distribution

- Bayes tree corresponds to the square root information matrix $\mathcal{I} = R^T R$

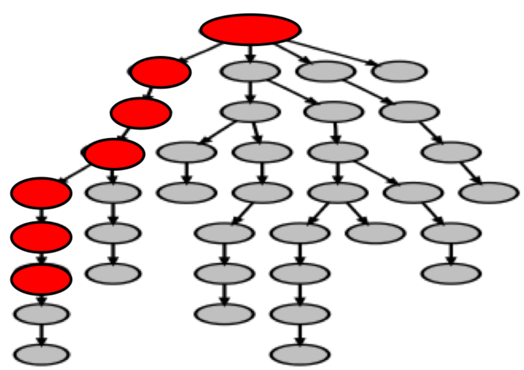
$$R = \begin{array}{cc} \begin{bmatrix} \times & \times & & & & \\ & \times & & & \times & \\ & & \times & & & \\ & & & \times & \times & \\ & & & & \times & \times \\ & & & & & \times \end{bmatrix} & \begin{array}{c} x_6 \\ x_5 \\ x_1 \\ x_4 \\ x_3 \\ x_2 \end{array} \\ \begin{array}{cccccc} x_6 & x_5 & x_1 & x_4 & x_3 & x_2 \end{array} & \end{array}$$



- **Factorization\Elimination** (i.e. calculation of R) corresponds to constructing the Bayes tree
  - Performed from bottom upwards
- Solving for $\hat{X}$:
  - Performed by **back-substitution,** from root of Bayes tree downwards

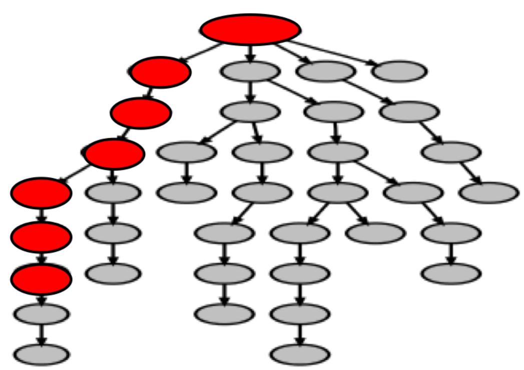
Efficient Bayes Tree Updates [Kaess et al IJRR 12]

- Key Insights
  - **Incorporating new measurements can be done efficiently**
  - Affects only variables involved in the measurement model and their ancestors
    - Only affects variables in the path to the top of the tree
    - Branches remain unaffected



Efficient Bayes Tree Updates [Kaess et al IJRR 12]

- Key Insights
  - **Many variable orderings exist**
  - Ordering affects:
    - Tree structure - different factorization of $p(X) = \prod_i p(X_i | S_i)$
    - Number of variables in each node / Computational complexity
    - Does not affect the solution



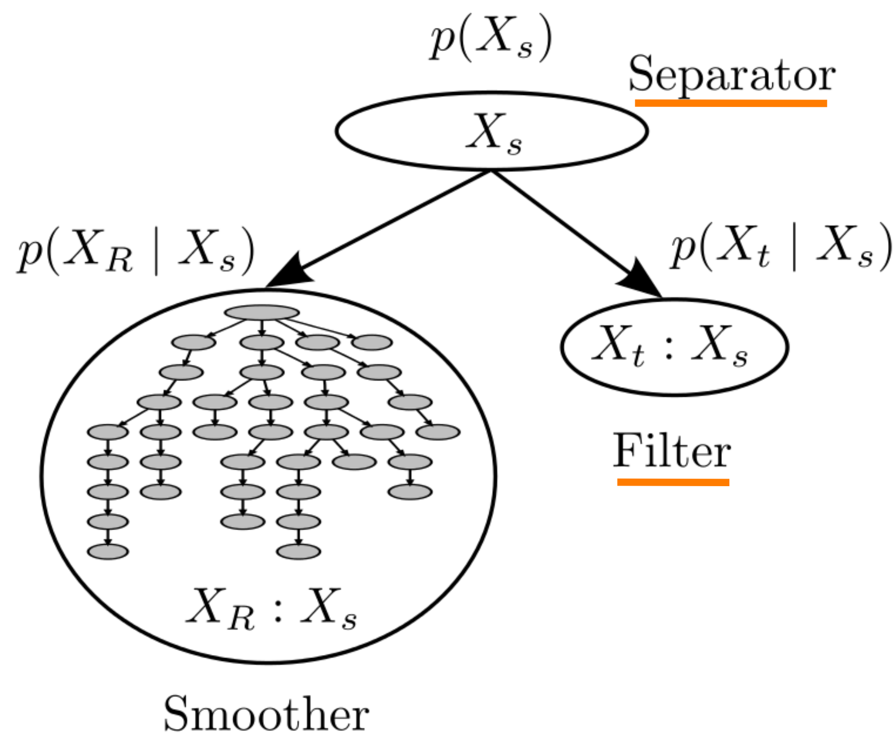
Parallelizing Filtering and Smoothing

- Factorization based on suitable variable ordering:

$$p\left(X\right)=p\left(X_R \mid X_s\right)p\left(X_s\right)p\left(X_t \mid X_s\right)$$

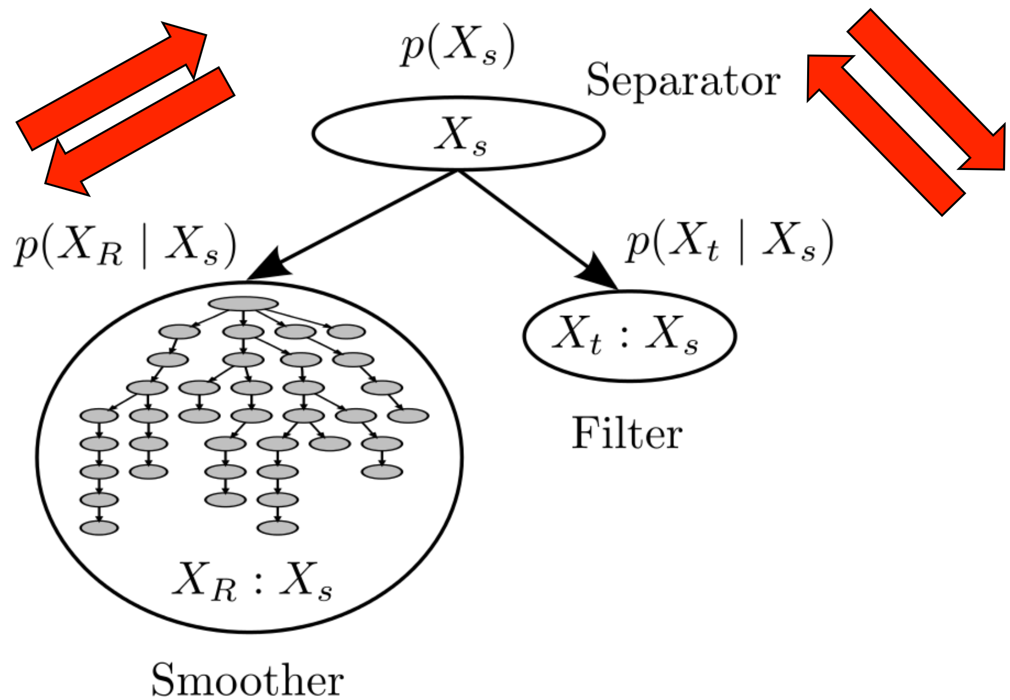
- Corresponding Bayes tree:

- Allows concurrent updates to filter and smoother!



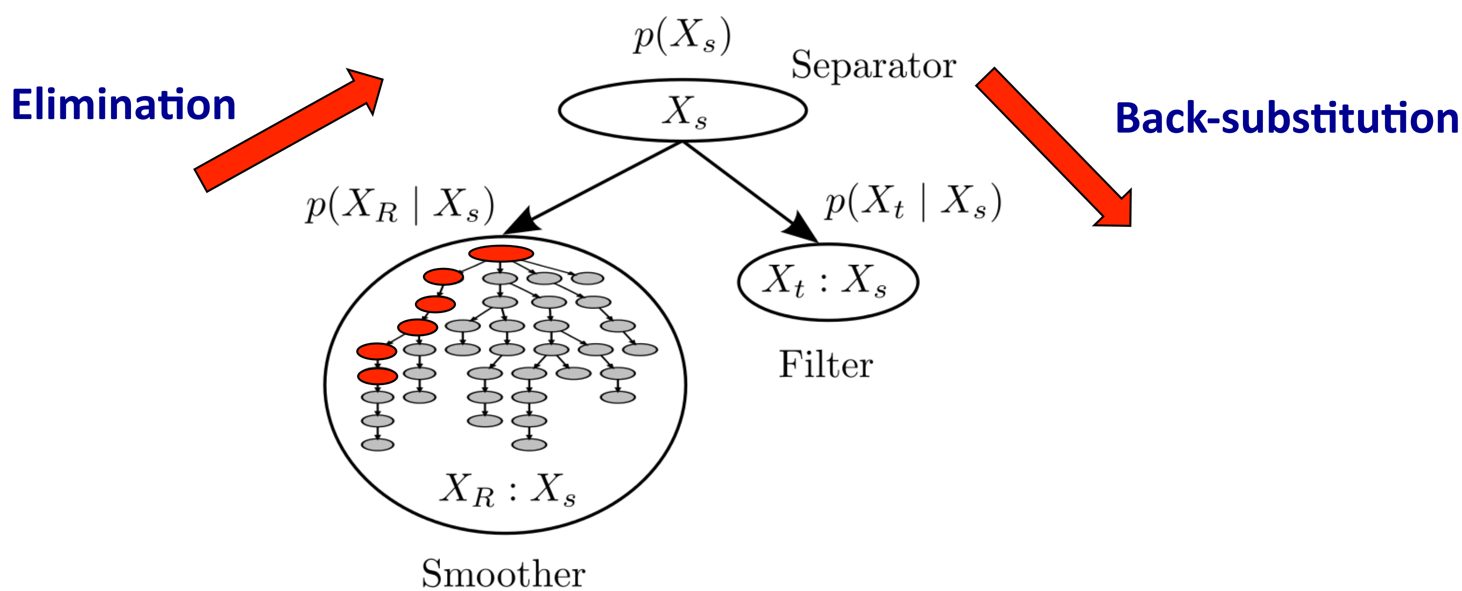
Parallelizing Filtering and Smoothing

- Filter and smoother are kept periodically synchronized
  - Information flows between the smoother and the filter via the separator



Parallelizing Filtering and Smoothing

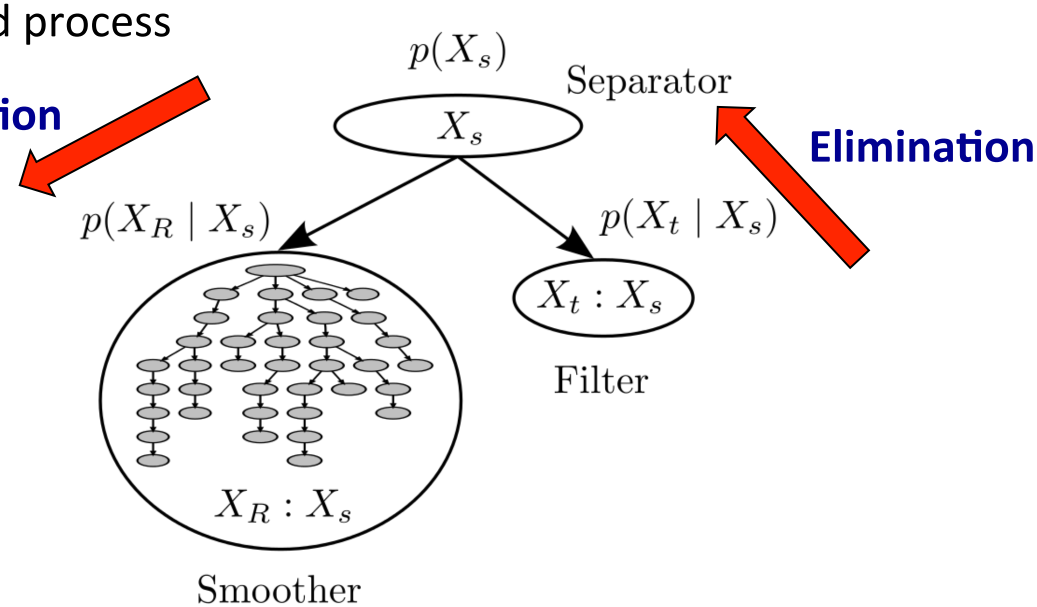
- **Loop closure measurements are added to the smoother**
  - Smoother processes new measurements in a background process
  - Upon completion, the separator (root of the Bayes tree) is updated
  - Updates from the root are propagated to the filter in a fast process
    - Involves only a small number of variables



Parallelizing Filtering and Smoothing

- **High-rate measurements are added to the filter**
  - Filter processes measurements in real time
    - Separator (root) is updated as part of this process
  - The separator accumulates updates
  - When the smoother is available:
    - Propagate these updates to the smoother
    - Performed in a background process

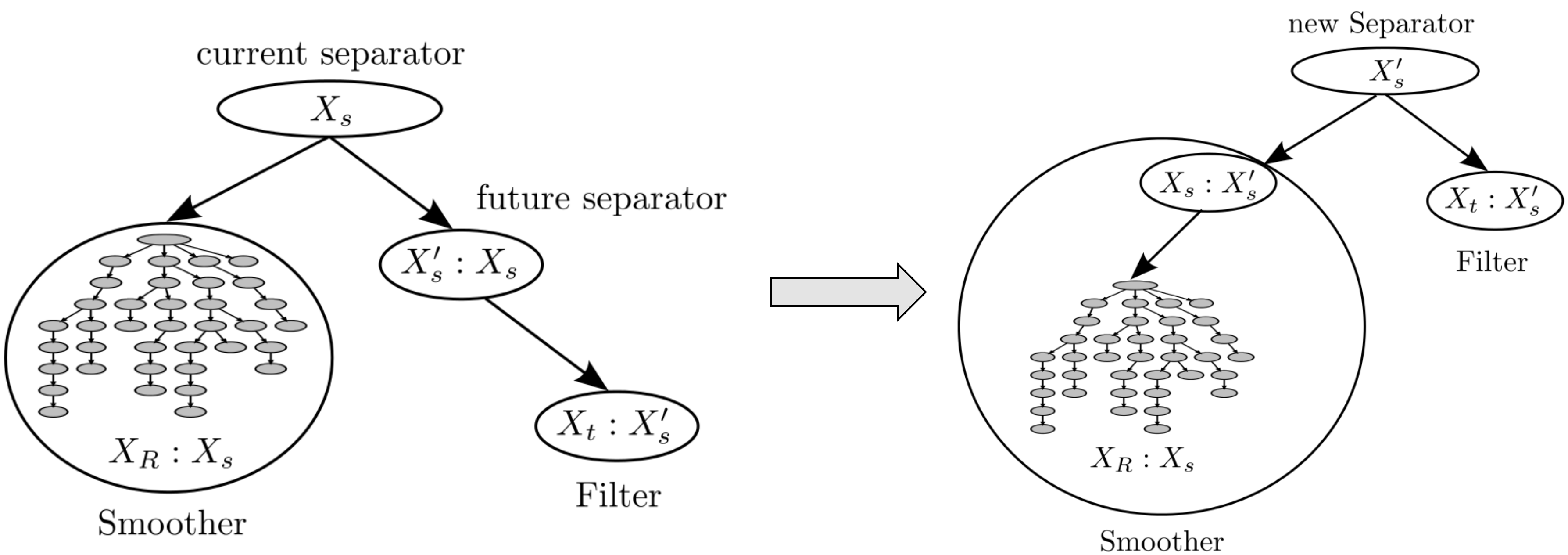
Back-substitution



How to move variables from the filter to the smoother?

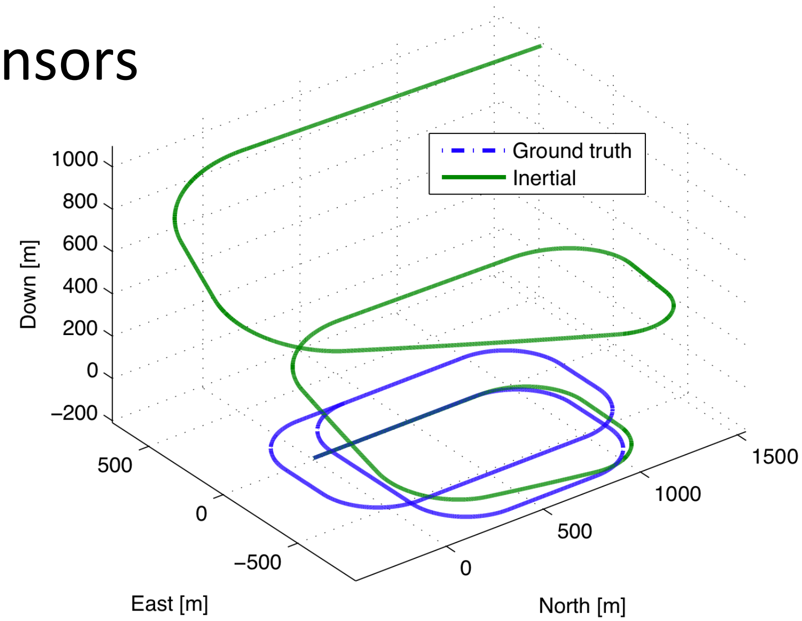
Moving Variables to the Smoother

- As time progresses, the filter maintains a sparse set of variables
- These variables form a chain of nodes on the filter side of the Bayes tree
- Whenever the smoother is available:
  - The current separator (root) node is moved to the smoother
  - A new separator is formed from the chain on the filter side
  - This is achieved by choosing a new variable ordering for these two nodes



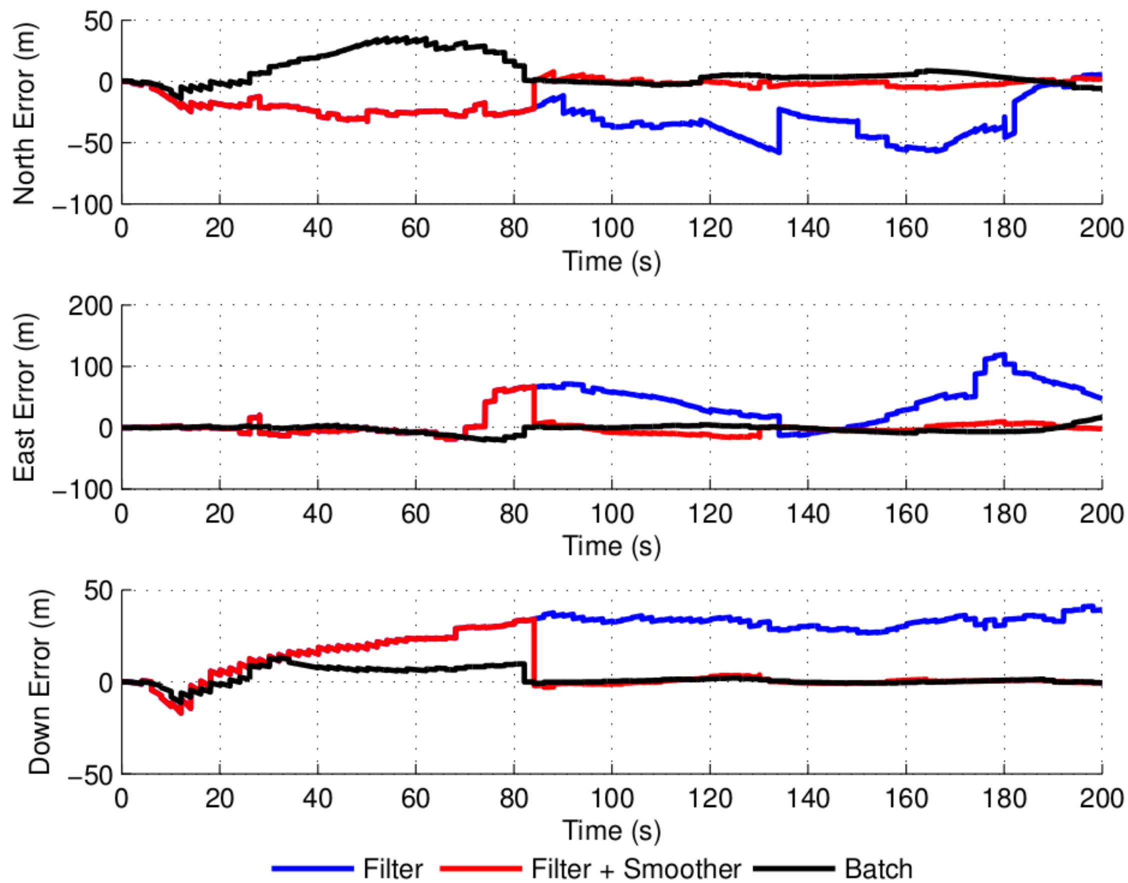
Evaluation: Simulation

- Simulated flight of an aerial vehicle
  - Velocity: 40 m/s
  - Constant height: 200 m above mean ground level
  - Ground elevation: $\pm$50 m
- Synthetic measurements of different sensors
  - IMU at 100 Hz
  - Stereo camera at 0.5Hz
    - Produces **relative pose** measurements
  - Sparse loop closures found in camera data
    - 22 loop closure events identified
    - Provided directly to the smoother



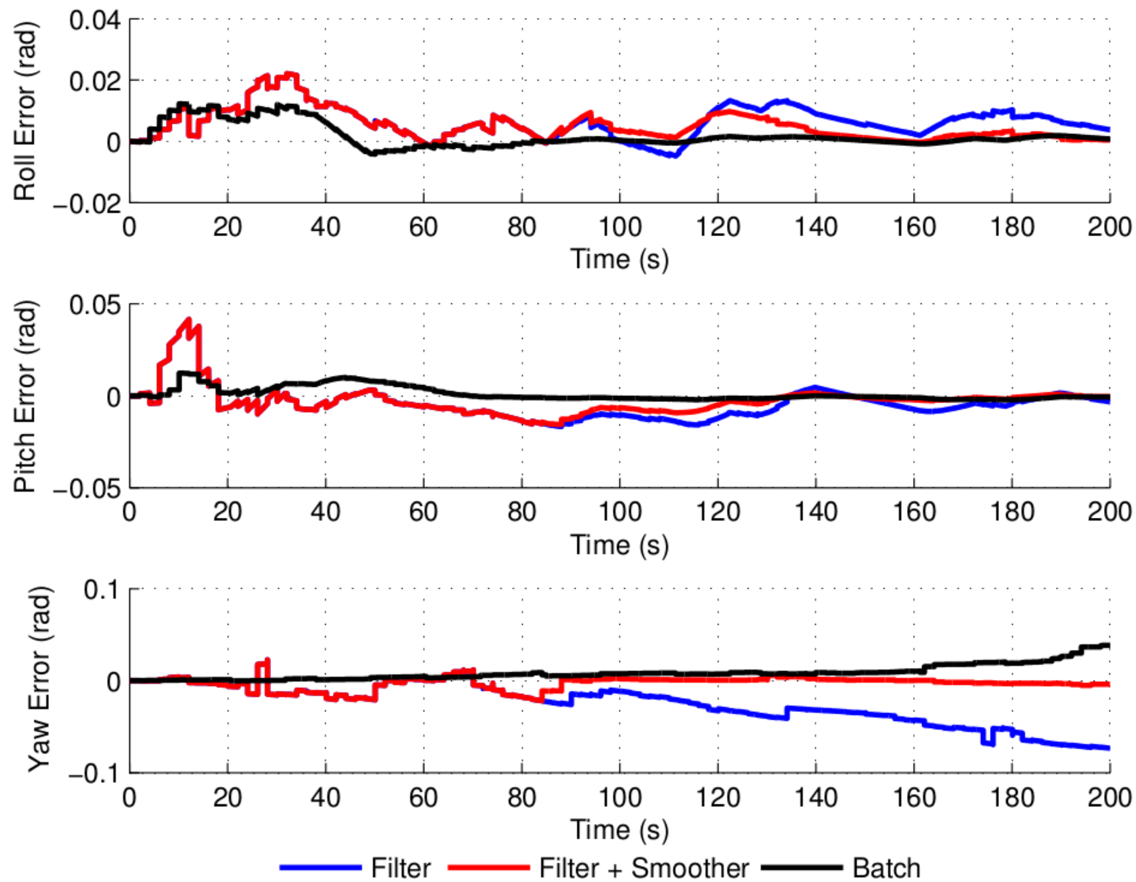
Evaluation: Positional Error

- Compared methods:
  - **Filter alone**
  - **Concurrent Filter and Smoother**
    (Our approach)
  - **Incremental Batch**



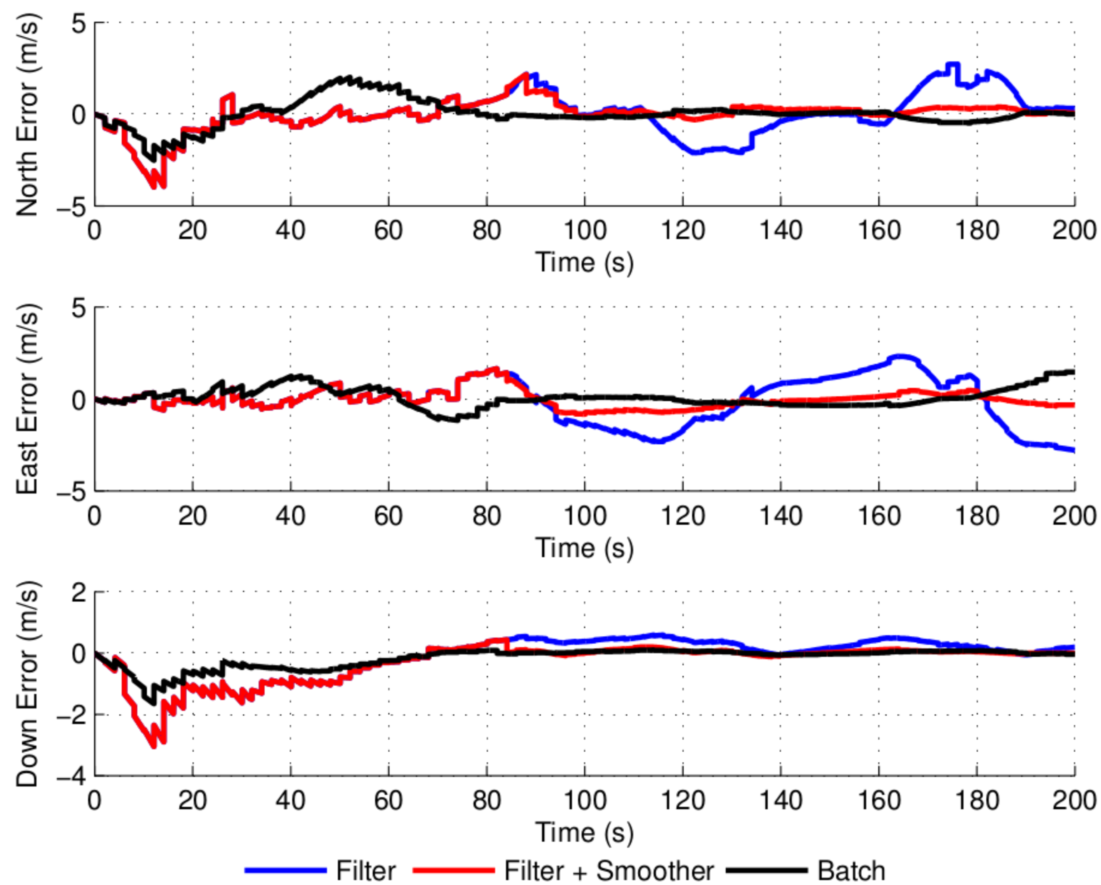
Evaluation: Rotational Error

- Compared methods:
  - **Filter alone**
  - **Concurrent Filter and Smoother**
    (Our approach)
  - **Incremental Batch**



Evaluation: Velocity Error

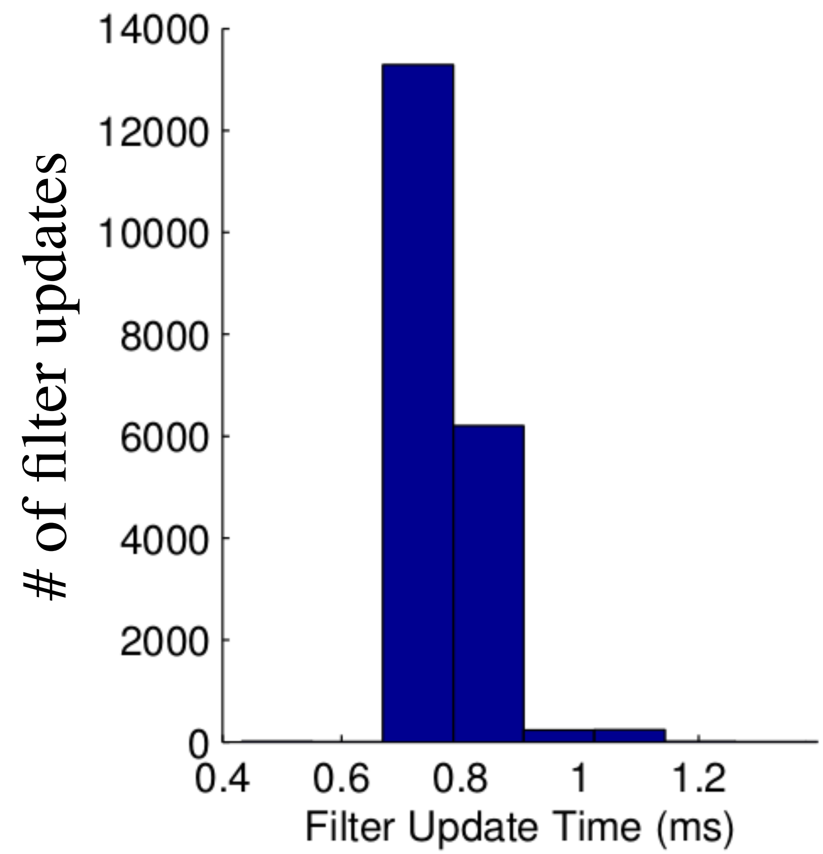
- Compared methods:
  - **Filter alone**
  - **Concurrent Filter and Smoother**
    (Our approach)
  - **Incremental Batch**



Timing: Filter Update

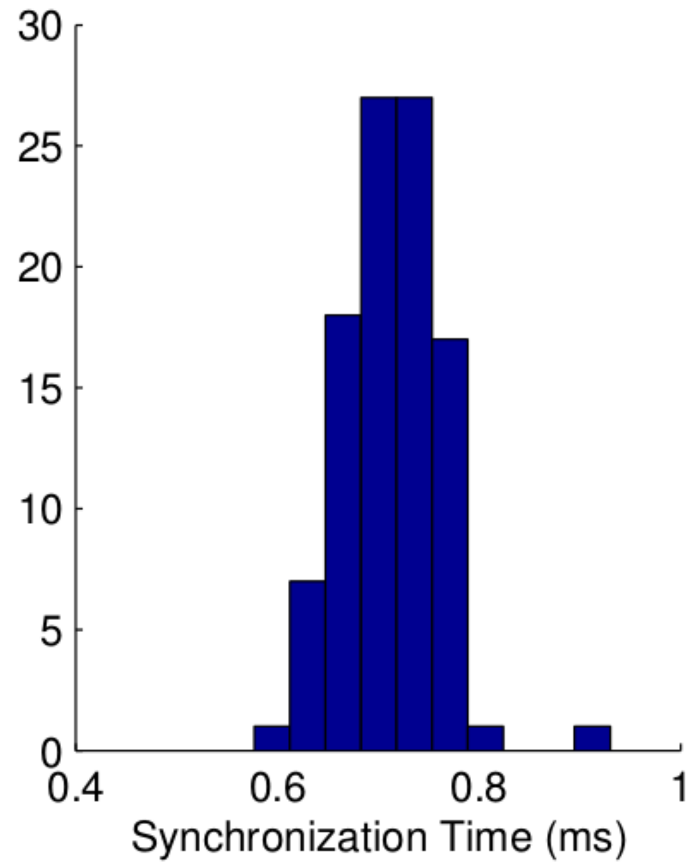
- Filter updates in constant time

- Confirmed by evaluation:



Timing: Synchronization

- Synchronization only affects top of tree
  - Performed after smoother is done
- Depends on size of root clique
- Evaluation shows very fast synchronization



Future Work

- Relinearization
  - Filter produces linear conditionals
  - “Lifting” of linear constraints (Konolige, TRO 2008)
- Bound complexity of smoother
  - Fixed-lag
  - Sparsification (SLAM)

Summary

- We combined filtering and smoothing
  - Parallel formulation in Bayes tree
  - Constant time filtering
  - Loop closing capability

- Tomorrow:

"Factor Graph Based Incremental Smoothing in Inertial Navigation Systems"

V. Indelman, S. Williams, M. Kaess, F. Dellaert