Distributed Vision-Aided Cooperative Navigation Based on Three-View Geometry

Vadim Indelman, Pini Gurfil

Distributed Space Systems Lab,

Aerospace Engineering, Technion

Ehud Rivlin

Computer Science, Technion

Hector Rotstein

RAFAEL – Advanced Defense Systems

March 2011

Contents

- Introduction
- Three-View Constraints
- Fusion with Navigation System
- Results
- Conclusions

Introduction

- A group of cooperative platforms is considered
  - Required to autonomously perform different missions
  - Navigation is an essential capability
- Dead reckoning \ inertial navigation errors have to be compensated
  - External sensors (e.g.: GPS, camera, range sensor)
  - Additional information (e.g.: DTM)
- What happens if GPS is unavailable or unreliable?

- <u>This work</u>:
  - Vision-based approach for cooperative navigation
  - Each platform is equipped only with: INS, single camera
    - No additional sensors or a priori information is required
    - Except for initial navigation solution and camera calibration parameters

Previous Work

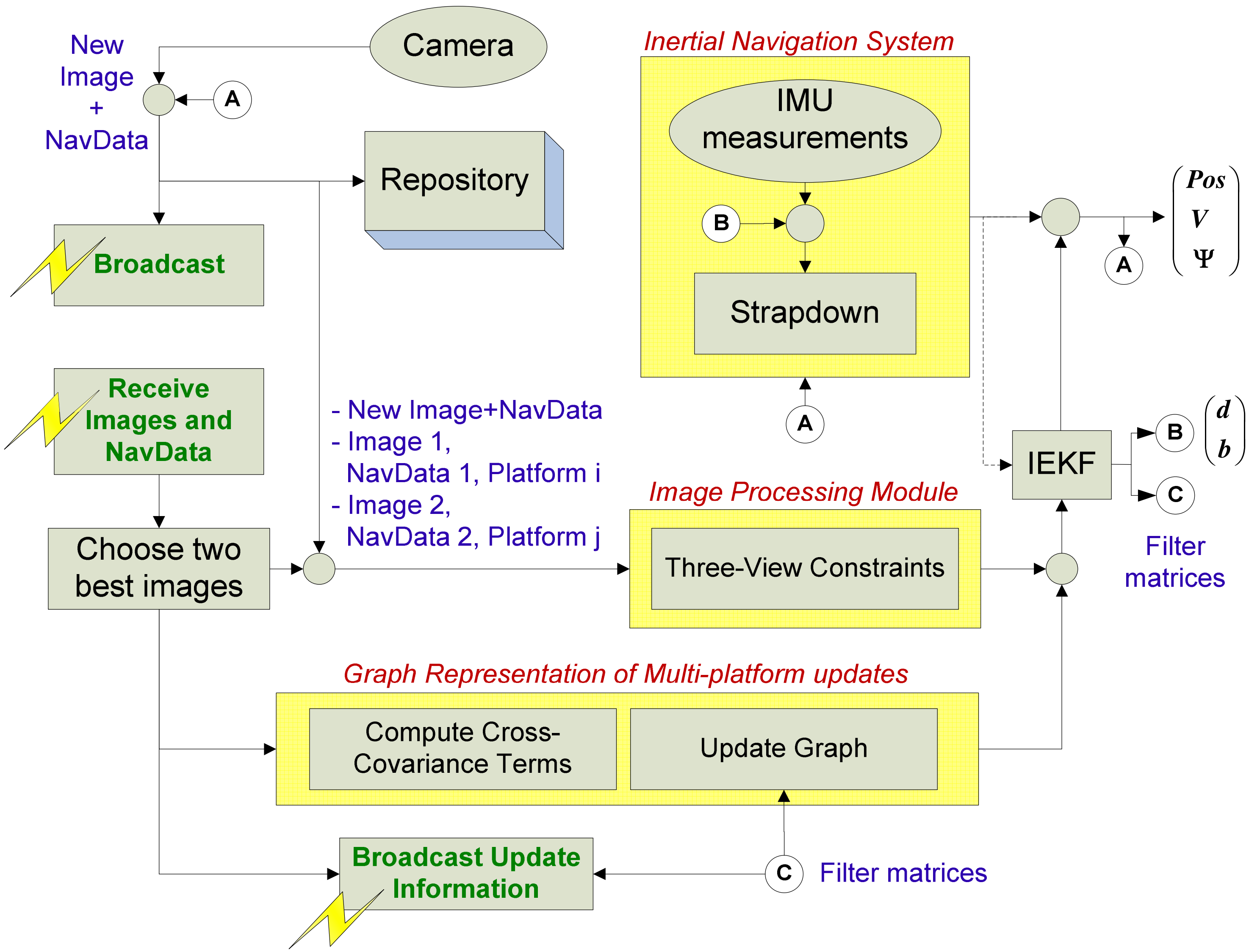
- Use some robots as landmarks: "Cooperative Positioning with Multiple Robots", Kurazume R. et al., 1994
- Relative pose measurements between pairs of robots: "Distributed Multirobot Localization", Roumeliotis S.I. and Bekey G.A., 2002
- Direct & indirect encounters between pairs of robots, nonlinear optimization: "Multiple Relative Pose Graphs for Robust Cooperative Mapping", Kim B. et al., 2010
- Vision-aided navigation based on three-view geometry: "Mosaic Aided Navigation: Tools, Methods and Results'', Indelman V. et al., 2010
- Consistent information fusion: "Consistent Cooperative Localization", Bahr A. et al., 2009

Concept

- Navigation update whenever the same scene is observed by three views
  - Possibly captured by different platforms
  - Not necessarily at the same time
  - The camera is not required to be aimed towards other platforms (in contrast to relative pose measurements)

- Setup
  - Each platform is equipped with its own
    - INS, Camera
    - Perhaps, additional sensors or a-priori information
  - All\some platforms maintain a repository of stored images associated with navigation information
  - The platforms are able to exchange navigation and imagery data
  - Each platform maintains a local graph, required for correlation calculation

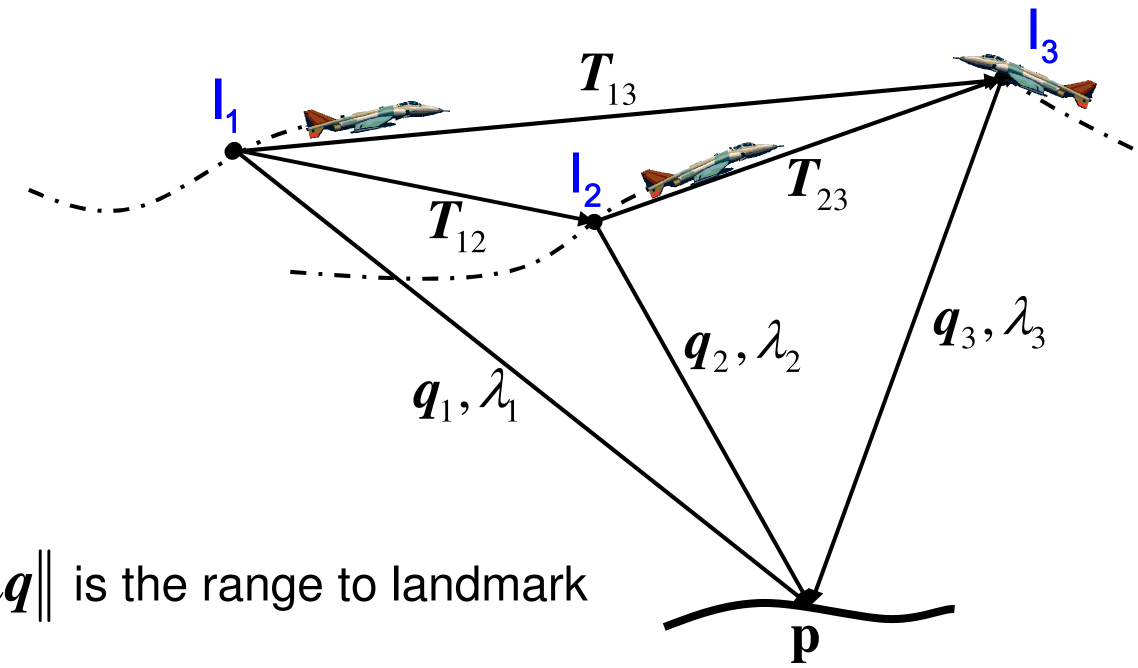
Overview



Camera
New
Image
+
NavData
A
Repository
Broadcast
Receive
Images and
NavData
Choose two
best images
- New Image+NavData
- Image 1,
NavData 1, Platform i
- Image 2,
NavData 2, Platform j
Inertial Navigation System
IMU
measurements
B
Strapdown
A
Pos
V
Ψ
A
IEKF
B
d
b
C
Filter
matrices
Image Processing Module
Three-View Constraints
Graph Representation of Multi-platform updates
Compute Cross-
Covariance Terms
Update Graph
Broadcast Update
Information
C
Filter matrices

Three-view Constraints

- Each image may be captured by a different platform
- The images are not necessarily captured at the same time
- Images are stored in repositories and retrieved upon demand



- p - static landmark
- $\boldsymbol{q}$ - line of sight (LOS)
- λ - scale parameter, s.t. $\|\lambda \boldsymbol{q}\|$ is the range to landmark
- $\boldsymbol{T}_{ij}$ - translation from i to j

Three-view Constraints (cont.)

$$\boldsymbol{q}_1^T\left(\boldsymbol{T}_{12}\times\boldsymbol{q}_2\right)=0$$

$$\boldsymbol{q}_2^T\left(\boldsymbol{T}_{23}\times\boldsymbol{q}_3\right)=0$$

$$\left(\boldsymbol{q}_2\times\boldsymbol{q}_1\right)^T\left(\boldsymbol{q}_3\times\boldsymbol{T}_{23}\right)=\left(\boldsymbol{q}_1\times\boldsymbol{T}_{12}\right)^T\left(\boldsymbol{q}_3\times\boldsymbol{q}_2\right)$$

- First two equations – epipolar constraints
- Third equation – relates between the magnitudes of $\boldsymbol{T}_{12}$ and $\boldsymbol{T}_{23}$
- Reformulating:

$$\Rightarrow \quad \begin{aligned}\left[\boldsymbol{g}^T\right]_{1\times3}\boldsymbol{T}_{12}&=0\\ \left[\boldsymbol{f}^T\right]_{1\times3}\boldsymbol{T}_{23}&=0\\ \left[\boldsymbol{u}^T\right]_{1\times3}\boldsymbol{T}_{23}&=\left[\boldsymbol{w}^T\right]_{1\times3}\boldsymbol{T}_{12}\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{g}&=\boldsymbol{g}\left(\boldsymbol{q}_1,\boldsymbol{q}_2\right)\\ \boldsymbol{f}&=\boldsymbol{f}\left(\boldsymbol{q}_2,\boldsymbol{q}_3\right)\\ \boldsymbol{u}&=\boldsymbol{u}\left(\boldsymbol{q}_1,\boldsymbol{q}_2,\boldsymbol{q}_3\right)\\ \boldsymbol{w}&=\boldsymbol{w}\left(\boldsymbol{q}_1,\boldsymbol{q}_2,\boldsymbol{q}_3\right)\end{aligned}$$

Three-view Constraints (cont.)

- Multiple features
  - Matching pairs between 1st and 2nd view $\left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2} \right\}_{i=1}^{N_{12}}$
  - Matching pairs between 2nd and 3rd view $\left\{ \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{23}}$
  - Matching triplets between the three views $\left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{123}}$

$$
\begin{aligned}
\left[\boldsymbol{u}_i^T \right]_{1\times 3} \boldsymbol{T}_{23} &= \left[\boldsymbol{w}_i^T \right]_{1\times 3} \boldsymbol{T}_{12} & i &= 1, \ldots, N_{123} \\
\left[\boldsymbol{f}_j^T \right]_{1\times 3} \boldsymbol{T}_{23} &= 0 & j &= 1, \ldots, N_{23} \\
\left[\boldsymbol{g}_k^T \right]_{1\times 3} \boldsymbol{T}_{12} &= 0 & k &= 1, \ldots, N_{12}
\end{aligned}
\quad \Longrightarrow \quad
\begin{bmatrix} U \\ F \\ 0 \end{bmatrix}_{N\times 3} \boldsymbol{T}_{23} = \begin{bmatrix} W \\ 0 \\ G \end{bmatrix}_{N\times 3} \boldsymbol{T}_{12}
$$

$$N = N_{123} + N_{12} + N_{23}$$

Fusion with Navigation using Implicit Extended Kalman Filter (IEKF)

- Residual Measurement

$$z \equiv \begin{bmatrix} U \\ F \\ 0 \end{bmatrix}_{N\times 3} \boldsymbol{T}_{23} - \begin{bmatrix} W \\ 0 \\ G \end{bmatrix}_{N\times 3} \boldsymbol{T}_{12}$$

- Recall
  - All original LOS vectors are expressed in camera system of the appropriate view
  - $\boldsymbol{T}_{23}, \boldsymbol{T}_{12}$ are functions of $\boldsymbol{Pos}_3, \boldsymbol{Pos}_2, \boldsymbol{Pos}_1$

$$\boxed{\boldsymbol{Pos}_i \equiv \boldsymbol{Pos}_i\left(t_i\right)}$$

$$\Downarrow$$

$$z = \boldsymbol{h}\left(\boldsymbol{Pos}_3, \Psi_3, \boldsymbol{Pos}_2, \Psi_2, \boldsymbol{Pos}_1, \Psi_1, \left\{\boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3}\right\}\right)$$

Fusion with Navigation using IEKF (cont.)

- State vector definition: $\boldsymbol{X} = \begin{bmatrix} \Delta \boldsymbol{P}^T & \Delta \boldsymbol{V}^T & \Delta \boldsymbol{\Psi}^T & \boldsymbol{d}^T & \boldsymbol{b}^T \end{bmatrix}^T$
- Inertial navigation error of the i-th platform: $\boldsymbol{X}_i\left(t_b\right) = \Phi^i_{t_a \to t_b} \boldsymbol{X}_i\left(t_a\right) + \boldsymbol{\omega}^i_{t_a \to t_b}$
- Linearization of $\boldsymbol{z}$

$$\boldsymbol{z} = \boldsymbol{h}\left(\boldsymbol{Pos}_3, \Psi_3, \boldsymbol{Pos}_2, \Psi_2, \boldsymbol{Pos}_1, \Psi_1, \left\{\boldsymbol{q}^{C_1}_{1_i}, \boldsymbol{q}^{C_2}_{2_i}, \boldsymbol{q}^{C_3}_{3_i}\right\}\right)$$
$$\cong H_3 \boldsymbol{X}_{\mathbf{III}}\left(t_3\right) + H_2 \boldsymbol{X}_{\mathbf{II}}\left(t_2\right) + H_1 \boldsymbol{X}_{\mathbf{I}}\left(t_1\right) + D\boldsymbol{v} + H.O.T.$$

  - $\boldsymbol{X}_{\mathbf{III}}\left(t_3\right), \boldsymbol{X}_{\mathbf{II}}\left(t_2\right), \boldsymbol{X}_{\mathbf{I}}\left(t_1\right)$ represent navigation errors of different platforms at different time instances.
    - **None of these are known a-priori**
    - **Can be correlated**
  - Theoretically, all the participating platforms can be updated

Fusion with Navigation (cont.)

- The measurement update step involves cross-covariance terms
  - E.g., if only platform III is updated: $K = P_{X_{III}(t_3)z(t_3,t_2,t_1)} P^{-1}_{z(t_3,t_2,t_1)}$
    - with

$$P_{X(t_3)z(t_3,t_2,t_1)} = P_3 H_3^T + P_{32} H_2^T + P_{31} H_1^T$$

$$P_{z(t_3,t_2,t_1)} = H_3 P_3 H_3^T + \begin{bmatrix} H_2 & H_1 \end{bmatrix} \begin{bmatrix} P_2 & P_{21} \\ P_{21}^T & P_1 \end{bmatrix} \begin{bmatrix} H_2 & H_1 \end{bmatrix}^T + DRD^T$$

    - where

$$P_{ij} \equiv E\left[\tilde{X}_i(t_i) \tilde{X}_j^T(t_j) \right]$$

- Maintaining all the possible cross-covariance terms – impractical
  - In contrast to relative pose measurements
- Therefore: either neglect, or calculate upon-demand

Explicit Calculation of Cross-covariance terms - Concept

- Algorithm:
  - More details: "Graph-based Distributed Cooperative Navigation", Indelman V., et al., ICRA 2011
  - Allows updating only one platform

- Concept:
  1. Store covariance and cross-covariance terms from all the past three-view measurement updates
  2. Express $\tilde{\boldsymbol{X}}_i(t_i)$ and $\tilde{\boldsymbol{X}}_j(t_j)$ according to the history of MP measurement updates
  3. Calculate $\boldsymbol{E}\left[\tilde{\boldsymbol{X}}_i(t_i)\tilde{\boldsymbol{X}}_j^T(t_j)\right]$ based on expressions from step 2.

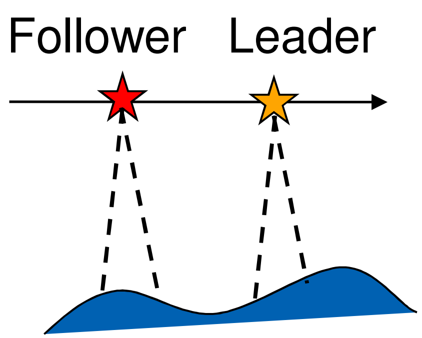
- Automation of the above for general scenarios using graph representation

Simulation Results – Leader-Follower Scenario

- 2 platforms: Leader, Follower
  - Leader is equipped with a better IMU
  - Initial navigation errors and IMU errors:

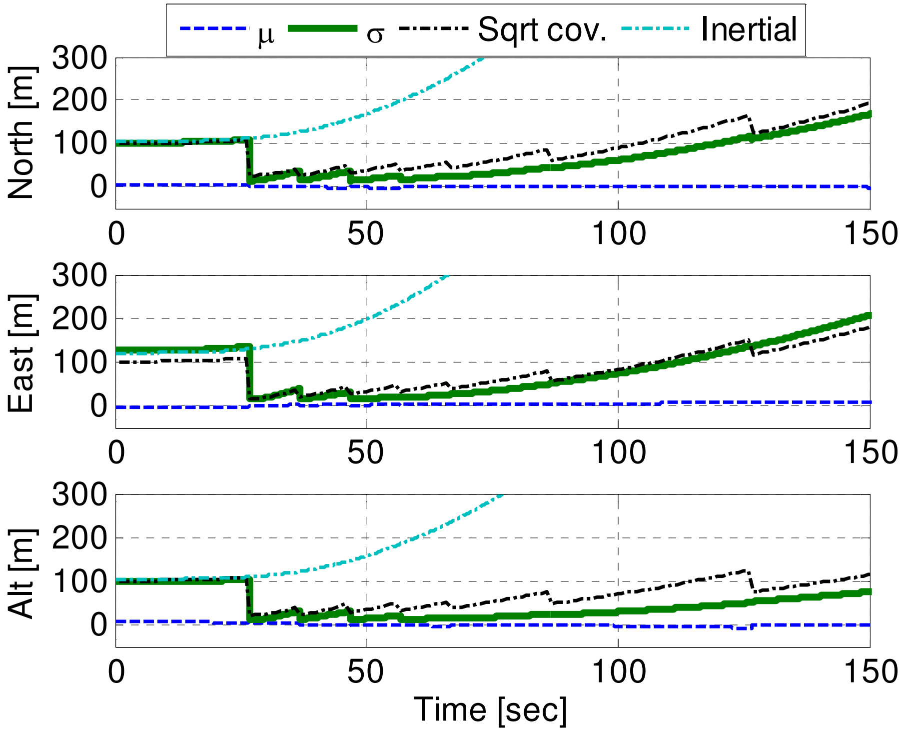
Parameter	Description	Leader	Follower	Units
$\Delta\mathbf{P}$	Initial position error (1σ)	$(10,10,10)^T$	$(100,100,100)^T$	m
$\Delta\mathbf{V}$	Initial velocity error (1σ)	$(0.1,0.1,0.1)^T$	$(0.3,0.3,0.3)^T$	m/s
$\Delta\mathbf{\Psi}$	Initial attitude error (1σ)	$(0.1,0.1,0.1)^T$	$(0.1,0.1,0.1)^T$	deg
$\mathbf{d}$	IMU drift (1σ)	$(1,1,1)^T$	$(10,10,10)^T$	deg/hr
$\mathbf{b}$	IMU bias (1σ)	$(1,1,1)^T$	$(10,10,10)^T$	mg

- Trajectory: Straight and level, north heading flight
  - Velocity: 100 m/s
  - Leader is 2000 m ahead (20 second delay)
  - Height above ground level: 2000±200m
- Follower is updated every 10 seconds
- Leader is not updated (inertial navigation)
- Synthetic imagery

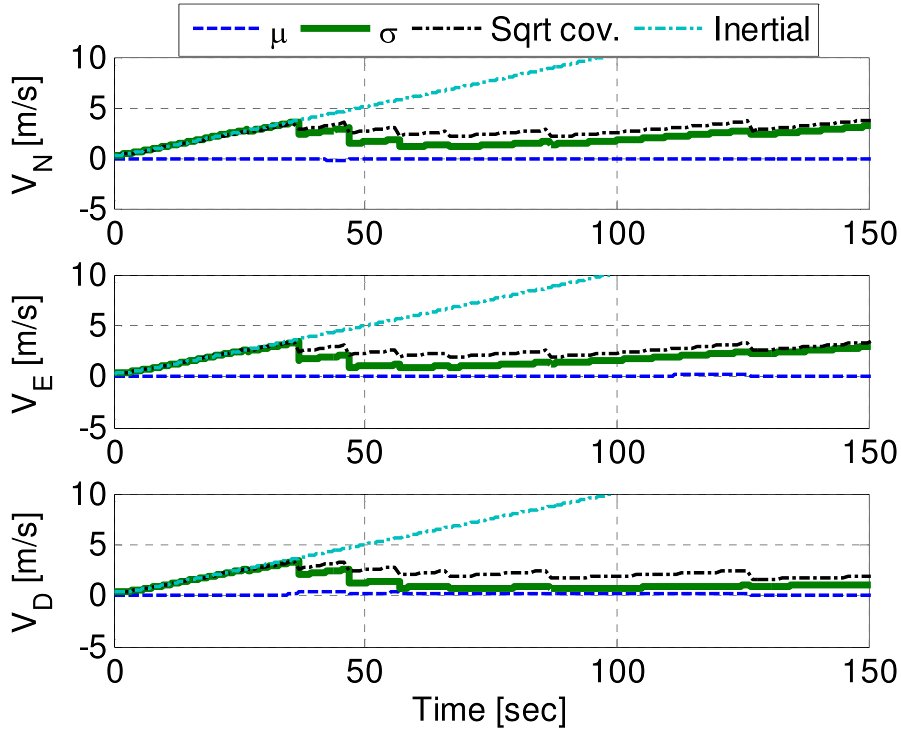


Simulation Results – Leader-Follower Scenario (cont.)

Monte Carlo results (1000 runs): Follower’s navigation errors



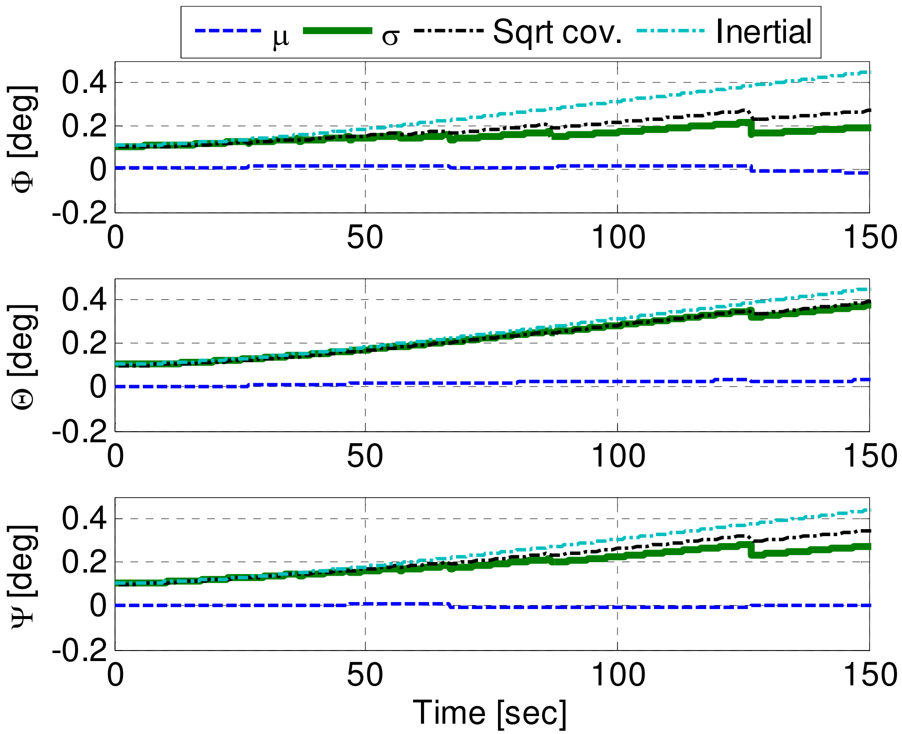
Position errors



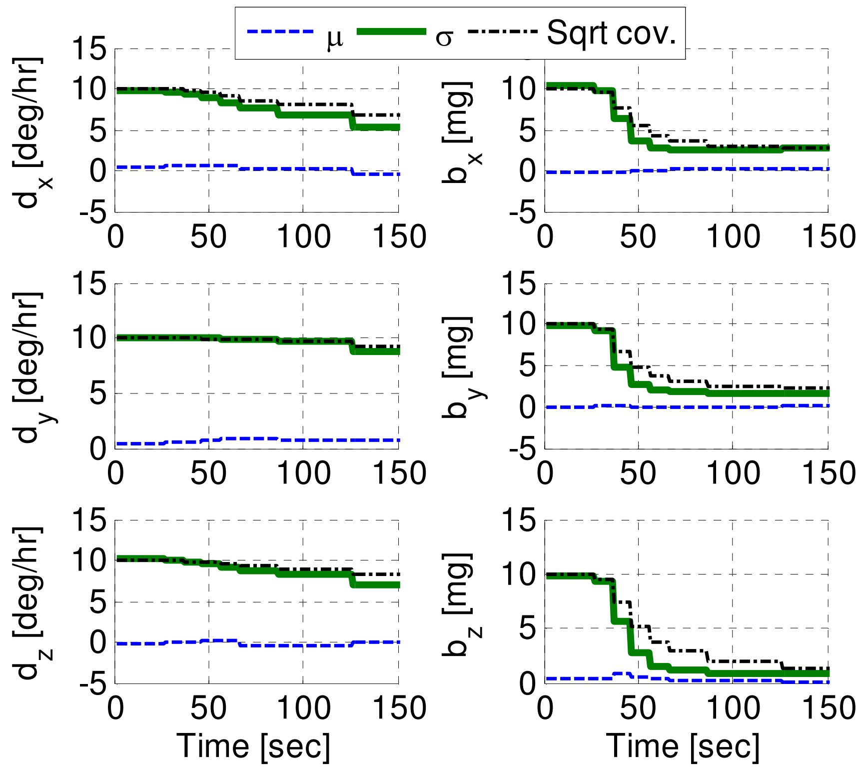
Velocity errors

Simulation Results – Leader-Follower Scenario (cont.)

Monte Carlo results (1000 runs): Follower's navigation errors



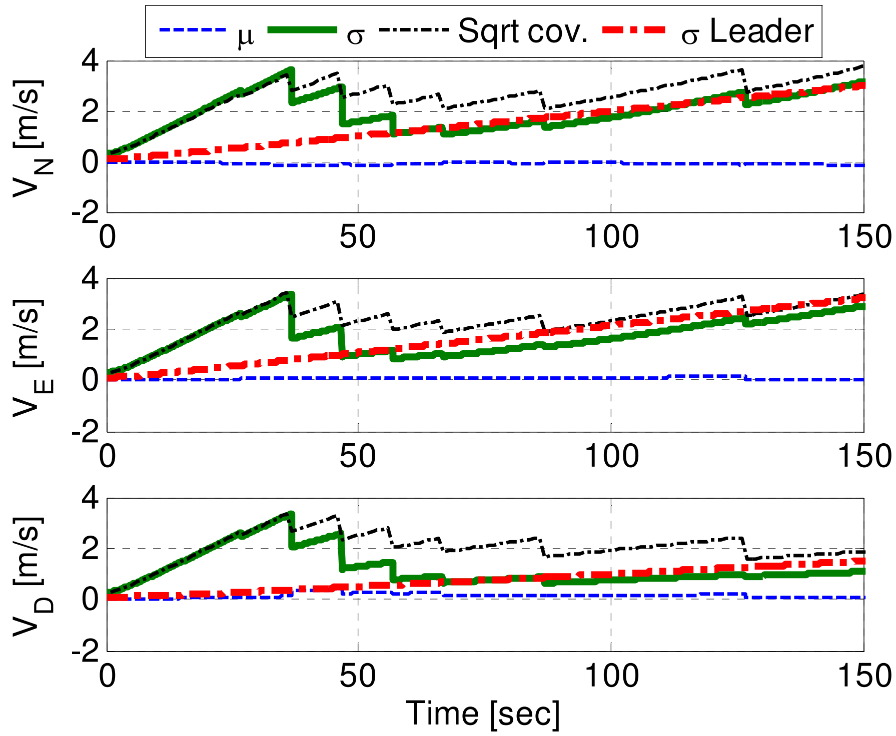
Euler angle errors



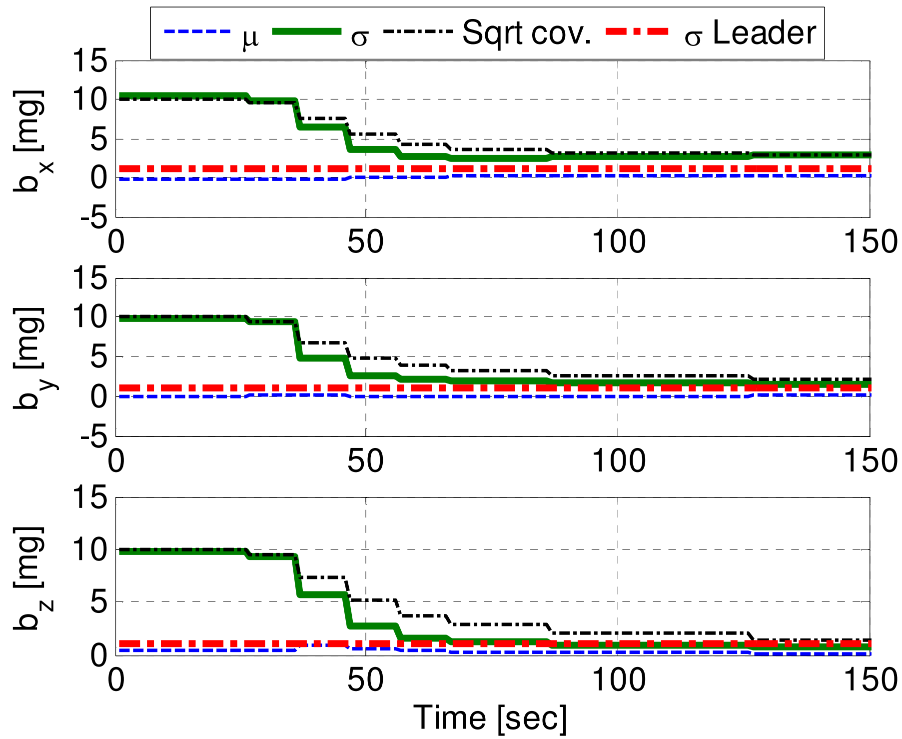
Drift and bias estimation errors

Simulation Results – Leader-Follower Scenario

Monte Carlo results (1000 runs): Follower's vs. Leader's navigation errors



Velocity errors



Bias estimation errors

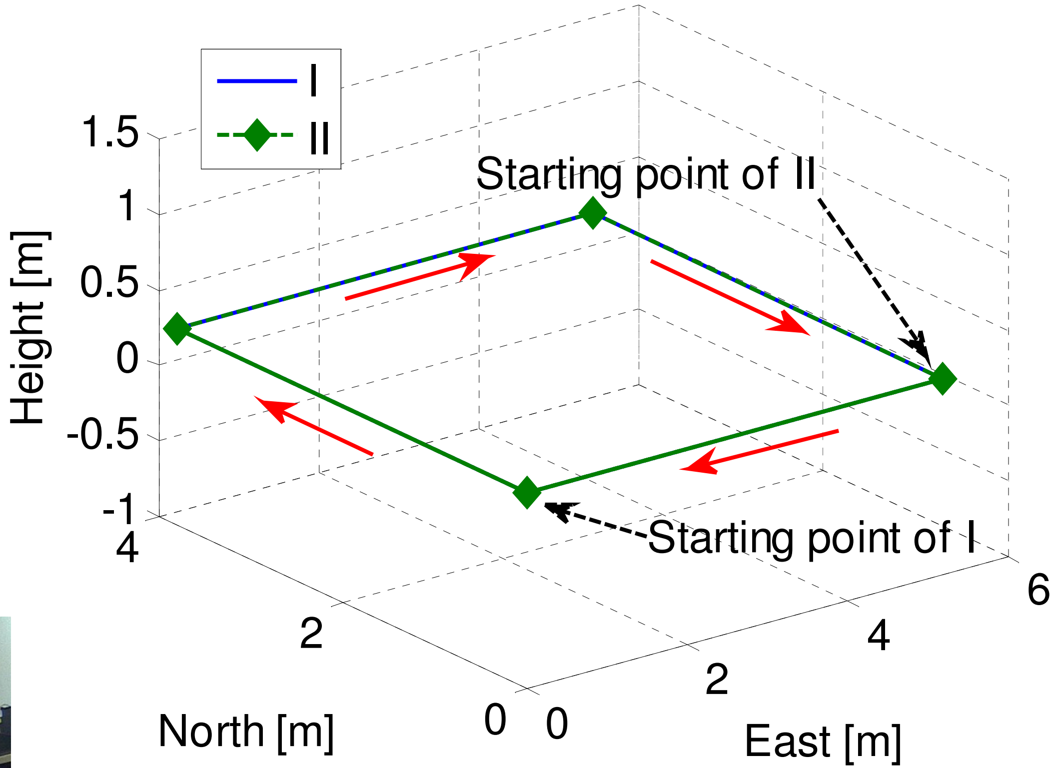
Experiment Results – Pattern Holding Scenario

- Experiment Setup
  - An IMU and a camera were mounted on top of a ground vehicle
  - IMU\INS: Xsens MTi-G
  - Camera: Axis 207MW
- IMU data and captured images were stored and synchronized
  - IMU data @ 100Hz
  - Imagery data @ 15Hz
- The method was applied in two modes:
  - ◇ Multi-platform update
  - × Self update (all images from the same platform)

Experiment Results – Pattern Holding Scenario (cont.)

- Two different trajectories
- IMU and camera were turned off in between

<u>Two</u> platforms with identical hardware (camera + IMU)



Recorded imagery

Experiment Results – Pattern Holding Scenario (cont.)

Example

Image 1

Image 2

Image 3

Matching Triplets

- Implementation details (simplified)
  - Fundamental matrix model
    - The Fundamental matrix is not required elsewhere

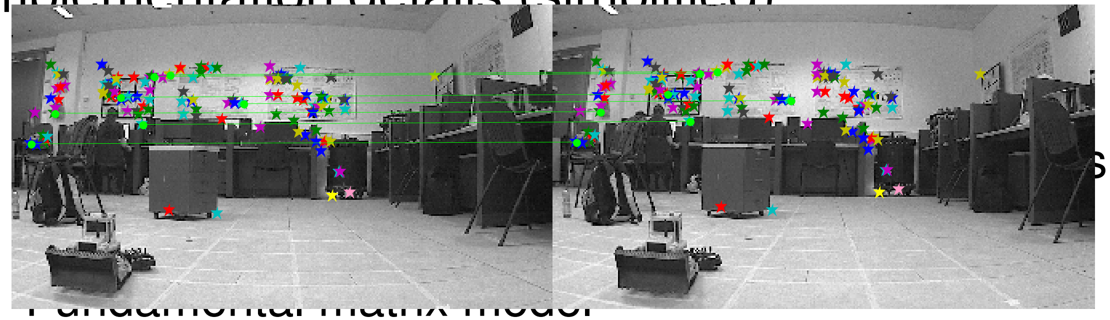


Image 1 Image 2

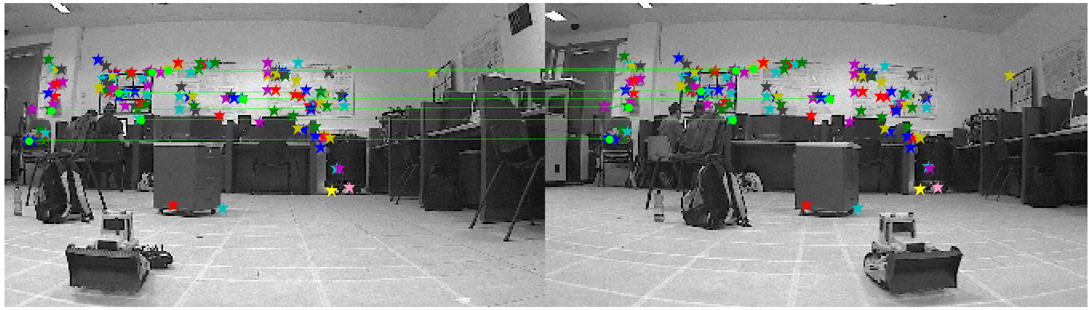


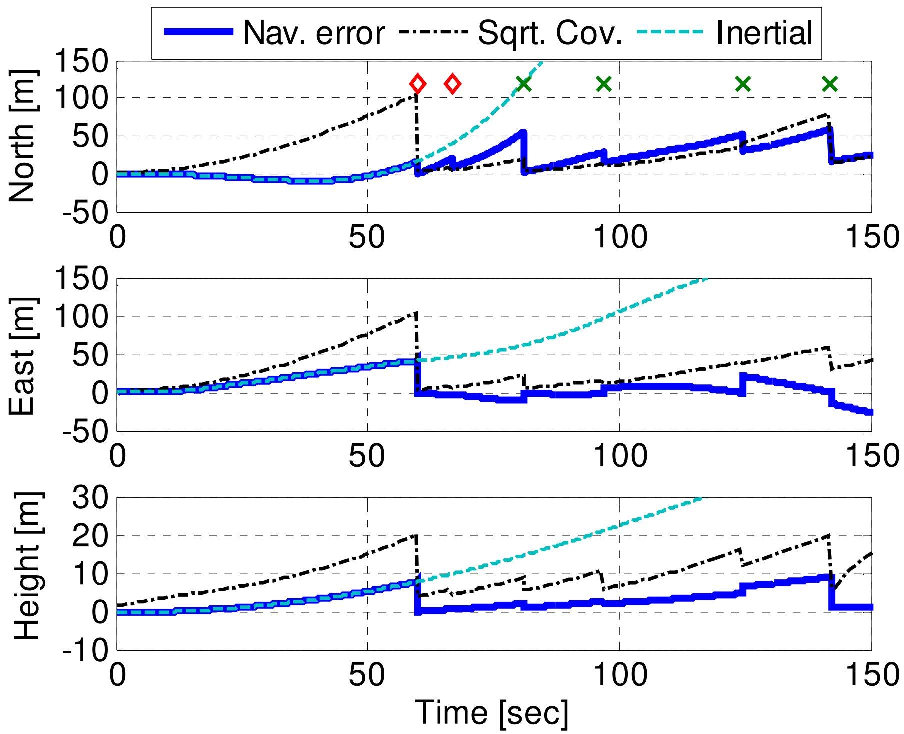
Image 2 Image 3

$$\left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{123}}, \left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2} \right\}_{i=1}^{N_{12}}, \left\{ \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{23}}$$

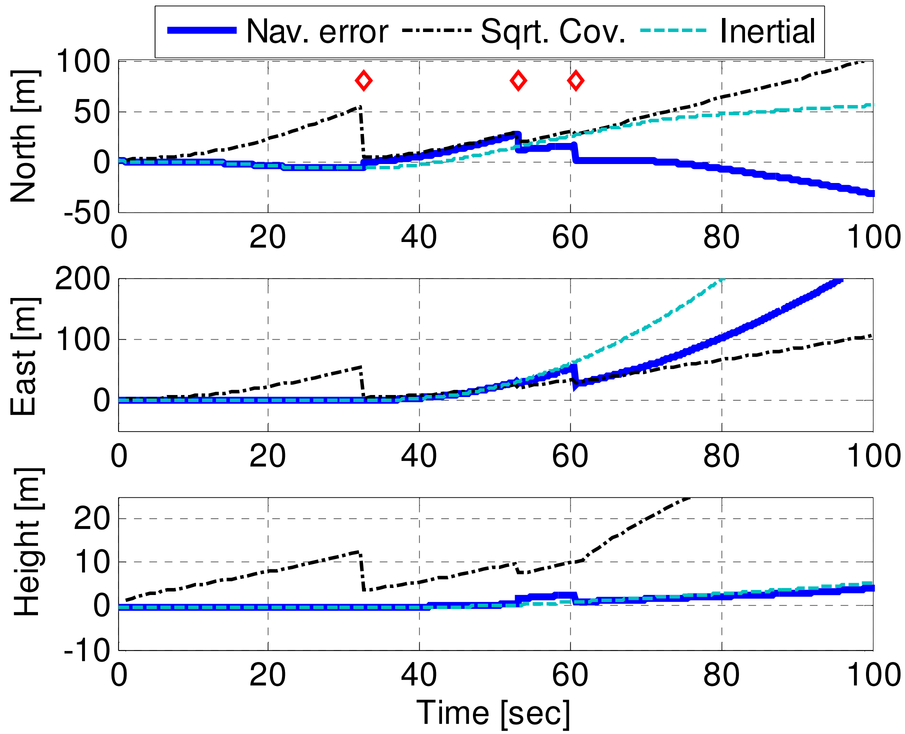
Experiment Results – Pattern Holding Scenario (cont.)

◇ Multi platform update
× Self update

Position errors



Platform I



Platform II

Conclusions

- Distributed cooperative navigation aiding
  - Three-view constraints are formulated whenever the same scene is observed by several platforms
    - The camera is no more required to be aimed towards other platforms (as in relative pose measurements)
    - Range sensor is not required
    - The views are not necessarily captured at the same time
  - Allows reduction of navigation errors in some platforms based on other platforms in the group
    - Including position and velocity errors in all axes