Data Fusion of Dual Foot-Mounted INS to Reduce the Systematic Heading Drift

G.V. Prateek*, Girisha R*, K.V.S. Hari* and Peter Händel†

prateekgv@ece.iisc.ernet.in, girishr@ece.iisc.ernet.in, hari@ece.iisc.ernet.in and ph@kth.se

http://ece.iisc.ernet.in/~ssplab

*Statistical Signal Processing Laboratory, Department of ECE
Indian Institute of Science, Bangalore, India.

†Signal Processing Lab, ACCESS Linnaeus Centre,
KTH Royal Institute of Technology, Stockholm, Sweden.

ISMS 2013, Bangkok, Thailand
29th January 2013

Funded by Department of Science and Technology, Govt. of India and VINNOVA, Sweden.
Figure: First Responder System
The different systems tracks the states of different points on the body.

There is a non-rigid relationship between the navigation points.

There is an upper limit $\gamma$ how spatially separated the systems can be.

**Figure:** Illustration of the possible placements of the subsystem in a pedestrian navigation system and the maximum spatial separation $\gamma$ between the subsystems.

Gait Cycle Phases

**Figure:** Gait cycle phases during walking and running - (blue) push-off, (green) swing, (red) heel-strike, (orange) stance

Problem Description

1. The main drawback of the existing foot-mounted ZUPT aided INS is the Systematic Heading Drift.

2. The estimated trajectories drift away from the actual path as time progresses (despite having a calibration phase).

3. One possible way these errors can be mitigated is to use foot-mounted INS on both feet such that the symmetrical modeling errors cancel out.

4. In our paper, we have assumed the maximum separation between the two foot-mounted IMUs as $\gamma$.

For more details visit http://openshoe.org
**Figure:** OpenShoe unit mounted on the left foot of a user asked to walk on a level path in the first floor of the Signal Processing Building, Indian Institute of Science, Bangalore, India.
Observations

- The duration of the phases\(^1\) of the gait cycle for walkers shows that for more than 50% of the time the foot occupies Heel-strike and Stance phase.
- And the errors are minimized when the foot is stationary. (ZUPT occurrences)

![Figure: Illustration of the motion of the feet during motion. The sphere indicates the range constraint on the spatial separation between the two feet with the feet that is stationary as the center of the sphere.](image)

---

Zero-Velocity Updates

**Figure:** A snapshot of the ZUPTs occurrences for left and right foot from time instance 2.5[s] to 5[s] for a Straight Path trajectory using the GLRT algorithm\(^2\)

Figure: Cross section of a sphere of radius $\gamma$, which is the maximum possible spatial separation between the two feet.
Let $d_{ik} = \text{norm}([\hat{x}_{ik}^i]_{1:3} - [\hat{x}_{jk}^j]_{1:3})$ represent the separation between the two navigation systems $i, j \in \{l, r\}$ and $i \neq j$, at any given instance of time $k$ where $\hat{x}_{ik}^i$ is a 9-state vector consisting of position, velocity and attitude information.

If the $i^{th}$ navigation systems is in stance phase (ZuPT is ON), the $j^{th}$ navigation system is not in stance phase and the separation between them is $d_{ik}^j > \gamma$, then the new position coordinates of the $j^{th}$ navigation system is obtained as follows

$$\hat{p}_{jk}^i = \frac{1}{d_{ik}^j} \left( (d_{ik}^j - \gamma)[\hat{x}_{ik}^i]_{1:3} + \gamma[\hat{x}_{jk}^j]_{1:3} \right).$$

Equation (1) represents the orthogonal projections of the position estimates of the foot that is in motion on to the surface of the sphere.
Existing Algorithm - Pseudo Code

Pseudo code for the algorithm without range constraint on the spatial separation of the two navigation systems \( i, j \) where \( i, j \in \{l, r\} \) and \( i \neq j \).

1: \( k \leftarrow 0 \)
2: \( P_k^i, Q_k^i, R_k^i, H_k^i \leftarrow \text{Process}\{\text{Initialize Filter}\} \)
3: \( P_k^j, Q_k^j, R_k^j, H_k^j \leftarrow \text{Process}\{\text{Initialize Filter}\} \)
4: \( \hat{x}_k^i \leftarrow \text{Process}\{\text{Initial Nav State}\} \)
5: \( \hat{x}_k^j \leftarrow \text{Process}\{\text{Initial Nav State}\} \)
6: loop
7: \( k \leftarrow k + 1 \)
8: \( \hat{x}_k^i \leftarrow \text{Process}\{\text{Nav Equations}\} \)
9: \( \hat{x}_k^j \leftarrow \text{Process}\{\text{Nav Equations}\} \)
10: \( P_k^i \leftarrow F_k^i P_{k-1}^i F_k^i T + G_k^i Q_k^i G_k^i T \)
11: \( P_k^j \leftarrow F_k^j P_{k-1}^j F_k^j T + G_k^j Q_k^j G_k^j T \)
12: for \( s \in \{l, r\} \) do
13:   if zupt\(_s^k\) is on then
14:     \( K_s^k \leftarrow P_k^s H_s^k T [H_s^k P_k^s H_s^k T + R_s^k]^{-1} \)
15:     \( \delta \hat{x}_s^k \leftarrow -K_s^k [\hat{x}_s^k]_{4:6} \)
16:     \( \hat{x}_s^k \leftarrow \text{Process}\{\text{Correct Nav States}\} \)
17:     \( P_s^k \leftarrow [I - K_s^k H_s^k] P_k^s \)
18:   end if
19: end for
20: end loop

- \( k \) \leftarrow \text{sample index.} \)
- \( P_k \) \leftarrow \text{9-state covariance matrix.} \)
- \( Q_k = \mathbb{E}\{w_1^k(w_1^k)^T\} \) is the process noise due to gyroscope and accelerometer and \( w_1^k \in \mathbb{R}^6 \).
- \( R_k = \mathbb{E}\{w_2^k(w_2^k)^T\} \) is the zupt occurrence measurement noise and \( w_2^k \in \mathbb{R}^3 \).
- \( H_k = [0_3 \ 1_3 \ 0_3] \) is the observation matrix for zupt algorithm.
- \( \hat{x}_k \) is the estimated 9-state vector containing position, velocity and attitude estimates.
- \( F_k \) and \( G_k \) define the state space model.
- \( l \) and \( r \) represent the left and right navigation system respectively.
Data Collection

- Collected the data on the first floor of Signal Processing Building, Indian Institute of Science, Bangalore, India.
- The tiles in the building were used as marker beacon to collect data in a controlled manner. Each tile of length 2 feet $\times$ 2 feet.
- The maximum stride length never exceeded 0.6096[m].

Assumptions

- The two IMUs are aligned in the same direction.
- Sampling happens at full speed (820 Hz).

Figure: A layout of the first floor of the Signal Processing Building. It is in inverted U shape. The parallel arms are 34[m] in length and the perpendicular arm is 23[m] in length.
Simulation Results - Existing Algorithm

(a) Left and Right foot trajectory for Inverted ‘L’ Path along segment AB and BC.

(b) Left and Right foot trajectory for Inverted ‘U’ Path along segment AB, BC and CD

Figure: Trajectories obtained after applying the algorithm without any range constraint on the spatial separation between two feet. Initial heading value is equal to $0^\circ$ for all data sets.
Proposed Algorithm - Sphere Limit Method - Pseudo Code

1: \( k \leftarrow 0 \)
2: \( \mathbf{P}_k^i, \mathbf{Q}_k^i, \mathbf{R}_k^i, \mathbf{H}_{k}^i, \mathbf{H}'_{k}^i \leftarrow \text{Process} \{\text{Initialize Filter}\} \)
3: \( \mathbf{P}_k^j, \mathbf{Q}_k^j, \mathbf{R}_k^j, \mathbf{H}_{k}^j, \mathbf{H}'_{k}^j \leftarrow \text{Process} \{\text{Initialize Filter}\} \)
4: \( \dot{x}_{k}^i \leftarrow \text{Process}\{\text{Initial Navigation State}\} \)
5: \( \dot{x}_{k}^j \leftarrow \text{Process}\{\text{Initial Navigation State}\} \)
6: loop
7: \( k \leftarrow k + 1 \)
8: \( \dot{x}_{k}^i \leftarrow \text{Process}\{\text{Navigation Equations}\} \)
9: \( \dot{x}_{k}^j \leftarrow \text{Process}\{\text{Navigation Equations}\} \)
10: \( \mathbf{P}_k^i \leftarrow \mathbf{F}_k^i \mathbf{P}_k^{i-1} \mathbf{F}_k^i T + \mathbf{G}_k^i \mathbf{Q}_k^i \mathbf{G}_k^i T \)
11: \( \mathbf{P}_k^j \leftarrow \mathbf{F}_k^j \mathbf{P}_k^{j-1} \mathbf{F}_k^j T + \mathbf{G}_k^j \mathbf{Q}_k^j \mathbf{G}_k^j T \)
12: for \( i, j \in \{l, r\} \) and \( i \neq j \) do
13: if zupt\(_k^j\) is on then
14: \( \mathbf{K}_{k}^j \leftarrow \mathbf{P}_k^j \mathbf{H}_{k}^j T [\mathbf{H}_{k}^j \mathbf{P}_k^j \mathbf{H}_{k}^j T + \mathbf{R}_k^j]^{-1} \)
15: \( \delta x_{k}^j \leftarrow -\mathbf{K}_{k}^j [\dot{x}_{k}^j]_{4:6} \)
16: \( \dot{x}_{k}^j \leftarrow \text{Process}\{\text{Correct Nav. States}\} \)
17: \( \mathbf{P}_k^j \leftarrow [\mathbf{I} - \mathbf{K}_{k}^j \mathbf{H}_{k}^j] \mathbf{P}_k^j \)
18: if zupt\(_k^j\) is off and \( \delta x_{k}^j > \gamma \) then
19: \( \hat{\mathbf{p}}_k^j \leftarrow \text{Process}\{\text{Correct Position}\} \)
20: \( \mathbf{K}_{k}^j \leftarrow \mathbf{P}_k^j \mathbf{H}_{k}^j T [\mathbf{H}_{k}^j \mathbf{P}_k^j \mathbf{K}_{k}^j \mathbf{H}_{k}^j T + \mathbf{R}'_{k}^j]^{-1} \)
21: \( \delta x_{k}^j \leftarrow \mathbf{K}_{k}^j (\hat{\mathbf{p}}_k^j - [\dot{x}_{k}^j]_{1:3}) \)
22: \( \dot{x}_{k}^j \leftarrow \text{Process}\{\text{Correct Nav. States}\} \)
23: \( \mathbf{P}_k^j \leftarrow [\mathbf{I} - \mathbf{K}_{k}^j \mathbf{H}_{k}^j] \mathbf{P}_k^j \)
24: end if
25: end if
26: end for
27: end loop

- \( k \) \( \leftarrow \) sample index.
- \( \mathbf{P}_k \) \( \leftarrow \) 9-state covariance matrix.
- \( \mathbf{Q}_k = \mathbb{E}\{\mathbf{w}_k^1 (\mathbf{w}_k^1)^T\} \) is the process noise due to gyroscope and accelerometer and \( \mathbf{w}_k^1 \in \mathbb{R}^6 \).
- \( \mathbf{R}_k = \mathbb{E}\{\mathbf{w}_k^2 (\mathbf{w}_k^2)^T\} \) is the zupt occurrence measurement noise and \( \mathbf{w}_k^2 \in \mathbb{R}^3 \).
- \( \mathbf{R}'_k = \mathbb{E}\{\mathbf{w}_k^3 (\mathbf{w}_k^3)^T\} \) is the position correction measurement noise and \( \mathbf{w}_k^3 \in \mathbb{R}^3 \).
- \( \mathbf{H}_k = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \) is the observation matrix for zupt algorithm.
- \( \mathbf{H}'_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \) is the observation matrix for position correction algorithm.
- \( \dot{x}_k \) is the estimated 9-state vector containing position, velocity and attitude estimates.
- \( \mathbf{F}_k \) and \( \mathbf{G}_k \) define the state space model.
- \( l \) and \( r \) represent the left and right navigation system respectively.
Simulation Results - Sphere Limit Method - $0^\circ$ Initial Heading

(a) Left and Right foot trajectory for Inverted ‘L’ Path along segment AB and BC using proposed algorithm.

(b) Left and Right foot trajectory for Inverted ‘U’ Path along segment AB, BC and CD using proposed algorithm

Figure: Trajectories obtained after applying the proposed algorithm with initial heading value equal to $0^\circ$. $\gamma = 0.6096$[m] for all the above trajectories.
Simulation Results - Sphere Limit Method - Estimated Initial Heading

(a) Left and Right foot trajectory for Inverted ‘L’ Path along segment AB and BC using proposed algorithm with initial heading for left equal to $-10^\circ$ and initial heading for right equal to $10^\circ$.

(b) Left and Right foot trajectory for Inverted ‘U’ Path along segment AB, BC and CD using proposed algorithm with initial heading for right equal to $-10^\circ$ and initial heading for left equal to $15^\circ$.

Figure: Trajectories obtained after applying the proposed algorithm with an estimate of initial heading value available before hand. $\gamma = 0.6096$[m].
A user was equipped two OpenShoe navigation system and asked to walk along a straight line for 110[m].

As reference points plates with imprints of the shoes were positioned at 0[m], 10[m], and 110[m].

Twenty trajectories with 4 different OpenShoe units were collected.

The data was the processed with the proposed method and compared with the existing methods.

**Figure:** Illustration of experimental setup.

---

Simulation Results

**Figure:** Scatter plot of end position of the two systems with and without range constraint for existing and proposed algorithm from walking along a 110[m] straight line. The scatter plots obtained are for $\gamma = 1[m]$ for all the datasets for the existing and proposed algorithm. The heading estimate is obtained at 10[m].

---

**Figure:** Two OpenShoe units without the proposed algorithm.
**Figure:** Two OpenShoe units with the proposed Sphere Limit Method.
Real-time Simulation - C++ Implementation - Comparison

(a) Without any Data Fusion algorithm  
(b) With Sphere Limit Algorithm

Figure: Final plot of the trajectory obtained for data collected on SP Building roof top, IISc Bangalore.
Thank you