CS664A Project Report - Correcting systematic heading drift using dual foot-mounted configuration

Shashwat Ranjan Chaurasia 14641
Akshay Bhatia 160079
Vasu Bansal 160776

18 July 2018

1 Dead Reckoning

In navigation, dead reckoning is the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course. Oblu, a foot-mounted sensor detects steps, computes displacement and heading of each detected step with respect to the previous one and transmits it over Wifi (or using any other data transmission technology) to the application platform for construction of the tracked path.

1.1 ZUPT(Zero Velocity Update):

In the whole phases of a stride during normal walking T , point A on the bottom of the sole is in contact with the ground for a short period of time, called still-phase. During this time, 'A' is not moving relative to the ground and the velocity of 'A' is zero.
As a result, the velocity error should be reset when zero condition is detected. In this way the accumulated errors from the accelerometer output could be effectively removed. This method is called "Zero velocity update" (ZUPT).

1.2 Why is Oblu fusion data better than a single Oblu data:

One of the drawbacks of the existing foot-mounted ZUPT-aided INS is the Systematic Heading Drift. The estimated trajectories drift away from the actual path as time progresses. Despite having a calibration phase before the walking starts, systematic heading drifts are still persistent. Another important observation to be made is that the drift obtained are symmetrical.

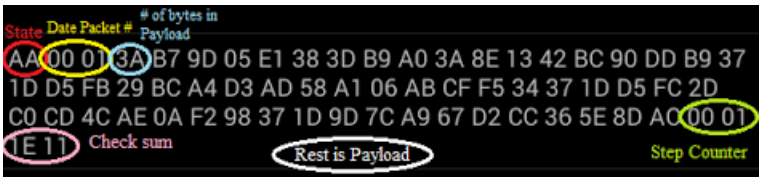
1.3 Structure of Oblu's Data Packet

Oblu sends data packet containing displacement and orientation information for each stride, at every step, to the application platform.

Oblu transmits 4 bytes acknowledgement in response to START command from the application platform.

"A0 34 00 D4"
A0: Acknowledgement state
34 = Start command state
00 D4 = Checksum

Followed by 4-byte acknowledgement, oblu sends out 64-bytes tracking data in form of packets via wireless communication.

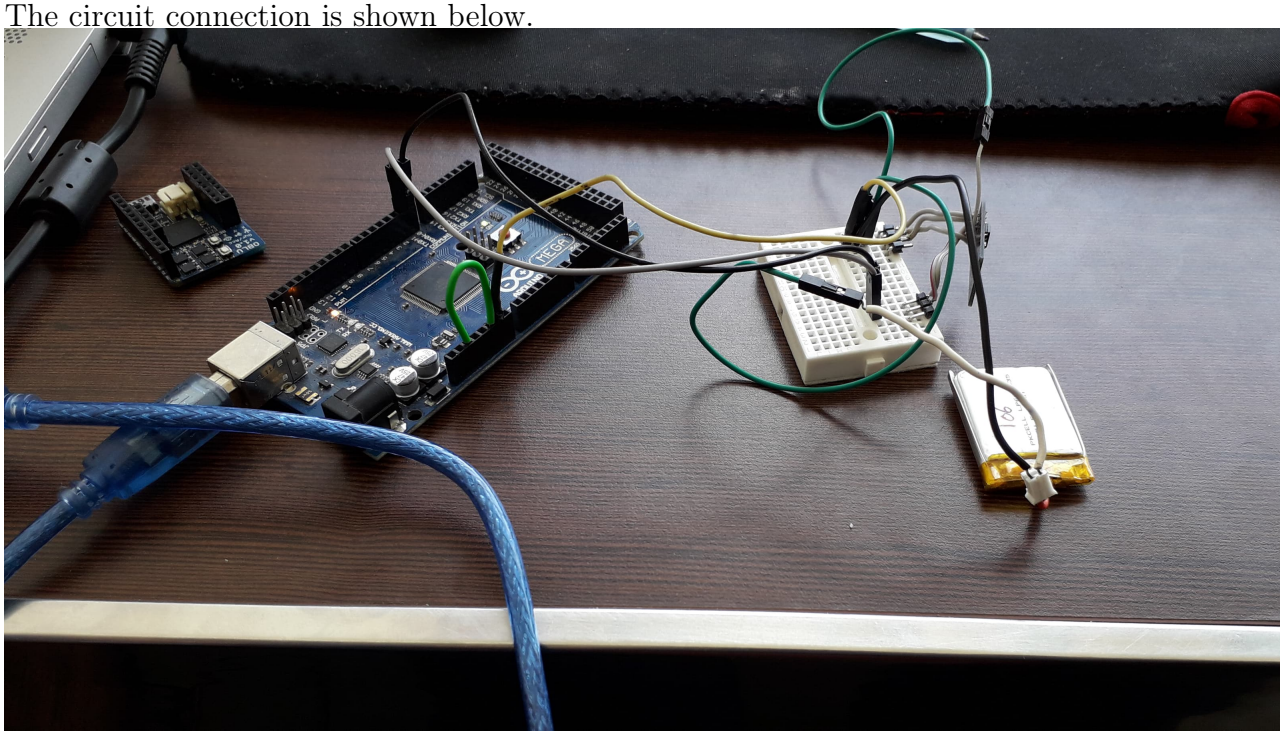


B00: state of the header
B01-B02: data packet number
B03: Number of Bytes in the payload
B04-B07: Displacement in x (Type float)
B08-B11: Displacement in y (Type float)
B12-B15: Displacement in z (Type float)
B16-B19: Change in angle around z axis (Type float)
B20-B59: 10 entries 4 bytes each of 4x4 symmetric covariance matrix
B60-B61: Step counter

B62-B63: Checksum

2 ESP8266 WiFi Module Programming

We programmed ESP8266 with the help of Arduino UNO so as to configure it for connecting with 'iitk' network.
The code was provided by instructor and bootloaded into the ESP8266's flash memory.
The circuit connection is shown below.



3 FUSION ALGORITHM

3.1 KALMAN FILTER

Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a

joint probability distribution over the variables for each timeframe.

Basic operations for a Kalman Filter include :

Predict:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}$$

Update:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$$

The predict equation uses the posterior from the previous time-step **k-1** together with the motion model to predict what the current state **x** will be. This belief is then updated via the update equation by using Bayes' theorem to combine the observed measurement **y** with the measurement model and the predicted state.

For applying Kalman filter for data fusion from Inertial Measurement Units, we need to follow the given steps:

Predict

1. $\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1}$
2. $P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^T + Q_{k-1}$

Update

3. $v_k = y_k - H_k\hat{x}_{k|k-1}$
4. $S_k = H_kP_{k|k-1}H_k^T + R_k$
5. $K_k = P_{k|k-1}H_k^TS_k^{-1}$
6. $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_kv_k$
7. $P_{k|k} = P_{k|k-1} - K_kS_kK_k^T$

Note: The posterior (magenta colored) mean is the optimal estimate of the state, and the posterior (magenta colored) covariance is the uncertainty of the estimate.

1. Predicted mean is calculated by taking the past posterior mean and multiplying it with the matrix A_{k-1}.

2. The predicted covariance is computed in a similar manner, where we multiply the past posterior covariance with

$$A_{k-1}$$

twice and add

$$Q_{k-1}$$

.

3.

$$v_k$$

is the difference, and can be seen as a representation of the new information that's gained when comparing the real measurement with the predicted measurement that we get from the measurement model.

4.

$$S_k$$

represents the predicted measurement covariance.

$$R_k$$

represents the measurement uncertainty in the measurement model

5.

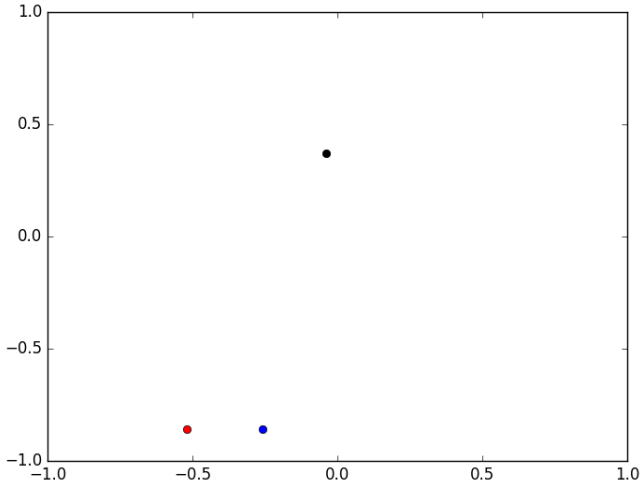
$$K_k$$

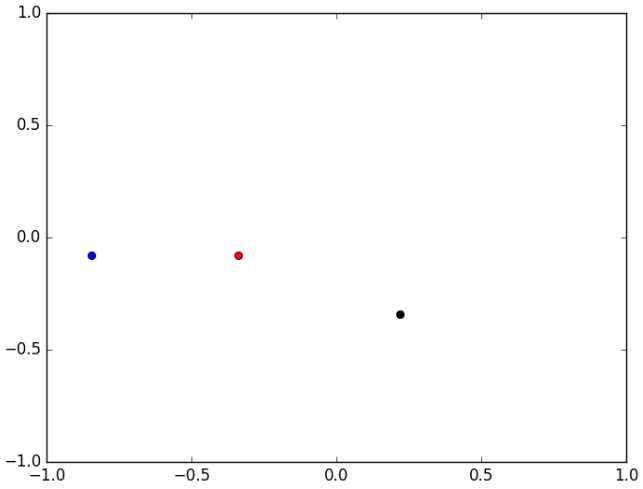
is called the Kalman gain and represents how much the predicted state and covariance should be adjusted with the new information gained from the measurement.

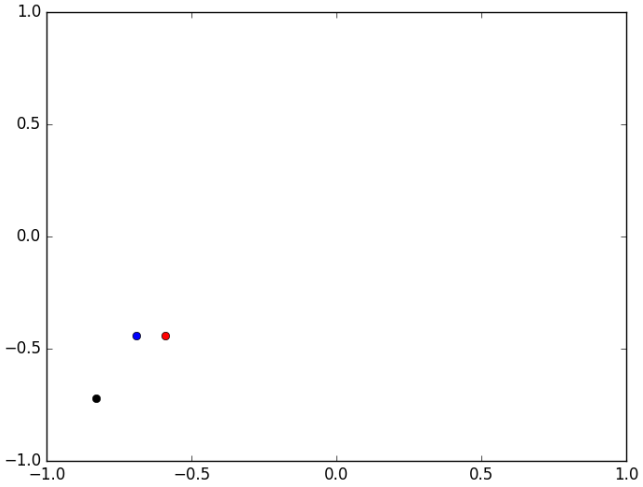
In real world applications we don't know the model parameters beforehand (Q and R), as a result, we choose R from the **covariance matrix obtained from oblu's data packet**, and finetune Q accordingly.

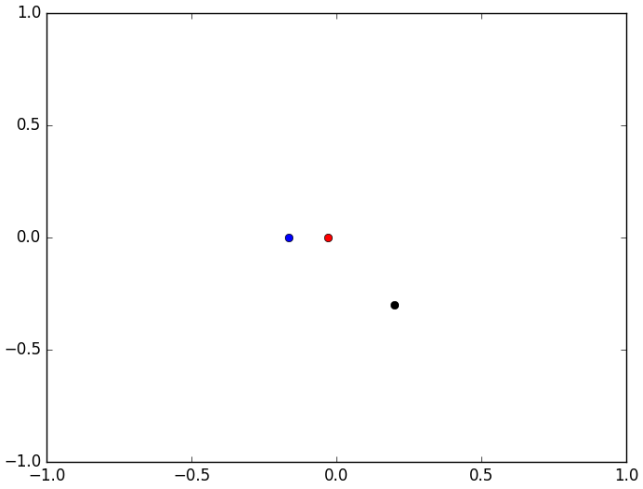
3.2 Results

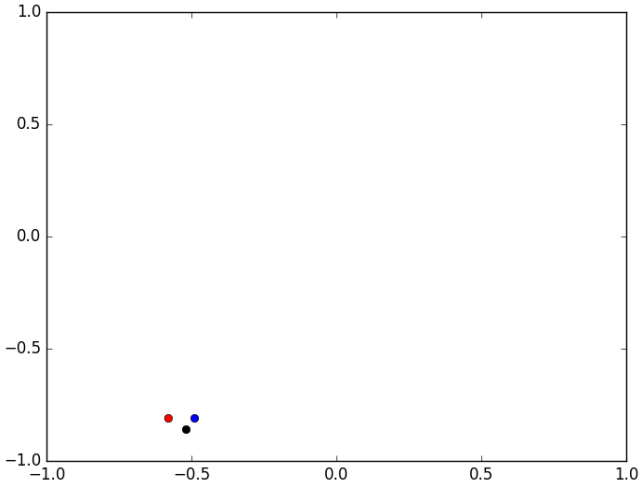
We generated random motion data points from two IMU's and applied our algorithm. Results are shown
Black - Data Point 1
Red - Data Point2
Blue - Algorithm Performance

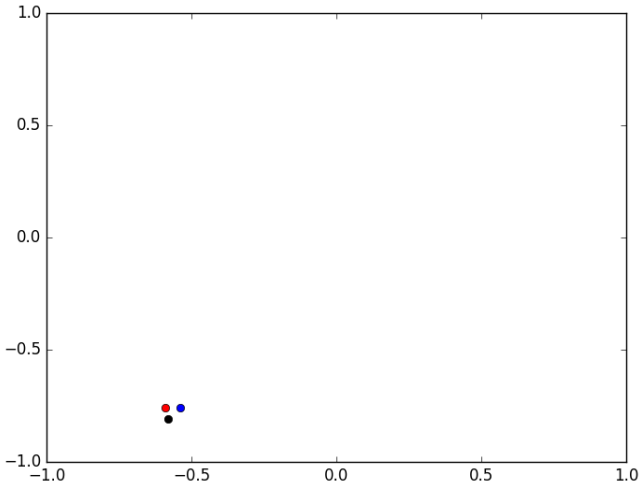


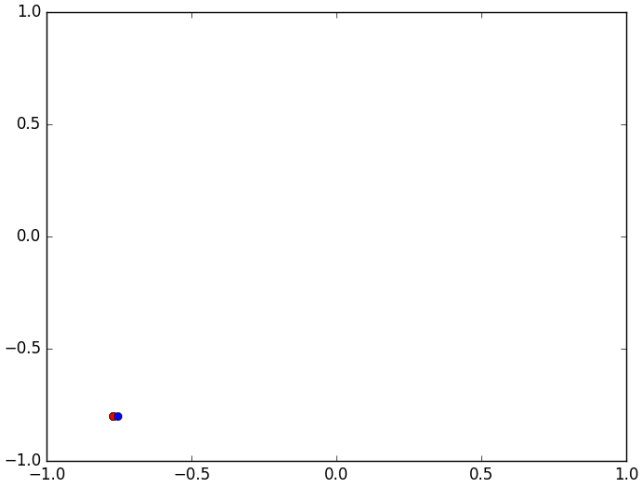

1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0





4 Connection to Cloud

We were not able to program ESP8266 to transmit to Cloud Server. In addition we were also not able to find any running instances so that we could run our algorithm on AWS servers.

Hence, this project report is incomplete.

5 References

Medium Blog
Kalman Filter Wiki
https://www.linkedin.com/pulse/pedestrian-dead-reckoning-simplified-amit-k-gupta-1/
https://ieeexplore.ieee.org/document/7847379/
Application Note, Wireless Stepwise Dead Reckoning, "PDR with Oblu"