Pedestrian Navigation System

Sai Venkatesh (IMT2017012)

May 19, 2020

1 Introduction

Pedestrian navigation systems (PNS) are used in a range of applications, from pure navigation and guidance tools, healthcare assistance systems and more generally location based services. The main goal of a PNS is to have an accurate and reliable position estimate. This can be achieved by many methods, we will look into some of them in later sections. Generally for navigation system GPS based systems are used, but in indoor environments GPS signals are very weak. So using GPS indoors will result less accurate position estimations. To overcome this IMU data is being used to estimate position indoors.

2 Dataset

An IMU (Inertial Measurement Unit) is a device that uses different types of sensors which are used to determine the acceleration, angular velocity, body orientation, etc. The data was collected from 17 IMUs which are placed in a body suit.

2.1 Placement of trackers

Out of 17 units

- 13 are Xsens MVN Motion Capture units.
  - 1 on Head
  - 2 on Shoulders
  - 2 on Upper Arms
  - 2 on Fore Arms

- 2 on Upper Legs
  - 2 on Lower Legs
  - 2 on Foots
- 4 are smart phones in different positions.
  - 1 in backpack
  - 1 in front pocket
  - 1 in back pocket
  - 1 in fixed/swinging hand

2.2 Activity Modes

The 3 activity modes which are taken into consideration are

- Standing Still
- Walking
- Running

2.3 Data Collected

The data is collected with both mobile phones and motion trackers (MTws) and it includes the following variables, 3D AccR, 3D AngVelR, 3D OriR, 3D MagR, GPS, Pressure, Mtw Position, Velocity, 3D Acc, 3D AngVel, 3D Ang.Acc, Ori, and ground truth.

2.4 Scenarios

The data has been collected in 4 scenarios.

- Scenario 1: The data is collected outdoors. It covers all 3 activity modes and all all four device modes.

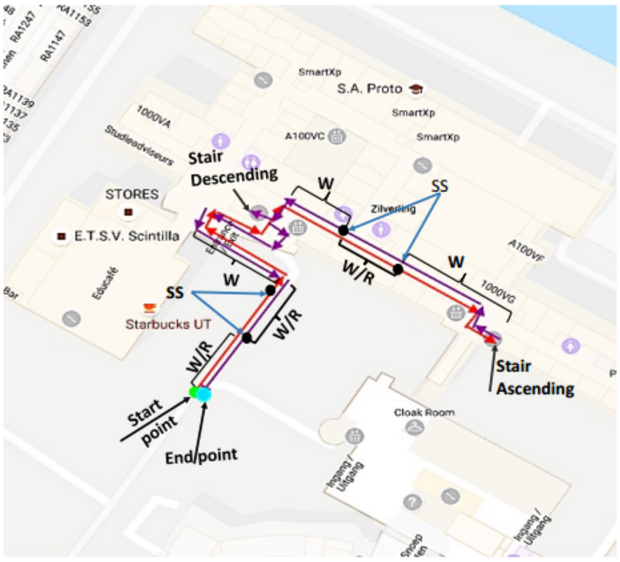
(a) Outdoor-only scenario with all classes (Case 1).

- Scenario 2: This data is collected in both indoors and outdoors while holding phone in fixed position and all other phone placements.

(b) Outdoor-Indoor scenario containing the W1 class (Case 2).

- Scenario 3: This data is collected while walking in both indoors and outdoors additionally it also includes climbing up down a stairs. This is also includes
  - phone in fixed hand and in all other positions
  - phone in swinging hand and in all other positions

- Scenario 4: It basically covers all the devise modes and all activity modes in both outdoor and indoor environment.



(c) Outdoor-Indoor scenario containing all classes. Forward and backward paths are indicated with red and purple colors, respectively (Case 3 and Case 4).

3 Patient Tracking in Hospitals (PNS Application)

4 Goal:

The Goal of this application to track and monitor the movements of visitors in a hospital. Generally their always some patients who need strict supervision, in those cases this application will be helpful for doctors/nurses to track the patient. This is also useful to notify the authorities, if a patient goes far away from his/her designated area.

5 Design:

5.1 Hardware:

This application requires a foot mounted IMU device. Whose sampling rate should be atleast 50Hz. Fortunately the provided data set includes data sampled from foot mounted IMU device.

5.2 Software:

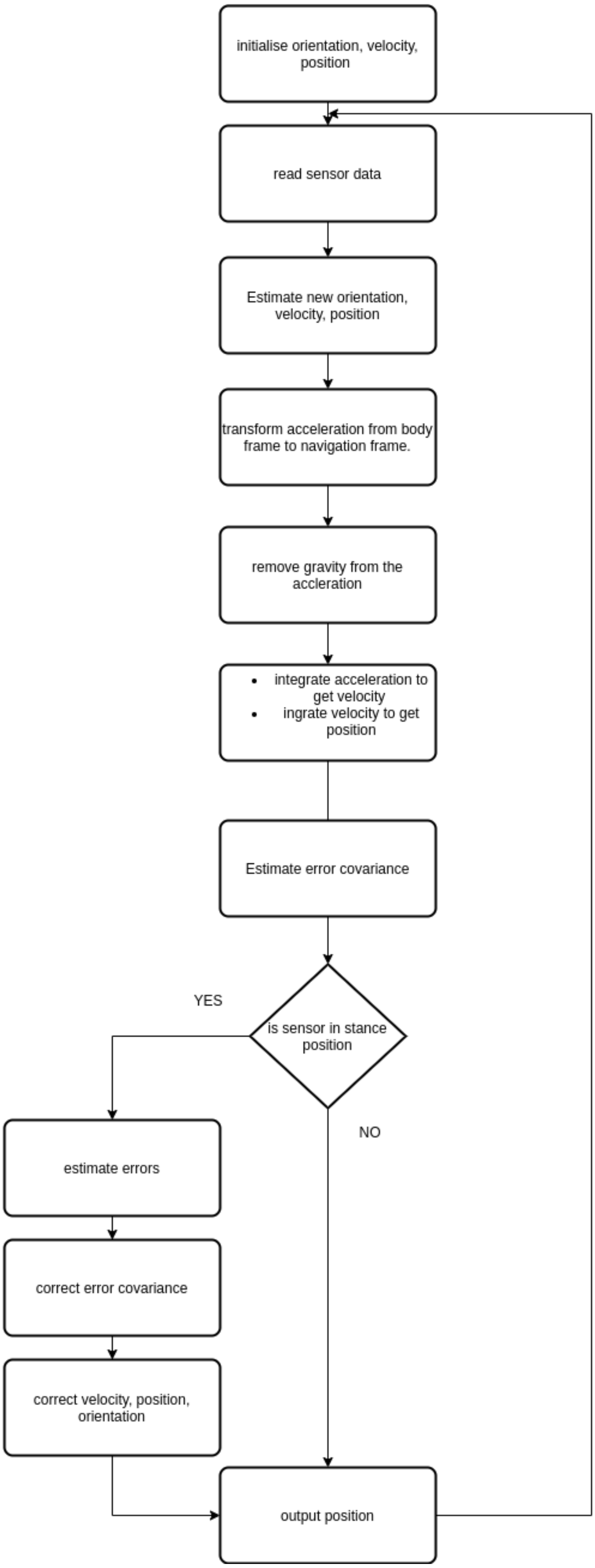
5.2.1 Existing Methods:

- **Infrastructure based Methods**: These methods require a pre-existing infrastructure for position estimation. They generally work by measuring distances for triangulation from various beacons. Generraly these methods have very less errors. These methods generally use WiFi, Zigbee, LoRa as beacons. Thier major drawback is that they cannot be used everywhere and setting up infrastructure is costly.

- **Dead-Reckoning(PDR)** is a process of estimating an object's position by tracking its movement relative to a known starting point. This method can be used to estimate the position of the IMU sensor. But this method accumulates huge errors in very small amount of time. To solve this problem many methods have been proposed, few of them are:

  - **Using magnetometer**: This method corrects the heading errors using magnetic field of the earth.
  - **ZUPT(Zero Velocity Updates)**: This method is used for foot mounted IMU units. The principle behind this method is that, while walking for each step the foot remains stationary for certain amount of time. This is called stance phase. When this happens the predicted velocity is considered as error and it is fed to Kalman Filter to get corrected bias, which will be later used to correct the predicted position, velocity, acceleration etc.
  - **Simulated ZUPT**: This method is used for hand held mobile phones. In this method they apply ZUPT between steps.
  - **Straight line detection**: This method assumes that in normal conditions a person will be walking in straight line rather than in zigzag fashion. This method changes the heading only if angular velocity is above some threshold.
  - **Use of more sensors**: Many other methods use extra sensors like ultra sound, Lidar to detect walls and other obstacles to predict the positions.
  - Sometimes combination of these methods is also used for better position estimate.

For our application we will be using ZUPT for position estimation. The following diagram will briefly explain how PDR-ZUPT works.


initialise orientation, velocity, position
read sensor data
Estimate new orientation, velocity, position
transform acceleration from body frame to navigation frame.
remove gravity from the accleration
integrate acceleration to get velocity
ingrate velocity to get position
Estimate error covariance
is sensor in stance position
YES
NO
estimate errors
correct error covariance
correct velocity, position, orientation
output position

5.3 Implementation

In this section some of the boxes in the above flow chart are explained

5.3.1 position update

- Roll, Pitch, Yaw : these are the euler's angles which describe the orientation of the sensor in navigation frame.
- rotation matrix (C): This matrix is used to change a vector from body frame(sensor) to navigation frame. This is formed using Roll, Pitch, Yaw.

$$C_k = \begin{pmatrix} cos(p)*cos(y) & (sin(r)*sin(p)*cos(y)) - (cos(r)*sin(y)) & (cos(r)*sin(p)*cos(y)) + (sin(r)*sin(y)) \\ cos(p)*sin(y) & (sin(r)*sin(p)*sin(y)) + (cos(r)*cos(y)) & (cos(r)*sin(p)*sin(y)) - (sin(r)*cos(y)) \\ -sin(p) & sin(r)*cos(p) & cos(r)*cos(p) \end{pmatrix}$$

- update equations (estimate):

$$C_k = C_{k-1} * (2 * I + \Omega_k * \delta t) * (2 * I - \Omega_k * \Delta t)^{-1}$$

where Ω is skew-symmetric angular rate matrix

$$a_k^n = (C_k + C_{k-1})/2 * a_k^b$$

$$v_k = v_{k-1} + (a_k^n + a_{k-1}^n - 2 * [0, 0, g]) * \Delta t/2$$

$$p_k = p_{k-1} + (v_k + v_{k-1})/2 * \Delta t$$

5.3.2 stance detection:

there are 3 conditions which should be satisfied when sensor is in stance position.

- The magnitude of acceleration should be between $9m/s^2$ and $11m/s^2$.

$$C1 = \begin{cases} 1 & 9 < a_k < 11 \\ 0 & \text{otherwise} \end{cases}$$

- Local variance (along a window size w) should be greater than $3m/s^2$.

$$\sigma^2_{a_k^b} = \frac{1}{2S+1} \Sigma_{q=k-s}^{k+s} (a_k^q - \bar{a_k^b})$$

$$\bar{a_k^b} = \frac{1}{2S+1}\Sigma_{q=k-s}^{k+s} a_q$$

$$C2 = \begin{cases} 1 & \sigma_{a_k^b} > 3 \\ 0 & \text{otherwise} \end{cases}$$

- Magnitude of angular acceleration should be less than 50^o.

$$C3 = \begin{cases} 1 & \omega_k < 50 \\ 0 & \text{otherwise} \end{cases}$$

- Stance phase is detected when all 3 conditions are satisfied.

$$stance = \begin{cases} 1 & C1 = 1\&C2 = 1\&C3 = 1 \\ 0 & \text{otherwise} \end{cases}$$

5.3.3 ZUPT

This applied when stance phase is detected. During the stance phase velocity of the sensor should be zero. But that never happens due to accumulation of errors. So instead of directly making velocity as zero, Kalman filter is applied. This filter takes the velocity during the stance phase as error, according to that it updates the error state vector.

- Kalman Gain:

$$K_k = P_k * H^T(HP_kH^T + R)^{-1}$$

  where P_k is state estimation co-variance matrix , H is measurement matrix, R is measurement noise co-variance matrix.

- error estimation

$$error = (error\_c, error\_p, error\_v)^T = K_k * v_k$$

- error correction

$$C_k = ((2 * I + \Omega_{error\_c,k})/(2 * I - \Omega_{error\_c,k})) * C_k$$

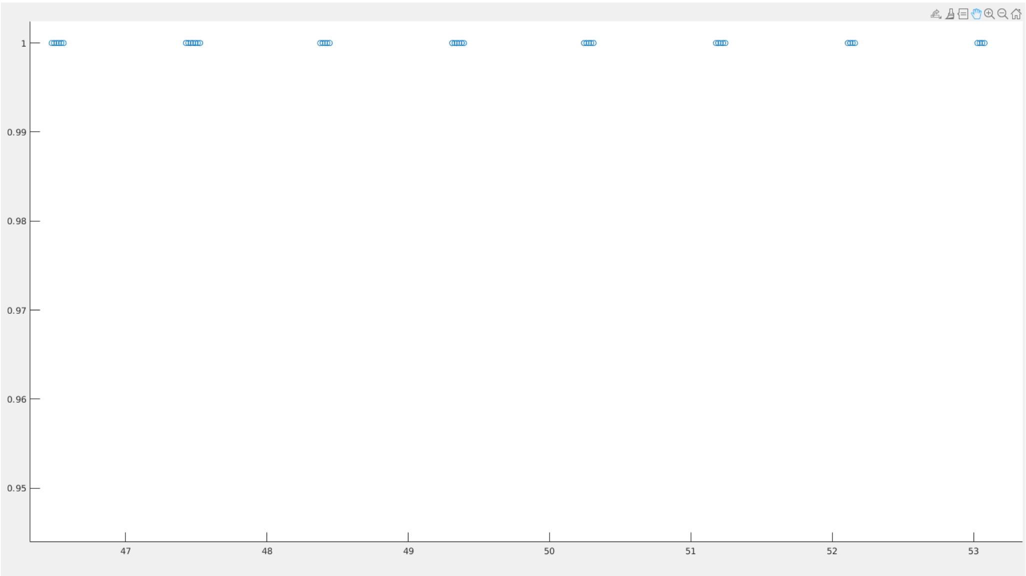
$$P_k = P_k - error\_p$$

$$V_k = V_k - error\_v$$

6 Results

6.1 Stance detection

The Blue dots in the below figure shows at what time, stance phase is occurring. As the subject was walking at a constant phase, we can see that stance phase occurs regular interval of time (which approx 1sec in this case).



stance Vs timeline

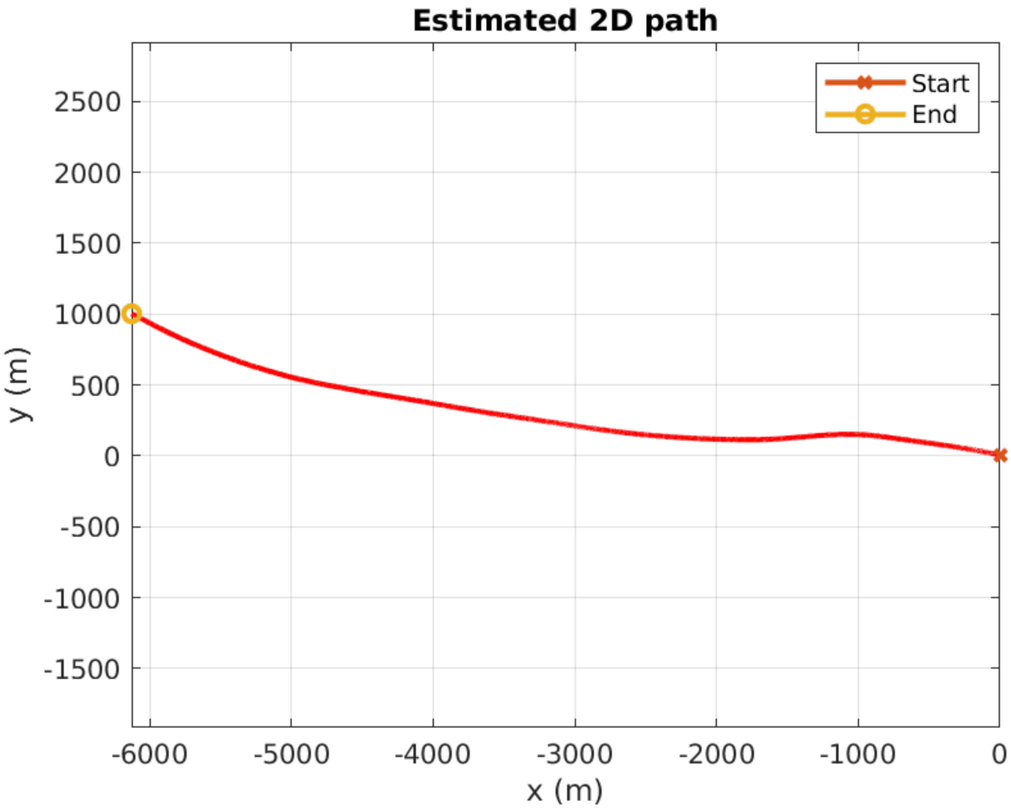
6.2 path prediction

6.3 Original Path

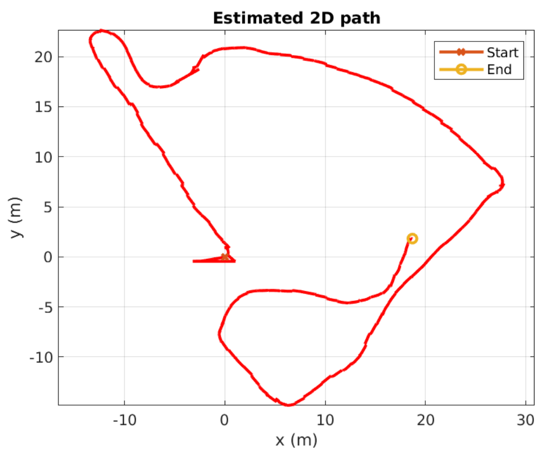
(b) Outdoor-Indoor scenario containing the W1 class (Case 2).

stance Vs timeline

6.3.1 Without ZUPT



6.3.2 With ZUPT



(b) Outdoor-Indoor scenario containing the W1 class (Case 2).

Figure 1:

7 Conclusion

We this report we saw that how ZUPT significantly increases the accuracy of the path predictions. But this can be improved further using additional sensors and using other heading correction methods explained in section 5.2.1. The main challenge faced by me was that the bias and variance of the sensor was not given in the dataset, which are essential for working of Kalman filter. So I have used a kind of brut-force method to find the variance and bias (by seeing which value is giving better predictions).

8 References

- Matlab online resources
- https://ieeexplore.ieee.org/document/8115956
- https://www.microsoft.com/en-us/research/uploads/prod/2017/12/Liqiang_Zhang_2018.pdf
- https://ieeexplore.ieee.org/document/6379901
- https://ieeexplore.ieee.org/document/6127851
- https://ieeexplore.ieee.org/document/1528431
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6210328/
- https://www.ncbi.nlm.nih.gov/pubmed/30658458
- https://www.researchgate.net/publication/224199246/