Practical Training Report

EEE3000X: Practical Training
Nicholas H Anderson - ANDNIC019
andnic019@myuct.ac.za
25 January 2019

***Abstract*—This report details the investigation of a system designed to track the performance and identify the movements of an ultimate frisbee athlete using the Adafruit BNO055 IMU and the NEO M8N Mini GPS module as part of the 6 week compulsory practical training required for EEE3000X. The project was conducted at UCT under the supervision of Dr Yunus Gaffar. To all report purists, be warned, this report isn't exactly "traditional", but I've tried to turn a helter-skelter project into a neat report.**

I. Introduction

Performance tracking in sport is becoming increasingly popular among professional sports teams across many different sporting codes. One such sport that does not utilize performance trackers is ultimate, more commonly known as ultimate Frisbee. This is a high intensity sport with sharp turns and sprints, so there may be a place for improving the margins that govern such intense movements.

The problem identified is that the existing wearable devices are relatively expensive and do not output certain ultimate specific metrics. Thus, a cheap and ultimate specific unit is needed to be developed.

Some of the performance attributes identified that are specific to ultimate include:

- The acceleration experienced when changing direction.
- The athletes maximum speed.
- The duration of sprints compared to jogging or walking.
- The path taken when "cutting". This is your running path to catch the disc.

Existing wearable devices, such as the STATSports' Viper Pod device shown in Figure 1, are capable of tracking up to 50 different performance metrics. The metrics provided by the Apex Pro Series device that are applicable to ultimate are current and max speed, total distance covered, acceleration, deceleration, the number and duration of sprints, and the athlete's position and running path on a field.

II. Methodology

Since the capabilities of the system were to be investigated and were largely unknown, the methodology was somewhat dynamic and exploratory.

A. Hardware

The Adafruit BNO055 IMU and the NEO M8N Mini GPS module were chosen as the core hardware peripherals, and the Arduino Nano was chosen as the micro controller used for testing.

Fig. 1. STATSports' Viper Pod worn by FC Barcelona's Lionel Messi [1].

The Adafruit BNO055 is a 9 DOF breakout board that can provide readings of acceleration, linear acceleration, absolute orientation, gravity and gyroscope acceleration at 100Hz.

The NEO M8N Mini GPS module can provide standard NMEA GPS data at 10Hz with the readings of interest being position, speed and bearing.

B. Implementation

The IMU and GPS module were connected to the Arduino using I2C and UART respectively, and the Arduino was connected to a laptop using the USB serial port as shown in figure 2. Matlab was used to process the data from the peripherals. The Arduino was set to read the IMU at 90Hz and the GPS module at 10Hz.

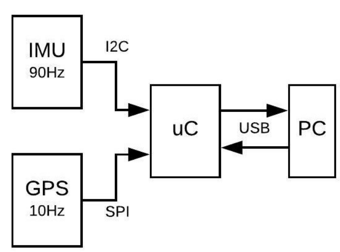

Fig. 2. Simplified block diagram of the system used for testing.

C. Experiment Procedure

The experimental procedure for investigating the capabilities of the IMU was to do a set of movements with the IMU placed on one's upper back, read a set number of samples from the peripheral under investigation to the Arduino, and write the results to the serial monitor. The data was then copied from the serial monitor to a .csv file on the PC, and Matlab was used to process the data to see if movements and or performance metrics could be identified.

The IMU's capabilities were tested by recording a range of different movements on an athlete such as walking, jogging, jumping etc. The recordings were then processed with Matlab to attempt to be able to identify those movements from the athlete in future tests.

The GPS module was tested by walking a known path and checking how accurately the corresponding GPS coordinates, velocity and bearing track the path.

III. Results

This section details the analysis of the capabilities of the hardware peripherals and signal processing methods.

A. Step Counting Algorithm

A step counting algorithm was investigated for applications where the wearable device is placed on your arm or your upper back.

1) Magnitude Algorithm with Moving Threshold: The first algorithm implement used the peaks of the magnitude of the linear accelerometer data over a certain threshold and bandpass filtering to the frequency range at which humans walk to identify steps. The results of this algorithm are shown in figure 3. The algorithm was largely successful when tested with the accelerometer positioned on the arm, but this is not the placement being investigated. On the upper back, it was highly accurate at picking up steps, with very few false positives.

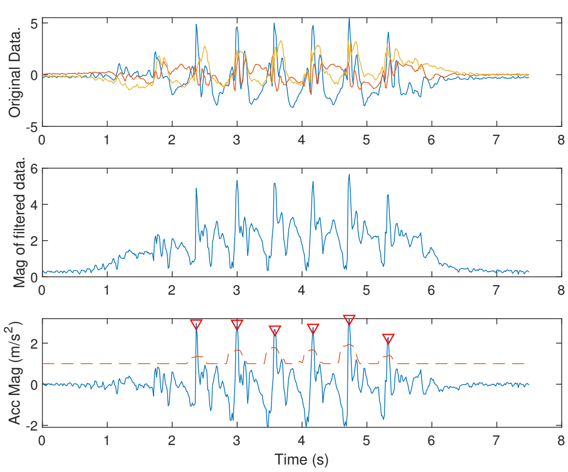

Fig. 3. Plots of each stage of the step counter algorithm, 6 correct step identifications.

2) Convolution Algorithm: The second algorithm investigated attempted to use match filters and convolution to calculate the number of steps. To achieve this, a single step was identified and used in convolution. This algorithm was unsuccessful as the frequency of steps taken (speed of walking) is prone to change.

B. Movement Metrics

The Movement metrics investigated were running, walking, jumping, accelerating and decelerating with the sensor placed on one's upper back. The first challenge encountered was identifying the signature of a step, jump etc. In order to improve this, the Euler angles and gravity readings provided by the IMU were used to create a transformation matrix that transformed all the IMU data to a known orientation such that the positive x-axis corresponded with upwards no matter what orientation the system was in, as shown in figure 4. Using this, the movements of the athlete were made more clearly distinguishable and the movement metrics could be calculated.

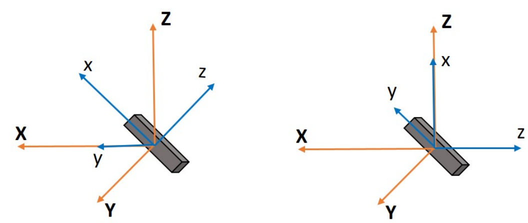

Fig. 4. Diagram showing the transformation of the IMU axis (blue) to correspond with the direction of the global axis (orange), with the positive x-axis corresponding with upwards.

The first metric investigated was jumping. The data was Transformed allowing the vertical acceleration of the system to be isolated and measured. This is shown in figure 5, which illustrates how a single jump was more distinguishable after the data was transformed.

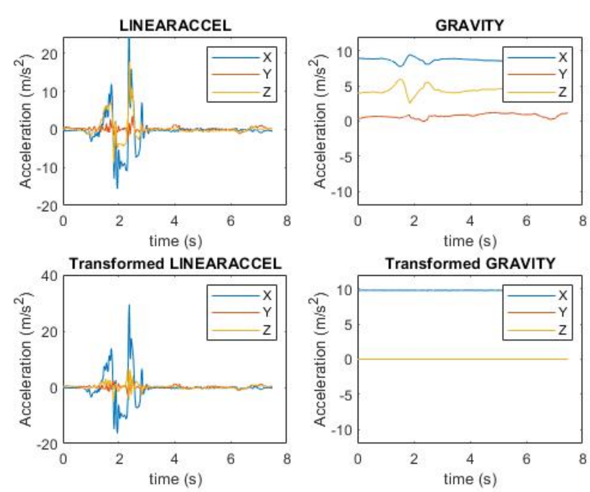

Fig. 5. The IMU data was transformed into a known orientation with gravity acting down the x-axis only to identify the vertical acceleration (x-axis) of the jump movement.

The transformed IMU data was used to isolate the acceleration along the plane of the Earth, and from this, the speed, distance and position was calculated from the transformed linear acceleration readings. Bandpass filtering was used to remove high frequency noise and accelerometer drift. This is shown in figure 6 where a 5m jog was measured and the speed, position and distance was measured. It is clear from figure 6 that the accelerometer alone is not accurate enough to measure speed and position as a result of accelerometer drift.

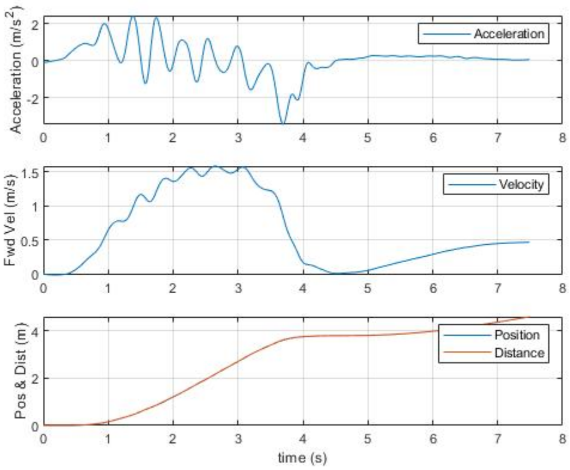

Fig. 6. Plots of linear acceleration, velocity, and position and distance of the running and stopping motion. It is evident that there is still unrectified accelerometer drift present.

C. Position on Field

Initially, the GPS was tested by walking from pole to pole on the rugby field at UCT. The results were plotted onto a map as shown in figure 7. Interestingly, the module was stationary, the GPS measured a relatively accurate reading for its position, but as soon as movement was introduced, the GPS measurement began to drift significantly. The heading and velocity readings were then used to approximate the trajectory, starting at the GPS coordinates where the actual GPS coordinates started, as shown in equation 1 and equation 2.

$$GPS_{x_{approx}} = GPS_{x_0} + v_i \cdot \Delta T \cdot \cos\theta_i \tag{1}$$

$$GPS_{y_{approx}} = GPS_{y_0} + v_i \cdot \Delta T \cdot \sin\theta_i \tag{2}$$

Where GPS_{x_0} and GPS_{y_0} are the initial measured GPS coordinates converted to meters using the Haversine equation, v_i is the velocity at a specific sample, ΔT is the sample time, and θ_i is the angle that the heading makes with the positive x-axis.Using these equations, the approximated data is shown on the right map of figure 7.

It can be observed from the left map in figure 7 that the GPS coordinates suffers from severe inaccuracies and drift. It is not remotely accurate enough to track an athletes movement on the field. However, it can be seen on the right map that the approximate calculated GPS coordinates track the line with a greater accuracy. Unfortunately, the velocity reading from the GPS was found to be noisy and inaccurate as shown in figure 8 where the module was moved across the field at approximately 5km/h and the GPS measurements ranged from 2km/h to 10km/h.

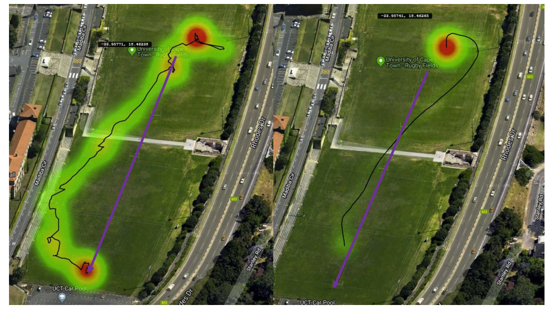

Fig. 7. GPS tests to track the path shown in purple. Left: GPS coordiates. Right: Approximation of GPS coordinates using bearing and speed.

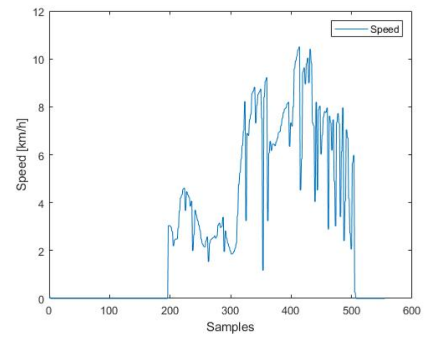

Fig. 8. GPS measurement of speed for a test where the sensor started stationary, moved at approximately 5km/h, then stopped again.

It would be useful to use the accelerometer readings from the IMU transformed onto the global coordinate system in order to approximate the bearing and velocity of the system. To transform the measurements into the global coordinate system, the quaternion measurements were converted to pitch, roll and yaw angles to create a transformation matrix, and applied to the accelerometer data as shown in equation 3.

$$A_{linear,XYZ} = R(\phi_{pitch}, \sigma_{roll}, \theta_{yaw}) \cdot A^T_{linear,xyz} \tag{3}$$

where

$$A^T_{linear,xyz} = \begin{pmatrix} a_{x,1} & a_{x,2} & \dots & a_{x,n} \\ a_{y,1} & a_{y,2} & \dots & a_{y,n} \\ a_{z,1} & a_{z,2} & \dots & a_{z,n} \end{pmatrix} \tag{4}$$

This sparked the idea of using a Kalman filter to combine the two sets of GPS coordinates (measured and approximated) to find the best approximation of the actual path, provided the path is a straight line. I spent a few days looking into Kalman filters and how to implement them with a GPS module. Unfortunately, I reached the end of my 6 week working period before I could implement the filter, but I did manage to learn how it works on a relatively high level.

IV. Conclusion

During the 6 week practical training, it was found that the task of developing a performance tracking system was not a trivial process. Problems such as sensor noise and sensor drift caused inaccurate results and consequent slow progress. Nonetheless, methods to account for non-ideal characteristics of the sensors were investigated, although not implemented due to time constraints. It is recommended that Kalman Filters, specifically the Extended Kalman Filter, are investigated to improve the GPS readings using a combination of the linear acceleration data from the IMU, and the bearing and speed data from the GPS.

V. Some More Fun Things I've learned

This section is not part of the formal report, but rather, I thought I would list some of the other interesting things I did while waiting for large software downloads or if I just wanted to look at something different for a change.

A. LaTeX

I spent quite a bit of time further familiarizing myself with LaTeX. I wrote this report in LaTeX to practice the skills I was learning (and because it is beautiful). Below is just one of the really nifty tricks I learned, which is including code snippets into a report.

```
% Looking at how effective correlation is at detecting steps.
% Method:   1) find a "step template" from the recorded data.
%           2) move the step to the origin
%           3) convolve the step with the whole data set.
%           4) Use threshold peak detection.
% ------------------------------------------------------------

sFile = 'IMU_Tests/IMU_JOG_144bpm.csv';
sFile2 = 'IMU_Tests/IMU_WALK_100bpm.csv';
disp(['Reading from ', sFile]);
IMUdata = csvread(sFile);   %Read IMU data from .csv
IMUdata2 = csvread(sFile2);
X_LINACC = IMUdata(:,2);    %Linear acceleration data
X_LINACC2 = IMUdata2(:,2);
len_step = numel(IMUdata(N_0:N_1,2)');
```

B. Verilog & FPGA

I was interested in Verilog coding and FPGA's since we touched on it in Embedded Systems II with Dr Simon Winberg, so I decided to give it a little bit of my time. I learned how to control the LEDs and switches on the Nexys 4 DDR board, as well as how to create and apply a PWM to a RGB LED.

VI. Some Quick Thank You's

I would like to say thank you to Dr. Gaffar for approving of this vacwork and supervising the work, the radar lab for allowing me to work in their space, and Willem Francois van Zyl for approaching me with the project idea and providing me with endless help along the way.

References

[1] STATSport, "Barcelona's Lionel Messi wearing a Viper pod [image]," https://www.skysports.com, 2016.