IMU Assessment on eo SwimBETTER – PILOT

Dr Neil Baker, Dr Kenneth Graham

AIM and Background

The validity/reliability/accuracy of sensors used in sport are important to allow for confidence in the quantitative changes in values measured and recorded. The use of first principle physical techniques for evaluating the values recorded by sensors allows for both tracing measures back to known standards and easy replication by others. This series of measures is designed to assess the accuracy and reliability of the IMU sensors to detect displacement (distance) in a controlled condition. Other research groups (Walker et al 2017).

Methods

Prior to data collection the **eo** SwimBETTER handsets containing the IMUs to be assessed were connected to a wooden rod which itself was affixed to a cordless electric drill

The handsets were turned on and orientated as per a normal swim session. Handsets were then located and affixed to the wooden rod using rubber restraints.

Prior to each trial the handsets were then started and the drill set at the slowest consistent rotational velocity (approx. 1Hz). Drill rotational velocity was confirmed by manually timing 10 revolutions. At the end of each trial the drill and handsets were stopped, and the data downloaded.

Two trials, of 40-50 seconds, at each of three distances from the centre of rotation were conducted. Distances used were 69, 50 and 30 centimetres.

Rotational values presented in the graphs are 0.98Hz (69cm), 1.07Hz (50cm) and 0.96Hz (30cm).

Adjustment of the drill controller on starting and stopping resulted in perturbations in the data that are seen at the beginning and end of recording.

Data Processing

Along with the time of each data sample, the raw IMU data output for *Accel(x, y, z)* and *Quaternion* values, were extracted from the handsets. Since the accelerometer data is relative to IMU chip axis (x, y, z), a vector rotation was applied to transform the Accelerometer data into the “pool” frame of reference, annotated in the image below as *fwd, lat,* and *vert.*

To ensure the the data sampling rate of the handset is uniform, the data was resampled to ensure a uniform sampling rate prior to filtering away any high frequency noise above 50Hz.

Separating the acceleration into constituent components, Accel_fwd, Accel_lat, Accel_vert, the data was numerically integrated over time and spectrally filtered to remove the DC offsets, resulting in the relative velocities *vel_fwd, vel_lat, vel_vert*.

Repeating the above process returns the relative displacement *displ_fwd, displ_lat, displ_vert*.

The plots below show the relative displacement measured across all 3 axis, along with the resultant magnitude of displacement.

As the positioning of the handset on the rotating arm changes, it can be seen the relative displacement measured by the handset changes accordingly.

Results and Discussion

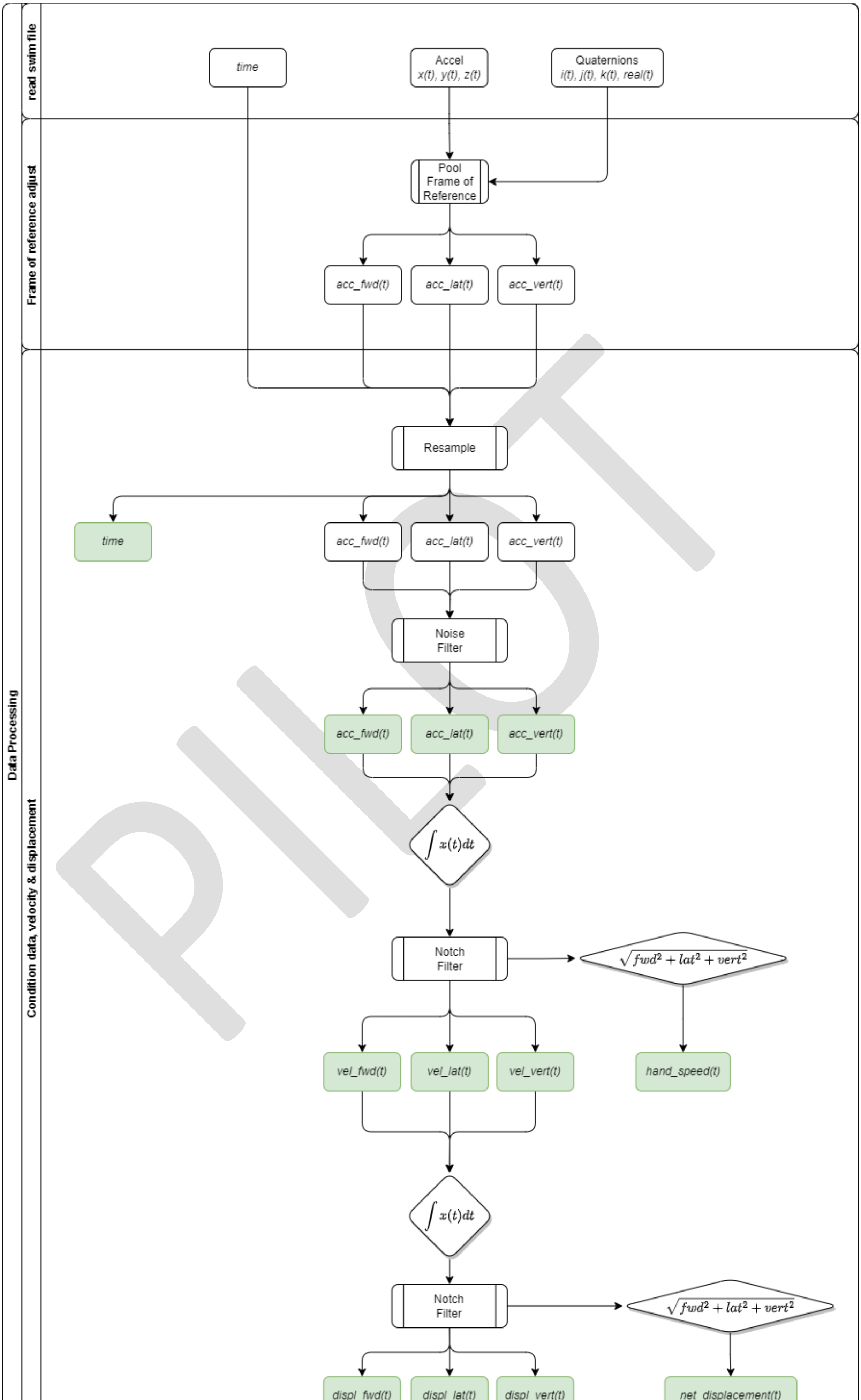
The results are presented in the attached graphs. One trial for each of the three distances are presented.

As can be seen in the data, primary motion is the coupling of forward/backward motion to vertical up/down motion, consistent with the experimental setup of the handset spinning in a circular motion oriented vertically. Some ‘wobble’, side to side motion is also evident, but minimal in magnitude. The net magnitude of displacement is shown in yellow.

The results of these trials show that the **eo** SwimBETTER IMUs reliably and accurately measure displacement.

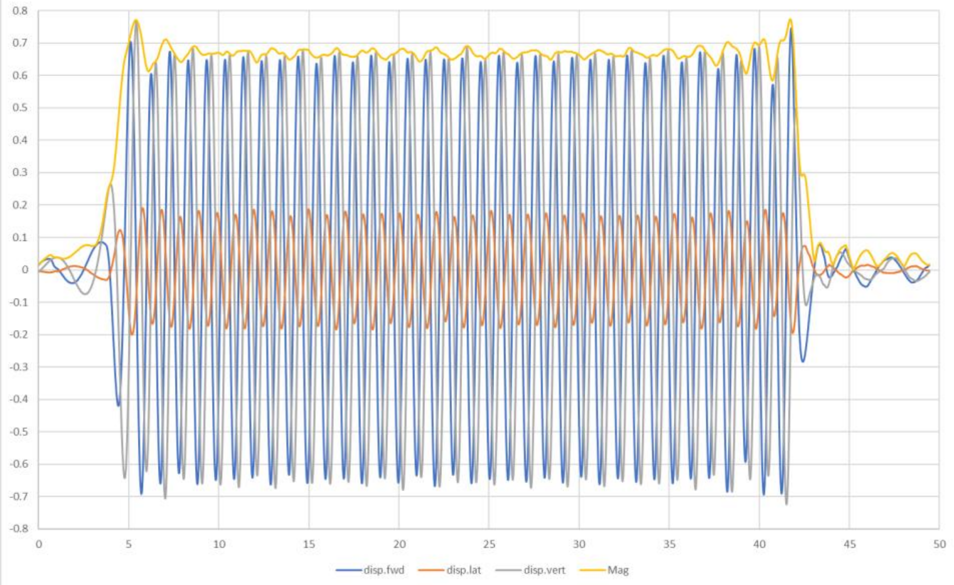
REFERENCES

Sports Biomech. 2017 Nov;16(4):485-500. doi: 10.1080/14763141.2016.1246596. Epub 2016 Oct 20.



Data Processing
read swim file
Frame of reference adjust
Condition data, velocity & displacement
time
Accel
x(t), y(t), z(t)
Quaternions
i(t), j(t), k(t), real(t)
Pool Frame of Reference
acc_fwd(t)
acc_lat(t)
acc_vert(t)
Resample
time
acc_fwd(t)
acc_lat(t)
acc_vert(t)
Noise Filter
acc_fwd(t)
acc_lat(t)
acc_vert(t)
$\int x(t)dt$
Notch Filter
$\sqrt{fwd^2 + lat^2 + vert^2}$
vel_fwd(t)
vel_lat(t)
vel_vert(t)
hand_speed(t)
$\int x(t)dt$
Notch Filter
$\sqrt{fwd^2 + lat^2 + vert^2}$
displ_fwd(t)
displ_lat(t)
displ_vert(t)
net_displacement(t)
PILOT

1. R Displacement r ≈69cm



4. Displacement r≈50cm