Inertial navigation for divers

Problem presented by

Nick Bushell

VR Technology

Executive Summary

A SCUBA diver would like to know his/her position underwater, relative to the dive start position. GPS is not an option as the signals do not travel underwater. Transponder/pinger systems have already been developed, but these are expensive and require configuration by the boat operator and the diver.

A simple standalone affordable dead-reckoning system is needed, but it is inherently difficult to determine the diver's movements. The system needs to be very cheap, so that price is not a barrier to its deployment. The system needs to be small.

The Study Group investigated the possibility of such a system making use of cheaply available accelerometer, gyroscope and magnetic field sensors, similar to those included in many modern smart phones.

The Study Group captured data using an Android device strapped to a skateboard to simulate the type of movements a diver might make underwater. Different modes of movement were evident from filtered versions of the sensor outputs, so indicating that a pedometry based solution ought to be feasible. The Study Group then formulated and investigated the feasibility of a generic dead-reckoning system. Although sensors provide more data than is strictly necessary, significant errors arise from imperfect calibration and from noise for which the Study Group derived estimates of the resulting drift in position over time. The accuracy of practical numerical integration schemes in the context of rotating frames was investigated, and a Kalman filter was used to reduce error in the orientation data by combining accelerometer and gyroscopic data.

Version 1.1
July 18, 2012
iii+21 pages

Report author

Lorcán Mac Manus and Melvin Brown (Industrial Mathematics KTN)

Contributors

Alexander Lewis (University of Oxford)
Andrey Melnik (University of Oxford)
Angel Manuel Ramos (Universidad Complutense de Madrid)
Charles Brett (University of Warwick)
David Wood (University of Warwick)
Ingrid von Glehn (University of Oxford)
Jeff Dewynne (University of Oxford)
Klemens Katterbauer (Department of Electrical Engineering, King Abdullah University of Science and Technology)
Lorenz Berger (University of Oxford)
Marcin Bulkowski (Polish Academy of Sciences)
Paul Roberts (University of Oxford)
Robert Whittaker (University of East Anglia)
Scott Grandison (UEA, Computing Sciences)
Stephen O'Keeffe (University of Oxford)

ESGI85 was jointly organised by
The University of East Anglia
The Knowledge Transfer Network for Industrial Mathematics

and was supported by
The Engineering and Physical Sciences Research Council

Contents

1 Introduction

1.1 Background and scope

(1.1.1) A SCUBA diver would like to know his/her position underwater, relative to the dive start position. This requires a gyro system optimised for underwater use, utilising low cost hardware sensor inputs. GPS is not an option as the signals do not travel underwater. Transponder/pinger systems have already been developed, but these are expensive and require configuration by the boat operator and the diver.

(1.1.2) A simple standalone affordable dead-reckoning system is needed, but it is inherently tricky to determine the diver's movements. The system needs to be very cheap, so that price is not a barrier to its deployment. The system needs to be small.

(1.1.3) The study group sought to address the following questions: how should the position be calculated and what accuracy can be obtained? This led to considerations of which characteristics of motion are important.

(1.1.4) A further consideration was to think about the likely sources of errors and deviations from idealised behaviour. Some technical information on these device outputs is provided by Anscamobile, who market the Corona SDK for mobile platform development [2]. Some basic information on MEMS type accelerometers, including the likely source of error is covered by a Texas Instruments briefing note [3].

(1.1.5) As a practical focus for the study group, a modest land–based solution was implemented in an android app. As accelerometers and compasses are standard in so many phone/tablet models, this was deemed to be a suitable test platform for practical ideas.

2 Understanding the available data

2.1 Feasibility of using data from an Android device

(2.1.1) To establish the feasibility of the proposed methods, an Android device was pushed along the floor on a skateboard and recorded data for the acceleration $\boldsymbol{a}$ and angular velocity $\boldsymbol{\omega}$. This data was extracted from the device and was used as input data to solve equations (1) to (3) numerically using MATLAB. Results are shown in figures 1 and 2, and provide an encouraging demonstration of the feasibility of the methods.

(2.1.2) The device was placed flat on the skateboard, with the z axis aligned approximately downwards. The movement along x and y axes is shown, with initial condition (0,0). The units are normalised, and we consider

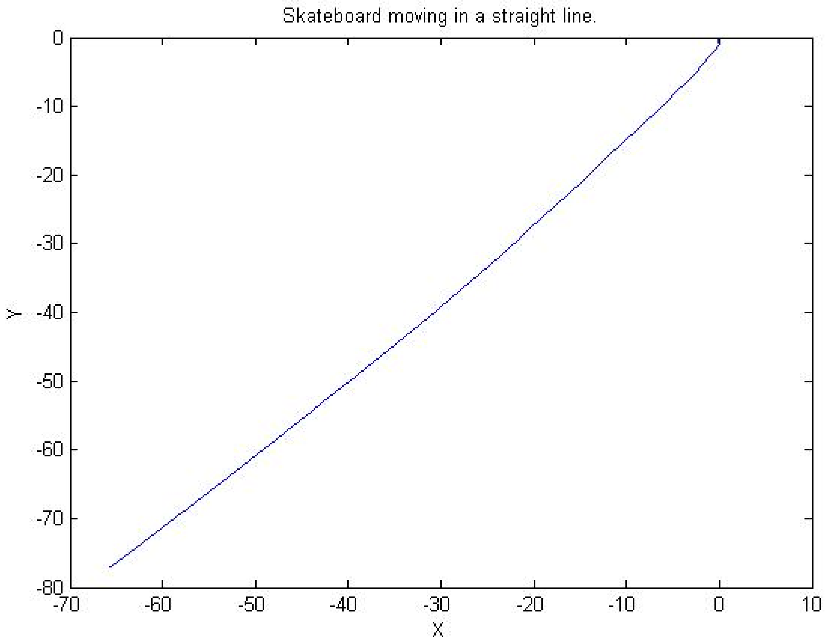

Figure 1: Skateboard moving approximately in a straight line.

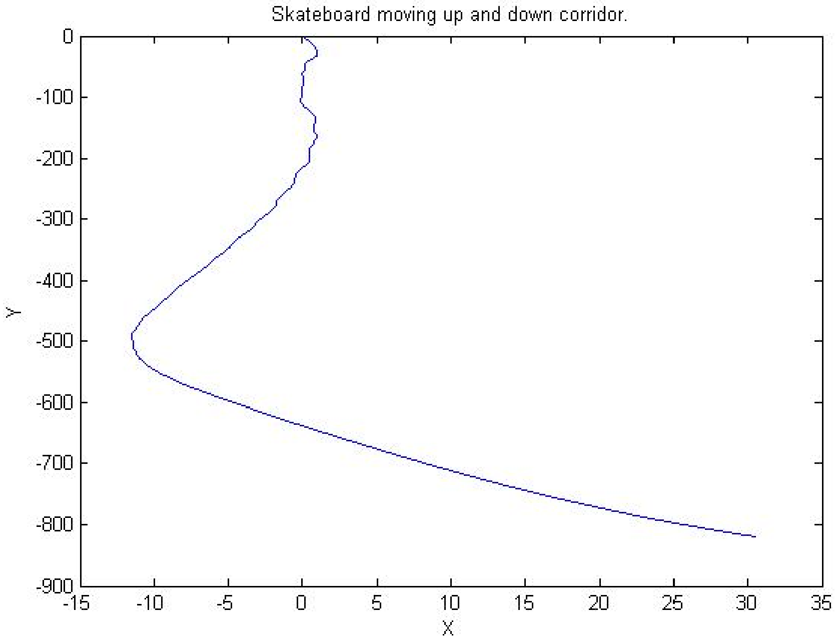

Figure 2: Skateboard moving along a corridor.

only qualitative features of the calculation. In Figure 2, x and y are approximately aligned to the corridor width and length respectively, while in Figure 1, the device is placed approximately at a 45 degree angle to the corridor length. Both figures show roughly straight lines as expected, with Figure 2 demonstrating more deviation after a longer period of time.

2.2 Movement scenarios and data captured

(2.2.1) In order to generate realistic data for analysis, an iPhone app was used to log data streams from the iPhone's inertial sensors, namely the accelerometers, gyroscopes and magnetometer. A number of different motion scenarios were used, with the device attached to a skateboard; the device hand–held while walking; and the device at rest on the window sill of the building (in order to estimate background noise in the sensors).

(2.2.2) With the iPhone attached to the skateboard, two sub-scenarios were used: the skateboard moved with an obvious push and allowed to decelerate noticeably between pushing actions and the skateboard moved at a near constant speed.

(2.2.3) In total, three data sets were captured for the skateboard with each of a pushing motion, the walking experiment, and the iPhone at rest. Two data sets were captured for the skateboard at a (near) constant speed.

2.3 Making sense of the data

(2.3.1) The strongest signal in the captured accelerometer data was that due to gravity. This signal may play an important role in determining an absolute reference frame, but for the study group we considered it a nuisance signal. A bandpass filter with a characteristic shown in Figure 3 was applied to the three accelerometer axis signals. The characteristics of this filter were that the passband be between 0.5Hz and 4Hz. This was chosen on the basis that anything below 0.5Hz was likely to be dominated by the gravitational signal and that anything above 4Hz might well be noise.

(2.3.2) An example application of this filter to one of the skateboard pushing datasets resulted in the axial accelerations in Figure 4. Compass heading information from the magnetometer is included in this figure also (although this signal has not been filtered). As can be seen, each of the accelerometer readings is centred around zero; this is to be expected as the filter has suppressed the constant signals in the accelerometer readings. One can also see impulses in the acceleration signals, as might be expected by the nature of the scenario (the skateboard being pushed and allowed to slow down noticeably between pushes). The acceleration peaks in both the positive and negative directions indicate maximum acceleration and deceleration.

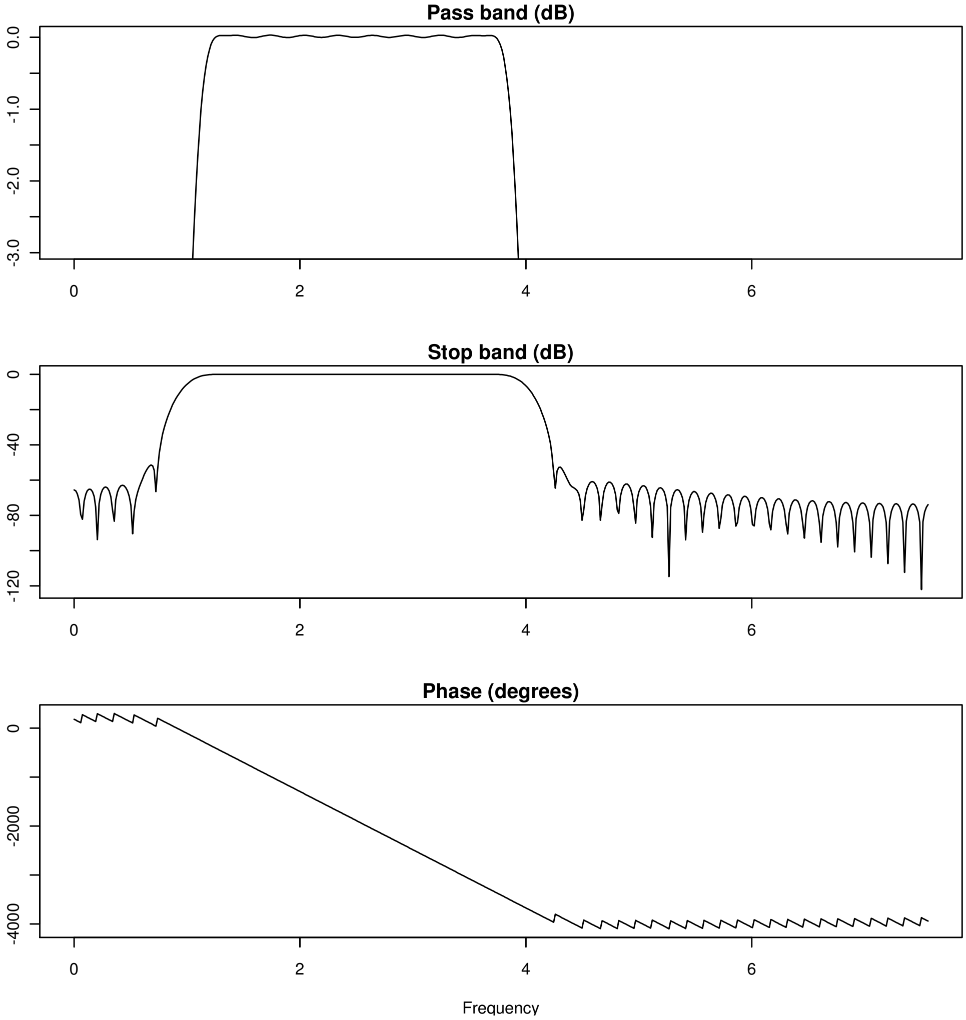

Figure 3: The pass band response, stopband response and phase response of the bandpass filter used to suppress gravity and higher frequency noise. Frequency is in Hz.

(2.3.3) Next we considered the magnitude of the accelerations, i.e.

$$|\boldsymbol{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}.$$

The results of this are in Figure 5 for all 11 datasets captured, although their number is obscured by noise in the data. Colour coding is used to distinguish static (green), walking (red), skateboard being pushed (blue) and skateboard at a near-constant speed (black).

(2.3.4) In order to see these magnitudes more clearly, a low pass filter with a cut-off of around 0.5Hz was applied to the magnitude outputs. The response of this filter is shown in Figure 6. The resulting smoothed magnitudes are shown in Figure 7.

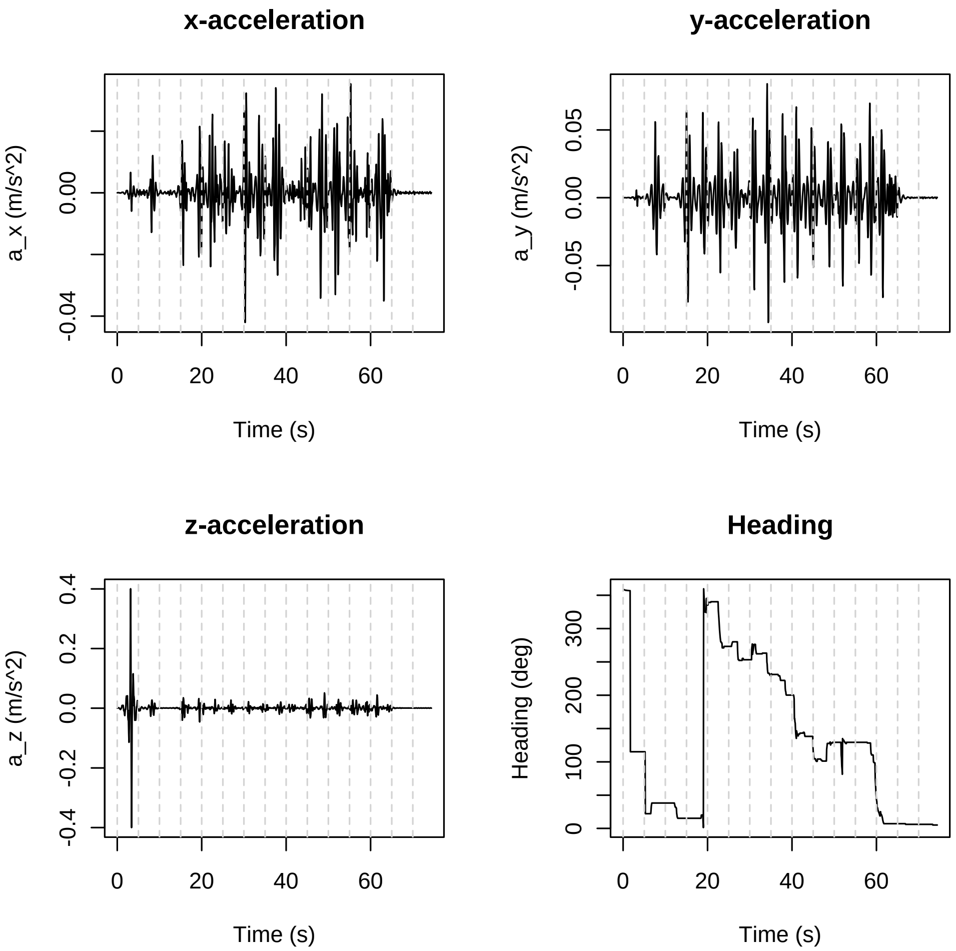

Figure 4: Accelerometer readings in the x, y and z directions, filtered with the bandpass filtered described in Figure 3. Note the different scale on the z-acceleration component. Heading information from the magnetometer is also included, showing how directional information changed as the orientation of the device changed throughout the data capture experiment.

2.4 Counter-intuitive filtering artefacts

(2.4.1) The filtering adds some artefacts that may make the acceleration plots of Figures 5 and 7 seem incorrect. In particular, the acceleration does not hit the baseline value (i.e. the ‘at rest’ acceleration magnitude corresponding to $\boldsymbol{a} = 0$) periodically, as might be expected for the push scenarios[1]. This is due to the removal of high-frequency terms in the filtering. Refer back to Figure 4 to see that the accelerations do indeed oscillate between positive and negative values.

[1] Bear in mind that pushing the skateboard implies that $\boldsymbol{a}$ must periodically change sign.

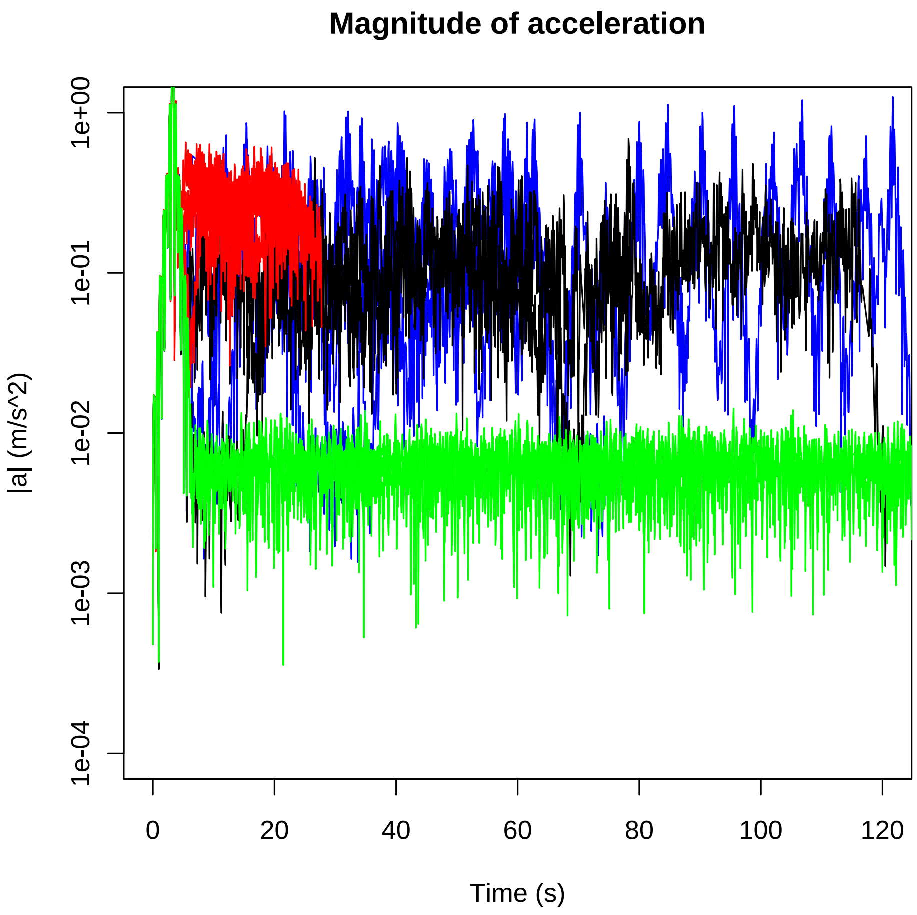

Figure 5: Accelerometer magnitudes for each of the 11 data sets captured (overlayed and partially obscured in noise), following prefiltering of axial components using the bandpass filter of Figure 3. Colour coding is used to distinguish static (green), walking (red), skateboard being pushed (blue) and skateboard at a near-constant speed (black).

(2.4.2) Both the lowpass and bandpass filters are 100 point FIR filters, designed using R's `fir1` command. The sampling rate of the iPhone accelerometers is around 15Hz. Therefore it can be readily seen that the extended transient spikes in the first 10-15 seconds are caused by these filter settle times.

2.5 Towards a V1.0 dead-reckoning system

(2.5.1) Each of the transportation modes of Figure 7 can be distinguished by eye. In particular, there is a clear oscillation in the blue signals corresponding to the pushing action on the skateboard (blue); the near-constant scenario on the skateboard (black) is not quite as constant as walking (red),

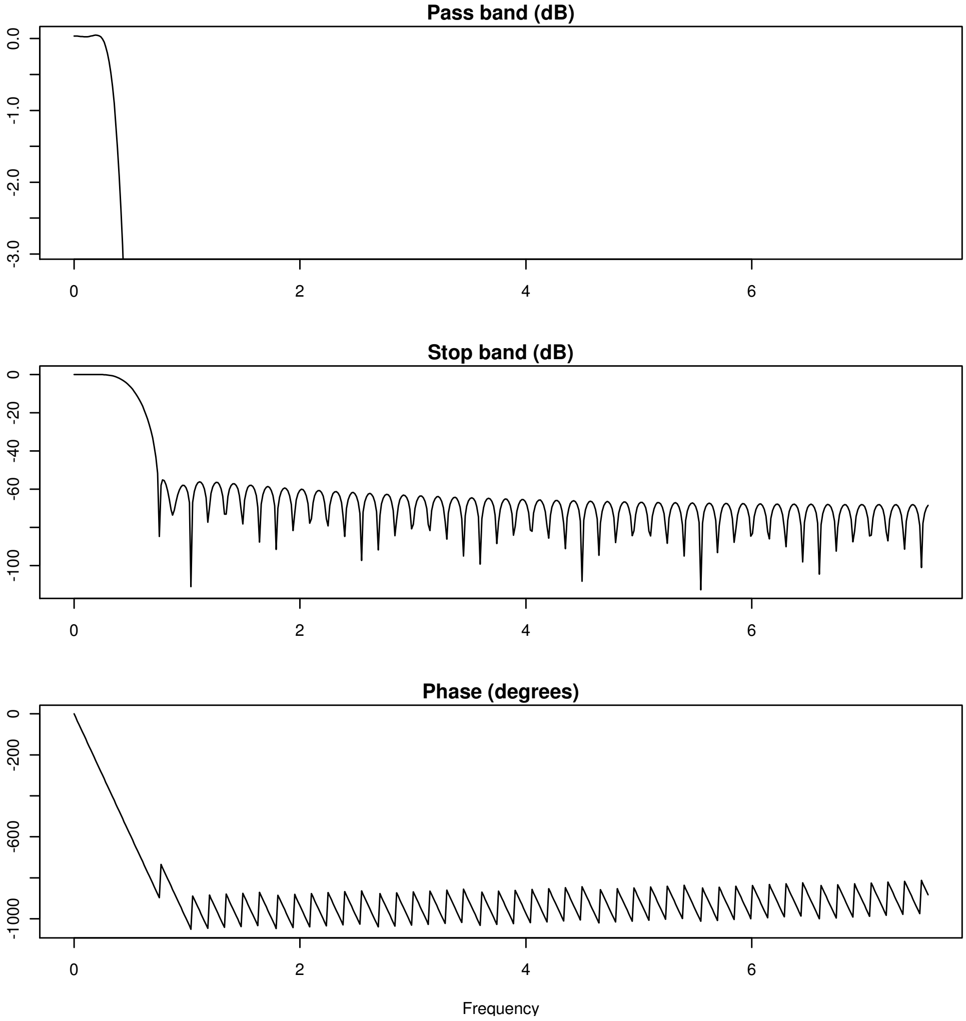

Figure 6: Lowpass filter with a cut-off of 0.5Hz applied to the acceleration magnitudes, chosen to display the coarse features of the motion more clearly.

which might be expected; there is a consistent baseline in the static signal (green).

(2.5.2) These observations suggest that different modes of movement can be distinguished in the accelerometer readings using fairly simple techniques. As a minimum, it should be possible to estimate diver speeds using accelerometer readings and to combine this with magnetometer readings to develop a dead-reckoning system based on pedometery principles.

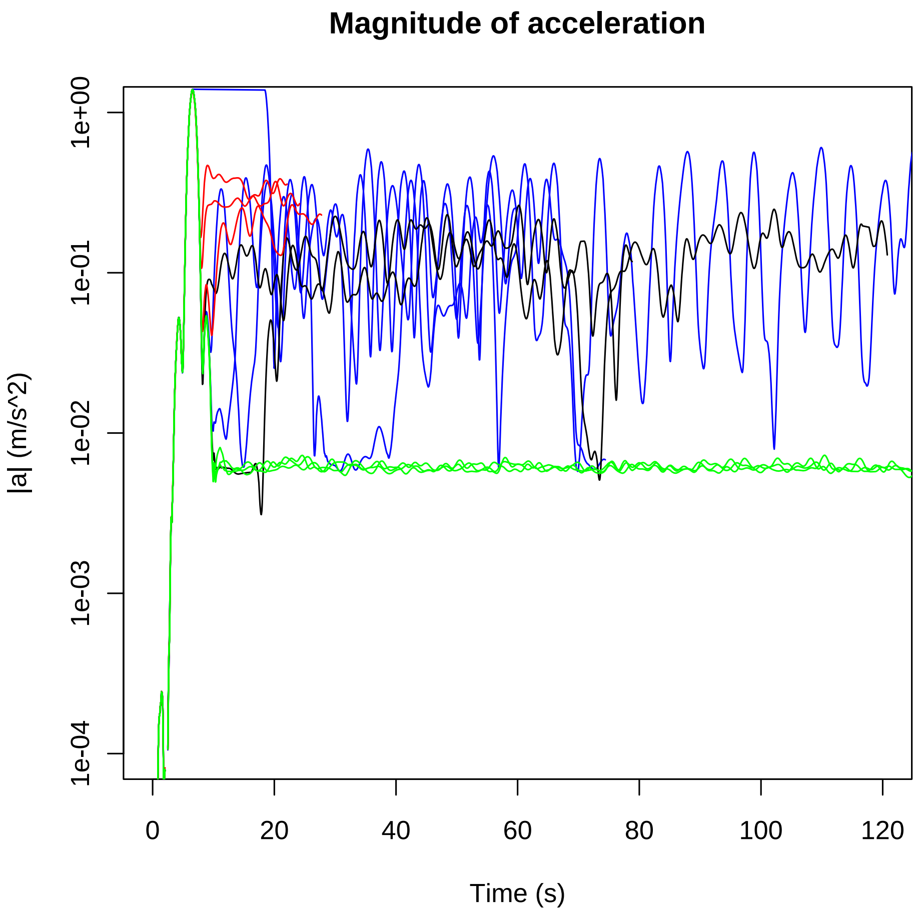

Figure 7: Smoothed accelerometer magnitudes for each of the eleven data sets captured (see text for details). Colour coding is used to distinguish static (green), walking (red), skateboard being pushed (blue) and skateboard at a near-constant speed (black). Time is in seconds.

3 Coordinate frames and equations of motion

3.1 Reference frames, measurements and objectives

(3.1.1) The problem has two frames of reference: the fixed (global) reference frame for known earth-based quantities such as the magnetic field vector and the gravitational acceleration vector; and the diver's reference frame for quantities measured by a device attached to the diver.

(3.1.2) The two frames of reference are depicted in Figure 8. A device orientation matrix relates the coordinates of a point expressed in the diver's reference frame to its coordinates in the global reference frame.

(3.1.3) In the diver's local coordinate frame, the device can measure the following quantities: $\boldsymbol{a}(t)$ – acceleration vector from accelerometers; $\boldsymbol{\omega}(t)$ – angular

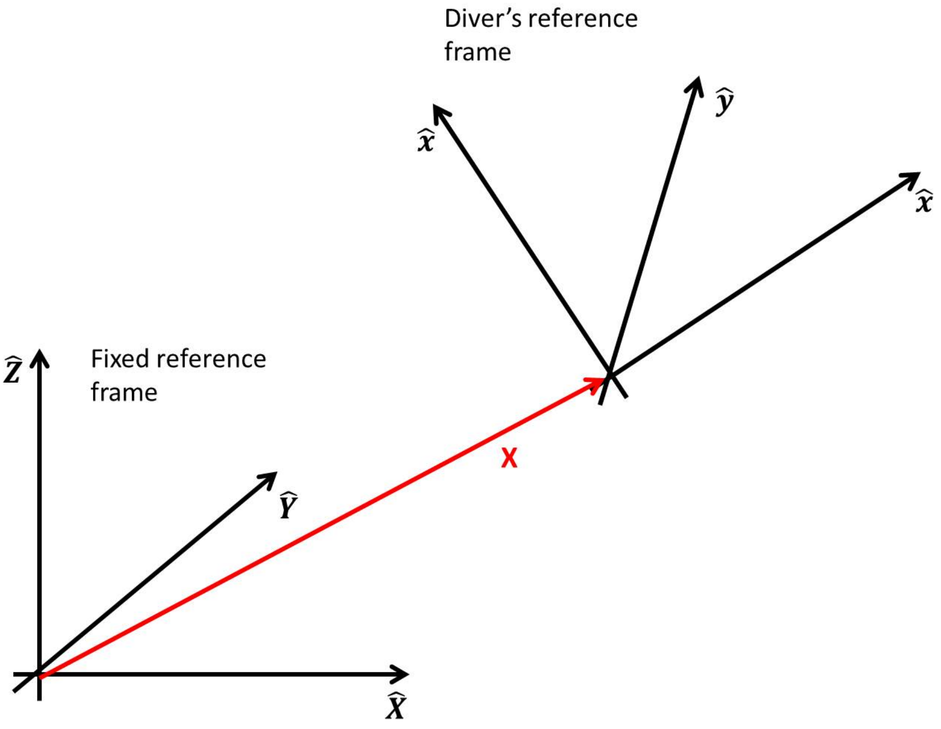

Figure 8: The fixed and diver reference frames and their notation.

velocity vector from gyroscopes; $\boldsymbol{b}(t)$ – magnetic field vector from sensor; and $h(t)$ – depth from pressure transducer.

(3.1.4) The constant magnetic field of the earth $\boldsymbol{B}$ and gravity $-g\hat{\boldsymbol{Z}}$ are fixed in the global coordinate system.

(3.1.5) The objective is to estimate the following quantities: $\hat{\boldsymbol{X}}(t)$ – diver position in global coordinate system; $\boldsymbol{U}(t)$ – diver velocity in global coordinate system; and $\boldsymbol{M}(t)$ – device orientation matrix (orthogonal rotation).

3.2 Formal mathematical relationships

(3.2.1) In an idealised world, this problem may be solved by integrating the rate equations for the device orientation matrix $\boldsymbol{M}$, and for the diver velocity $\boldsymbol{U}$ and position $\hat{\boldsymbol{X}}$ in the global reference frame, using the accelerometer readings $\boldsymbol{a}$ and gyro readings $\boldsymbol{\omega}$ as input vectors:

$$\dot{\boldsymbol{M}} = -\boldsymbol{\omega} \wedge \boldsymbol{M} \qquad \boldsymbol{M}(0) = \boldsymbol{I} \tag{1}$$

$$\dot{\boldsymbol{U}} = \boldsymbol{M}^T \cdot \boldsymbol{a} + g\hat{\boldsymbol{Z}} \qquad \boldsymbol{U}(0) = \boldsymbol{0} \tag{2}$$

$$\dot{\boldsymbol{X}} = \boldsymbol{U} \qquad \boldsymbol{X}(0) = \boldsymbol{0} \tag{3}$$

(3.2.2) Two additional equations relate data (magnetic field and depth) in the diver's frame to the corresponding data in the global frame:

$$\boldsymbol{M}^T \cdot \boldsymbol{b} = \boldsymbol{B} \tag{4}$$

$$\boldsymbol{X} \cdot \hat{\boldsymbol{Z}} = -h. \tag{5}$$

(3.2.3) A reasonable plan then is to numerically integrate (1-3) and to use (4-5) to correct as much of the noise and drift errors as possible. A scheme for using the magnetometer information to correct errors in the rotation matrix is outlined in Appendix C.

4 Theoretical treatment of errors

4.1 Noise and the integration of the circular motion

(4.1.1) Noise in the measured data can lead to the build up of error during the integration of the equations of motion. To gain insight into this effect, a simple case of circular motion, expressed in the notation of the global and diver reference frames was defined as follows.

(4.1.2) The initial conditions of motion were

$$\mathbf{X}(0) = (1, 0, 0), \tag{6}$$

$$\mathbf{U}(0) = (0, 1, 0), \tag{7}$$

$$\mathbf{M}(0) = \mathbf{I}_{3\times 3}, \tag{8}$$

and random noise was added to the constant angular velocity and acceleration that define circular motion:

$$\boldsymbol{\omega}(t) = (0, 0, 1) + \epsilon_1 \mathbf{n}(t), \tag{9}$$

$$\mathbf{a}(t) = (-1, 0, 0) + \epsilon_2 \mathbf{m}(t). \tag{10}$$

Here, $\mathbf{n}(t)$ and $\mathbf{m}(t)$ are vectors whose entries are random numbers drawn from a standard normal distribution, $N(0, 1)$.

(4.1.3) The above equations were numerically integrated in MATLAB, for four different regimes of added noise; the results of this investigation are shown in Figure 9. They show how random noise introduces systematic drift into the integrated solutions of the equations of motion. .

4.2 Estimating the drift in position

(4.2.1) An analytical estimate for the drift in position arising from random noise in the acceleration data can be obtained for linear motion as follows. The analysis is a summary (in one dimension of linear motion) of the more detailed analysis given in Appendix A.

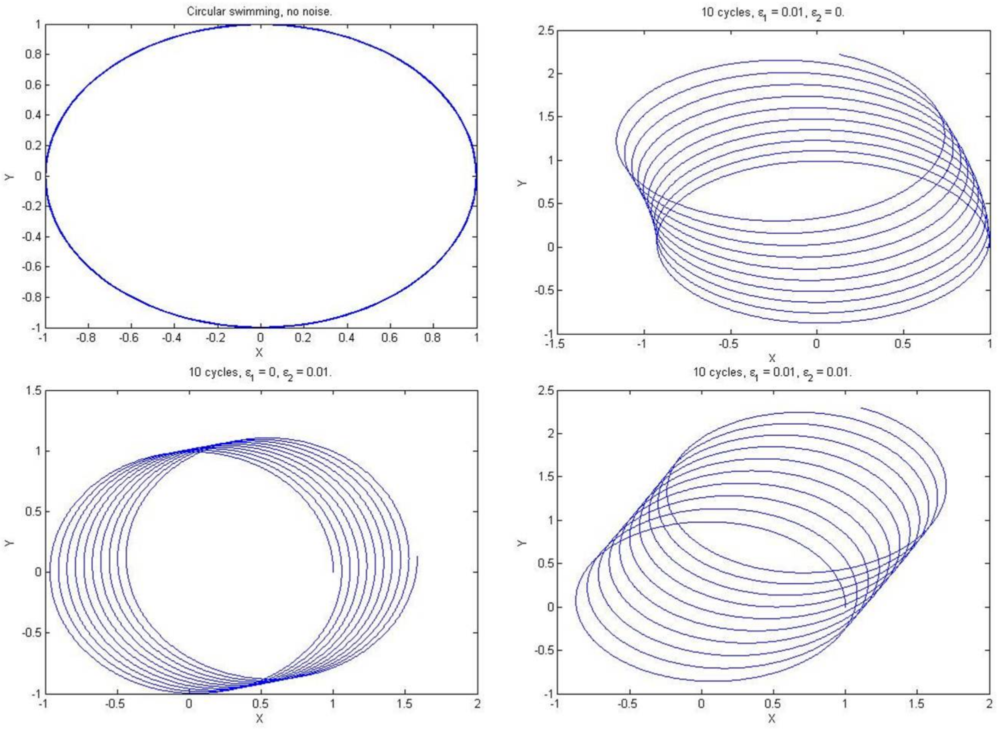

Figure 9: Trajectories showing the drift caused by the effect of random noise upon the numerical integration of the circular motion equations (9) and (10).

(4.2.2) Accelerometer data includes random noise, with an amplitude that depends on the sampling interval δt. If we choose to model errors in the acceleration a at each time-step as i.i.d. random variables, then

$$\hat{a}_n \sim \frac{\epsilon}{\delta t^{1/2}} \mathrm{N}(0,1) \equiv \mathrm{N}\left(0, \frac{\epsilon^2}{\delta t}\right)$$

Typically, for the sensors used in the Study Group, $\epsilon = 10^{-3}\,\mathrm{m\,s^{-2}\,Hz^{-1/2}}$ [4].

(4.2.3) We integrate $\ddot{x} = a$ using a simple Euler scheme to find cumulative errors in x at time $t = n\delta t$:

$$\hat{x}_n = \delta t^2 \sum_{j=1}^{n} j\,\hat{a}_{n-j} \quad \Rightarrow \quad \hat{x}_n \sim \mathrm{N}\left(0, \epsilon^2 \delta t^3 \frac{n(n+1)(2n+1)}{6}\right)$$

So for large n, the standard deviation of position error is approximately

$$\sigma = \frac{\epsilon t^{3/2}}{\sqrt{3}} \propto t^{3/2}$$

which grows with time. For $\epsilon = 10^{-3}$ this gives $\sigma \approx 45\,\mathrm{m}$ after 30 minutes, and $\sigma \approx 125\,\mathrm{m}$ after 1 hour, so illustrating the significance of noise.

(4.2.4) The Study Group did not complete an analysis of the effects of angular velocity measurement noise upon the estimation of orientation. The analysis would proceed along similar lines to that above and in Appendix A.

4.3 Numerical integration schemes and their effects

(4.3.1) In a real time implementation, MATLAB integration routines (e.g. `ode45`) cannot be used, rather low-order and fast integration schemes must be explicitly coded and executed in real-time software. An understanding of how best to implement such schemes and of their accuracy and execution speed is required. The Study Group examined the performance of first and second order in time schemes, and the management of error accumulation in the context of circular motion.

(4.3.2) The Study Group proposed a similar approach to that proposed by Woodman (2007) in [4], seeking to compute $\boldsymbol{C}(t) = \boldsymbol{C}(0)\exp(\int_0^t \boldsymbol{\Omega}(\tau)\mathrm{d}\tau)$ where $\boldsymbol{C} = \boldsymbol{M}^{\mathrm{T}}$ and $\boldsymbol{\Omega}$ is the skew symmetric form of the angular rate vector $\boldsymbol{\omega}$.

(4.3.3) Given a numerical time step δt, this matrix can be approximated, as proposed in [4], by using $\sigma = \delta t|\omega|$ and

$$C(t+\delta t) = C(t)\left(I + \frac{\sin\sigma}{\sigma}\delta t\Omega + \frac{1-\cos\sigma}{\sigma^2}(\delta t)^2\Omega^2\right),$$

with the angular velocity values corresponding to the interval $(t, t+\delta t)$.

(4.3.4) Then, we can compute an approximation of the velocity and trajectory by using the backward Euler scheme (first order accurate in time), as in [4] or use the following second-order accurate time discretization scheme:

$$U(\delta t) = U(0) + \frac{2\delta t}{3}\left(C * a(\delta t) - g\right) + \frac{\delta t}{3}\left(C * a(0) - g\right),$$

$$X(\delta t) = X(0) + \frac{2\delta t}{3}U(\delta t) + \frac{\delta t}{3}U(0),$$

and then,

$$U(t+\delta t) = \frac{4}{3}U(t) - \frac{1}{3}U(t-\delta t) + \frac{2\delta t}{3}\left(C * a(t+\delta t) - g\right),$$

$$X(t+\delta t) = \frac{4}{3}X(t) - \frac{1}{3}X(t-\delta t) + \frac{2\delta t}{3}U(t+\delta t).$$

(4.3.5) This is a well-known scheme that is used in many applied numerical works (see, for instance, [5]). Results for a circular trajectory are shown in Figure 10. This scheme has also been found to work for an ellipse.

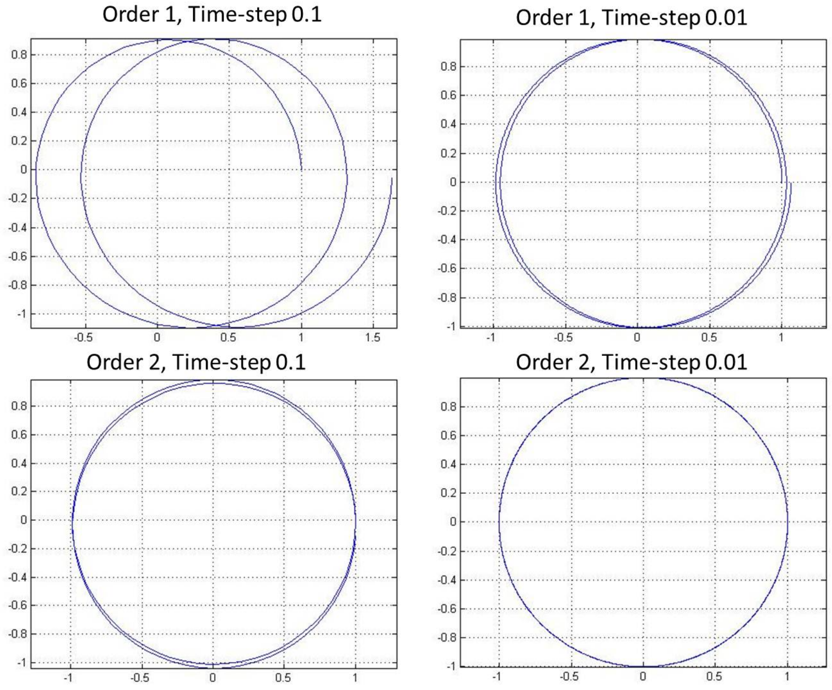

Figure 10: Top: Results for method described in Woodman (2007) [4] (order 1). Bottom: Results for second order method described above. In both cases we consider time steps 0.1 and 0.01.

(4.3.6) Some caution is required in using the integration scheme in Woodman (2007) [4]. While $\frac{dC}{dt} = \boldsymbol{C\Omega}$ (i.e. equation (33) in Woodman (2007)) is correct, $\boldsymbol{C}(t) = \boldsymbol{C}(0)\exp(\int_0^t \boldsymbol{\Omega}(\tau)\mathrm{d}\tau)$ (i.e equation (34) in [4]) is incorrect because 3D rotations do not commute. For example, if $\boldsymbol{\Omega} = \boldsymbol{\Omega}_1$ for an interval $\Delta_1 t$, followed by $\boldsymbol{\Omega} = \boldsymbol{\Omega}_2$ for an interval Δt_2, then that has a different rotation matrix from that obtained had $\boldsymbol{\Omega} = \boldsymbol{\Omega}_2$ for Δt_2 been followed by $\boldsymbol{\Omega} = \boldsymbol{\Omega}_1$ for Δt_1. However, the exponential formula would give the same result both ways because integral $\int_0^t \boldsymbol{\Omega}(\tau)\mathrm{d}\tau$ would be the same either way. Even the equation $\boldsymbol{C}(t+\delta t) = \boldsymbol{C}(t)\exp(\int_t^{t+\delta t} \boldsymbol{\Omega}(\tau)\mathrm{d}\tau)$ (i.e equation (35) in [4]) would seem to have errors of the order of δt^2 because of non-commutativity, so reducing equation (41) of [4] to being no better than the forward-Euler formula $C(t+\delta t) = C(t)(1+\boldsymbol{\Omega(t)}\delta t)$.

(4.3.7) The Study Group reviewed the quaternionic representation of 3-D rotations as means of accommodating non-commutativity and of correcting the rotation matrix to ensure its orthogonality during the course of numerical integration. The quaternionic representation is briefly described in Appendix B. The Study Group did not have time to implement the quaternionic scheme, but it should be implemented and tested in subsequent work.

5 Improving the estimates of the motion: correlation and filtering

5.1 Kalman filtering of compass and gyroscope data

(5.1.1) Determining an accurate orientation of the diver with respect to the global coordinate system is just as important as determining accurately the magnitudes of the displacements. If the displacements of the diver are in the wrong direction, this will negate the utility of the navigation device.

(5.1.2) There are different and independent measurements available that are able to provide information about the orientation of the diver, including: the gyroscope, which provides angular velocity ω; the accelerometer, which provides orientation relative to the direction of gravity g; and the compass, which provides orientation relative to the direction of the earth's magnetic field b.

(5.1.3) All these device measurements contain some amount of error. We propose to use a Kalman filter to combine all three measurements and get one optimal estimate of the three angles ϕ, θ and ψ that determine the rotation matrix R.

(5.1.4) The measured gravity vector can be obtained by low pass filtering the acceleration readings to filter out the non-gravity components. The same idea can be applied to calculate the rotation matrix for the measured magnetic force vector and the known magnetic field vector. Once the rotation matrix is known then the calculation of the three angles ϕ, θ and ψ is trivial.

(5.1.5) The Kalman filter matrices are derived in a similar way to [6]. The currently implemented Kalman filter is a discrete and linear Kalman filter. This works well for the 1D case, however if we wanted to implement the full 3D system we would prefer an extended Kalman filter to better handle the nonlinearity in the equations. For a good introduction to the Kalman filter see [7]. The only difference between our method and the method proposed by [6] is (11). Here we used a finite difference to estimate the time derivative.

$$\boldsymbol{x}(t+1) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(t) + \boldsymbol{v}(t) \tag{11}$$

(5.1.6) Some interesting plots showing how the Kalman filter improves the estimates of the diver orientation by combining compass and angular velocity measurements can be found in Figures 11 and 12.

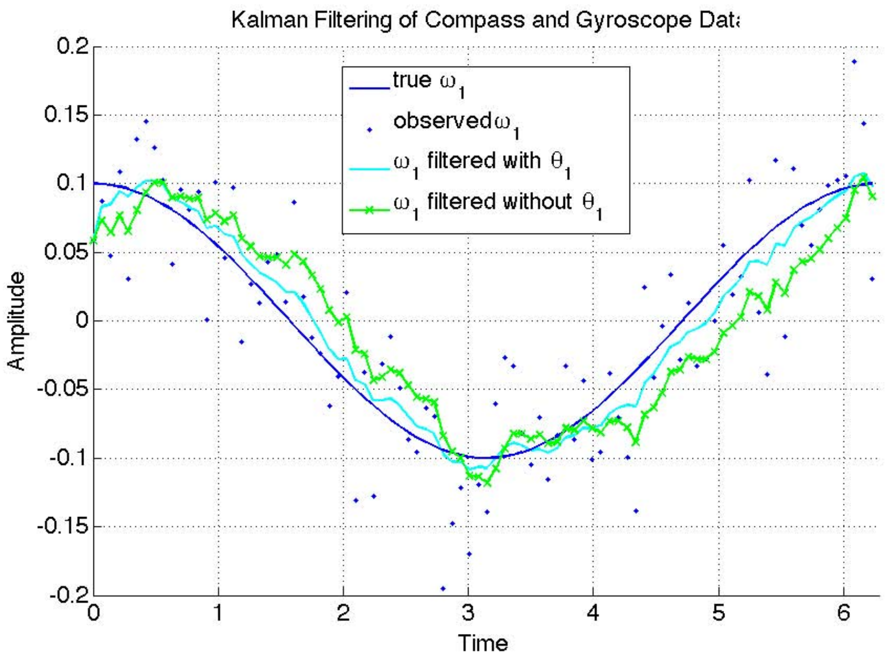

Figure 11: First example showing how the Kalman filter improves the estimates of the diver orientation by combining compass and angular velocity measurements.

6 Summary

6.1 Summary

(6.1.1) In this Study Group, we captured data using an Android device strapped to a skateboard to simulate the type of movements a diver might make underwater. Different modes of movement (non-movement, constant speed, impulsive movement) are evident from filtered versions of the sensor outputs, so indicating that a pedometry based solution ought to be feasible.

(6.1.2) We also investigated the feasibility of a more generic dead-reckoning system. Though sensors provide more data than is strictly necessary, noise and imperfect calibration lead to significant errors. Owing to the nature of the problem, these errors accumulate over time leading to significant drift errors over the time scales of interest. Using estimates of the white noise produced by the accelerometers we derived estimates of the drift in position over time. The accuracy of practical numerical integration schemes was also investigated. By combining accelerometer and gyroscopic data we showed how a Kalman filter can be used to reduce error in the orientation data.

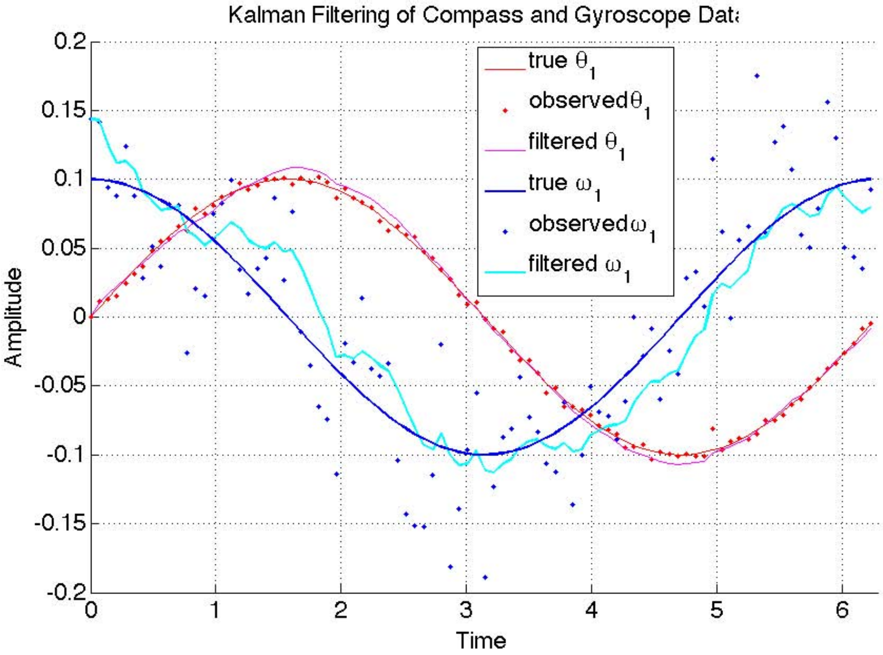

Figure 12: Second example showing how the Kalman filter improves the estimates of the diver orientation by combining compass and angular velocity measurements.

A An estimate of position drift for linear motion

A.1 One–dimensional case

(A.1.1) Consider a one-dimensional problem in which the device measures the acceleration a_i at intervals of δt with independent normally distributed random errors

$$\hat{a}_i \sim \frac{\epsilon}{\delta t^{1/2}} \mathrm{N}(0,1) \equiv \mathrm{N}(0, \epsilon^2/\delta t) \tag{12}$$

The dependence on δt is a known property of the device. (We assume that this occurs because the sensor contains some internal averaging between polling times.)

(A.1.2) Now consider a simple Euler integration scheme, in which we write:

$$u_{i+1/2} = u_{i-1/2} + \delta t\, a_i\,, \tag{13}$$

$$x_{i+1} = x_i + \delta t\, u_{i+1/2}\,, \tag{14}$$

starting from exact initial conditions $x_0 = u_{-1/2} = 0$. Then it is easy to see that

$$u_{n+1/2} = \delta t \sum_{i=0}^{n} a_i \,, \tag{15}$$

$$x_{n+1} = \delta t^2 \sum_{i=0}^{n} (n+1-i) a_i \,, \tag{16}$$

Hence the error in each position is given by

$$\hat{x}_n = \delta t^2 \sum_{i=0}^{n-1} (n-i)\hat{a}_i = \delta t^2 \sum_{j=1}^{n} j\, \hat{a}_{n-j} \,. \tag{17}$$

(A.1.3) Now from (12) we have that

$$\delta t^2 j\, \hat{a}_{n-j} \sim \mathrm{N}(0, \epsilon^2 \delta t^3 j^2) \tag{18}$$

Since the $\hat{a}_i$'s are independent, on summing we add the variances, and so obtain

$$\hat{x}_n \sim \mathrm{N}\Big(0, \epsilon^2 \delta t^3 \textstyle\sum_{j=1}^{n} j^2\Big) \equiv \mathrm{N}\Big(0, \frac{1}{6}\epsilon^2 \delta t^3 n(n+1)(2n+1)\Big) \,. \tag{19}$$

(A.1.4) So for large n, the distribution is approximately

$$\hat{x}_n \sim \mathrm{N}\Big(0, \tfrac{1}{3}\epsilon^2 t^3\Big) \tag{20}$$

where $t = n\delta t$ is the elapsed time. So the standard deviation $\sigma(t)$ of the error in position at time t is given by

$$\sigma(t) \approx \frac{\epsilon\, t^{3/2}}{\sqrt{3}} \,, \tag{21}$$

independent of δt.

(A.1.5) From [4] typical errors on the sensor are

$$\epsilon \approx 0.001\,\mathrm{ms}^{-2}\mathrm{Hz}^{-1/2} \tag{22}$$

If this is the case, then after an hour σ will be around 125 m. After two hours, this will rise to over 350 m

A.2 Three–dimensional case

(A.2.1) In three dimensions, there will be three independent errors aligned with the coordinate axes, each being distributed as above:

$$\hat{x}, \hat{y}, \hat{z} \sim \mathrm{N}\Big(0, \tfrac{1}{3}\epsilon^2 t^3\Big) \tag{23}$$

(A.2.2) The absolute error $\hat{r} = \sqrt{\hat{x}^2 + \hat{y}^2 + \hat{z}^2}$ is then a Chi distribution

$$\hat{r} \sim \chi\left(\tfrac{1}{\sqrt{3}}\epsilon t^{3/2}\right), \qquad (24)$$

and its expected value and standard deviation are

$$\mu_{\hat{r}} = 2\sqrt{\frac{2}{3\pi}}\,\epsilon t^{3/2}, \qquad \sigma_{\hat{r}} = \sqrt{1 - \frac{8}{3\pi}}\,\epsilon t^{3/2}. \qquad (25)$$

(A.2.3) With $\epsilon = 0.001\,\mathrm{ms}^{-2}\mathrm{Hz}^{-1/2}$ as above, the values of $\mu_{\hat{r}}$ after one and two hours are roughly 200 m and 560 m respectively.

(A.2.4) A similar calculation could be done to estimate likely errors in orientation, though the form would be more complicated.

B Quaternionic representation

B.1 Theory

(B.1.1) Quaternionic representations for rotations are well known to help reduce errors that accumulate during numerical calculations. Rather than having to re-orthogonalise rotation matrices as rounding errors creep in on repeated multiplications, the quaternion representing the combination of rotations needs to be kept unit length.

B.2 Implementation

(B.2.1) We denote the quaternion $\boldsymbol{q} = q_0 + q_1 i + q_2 j + q_3 k$ by the more convenient (for our application) notation $(q_0, \boldsymbol{p})$. Quaternions add in the obvious manner, $\boldsymbol{q}^c = (q_o, -\boldsymbol{p})$, $\boldsymbol{q}^{-1} = (q_0, -\boldsymbol{p})/\boldsymbol{q}\boldsymbol{q}^c$ and for two quaternions $\boldsymbol{q_1} = (a_1, \boldsymbol{b_1})$ and $\boldsymbol{q_2} = (a_2, \boldsymbol{b_2})$ multiplication looks like

$$\boldsymbol{q_1}\boldsymbol{q_2} = (a_1 a_2 - \boldsymbol{b_1}.\boldsymbol{b_2}, a_1\boldsymbol{b_2} + a_2\boldsymbol{b_1} + \boldsymbol{b_1} \times \boldsymbol{b_2}).$$

(B.2.2) Taking the unit quaternion $\boldsymbol{q} = (\cos(\theta/2), \sin(\theta/2)\hat{\boldsymbol{p}})$ and the vector $\boldsymbol{v} = (0, \boldsymbol{w})$, the product $\boldsymbol{v'} = \boldsymbol{q}\boldsymbol{v}\boldsymbol{q}^{\boldsymbol{-1}}$ gives $(0, \boldsymbol{w'})$ where in $\boldsymbol{R}^3$ the vector $\boldsymbol{w'}$ is the vector $\boldsymbol{w}$ rotated about the axis $\hat{\boldsymbol{p}}$ through angle θ.

(B.2.3) So now if $\boldsymbol{W}(t) = (0, \boldsymbol{\omega}(t))$ then

$$\frac{d\boldsymbol{q}}{dt} = \frac{1}{2}\boldsymbol{W}(t)\boldsymbol{q}(t)$$

$$(0, \dot{\boldsymbol{U}}) = \boldsymbol{q}^{-1}(t)(0, \boldsymbol{a})\boldsymbol{q} + (0, g\hat{\boldsymbol{Z}})$$

$$(0, \dot{\boldsymbol{X}}) = (0, \boldsymbol{U})$$

(B.2.4) When integrating the vector $\boldsymbol{q}$ has to be renormalised every step, a discussion on this may be found in [1].

C Correction of the rotation matrix using heading information

C.1 Outline of approach

(C.1.1) A scheme for correction to orientation to ensure that (4) holds is now outlined.

(C.1.2) Recall that $\boldsymbol{B}$ is the magnetic field in the global coordinate system and $\boldsymbol{b}$ is the field measured by the magnetic field sensor at any given time. Note that $\boldsymbol{B}$ is constant, and is known from how the global system is defined relative to the earth when the diver begins his dive. Also recall that $\boldsymbol{M}$ is the rotation matrix describing the calculated orientation of the diver at any particular time.

(C.1.3) We now define $\boldsymbol{b'}$ to be the magnetic field we would expect to measure given the global field $\boldsymbol{B}$ and the assumed orientation $\boldsymbol{M}$, i.e.

$$\boldsymbol{b'} = \boldsymbol{M} \cdot \boldsymbol{B}\,. \tag{26}$$

(C.1.4) In general, $\boldsymbol{b}$ will not be exactly equal to $\boldsymbol{b'}$, because of errors in both the measured $\boldsymbol{b}$ and drift errors in the computed orientation $\boldsymbol{M}$. We aim to fix the discrepency by adjusting $\boldsymbol{M}$ so that the measured field matches that implied by the orientation.

(C.1.5) In other words, we wish to find a new orientation matrix $\boldsymbol{M'}$, which is close to the original $\boldsymbol{M}$, but which satisfies

$$\boldsymbol{b} = \boldsymbol{M'} \cdot \boldsymbol{B}\,. \tag{27}$$

C.2 Implementation

(C.2.1) We first normalise the measured magnetic field $\boldsymbol{b}$ so that $|\boldsymbol{b}| = |\boldsymbol{B}|$. Since $\boldsymbol{M}$ is orthogonal by definition, we already have $|\boldsymbol{b'}| = |\boldsymbol{B}|$.

(C.2.2) Let $\boldsymbol{M'} = \boldsymbol{Q} \cdot \boldsymbol{M}$ for some $\boldsymbol{Q}$. Then we must find an orthogonal $\boldsymbol{Q}$ such that,

$$\boldsymbol{b} = (\boldsymbol{Q} \cdot \boldsymbol{M}) \cdot \boldsymbol{B} = \boldsymbol{Q} \cdot \boldsymbol{b'} \tag{28}$$

i.e. $\boldsymbol{Q}$ should be a rotation that takes $\boldsymbol{b'}$ to $\boldsymbol{b}$. For $\boldsymbol{M'}$ to be close to $\boldsymbol{M}$ we want $\boldsymbol{Q}$ to be close to the identity $\boldsymbol{I}$.

(C.2.3) There are many rotations $\boldsymbol{Q}$ that will satisfy (28), but we shall take the one with the smallest angle, so that it is closest to the idenity. This will be rotation normal to the plane containing $\boldsymbol{b'}$ and $\boldsymbol{b}$, i.e. one about the axis

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{b'} \wedge \boldsymbol{b}}{|\boldsymbol{b'} \wedge \boldsymbol{b}|}\,. \tag{29}$$

(C.2.4) The required rotation angle θ is given by the angle between the vectors $\boldsymbol{b'}$ and $\boldsymbol{b}$, so we have

$$\cos\theta = \frac{\boldsymbol{b'} \cdot \boldsymbol{b}}{\boldsymbol{B}^2}\,, \tag{30}$$

$$\hat{\boldsymbol{n}}\sin\theta = \frac{\boldsymbol{b'} \wedge \boldsymbol{b}}{\boldsymbol{B}^2}\,, \tag{31}$$

since $|\boldsymbol{b}| = |\boldsymbol{b'}| = |\boldsymbol{B}|$.

(C.2.5) The rotation matrix for a rotation of angle θ about an axis $\hat{\boldsymbol{n}}$ is given by

$$\boldsymbol{Q} = \boldsymbol{I} + (1 - \cos\theta)\hat{\boldsymbol{n}}\hat{\boldsymbol{n}} + (\hat{\boldsymbol{n}} \wedge \boldsymbol{I})\sin\theta\,. \tag{32}$$

Hence we have

$$\boldsymbol{Q} = \frac{(\boldsymbol{b'} \cdot \boldsymbol{b})\boldsymbol{I}}{\boldsymbol{B}^2} + \left(1 - \frac{\boldsymbol{b'} \cdot \boldsymbol{b}}{\boldsymbol{B}^2}\right)\frac{(\boldsymbol{b'} \wedge \boldsymbol{b})(\boldsymbol{b'} \wedge \boldsymbol{b})}{|\boldsymbol{b'} \wedge \boldsymbol{b}|^2} + \frac{(\boldsymbol{b'} \wedge \boldsymbol{b}) \wedge \boldsymbol{I}}{\boldsymbol{B}^2}\,. \tag{33}$$

(C.2.6) This approach assumes that the dominant error is in $\boldsymbol{M}$, and that the measured magnetic field $\boldsymbol{b}$ is relatively error free. This may not be the case in reality, so it may be preferable to not correct $\boldsymbol{M}$ completely, but to only adjust it by a rotation of angle $\alpha\theta$ for some fixed $\alpha \in (0, 1)$.

Bibliography

[1] *Quaternion-based rigid body rotation integration algorithms for use in particle methods*, Johnson et. al., Int. J. Numer. Meth. Eng 2008; 74.

[2] *Anscamobile accelerometer data*, Anscamobile (http://developer.anscamobile.com/reference/index/events/accelerometer)

[3] *Accelerometers and how they work*, Texas Instruments (http://www2.usfirst.org/2005comp/Manuals/Acceler1.pdf)

[4] *An introduction to inertial navigation*, Woodman, O.J., Technical Report 696, Computer Laboratory, University of Cambridge; UCAM-CL-TR-696, ISSN 1476-2986.

[5] A.M. Ramos, R. Glowinski and J. Periaux, *Pointwise Control of the Burgers Equation and Related Nash Equilibrium Problems: Computational Approach*, Journal of Optimization, Theory and Applications, Vol. 112, No. 3, pp. 499–516, 2002.

[6] Y. Suh, *Attitude estimation using low cost accelerometer and gyroscope*, Science and Technology, Proceedings KORUS 2003; The 7th Korea-Russia International Symposium on, volume 2, IEEE, pp. 423-427.

[7] G. Welch and G. Bishop, *An introduction to the Kalman filter, University of North Carolina at Chapel Hill, Chapel Hill, NC 7 (1995).*