An Efficient Orientation Filter for Inertial Measurement Units (IMUs) and Magnetic Angular Rate and Gravity (MARG) Sensor Arrays

Sebastian O.H Madgwick*, Ravi Vaidyanathan, and Andrew Harrison
s.madgwick@bristol.ac.uk; r.vaidyanathan@bristol.ac.uk; andrew.j.l.harrison@bristol.ac.uk

Department of Mechanical Engineering

University of Bristol

Bristol, BS8 1TR

United Kingdom

* corresponding author

This report should be referenced:

S.O.H. Madgwick, R. Vaidyanathan, A.J.L. Harrison, "An Efficient Orientation Filter for IMU and MARG Sensor Arrays", Department of Mechanical Engineering, University of Bristol, April 2010

A efficient orientation filter for IMUs and MARG sensor arrays

Sebastian O.H. Madgwick Andrew J.L. Harrison
Ravi Vaidyanathan

April 9, 2010

Abstract

This report presents a novel orientation filter applicable to IMUs consisting of tri-axis gyroscopes and accelerometers, and MARG sensor arrays that also include tri-axis magnetometers. The MARG implementation incorporates magnetic distortion and gyroscope bias drift compensation. The filter uses a quaternion representation, allowing accelerometer and magnetometer data to be used in an analytically derived and optimised gradient-descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative. The benefits of the filter include: (1) computationally inexpensive; requiring 109 (IMU) or 277 (MARG) scalar arithmetic operations each filter update, (2) effective at low sampling rates; e.g. 10 Hz, and (3) contains 1 (IMU) or 2 (MARG) adjustable parameters defined by observable system characteristics. Performance was evaluated empirically using a commercially available orientation sensor and reference measurements of orientation obtained using an optical measurement system. A simple calibration method is presented for the use of the optical measurement equipment in this application. Performance was also benchmarked against the propriety Kalman-based algorithm of orientation sensor. Results indicate the filter achieves levels of accuracy exceeding that of the Kalman-based algorithm; $< 0.6°$ static RMS error, $< 0.8°$ dynamic RMS error. The implications of the low computational load and ability to operate at low sampling rates open new opportunities for the use of IMU and MARG sensor arrays in real-time applications of limited power or processing resources or applications that demand extremely high sampling rates.

Contents

1 Introduction **3**

2 Quaternion representation **5**

3 Filter derivation **7**
3.1 Orientation from angular rate 7
3.2 Orientation from vector observations 8
3.3 Filter fusion algorithm 12
3.4 Magnetic distortion compensation 13
3.5 Gyroscope bias drift compensation 15
3.6 Filter gains 16

4 Experimentation **17**
4.1 Equipment 17
4.2 Orientation from optical measurements 18
4.3 Calibration of frame alignments 20
4.4 Experimental proceedure 23

5 Results **23**
5.1 Typical results 24
5.2 Static and dynamic performance 26
5.3 Filter gain vs. performance 27
5.4 Sampling rate vs. performance 28
5.5 Gyroscope bias drift 29

6 Discussion **30**

7 Conclusions **31**

A IMU filter implementation optimised in C **37**

B MARG filter implementation optimised in C **38**

1 Introduction

The accurate measurement of orientation plays a critical role in a range of fields including: aerospace [1, 2, 3], robotics [4, 5], navigation [6, 7] and human motion analysis [8, 9] and machine interaction [10]. Whilst a variety of technologies enable the measurement of orientation, inertial based sensory systems have the advantage of being completely self contained such that the measurement entity is constrained neither in motion nor to any specific environment or location. An IMU (Inertial Measurement Unit) consists of gyroscopes and accelerometers enabling the tracking of rotational and translational movements. In order to measure in three dimensions, tri-axis sensors consisting of 3 mutually orthogonal sensitive axes are required. A MARG (Magnetic, Angular Rate, and Gravity) sensor is a hybrid IMU which incorporates a tri-axis magnetometer. An IMU alone can only measure an attitude relative to the direction of gravity which is sufficient for many applications [4, 2, 8, 1]. MARG systems, also known as AHRS (Attitude and Heading Reference Systems) are able to provide a *complete* measurement of orientation relative to the direction of gravity and the earth's magnetic field.

A gyroscope measures angular velocity which, if initial conditions are known, may be integrated over time to compute the sensor's orientation [11, 12]. Precision gyroscopes, ring laser for example, are too expensive and bulky for most applications and so less accurate MEMS (Micro Electrical Mechanical System) devices are used in a majority of applications [13]. The integration of gyroscope measurement errors will lead to an accumulating error in the calculated orientation. Therefore, gyroscopes alone cannot provide an absolute measurement of orientation. An accelerometer and magnetometer will measure the earth's gravitational and magnetic fields respectively and so provide an absolute reference of orientation. However, they are likely to be subject to high levels of noise; for example, accelerations due to motion will corrupt measured direction of gravity. The task of an orientation filter is to compute a single estimate of orientation through the optimal fusion of gyroscope, accelerometer and magnetometer measurements.

The Kalman filter [14] has become the accepted basis for the majority of orientation filter algorithms [4, 15, 16, 17] and commercial inertial orientation sensors; xsens [18], micro-strain [19], VectorNav [20], Intersense [21], PNI [22] and Crossbow [23] all produce systems founded on its use. The widespread use of Kalman-based solutions are a testament to their accuracy and effectiveness, however, they have a number of disadvantages. They can

be complicated to implement which is reflected by the numerous solutions seen in the subject literature [3, 4, 15, 16, 17, 24, 25, 26, 27, 28, 29, 30, 31, 32]. The linear regression iterations, fundamental to the Kalman process, demand sampling rates far exceeding the subject bandwidth; for example, a sampling rate between 512 Hz [18] and 30 kHz [19] may be used for a human motion caption application. The state relationships describing rotational kinematics in three-dimensions typically require large state vectors and an extended Kalman filter implementation [4, 17, 24] to linearise the problem.

These challenges demand a large computational load for implementation of Kalman-based solutions and provide a clear motivation for alternative approaches. Many previous approaches to address these issues have implemented either fuzzy processing [2, 5] or fixed filters [33] to favour accelerometer measurements of orientation at low angular velocities and the integrated gyroscope measurements at high angular velocities. Such an approach is simple but may only be effective under limited operating conditions. Bachman *et al* [34] proposed an alternative approach where the filter achieves an optimal fusion of measurements data at all angular velocities. However, the process requires a least squares regression, which also brings in an associated computational load. Mahony *et al* [35] developed the complementary filter which is shown to be an efficient and effective solution; however, performance is only validated for an IMU.

This report introduces novel orientation filter that is applicable to both IMUs and MARG sensor arrays addressing issues of computational load and parameter tuning associated with Kalman-based approaches. The filter employs a quaternion representation of orientation (as in: [34, 17, 24, 30, 32]) to describe the coupled nature of orientations in three-dimensions and is not subject to the problematic singularities associated with an Euler angle representation[1]. A complete derivation and empirical evaluation of the new filter is presented. Its performance is benchmarked against an existing commercial filter and verified with optical measurement system. Innovative aspects of the proposed filter include: a single adjustable parameter defined by observable systems characteristics; an analytically derived and optimised gradient-descent algorithm enabling performance at low sampling rates; an on-line magnetic distortion compensation algorithm; and gyroscope bias drift compensation.

[1]Kuipers [36] offers a comprehensive introduction to this use of quaternions.

2 Quaternion representation

A quaternion is a four-dimensional complex number that can be used to represent the orientation of a ridged body or coordinate frame in three-dimensional space. An arbitrary orientation of frame B relative to frame A can be achieved through a rotation of angle θ around an axis ${}^{A}\hat{\boldsymbol{r}}$ defined in frame A. This is represented graphically in figure 1 where the mutually orthogonal unit vectors $\hat{\boldsymbol{x}}_A$, $\hat{\boldsymbol{y}}_A$ and $\hat{\boldsymbol{z}}_A$, and $\hat{\boldsymbol{x}}_B$, $\hat{\boldsymbol{y}}_B$ and $\hat{\boldsymbol{z}}_B$ define the principle axis of coordinate frames A and B respectively. The quaternion describing this orientation, ${}^{A}_{B}\hat{\boldsymbol{q}}$, is defined by equation (1) where r_x, r_y and r_z define the components of the unit vector ${}^{A}\hat{\boldsymbol{r}}$ in the x, y and z axes of frame A respectively. A notation system of leading super-scripts and sub-scripts adopted from Craig [37] is used to denote the relative frames of orientations and vectors. A leading sub-script denotes the frame being described and a leading super-script denotes the frame this is with reference to. For example, ${}^{A}_{B}\hat{\boldsymbol{q}}$ describes the orientation of frame B relative to frame A and ${}^{A}\hat{\boldsymbol{r}}$ is a vector described in frame A. Quaternion arithmetic often requires that a quaternion describing an orientation is first normalised. It is therefore conventional for all quaternions describing an orientation to be of unit length.

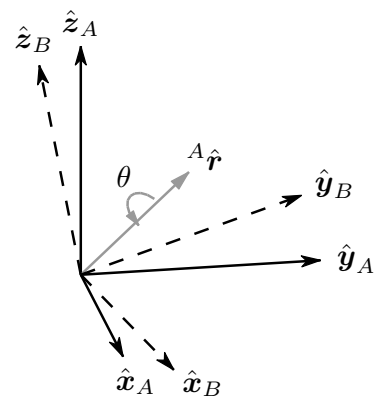

Figure 1: The orientation of frame B is achieved by a rotation, from alignment with frame A, of angle θ around the axis ${}^{A}\boldsymbol{r}$.

$$ {}^{A}_{B}\hat{\boldsymbol{q}} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} = \begin{bmatrix} -\cos\frac{\theta}{2} & r_x sin\frac{\theta}{2} & r_y sin\frac{\theta}{2} & r_z sin\frac{\theta}{2} \end{bmatrix} \tag{1} $$

The quaternion conjugate, denoted by *, can be used to swap the relative

frames described by an orientation. For example, ${}^{B}_{A}\hat{\boldsymbol{q}}$ is the conjugate of ${}^{A}_{B}\hat{\boldsymbol{q}}$ and describes the orientation of frame A relative to frame B. The conjugate of ${}^{A}_{B}\hat{\boldsymbol{q}}$ is defined by equation (2).

$$ {}^{A}_{B}\hat{\boldsymbol{q}}^{*} = {}^{B}_{A}\hat{\boldsymbol{q}} = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix} \tag{2} $$

The quaternion product, denoted by $\otimes$, can be used to define compound orientations. For example, for two orientations described by ${}^{A}_{B}\hat{\boldsymbol{q}}$ and ${}^{B}_{C}\hat{\boldsymbol{q}}$, the compounded orientation ${}^{A}_{C}\hat{\boldsymbol{q}}$ can be defined by equation (3).

$$ {}^{A}_{C}\hat{\boldsymbol{q}} = {}^{B}_{C}\hat{\boldsymbol{q}} \otimes {}^{A}_{B}\hat{\boldsymbol{q}} \tag{3} $$

For two quaternions, $\boldsymbol{a}$ and $\boldsymbol{b}$, the quaternion product can be determined using the Hamilton rule and defined as equation (4). A quaternion product is not commutative; that is, $\boldsymbol{a} \otimes \boldsymbol{b} \neq \boldsymbol{b} \otimes \boldsymbol{a}$.

$$ \begin{aligned} \boldsymbol{a} \otimes \boldsymbol{b} &= \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} \otimes \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix} \\ &= \begin{bmatrix} a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2 \\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1 \end{bmatrix}^{T} \end{aligned} \tag{4} $$

A three dimensional vector can be rotated by a quaternion using the relationship described in equation (5) [36]. ${}^{A}\boldsymbol{v}$ and ${}^{B}\boldsymbol{v}$ are the same vector described in frame A and frame B respectively where each vector contains a 0 inserted as the first element to make them 4 element row vectors.

$$ {}^{B}\boldsymbol{v} = {}^{A}_{B}\hat{\boldsymbol{q}} \otimes {}^{A}\boldsymbol{v} \otimes {}^{A}_{B}\hat{\boldsymbol{q}}^{*} \tag{5} $$

The orientation described by ${}^{A}_{B}\hat{\boldsymbol{q}}$ can be represented as the rotation matrix ${}^{A}_{B}\boldsymbol{R}$ defined by equation (6) [36].

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} 2q_1^2 - 1 + 2q_2^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & 2q_1^2 - 1 + 2q_3^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & 2q_1^2 - 1 + 2q_4^2 \end{bmatrix} \tag{6} $$

Euler angles ψ, θ and ϕ in the so called aerospace sequence [36] describe an orientation of frame B achieved by the sequential rotations, from alignment with frame A, of ψ around $\hat{\boldsymbol{z}}_B$, θ around $\hat{\boldsymbol{y}}_B$, and ϕ around $\hat{\boldsymbol{x}}_B$. This Euler angle representation of ${}^{A}_{B}\hat{\boldsymbol{q}}$ is defined by equations (7), (8) and (9).

$$\psi = \text{Atan2}\left(2q_2q_3 - 2q_1q_4, 2q_1^2 + 2q_2^2 - 1\right) \tag{7}$$

$$\theta = -\sin^{-1}\left(2q_2q_4 + 2q_1q_3\right) \tag{8}$$

$$\phi = \text{Atan2}\left(2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1\right) \tag{9}$$

3 Filter derivation

3.1 Orientation from angular rate

A tri-axis gyroscope will measure the angular rate about the x, y and z axes of the senor frame, termed ω_x, ω_y and ω_z respectively. If these parameters (in rads^{-1}) are arranged into the vector ${}^{S}\boldsymbol{\omega}$ defined by equation (10), the quaternion derivative describing the rate of change of orientation of the earth frame relative to the sensor frame ${}^{S}_{E}\dot{\boldsymbol{q}}$ can be calculated [38] as equation (11).

$${}^{S}\boldsymbol{\omega} = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix} \tag{10}$$

$${}^{S}_{E}\dot{\boldsymbol{q}} = \frac{1}{2}{}^{S}_{E}\hat{\boldsymbol{q}} \otimes {}^{S}\boldsymbol{\omega} \tag{11}$$

The orientation of the earth frame relative to the sensor frame at time t, ${}^{E}_{S}\boldsymbol{q}_{\omega,t}$, can be computed by numerically integrating the quaternion derivative ${}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t}$ as described by equations (12) and (13) provided that initial conditions are known. In these equations, ${}^{S}\boldsymbol{\omega}_t$ is the angular rate measured at time t, Δt is the sampling period and ${}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1}$ is the previous estimate of orientation. The sub-script ω indicates that the quaternion is calculated from angular rates.

$${}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t} = \frac{1}{2}{}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1} \otimes {}^{S}\boldsymbol{\omega}_t \tag{12}$$

$${}^{S}_{E}\boldsymbol{q}_{\omega,t} = {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1} + {}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t}\Delta t \tag{13}$$

3.2 Orientation from vector observations

A tri-axis accelerometer will measure the magnitude and direction of the field of gravity in the sensor frame compounded with linear accelerations due to motion of the sensor. Similarly, a tri-axis magnetometer will measure the magnitude and direction of the earth's magnetic field in the sensor frame compounded with local magnetic flux and distortions. In the context of an orientation filter, it will initially be assumed that an accelerometer will measure only gravity, a magnetometer will measure only the earth's magnetic field.

If the direction of an earth's field is known in the earth frame, a measurement of the field's direction within the sensor frame will allow an orientation of the sensor frame relative to the earth frame to be calculated. However, for any given measurement there will not be a unique sensor orientation solution, instead there will infinite solutions represented by all those orientations achieved by the rotation the true orientation around an axis parallel with the field. In some applications it may be acceptable to use an Euler angle representation allowing an incomplete solution to be found as two known Euler angles and one unknown [5]; the unknown angle being the rotation around an axis parallel with the direction of the field. A quaternion representation requires a *complete* solution to be found. This may be achieved through the formulation of an optimisation problem where an orientation of the sensor, ${}^{S}_{E}\hat{\boldsymbol{q}}$, is that which aligns a predefined reference direction of the field in the earth frame, ${}^{E}\hat{\boldsymbol{d}}$, with the measured direction of the field in the sensor frame, ${}^{S}\hat{\boldsymbol{s}}$, using the rotation operation described by equation (5). Therefore ${}^{S}_{E}\hat{\boldsymbol{q}}$ may be found as the solution to (14) where equation (15) defines the objective function. The components of each vector are defined in equations (16) to (18).

$$\min_{{}^{S}_{E}\hat{\boldsymbol{q}}\in\Re^4} \boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}) \tag{14}$$

$$\boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}) = {}^{S}_{E}\hat{\boldsymbol{q}}^{*} \otimes {}^{E}\hat{\boldsymbol{d}} \otimes {}^{S}_{E}\hat{\boldsymbol{q}} - {}^{S}\hat{\boldsymbol{s}} \tag{15}$$

$${}^{S}_{E}\hat{\boldsymbol{q}} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \tag{16}$$

$${}^{E}\hat{\boldsymbol{d}} = \begin{bmatrix} 0 & d_x & d_y & d_z \end{bmatrix} \tag{17}$$

$$ {}^{S}\hat{\boldsymbol{s}} = \begin{bmatrix} 0 & s_x & s_y & s_z \end{bmatrix} \tag{18} $$

Many optimisation algorithms exist but the gradient descent algorithm is one of the simplest to both implement and compute. Equations (19) describes the gradient descent algorithm for n iterations resulting in an orientation estimation of ${}^{S}_{E}\hat{\boldsymbol{q}}_{n+1}$ based on an 'initial guess' orientation ${}^{S}_{E}\hat{\boldsymbol{q}}_{0}$ and a step-size μ. Equation (20) computes the gradient of the solution surface defined by the objective function and its Jacobian; simplified to the 3 row vectors defined by equations (21) and (22) respectively.

$$ {}^{S}_{E}\boldsymbol{q}_{k+1} = {}^{S}_{E}\hat{\boldsymbol{q}}_{k} - \mu \frac{\nabla \boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}})}{\left\| \nabla \boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}) \right\|}, \ k = 0, 1, 2...n \tag{19} $$

$$ \nabla \boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}) = \boldsymbol{J}^{T}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}) \boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}) \tag{20} $$

$$ \boldsymbol{f}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}}) = \begin{bmatrix} 2d_x(\frac{1}{2} - q_3^2 - q_4^2) + 2d_y(q_1q_4 + q_2q_3) + \\ 2d_x(q_2q_3 - q_1q_4) + 2d_y(\frac{1}{2} - q_2^2 - q_4^2) + \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + \\ \qquad 2d_z(q_2q_4 - q_1q_3) - s_x \\ \qquad 2d_z(q_1q_2 + q_3q_4) - s_y \\ \qquad 2d_z(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix} \tag{21} $$

$$ \boldsymbol{J}({}^{S}_{E}\hat{\boldsymbol{q}}_{k}, {}^{E}\hat{\boldsymbol{d}}) = \begin{bmatrix} 2d_yq_4 - 2d_zq_3 & 2d_yq_3 + 2d_zq_4 & -4d_xq_3 + 2d_yq_2 - 2d_zq_1 & -4d_xq_4 + 2d_yq_1 + 2d_zq_2 \\ -2d_xq_4 + 2d_zq_2 & 2d_xq_3 - 4d_yq_2 + 2d_zq_1 & 2d_xq_2 + 2d_zq_4 & -2d_xq_1 - 4d_yq_4 + 2d_zq_3 \\ 2d_xq_3 - 2d_yq_2 & 2d_xq_4 - 2d_yq_1 - 4d_zq_2 & 2d_xq_1 + 2d_yq_4 - 4d_zq_3 & 2d_xq_2 + 2d_yq_3 \end{bmatrix} \tag{22} $$

Equations (19) to (22) describe the general form of the algorithm applicable to a field predefined in any direction. However, if the direction of the field can be assumed to only have components within 1 or 2 of the principle axis of the global coordinate frame then the equations simplify. An appropriate convention would be to assume that the direction of gravity defines the vertical, z axis as shown in equation (23). Substituting ${}^{E}\hat{\boldsymbol{g}}$ and normalised

accelerometer measurement ${}^{S}\hat{\boldsymbol{a}}$ for ${}^{E}\hat{\boldsymbol{d}}$ and ${}^{S}\hat{\boldsymbol{s}}$ respectively, in equations (21) and (22) yields equations (25) and (26).

$$ {}^{E}\hat{\boldsymbol{g}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \tag{23} $$

$$ {}^{S}\hat{\boldsymbol{a}} = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix} \tag{24} $$

$$ \boldsymbol{f}_g({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 + q_3q_4) - a_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - a_z \end{bmatrix} \tag{25} $$

$$ \boldsymbol{J}_g({}^{S}_{E}\hat{\boldsymbol{q}}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \tag{26} $$

The earth's magnetic field can be considered to have components in one horizontal axis and the vertical axis; the vertical component due to the inclination of the field which is between 65° and 70° to the horizontal in the UK [39]. This can be represented by equation (27). Substituting ${}^{E}\hat{\boldsymbol{b}}$ and normalised magnetometer measurement ${}^{S}\hat{\boldsymbol{m}}$ for ${}^{E}\hat{\boldsymbol{d}}$ and ${}^{S}\hat{\boldsymbol{s}}$ respectively, in equations (21) and (22) yields equations (29) and (30).

$$ {}^{E}\hat{\boldsymbol{b}} = \begin{bmatrix} 0 & b_x & 0 & b_z \end{bmatrix} \tag{27} $$

$$ {}^{S}\hat{\boldsymbol{m}} = \begin{bmatrix} 0 & m_x & m_y & m_z \end{bmatrix} \tag{28} $$

$$ \boldsymbol{f}_b({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}, {}^{S}\hat{\boldsymbol{m}}) = \begin{bmatrix} 2b_x(0.5 - q_3^2 - q_4^2) + 2b_z(q_2q_4 - q_1q_3) - m_x \\ 2b_x(q_2q_3 - q_1q_4) + 2b_z(q_1q_2 + q_3q_4) - m_y \\ 2b_x(q_1q_3 + q_2q_4) + 2b_z(0.5 - q_2^2 - q_3^2) - m_z \end{bmatrix} \tag{29} $$

$$ \boldsymbol{J}_b({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}) = \begin{bmatrix} -2b_zq_3 & 2b_zq_4 & -4b_xq_3 - 2b_zq_1 & -4b_xq_4 + 2b_zq_2 \\ -2b_xq_4 + 2b_zq_2 & 2b_xq_3 + 2b_zq_1 & 2b_xq_2 + 2b_zq_4 & -2b_xq_1 + 2b_zq_3 \\ 2b_xq_3 & 2b_xq_4 - 4b_zq_2 & 2b_xq_1 - 4b_zq_3 & 2b_xq_2 \end{bmatrix} \tag{30} $$

As has already been discussed, the measurement of gravity or the earth's magnetic field alone will not provide a unique orientation of the sensor. To do so, the measurements and reference directions of both fields may be combined as described by equations (31) and (32). Whereas the solution surface created by the objective functions in equations (25) and (29) have a minimum defined by a line, the solution surface define by equation (31) has a minimum define by a single point, provided that $b_x \neq 0$.

$$\boldsymbol{f}_{g,b}({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}, {}^{E}\hat{\boldsymbol{b}}, {}^{S}\hat{\boldsymbol{m}}) = \begin{bmatrix} \boldsymbol{f}_g({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{S}\hat{\boldsymbol{a}}) \\ \boldsymbol{f}_b({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}, {}^{S}\hat{\boldsymbol{m}}) \end{bmatrix} \tag{31}$$

$$\boldsymbol{J}_{g,b}({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}) = \begin{bmatrix} \boldsymbol{J}_g^T({}^{S}_{E}\hat{\boldsymbol{q}}) \\ \boldsymbol{J}_b^T({}^{S}_{E}\hat{\boldsymbol{q}}, {}^{E}\hat{\boldsymbol{b}}) \end{bmatrix} \tag{32}$$

A conventional approach to optimisation would require multiple iterations of equation (19) to be computed for each new orientation and corresponding senor measurements. Efficient algorithms would also require the step-size μ to be adjusted each iteration to an optimal value; usually obtained based on the second derivative of the objective function, the Hessian. However, these requirements considerably increase the computational load of the algorithm and are not necessary in this application. It is acceptable to compute one iteration per time sample provided that the convergence rate governed by μ_t is equal or greater than the physical rate of change of orientation. Equation (33) calculates the estimated orientation ${}^{S}_{E}\boldsymbol{q}_{\nabla,t}$ computed at time t based on a previous estimate of orientation ${}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1}$ and the objective function gradient $\nabla\boldsymbol{f}$ defined by sensor measurements ${}^{S}\hat{\boldsymbol{a}}_t$ and ${}^{S}\hat{\boldsymbol{m}}_t$ sampled at time t. The form of $\nabla\boldsymbol{f}$ is chosen according to the sensors in use, as shown in equation (34). The sub-script ∇ indicates that the quaternion is calculated using the gradient descent algorithm.

$${}^{S}_{E}\boldsymbol{q}_{\nabla,t} = {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1} - \mu_t \frac{\nabla\boldsymbol{f}}{\|\nabla\boldsymbol{f}\|} \tag{33}$$

$$\nabla\boldsymbol{f} = \begin{cases} \boldsymbol{J}_g^T({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1})\boldsymbol{f}_g({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1}, {}^{S}\hat{\boldsymbol{a}}_t) \\ \boldsymbol{J}_{g,b}^T({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1}, {}^{E}\hat{\boldsymbol{b}})\boldsymbol{f}_{g,b}({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1}, {}^{S}\hat{\boldsymbol{a}}, {}^{E}\hat{\boldsymbol{b}}, {}^{S}\hat{\boldsymbol{m}}) \end{cases} \tag{34}$$

An optimal value of μ_t can be defined as that which ensures the convergence rate of ${}^{S}_{E}\boldsymbol{q}_{\nabla,t}$ is limited to the physical orientation rate as this avoids overshooting due an unnecessarily large step size. Therefore μ_t can be calculated as equation (35) where Δt is the sampling period and ${}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t}$ is the

physical orientation rate measured by gyroscopes and α is an augmentation of μ to account for noise in accelerometer and magnetometer measurements.

$$\mu_t = \alpha \left\| {}^S_E\dot{\boldsymbol{q}}_{\omega,t} \right\| \Delta t,\ \alpha > 1 \tag{35}$$

3.3 Filter fusion algorithm

An estimated orientation of the sensor frame relative to the earth frame, ${}^S_E\boldsymbol{q}_{est,t}$, is obtained through the fusion of the orientation calculations, ${}^S_E\boldsymbol{q}_{\omega,t}$ and ${}^S_E\boldsymbol{q}_{\nabla,t}$; calculated using equations (13) and (33) respectively. The fusion of ${}^S_E\hat{\boldsymbol{q}}_{\omega,t}$ and ${}^S_E\boldsymbol{q}_{\nabla,t}$ is described by equation (36) where γ_t and $(1-\gamma_t)$ are weights applied to each orientation calculation.

$${}^S_E\boldsymbol{q}_{est,t} = \gamma_t {}^S_E\boldsymbol{q}_{\nabla,t} + (1-\gamma_t){}^S_E\boldsymbol{q}_{\omega,t},\ 0 \leq \gamma_t \leq 1 \tag{36}$$

An optimal value of γ_t can be defined as that which ensures the weighted divergence of ${}^S_E\boldsymbol{q}_{\omega}$ is equal to the weighted convergence of ${}^S_E\boldsymbol{q}_{\nabla}$. This is represented by equation (37) where $\frac{\mu_t}{\Delta t}$ is the convergence rate of ${}^S_E\boldsymbol{q}_{\nabla}$ and β is the divergence rate of ${}^S_E\boldsymbol{q}_{\omega}$ expressed as the magnitude of a quaternion derivative corresponding to the gyroscope measurement error. Equation (37) can be rearranged to define γ_t as equation (38).

$$(1-\gamma_t)\beta = \gamma_t \frac{\mu_t}{\Delta t} \tag{37}$$

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta} \tag{38}$$

Equations (36) and (38) ensure the optimal fusion of ${}^S_E\boldsymbol{q}_{\omega,t}$ and ${}^S_E\boldsymbol{q}_{\nabla,t}$ assuming that the convergence rate of ${}^S_E\boldsymbol{q}_{\nabla}$ governed by α is equal or greater than the physical rate of change of orientation. Therefore α has no upper bound. If α is assumed to be very large then μ_t, defined by equation (35), also becomes very large and the orientation filter equations simplify. A large value of μ_t used in equation (33) means that ${}^S_E\hat{\boldsymbol{q}}_{est,t\text{-}1}$ becomes negligible and the equation can be re-written as equation (39).

$${}^S_E\boldsymbol{q}_{\nabla,t} \approx -\mu_t \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \tag{39}$$

The definition of γ_t in equation (38) also simplifies as the β term in the denominator becomes negligible and the equation can be rewritten as equation (40). It is possible from equation (40) to also assume that $\gamma_t \approx 0$.

$$\gamma_t \approx \frac{\beta \Delta t}{\mu_t} \tag{40}$$

Substituting equations (13), (39) and (40) into equation (36) directly yields equation (41). It is important to note that in equation (41), γ_t has been substituted as both as equation (39) and 0.

$$ {}^{S}_{E}\boldsymbol{q}_{est,t} = \frac{\beta \Delta t}{\mu_t}\left(-\mu_t \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|}\right) + (1-0)\left({}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1} + {}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t}\Delta t\right) \tag{41}$$

Equation (41) can be simplified to equation (42) where ${}^{S}_{E}\dot{\boldsymbol{q}}_{est,t}$ is the estimated rate of change of orientation defined by equation (43) and ${}^{S}_{E}\dot{\hat{\boldsymbol{q}}}_{\epsilon,t}$ is the direction of the error of ${}^{S}_{E}\dot{\boldsymbol{q}}_{est,t}$ defined by equation (44).

$$ {}^{S}_{E}\boldsymbol{q}_{est,t} = {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1} + {}^{S}_{E}\dot{\boldsymbol{q}}_{est,t}\Delta t \tag{42}$$

$$ {}^{S}_{E}\dot{\boldsymbol{q}}_{est,t} = {}^{S}_{E}\dot{\boldsymbol{q}}_{\omega,t} - \beta {}^{S}_{E}\dot{\hat{\boldsymbol{q}}}_{\epsilon,t} \tag{43}$$

$$ {}^{S}_{E}\dot{\hat{\boldsymbol{q}}}_{\epsilon,t} = \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} \tag{44}$$

It can be seen from equations (42) to (44) that the filter calculates the orientation ${}^{S}_{E}\boldsymbol{q}_{est}$ by numerically integrating the estimated orientation rate ${}^{S}_{E}\dot{\boldsymbol{q}}_{est}$. The filter computes ${}^{S}_{E}\dot{\boldsymbol{q}}_{est}$ as the rate of change of orientation measured by the gyroscopes, ${}^{S}_{E}\dot{\boldsymbol{q}}_{\omega}$, with the magnitude of the gyroscope measurement error, β, removed in the direction of the estimated error, ${}^{S}_{E}\dot{\hat{\boldsymbol{q}}}_{\epsilon}$, computed from accelerometer and magnetometer measurements. Figure 2 shows a block diagram representation of the complete orientation filter implementation for an IMU.

3.4 Magnetic distortion compensation

Measurements of the earth's magnetic field will be distorted by the presence of ferromagnetic elements in the vicinity of the magnetometer. Investigations into the effect of magnetic distortions on an orientation sensor's performance

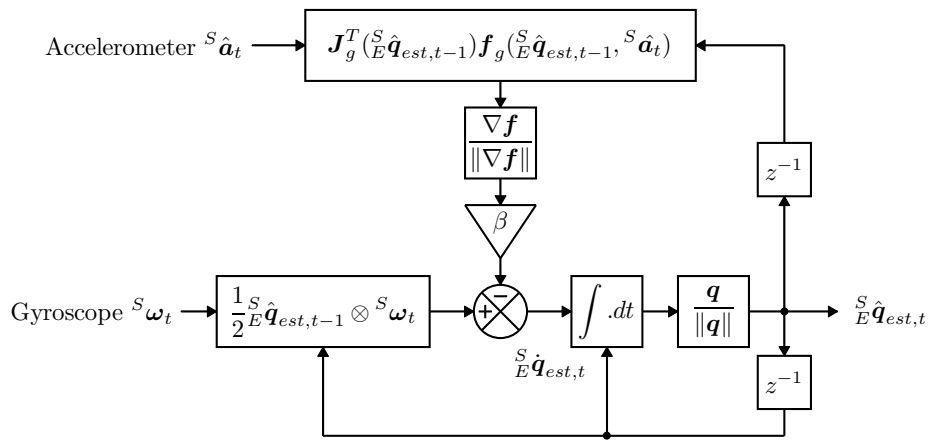

Figure 2: Block diagram representation of the complete orientation filter for an IMU implementation

have shown that substantial errors may be introduced by sources including electrical appliances, metal furniture and metal structures within a buildings construction [40, 41]. Sources of interference fixed in the sensor frame, termed hard iron biases, can be removed through calibration [42, 43, 44, 45]. Sources of interference in the earth frame, termed soft iron, cause errors in the measured direction of the earth's magnetic field. Declination errors, those in the horizontal plane relative to the earth's surface, cannot be corrected without an additional reference of heading. Inclination errors, those in the vertical plane relative to the earth's surface, may be compensated for as the accelerometer provides an additional measurement of the sensor's attitude.

The measured direction of the earth's magnetic field in the earth frame at time t, ${}^{E}\hat{\boldsymbol{h}}_t$, can be computed as the normalised magnetometer measurement, ${}^{S}\hat{\boldsymbol{m}}_t$, rotated by the estimated orientation of the sensor provided by the filter, ${}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1}$; as described by equation (45). The effect of an erroneous inclination of the measured direction earth's magnetic field, ${}^{E}\hat{\boldsymbol{h}}_t$, can be corrected if the filter's reference direction of the earth's magnetic field, ${}^{E}\hat{\boldsymbol{b}}_t$, is of the same inclination. This is achieved by computing ${}^{E}\hat{\boldsymbol{b}}_t$ as ${}^{E}\hat{\boldsymbol{h}}_t$ normalised to have only components in the earth frame x and z axes; as described by equation (46).

$$ {}^{E}\hat{\boldsymbol{h}}_t = \begin{bmatrix} 0 & h_x & h_y & h_z \end{bmatrix} = {}^{S}_{E}\hat{\boldsymbol{q}}_{est,t\text{-}1} \otimes {}^{S}\hat{\boldsymbol{m}}_t \otimes {}^{S}_{E}\hat{\boldsymbol{q}}^{*}_{est,t\text{-}1} \tag{45} $$

$$^{E}\hat{\boldsymbol{b}}_t = \begin{bmatrix} 0 & \sqrt{h_x^2 + h_y^2} & 0 & h_z \end{bmatrix} \tag{46}$$

Compensating for magnetic distortions in this way ensures that magnetic disturbances are limited to only affect the estimated heading component of orientation. The approach also eliminates the need for the reference direction of the earth's magnetic field to be predefined; a potential disadvantage of other orientation filter designs [17, 24].

3.5 Gyroscope bias drift compensation

The gyroscope zero bias will drift over time, with temperature and with motion. Any practical implementation of an IMU or MARG sensor array must account for this. An advantage of Kalman-based approaches is that they are able to estimate the gyroscope bias as an additional state within the system model [26, 30, 15, 24]. However, Mahony *et al* [35] showed that gyroscope bias drift may also be compensated for by simpler orientation filters through the integral feedback of the error in the rate of change of orientation. A similar approach will be used here.

The normalised direction of the estimated error in the rate of change of orientation, ${}^{S}_{E}\dot{\hat{\boldsymbol{q}}}_{\epsilon}$, may be expressed as the angular error in each gyroscope axis using equation (47); derived as the inverse to the relationship defined in equation (11). The gyroscope bias, ${}^{S}\boldsymbol{\omega}_b$, is represented by the DC component of ${}^{S}\boldsymbol{\omega}_{\epsilon}$ and so may removed as the integral of ${}^{S}\boldsymbol{\omega}_{\epsilon}$ weighted by an appropriate gain, ζ. This would yield the compensated gyroscope measurements ${}^{S}\boldsymbol{\omega}_c$, as shown in equations (48) and (49). The first element of ${}^{S}\boldsymbol{\omega}_c$ is always assumed to be 0.

$$^{S}\boldsymbol{\omega}_{\epsilon,t} = 2{}^{S}_{E}\hat{\boldsymbol{q}}^{*}_{est,t\text{-}1} \otimes {}^{S}_{E}\dot{\hat{\boldsymbol{q}}}_{\epsilon,t} \tag{47}$$

$$^{S}\boldsymbol{\omega}_{b,t} = \zeta \sum_{t} {}^{S}\boldsymbol{\omega}_{\epsilon,t}\Delta t \tag{48}$$

$$^{S}\boldsymbol{\omega}_{c,t} = {}^{S}\boldsymbol{\omega}_{t} - {}^{S}\boldsymbol{\omega}_{b,t} \tag{49}$$

The compensated gyroscope measurements, ${}^{S}\boldsymbol{\omega}_c$, may then be used in place of the of the gyroscope measurements, ${}^{S}\boldsymbol{\omega}$, in equation (11). The magnitude of the angular error in each axis, ${}^{S}\boldsymbol{\omega}_{\epsilon}$ is equal to a quaternion derivative of unit length. Therefore the integral gain ζ directly defines the

rate of convergence of the estimated gyroscope bias, ${}^{S}\boldsymbol{\omega}_b$, expressed as the magnitude of a quaternion derivative. As this process requires the use of the filter estimate of a *complete* orientation, ${}^{S}_{E}\hat{\boldsymbol{q}}_{est}$, it is only applicable to a MARG implementation of the filter. Figure 3 shows a block diagram representation of the complete filter implementation for a MARG sensor array, including the magnetic distortion and gyroscope bias drift compensation.

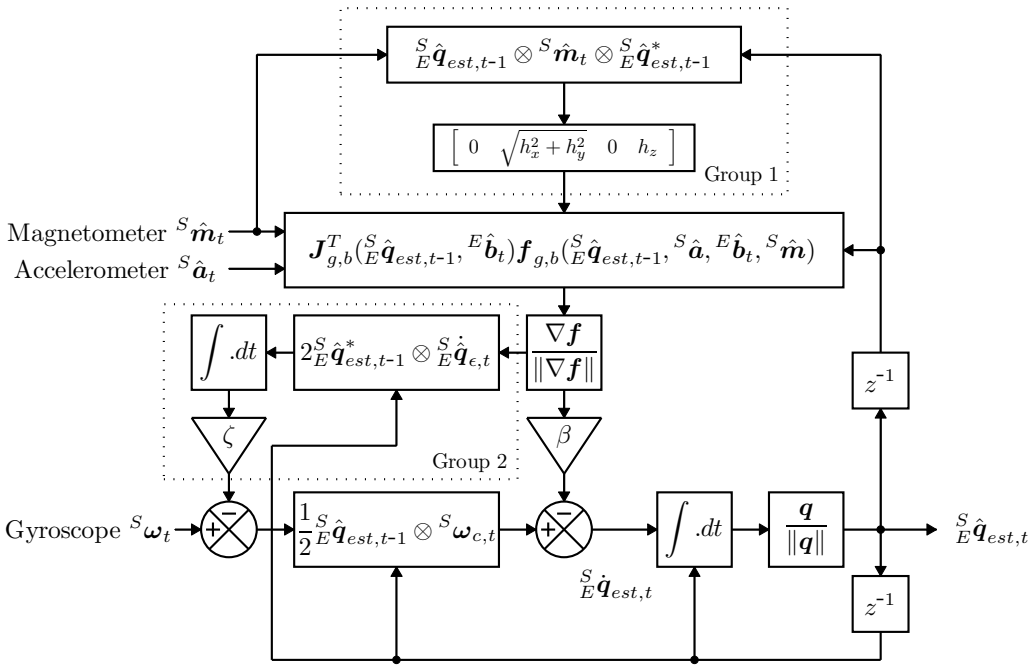

Figure 3: Block diagram representation of the complete orientation filter for an MARG implementation including magnetic distortion (Group 1) and gyroscope drift (Group 2) compensation

3.6 Filter gains

The filter gain β represents all mean zero gyroscope measurement errors, expressed as the magnitude of a quaternion derivative. The sources of error include: sensor noise, signal aliasing, quantisation errors, calibration errors, sensor miss-alignment, sensor axis non-orthogonality and frequency response characteristics. The filter gain ζ represents the rate of convergence to remove gyroscope measurement errors which are not mean zero, also expressed as the

magnitude of a quaternion derivative. These errors represent the gyroscope bias. It is convenient to define β and ζ using the angular quantities ω_β and $\dot{\omega}_\zeta$ respectively, where $\tilde{\omega}_\beta$ represents the estimated mean zero gyroscope measurement error of each axis and $\dot{\omega}_\zeta$ represents the estimated rate of gyroscope bias drift in each axis. Using the relationship described by equation (11), β may be defined by equation (50) where $\hat{\boldsymbol{q}}$ is any unit quaternion. Similarly, ζ may be described by equation (51).

$$\beta = \left\| \frac{1}{2}\hat{\boldsymbol{q}} \otimes \begin{bmatrix} 0 & \tilde{\omega}_\beta & \tilde{\omega}_\beta & \tilde{\omega}_\beta \end{bmatrix} \right\| = \sqrt{\frac{3}{4}}\tilde{\omega}_\beta \tag{50}$$

$$\zeta = \sqrt{\frac{3}{4}}\tilde{\dot{\omega}}_\zeta \tag{51}$$

4 Experimentation

4.1 Equipment

The filter was tested using the xsens MTx orientation sensor [18] containing 16 bit resolution tri-axis gyroscopes, accelerometers and magnetometers. The device and accompanying software offer a mode of operations where raw sensor data may be logged at a rate of 512 Hz and then post-processed to provide calibrated sensor measurements. The calibrated sensor measurements could then be processed by the proposed filter to provide the estimated orientation of the sensor. The software also incorporates a propriety Kalman-based orientation filter to provide an additional estimate orientation. As both the Kalman-based algorithm and proposed filter's outputs could be computed using identical sensor data, the performance of each algorithm could be evaluated relative to one-another, independent of sensor performance.

A Vicon system, consisting of 8 MX3+ cameras connected to an MX-ultranet server [46] and Nexus [47] software was used to provide reference measurement of the orientation sensor's actual orientation. The system is an array of IR (Infrared) sensitive cameras with incorporated IR flood lights. The cameras are fixed at calibrated positions and orientations so that the measurement subject is within the field of view of multiple cameras. The Cartesian positions of IR reflective optical markers fixed to the measurement subject may then be computed in the coordinate frame of the camera array. Cameras were fixed at a height of approximately 2.5 m, evenly distributed

around the perimeter of a 4 m by 4 m enclosure. Each camera was orientated to face toward the centre of the room, approximately 30° to 60° to horizontal. Experiments were conducted with the measurement subject in the centre of the room at a height of approximately 1 m. To measure the orientation of the sensor, it was fixed to an optical orientation measurement platform specifically designed for this application. The system was used to log the positions of optical markers at a rate of 120 Hz.

4.2 Orientation from optical measurements

The orientation measurement platform is comprised of 3 500 mm, mutually orthogonal rods, rigidly connected at the central position along each length. Optical markers were positioned at both ends of each rod and the orientation sensor fixed to a platform at the point where the rods coincide. The platform was constructed from an aluminium central hub, carbon fibre rods and assembled using adhesives to ensure that it had no magnetic properties that may interfere with the orientation sensor's magnetometer. Additional optical markers were placed at arbitrary but dissimilar positions along the lengths of the rods to break the rotational symmetry and aid the identification of each rod within the measurement data. Figure 4 shows an annotated photograph of the orientation measurement platform where ${}^{C}\boldsymbol{i}_{start}$, ${}^{C}\boldsymbol{i}_{end}$, ${}^{C}\boldsymbol{j}_{start}$, ${}^{C}\boldsymbol{j}_{end}$, ${}^{C}\boldsymbol{k}_{start}$ and ${}^{C}\boldsymbol{k}_{end}$ define the measured position of each marker within the camera frame. These positions may be used to define 3 mutually orthogonal unit vectors, ${}^{C}\hat{\boldsymbol{x}}_M$, ${}^{C}\hat{\boldsymbol{y}}_M$, and ${}^{C}\hat{\boldsymbol{z}}_M$ within the camera frame representing the direction the x, y and z axes of the orientation measurement platform coordinate frame; as described by equations (52), (53) and (54). These vectors define the rotation matrix describing the orientation of the measurement platform in the camera frame; as shown in equation (55).

$$ {}^{C}\hat{\boldsymbol{x}}_M = \frac{{}^{C}\boldsymbol{i}_{end} - {}^{C}\boldsymbol{i}_{start}}{\|{}^{C}\boldsymbol{i}_{end} - {}^{C}\boldsymbol{i}_{start}\|} \tag{52} $$

$$ {}^{C}\hat{\boldsymbol{y}}_M = \frac{{}^{C}\boldsymbol{j}_{end} - {}^{C}\boldsymbol{j}_{start}}{\|{}^{C}\boldsymbol{j}_{end} - {}^{C}\boldsymbol{j}_{start}\|} \tag{53} $$

$$ {}^{C}\hat{\boldsymbol{z}}_M = \frac{{}^{C}\boldsymbol{k}_{end} - {}^{C}\boldsymbol{k}_{start}}{\|{}^{C}\boldsymbol{k}_{end} - {}^{C}\boldsymbol{k}_{start}\|} \tag{54} $$

$$ {}^{C}_{M}\boldsymbol{R} = \begin{bmatrix} {}^{C}\hat{\boldsymbol{x}}_M & {}^{C}\hat{\boldsymbol{y}}_M & {}^{C}\hat{\boldsymbol{z}}_M \end{bmatrix} \tag{55} $$

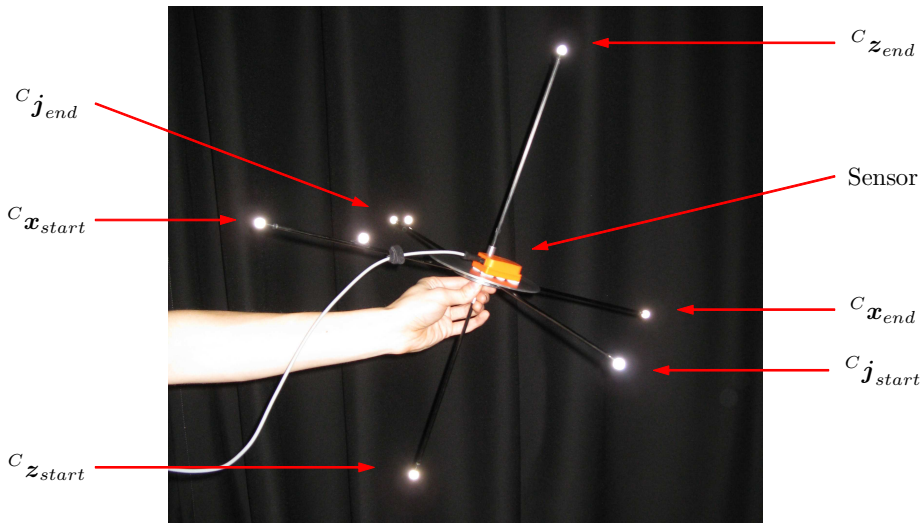

Figure 4: Photograph of the orientation measurement platform

Due to measurement errors and tolerances in the marker frame construction, the rotation matrix defined by equation (55) cannot be considered orthogonal and so does not represent a pure rotation. Bar-Itzhack provides a method [48] where by an optimal 'best fit' quaternion may be extracted from an imprecise and non-orthogonal rotation matrix. The method requires the construction of the symmetric 4 by 4 matrix, $\boldsymbol{K}$, defined by equation (56), where r_{mn} corresponds to the element of the mth row and n^{th} column of ${}^{C}_{M}\boldsymbol{R}$. The optimal quaternion ${}^{C}_{M}\hat{\boldsymbol{q}}$ is found as the normalised Eigen vector corresponding to the maximum Eigen value of $\boldsymbol{K}$. Equation (57) defines the optimal quaternion accounting the alternative quaternion element order convention assumed by the method, where v_1, v_2, v_3 and v_4 define elements of the normalised Eigen vector.

$$\boldsymbol{K} = \frac{1}{3}\begin{bmatrix} r_{11}-r_{22}-r_{33} & r_{21}+r_{12} & r_{31}+r_{13} & r_{23}-r_{32} \\ r_{21}+r_{12} & r_{22}-r_{11}-r_{33} & r_{32}+r_{23} & r_{31}-r_{13} \\ r_{31}+r_{13} & r_{32}+r_{23} & r_{33}-r_{11}-r_{22} & r_{12}-r_{21} \\ r_{23}-r_{32} & r_{31}-r_{13} & r_{12}-r_{21} & r_{11}+r_{22}+r_{33} \end{bmatrix} \tag{56}$$

$${}^{C}_{M}\hat{\boldsymbol{q}} = \begin{bmatrix} v_4 & v_1 & v_2 & v_3 \end{bmatrix} \tag{57}$$

4.3 Calibration of frame alignments

In order to compare the optical measurement of the platform orientation in the camera frame, ${}^{C}_{M}\hat{\boldsymbol{q}}$, and the orientation filter's estimated orientation of the earth in the sensor frame, ${}^{S}_{E}\hat{\boldsymbol{q}}_{est}$, it is necessary to know the alignment of the earth frame relative to the camera frame, ${}^{C}_{E}\hat{\boldsymbol{q}}$, and the alignment of the measurement platform relative to the sensor frame, ${}^{S}_{M}\hat{\boldsymbol{q}}$. Once these quantities are found the optical measurement of the sensor frame orientation in the earth frame, ${}^{S}_{E}\hat{\boldsymbol{q}}_{meas}$, may be defined by equation (58). Although the use of optical measurement equipment in this application is documentaed [26, 24, 41], little discussion is offered on the calibration of these two quantities.

$$ {}^{S}_{E}\hat{\boldsymbol{q}}_{meas} = {}^{C}_{E}\hat{\boldsymbol{q}} \otimes {}^{M}_{C}\hat{\boldsymbol{q}} \otimes {}^{S}_{M}\hat{\boldsymbol{q}} \tag{58} $$

The earth frame x and z axes are defined by the earth's magnetic and gravitational fields respectively. Measurements of these fields in the camera frame can be used to define the alignment ${}^{C}_{E}\hat{\boldsymbol{q}}$. The direction of gravity was measured using a pendulum constructed from a 1 m length of cotton thread with a small weight fixed to one end. Optical markers where fixed at either end of the length of cotton and the pendulum left to come to rest. Additional optical markers were required at arbitrary fixed positions relative to the static pendulum to break the rotational symmetry of the optical marker constellation. Figure 5 shows an annotated photograph of the pendulum where ${}^{C}\boldsymbol{p}_{start}$ and ${}^{C}\boldsymbol{p}_{end}$ define the position of the optical markers in the camera frame. The mean position of each marker over a period of time defines the direction of the pendulum in the camera frame, ${}^{C}\hat{\boldsymbol{p}}$. This directly defines the earth frame z axis in the camera frame, ${}^{C}\hat{\boldsymbol{z}}_{E}$; as shown in equation (59).

$$ {}^{C}\hat{\boldsymbol{p}} = \frac{{}^{C}\bar{\boldsymbol{p}}_{end} - {}^{C}\bar{\boldsymbol{p}}_{start}}{\|{}^{C}\bar{\boldsymbol{p}}_{end} - {}^{C}\bar{\boldsymbol{p}}_{start}\|} = {}^{C}\hat{\boldsymbol{z}}_{E} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \tag{59} $$

The direction of earth's magnetic field was measured using a magnetic compass constructed from a 1 m carbon-fibre rod with neodymium magnets fixed to each end; polarising each end to either magnetic north or south. The compass was hung from a 1 m length of cotton thread and left to come to rest. Optical markers were fixed to either end of the rod as well as to an arbitrary position along the length, but off-set form the rod's axis, to beak the rotational symmetry optical marker constellation. Figure 60 shows

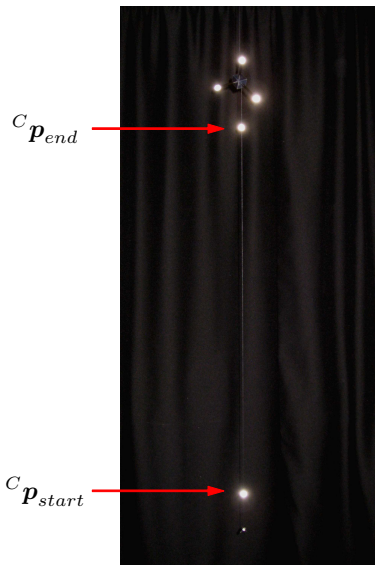

Figure 5: Photograph of the pendulum and optical markers used to measure the direction of gravity in the camera frame

an annotated photograph of the compass where ${}^{C}\boldsymbol{c}_{start}$ and ${}^{C}\boldsymbol{c}_{end}$ define the position of the optical markers in the camera frame. The mean position of each marker over a period of time define the direction of the compass in the camera frame, ${}^{C}\hat{\boldsymbol{c}}$; as shown in equation (60).

$$ {}^{C}\hat{\boldsymbol{c}} = \frac{{}^{C}\bar{\boldsymbol{c}}_{end} - {}^{C}\bar{\boldsymbol{c}}_{start}}{\|{}^{C}\bar{\boldsymbol{c}}_{end} - {}^{C}\bar{\boldsymbol{c}}_{start}\|} \tag{60} $$

Due to measurement errors and the imbalance of the suspended magnetic compass, ${}^{C}\hat{\boldsymbol{c}}$ cannot be assumed to be orthogonal to the direction of gravity defined by ${}^{C}\hat{\boldsymbol{p}}$ and so cannot be used to directly define the earth x axis. The non-orthogonal component of ${}^{C}\hat{\boldsymbol{c}}$ can be computed as the vector projection of ${}^{C}\hat{\boldsymbol{c}}$ on ${}^{C}\hat{\boldsymbol{p}}$. This can then be removed from ${}^{C}\hat{\boldsymbol{c}}$ to define the the earth x axis direction in the camera frame, ${}^{C}\boldsymbol{x}_{E}$; as shown in equation (61). Once normalised, this defines the earth x axis; ${}^{C}\hat{\boldsymbol{x}}_{E}$, as shown in equation (62).

$$ {}^{C}\boldsymbol{x}_{E} = {}^{C}\hat{\boldsymbol{c}} - \frac{{}^{C}\hat{\boldsymbol{c}} \cdot {}^{C}\hat{\boldsymbol{p}}}{\|{}^{C}\hat{\boldsymbol{p}}\|^{2}} {}^{C}\hat{\boldsymbol{p}} \tag{61} $$

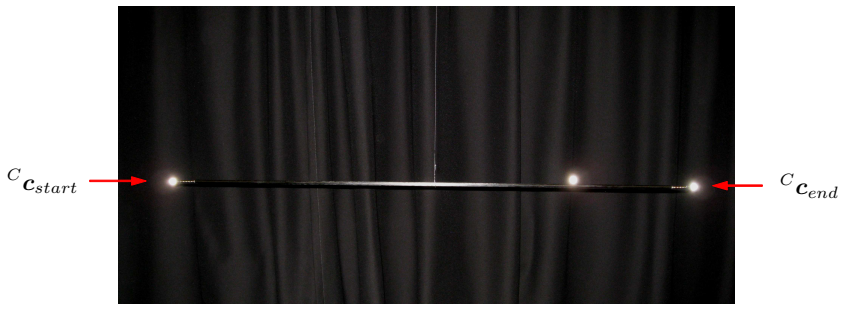

Figure 6: Photograph of the magnetic compass and optical markers used to measure the direction of the earth's magnetic field in the camera frame

$$ {}^{C}\hat{\boldsymbol{x}}_E = \frac{{}^{C}\boldsymbol{x}_E}{\|{}^{C}\boldsymbol{x}_E\|} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{62} $$

The earth frame y axis in the camera frame, ${}^{C}\hat{\boldsymbol{y}}_E$, can be calculated as the vector that is orthogonal to both ${}^{C}\hat{\boldsymbol{x}}_E$ and ${}^{C}\hat{\boldsymbol{z}}_E$ and so be defined by equation (63) where signs are chosen to satisfy the relative axis direction direction convention. The alignment of the earth frame may be defined as the rotation matrix ${}^{C}_{E}\boldsymbol{R}$, constructed from ${}^{C}\hat{\boldsymbol{x}}_E$, ${}^{C}\hat{\boldsymbol{y}}_E$, and ${}^{C}\hat{\boldsymbol{z}}_E$. The quaternion representation, ${}^{C}_{E}\hat{\boldsymbol{q}}$, can then be extracted form this using Bar-Itzhack's method [48].

$$ {}^{C}\hat{\boldsymbol{y}}_E = \begin{bmatrix} \pm\sqrt{1 - x_1^2 - z_1^2} \\ \pm\sqrt{1 - x_2^2 - z_2^2} \\ \pm\sqrt{1 - x_3^2 - z_3^2} \end{bmatrix}, \; {}^{C}\hat{\boldsymbol{x}}_E \cdot {}^{C}\hat{\boldsymbol{y}}_E = 0, \; {}^{C}\hat{\boldsymbol{z}}_E \cdot {}^{C}\hat{\boldsymbol{y}}_E = 0 \tag{63} $$

$$ {}^{C}_{E}R = \begin{bmatrix} {}^{C}\hat{\boldsymbol{x}}_E & {}^{C}\hat{\boldsymbol{y}}_E & {}^{C}\hat{\boldsymbol{z}}_E \end{bmatrix} \tag{64} $$

In order to find the alignment ${}^{S}_{M}\hat{\boldsymbol{q}}$ it is assumed that the static error of the orientation filter's Kalman-based algorithm is mean zero. The mean algorithm's output, ${}^{S}_{E}\hat{\hat{\boldsymbol{q}}}_{Kalman}$, was computed for the measurement platform held stationary for a period of approximate 10 seconds. This was used with the alignment ${}^{E}_{C}\hat{\boldsymbol{q}}$ and optical measurement ${}^{C}_{M}\hat{\boldsymbol{q}}$ to define the alignment of the measurement platform in the sensor frame, ${}^{S}_{M}\hat{\boldsymbol{q}}$, as equation (65).

$$ {}^{S}_{M}\hat{\boldsymbol{q}} = {}^{C}_{M}\hat{\boldsymbol{q}} \otimes {}^{E}_{C}\hat{\boldsymbol{q}} \otimes {}^{S}_{E}\hat{\hat{\boldsymbol{q}}}_{Kalman} \tag{65} $$

4.4 Experimental proceedure

The optical measurement data and raw orientation sensor data were logged simultaneously. The raw orientation sensor data was then processed by the accompanying software to provide the calibrated sensor data and Kalman-based algorithm output. This data then synchronised the optical measurement data, with the optical measurement data interpolated to match the higher sampling rate of the orientation sensor data. The calibrated sensor data was then processed through both the IMU and MARG implementations of the proposed orientation filter and calibrated orientation measurements were extracted from the optical measurement data using the methods described in sections 4.2 and 4.3.

The proposed filter's gain β was set to 0.033 for the IMU implementation and 0.041 for the MARG implementation. Trials summarised in section 5.3, found these values to provide optimal performance. However, an initial value of 2.5 was used for the first 10 seconds of any experiment to ensure the convergence of algorithm states from initial conditions. The gain ζ, applicable to the MARG implementation of the proposed filer, was set to 0 as the calibrated orientation sensor data was not subject to gyroscope bias drift.

Data was obtained for a sequence of rotations preformed by hand. The measurement platform was initially held stationary for 20 to 30 seconds to allow time for algorithm states to converge to steady-state values. The platform was then rotated 90° around its x axis, then 180° in the opposite direction, and then 90° to bring the platform back to the starting position. The platform was held stationary for 3 to 5 seconds between each rotation. This sequence was then repeated around the y and then z axes. The peak angular rate measured during each rotation was between 110°/s and 190°/s. The experiment was repeated 8 times to compile a dataset representative of system performance.

5 Results

It is common [24, 26, 18, 19, 20, 21] to quantify orientation sensor performance as the static and dynamic RMS (Root-Mean-Square) errors in the decoupled Euler parameters describing the pitch, roll, and heading components of an orientation. Pitch, ϕ, roll, θ and heading, ψ correspond to rotations around the sensor frame x, y, and z axis respectively. An Euler angle

representation has the advantage that the decoupled angles may be more easily interpreted or visualised. The disadvantage of an Euler representation is that it fails to described the coupling between each of the parameters and will subject to large and erratic errors if the Euler angle sequence reaches a singularity.

Euler parameters were computed directly from quaternion data using equations (7), (8) and (9). A total of 4 sets of of Euler parameters were computed, corresponding to the calibrated optical measurements of orientation, the Kalman-based algorithm estimated orientation and the proposed filter estimates orientation for both the MARG and IMU implementations. The errors of estimated Euler parameters, ϕ_ϵ, θ_ϵ and ψ_ϵ, were computed as the difference between the Euler parameters of the calibrated optical measurements and those of each of the corresponding estimated values.

5.1 Typical results

Figures 7, 8 and 9 show results typical of the 8 experiments for both the Kalman-based algorithm and the proposed filter MARG implementation. In each figure, the 3 traces of the upper plot represent the optically measured angle, the Kalman-based algorithm estimated angle, and the proposed filter estimated angle. The 2 traces of the lower plot represent the calculated error in each of the estimated angles.

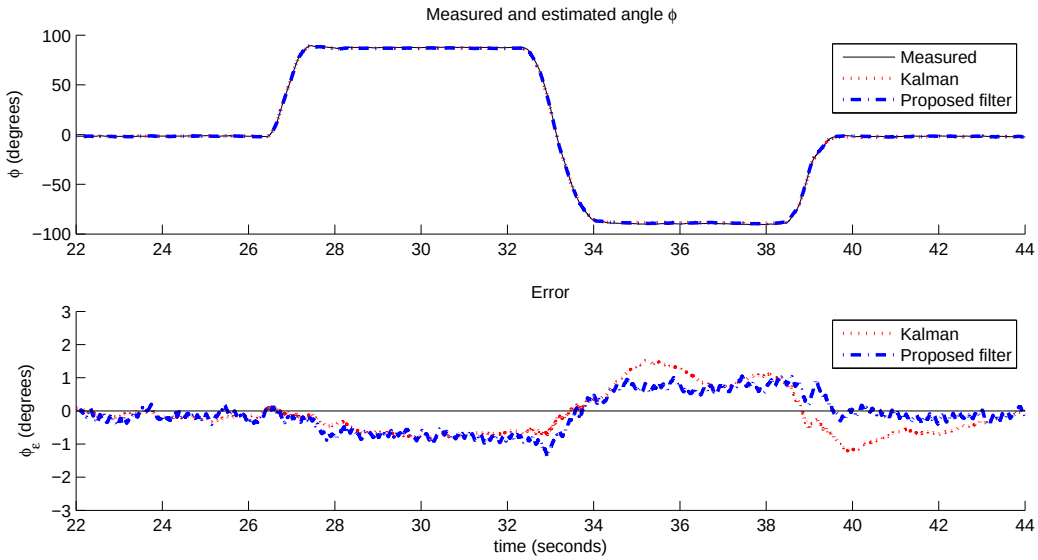

Figure 7: Typical results for measured and estimated angle ϕ (top) and error (bottom)

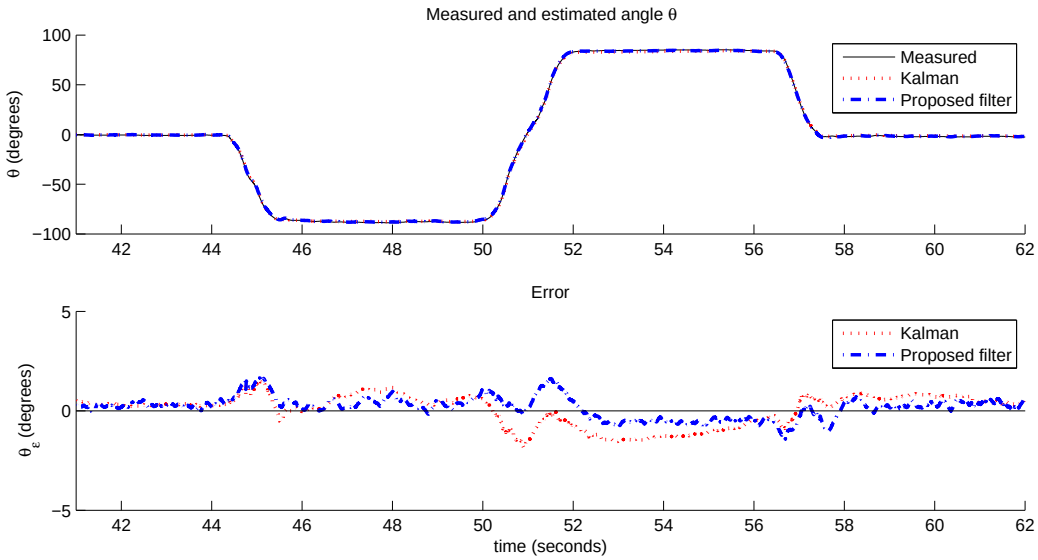

Figure 8: Typical results for measured and estimated angle θ (top) and error (bottom)

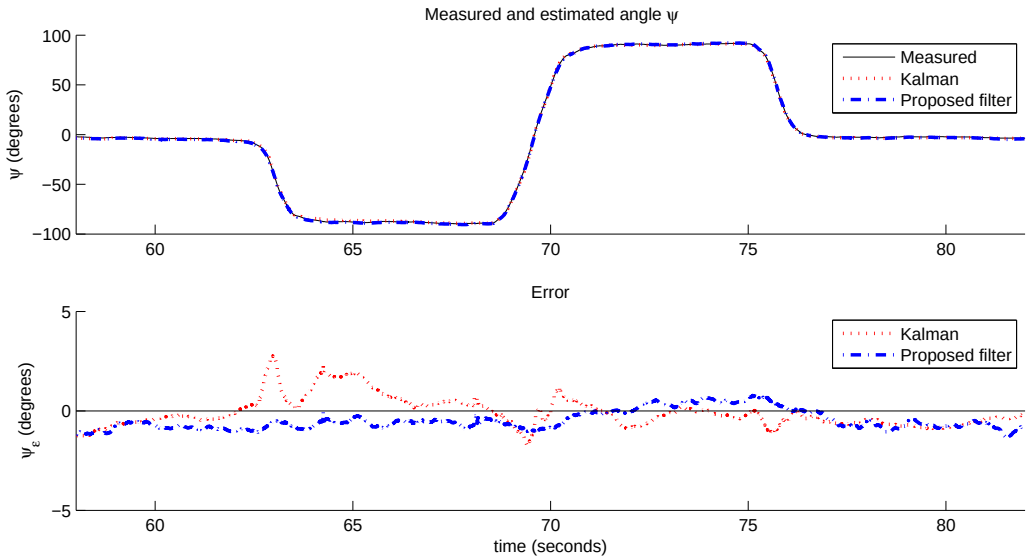

Figure 9: Typical results for measured and estimated angle ψ (top) and error (bottom)

5.2 Static and dynamic performance

The static and dynamic RMS values of ϕ_ϵ, θ_ϵ, and ψ_ϵ were calculated where a static state was assumed when the measured corresponding angular rate was $< 5°/\text{s}$, and a dynamic state when $\geq 5°/\text{s}$. This threshold was chosen to be suitably high enough above the noise floor of the data. Each RMS value was calculated for the period of time framing only the rotation sequence of the corresponding Euler parameter; as indicated in figures 7, 8 and 9. This was to prevent errors due to initial convergence or singularities in the Euler angle representation from corrupting results; that is, when $\theta = \pm 90$. The results are summarised in table 1. Each value, represents the mean of all 8 experiments. Values of RMS[ψ_ϵ] were not computed for the IMU implementation of the proposed filter as the filter cannot, nor is intended to, compensate for an accumulating error in this parameter.

Euler parameter	Kalman-based algorithm	Proposed filter (MARG)	Proposed filter (IMU)
RMS[ϕ_ϵ] static	0.789°	0.581°	0.594°
RMS[ϕ_ϵ] dynamic	0.769°	0.625°	0.623°
RMS[θ_ϵ] static	0.819°	0.502°	0.497°
RMS[θ_ϵ] dynamic	0.847°	0.668°	0.668°
RMS[ψ_ϵ] static	1.150°	1.073°	N/A
RMS[ψ_ϵ] dynamic	1.344°	1.110°	N/A

Table 1: Static and dynamic RMS error of Kalman-based algorithm and proposed filter IMU and MARG implementations

Results indicate that the proposed filter achieves higher levels of accuracy than the Kalman-based algorithm. The manufacturer of orientation sensor specify the typical performance of the Kalman-based algorithm as having a static RMS error of $< 0.5°$ in ϕ and θ, and $< 1°$ in ψ; and a dynamic RMS error of $< 2°$ in ϕ, θ, and ψ [18]. These values do not conform with those listed in table 1. Other studies [49] have shown that accuracy may far less than that quoted by the manufacture and that quoted levels of accuracy are only achieved once the devices is recalibrated. The lower levels of accuracy in the heading, ψ_ϵ, are to due characteristics of the sensor's measurements of the earth's magnetic field. The inclination of the earth's magnetic field during testing was between 65° and 70° to the horizontal [39]. As a consequence the component of the magnetic flux vector available as a reference of heading is relatively small. The larger component of the vector serves as an additional reference for pitch, ϕ, and roll, θ, alongside the reference measurement of gravity; hence errors in pitch, ϕ_ϵ, and roll, θ_ϵ, may be expected to be less than those in heading, ψ_ϵ. The magnetometer is specified [18] as having a bandwidth of 10 Hz which, relative to the 30 Hz and 40 Hz bandwidth of the accelerometer and gyroscope respectively, suggests an increased heading error, ψ_ϵ, in dynamic conditions.

5.3 Filter gain vs. performance

The results of an investigation into the effect of the filter adjustable parameter, β, on the proposed filter performance are summarised in Figure 10. The static and dynamic performance is quantified as the mean of the corresponding RMS values of ϕ_ϵ and θ_ϵ, for the IMU implementation and ϕ_ϵ, θ_ϵ and

ϕ_ϵ for the MARG implementation. The experimental data was processed though the separate proposed filter IMU and MARG implantations, using fixed values of β between 0 to 0.5. There is a clear optimal value of β for each filter implementation; high enough to minimises errors due to integral drift but sufficiently low enough that unnecessary noise is not introduced by large steps of gradient descent iterations.

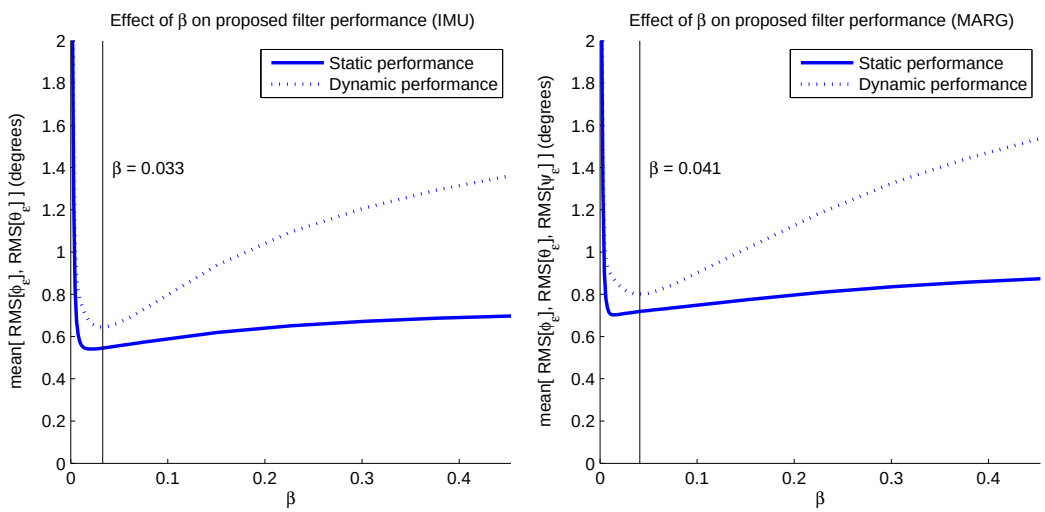

Figure 10: The effect of the adjustable parameter, β, on the performance of the proposed filter IMU (left) and MARG (right) implementations

5.4 Sampling rate vs. performance

The results of an investigation into the effect of sampling rate on filter performance is summarised in Figure11. The experimental data was processed though the separate proposed filter IMU and MARG implantations, using the previously defined, optimal values β where the experimental data was decimated to simulate sampling rates between 1Hz and 512 Hz. It can be seen from Figure11 that the propsoed filter acheives similar levels of performance at 50 Hz as at 512 Hz. Both filter implemntation are able to acheive a static error $< 2°$ and dynamic error $< 7°$ while sampling at 10 Hz. This level of accracuy may be sufficent be effective for applications such as human motion capture.

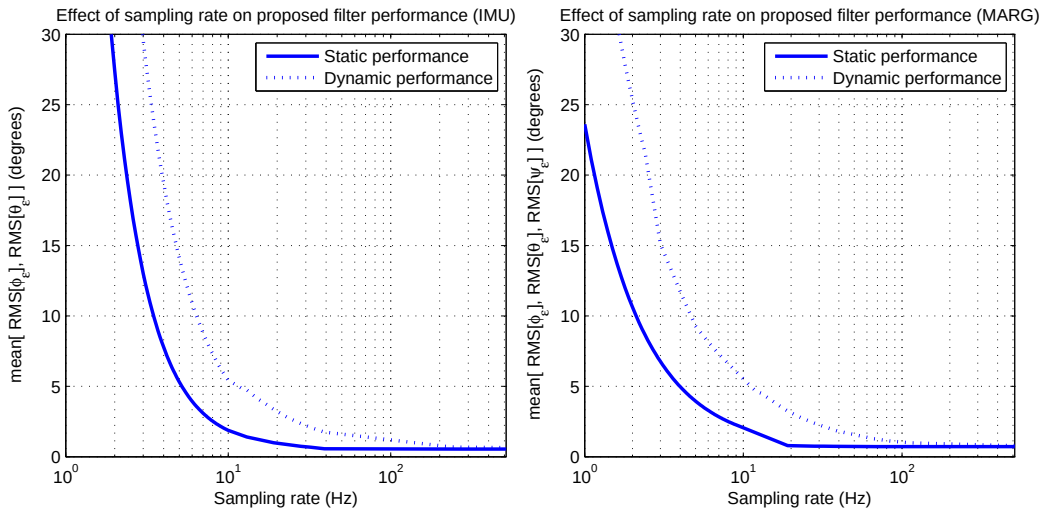

Figure 11: The effect of sampling rate on the performance of the proposed filter IMU (left) and MARG (right) implementations

5.5 Gyroscope bias drift

The calibrated gyrscope data used by the propsoed filter did not contain any bias errors. To investiagate the propsoed filter's ability to compensate for bias drift, errors were artifically introduced to the 8 experimental datasets. A constant drift of $0.2°/s/s$ was introduced to the gyroscope x axis measuremnts, ω_x and a constant bias error of $-0.2°/s$ as added to the gyroscope x axis measuremnts, ω_y. The filter gain ζ was set to 0 for the first 10 seconds of each experiment whilst the filter states converge from intial conditions. After this a value of 0.015 was used corrisponding a maximum convergene rate of the esitmated gyroscope bias of $1°/s/s$.

Figure 12 shows resutls typical of teh 8 experimetns, showing the gyroscope x and y axis bias estimated by the filter, plotted against the actual. From these results, the fitler can be seen to scuseffuly esitmate teh gyroscope biases witht eh rate limtied rate of convergese. The filter performance under these conditions was constent with that described in Table 1.

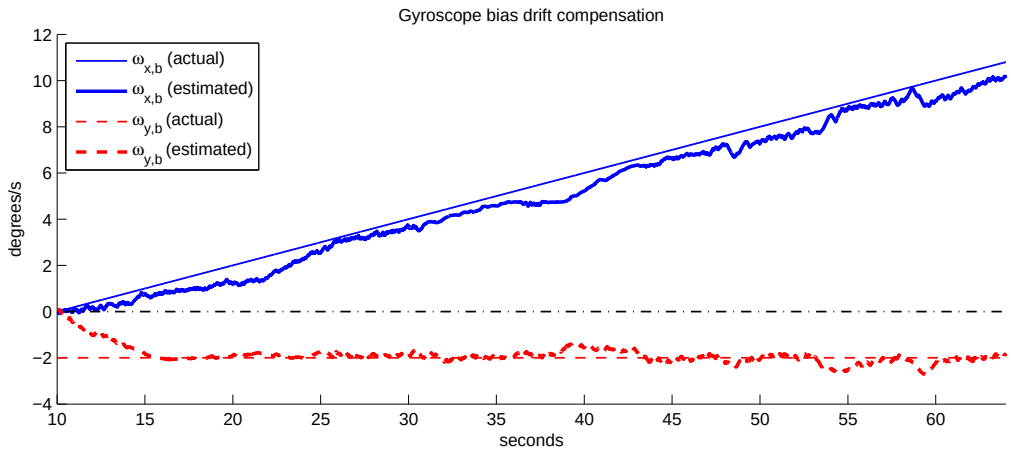

Figure 12: Filter tracking of gyroscope bias drift

6 Discussion

The derivation of the filter initially assumed that the accelerometer and magnetometer would only measure gravity and the earth's magnetic field. In practise, accelerations due to motion will result in an erroneous observed direction of gravity and so a potentially corrupt the estimated attitude and local magnetic distortions may still corrupt the estimated heading. In many applications it can be assumed that accelerations due to motion and local magnetic distortions are present for only short periods of time. Therefore the magnitude of the filter gain β may be chosen low enough that the divergence caused by the erroneous gravitational and magnetic field observations is reduced to an acceptable level over the period. The minimum acceptable value of β is limited by the gyroscope measurement error. In many applications it may be beneficial to use dynamic values of gains β and ζ. This will the influence accelerometers and magnetometers have on the estimated orientation to be reduced during potential problematic periods; for example, when large accelerations are detected. The use of large gains during the filter initialisation may also improve the filter's convergence from initial conditions. For example, it was found that a β and β gain of 10 enabled the filter states to converge within 5 seconds when initiated with a gyroscope bias error of 1000 deg/s in each axis.

The structure of the filter implantation for a MARG sensor array is sim-

ilar to that proposed by Bachman *et al* [34]. Both filters estimate the gyroscope measurement error as the gradient of a error surface created by the magnetometer and accelerometer measurements. Bachman's filter computes this using a Gauss-Newton approach which requires numerical differentiation and a matrix inversion. The filter proposed in this report uses an analytical derivation of the Jacobian and operates on a normalised gradient of the error surface. As a result, the filter proposed in this paper provides a substantial reduction in computational load and enables the derivation of an optimal filter gain based on system characteristics.

Appendix A and B describe the filter IMU and MARG implementations respectively, each implemented in C where the code has been optimised to reduce the number of arithmetic operations at the expense of data memory. The IMU implementation requires 108 arithmetic operations each filter update, and the MARG implementation requires 277 arithmetic operations per filter update; which includes the magnetic distortion and gyroscope drift compensation. These low computational requirements make the algorithm accessible for low power embedded systems systems enabling the use of low cost, low power hardware.

The experimental procedure used to evaluated the filter performances has a number of limitations; the filter performance was not evaluated for simultaneous rotations around more than one rotational axis and rotational velocities were limited to short periods and in magnitude. These limitations were necessary so that repeatable and quantifiable and practical.

7 Conclusions

In this work, we have introduced a novel orientation filter, applicable to both IMUs and MARG sensor arrays, that significantly ameliorates the computational load and parameter tuning burdens associated with conventional Kalman-based approaches. The filter is based on a Newton optimization using an analytic formulation of the gradient that is derived from a quaternion representation of motion. Novel aspects of the filter include:

- Analytic derivation of the Jacobian matrix, which eliminates significant computational load, allowing implementation at lower sampling frequencies and onto lower power, smaller platforms.
- The need to tune only one or two filter gains (β and ζ)), defined by the

gyroscope measurement error. Least squares curve fitting and complex tuning processes are therefore eliminated.

The filter derivation, magnetic distortion anf gyroscope bias drift compensation, and experimental testing have been detailed. Empirical testing and benchmarking has shown that the filter performs as well as a high quality commercial Kalman-based system, even with a full order of magnitude in reduction of sampling rate. The filter is both simple to implement and simple to tune. The implications of the low computational load and ability to operate at low sampling rates open a very wide range of new opportunities for the use of IMU and MARG sensor arrays in real-time applications. Applications where limited power or processing resources may be available are particularly well suited for the new filter. The filter also has great potential to alleviate computational load for applications that demand extremely high sampling rates.

References

[1] Mark Euston, Paul Coote, Robert Mahony, Jonghyuk Kim, and Tarek Hamel. A complementary filter for attitude estimation of a fixed-wing uav with a low-cost imu. In *6th International Conference on Field and Service Robotics*, July 2007.

[2] Sung Kyung Hong. Fuzzy logic based closed-loop strapdown attitude system for unmanned aerial vehicle (uav). *Sensors and Actuators A: Physical*, 107(2):109 – 118, 2003.

[3] A. Kallapur, I. Petersen, and S. Anavatti. A robust gyroless attitude estimation scheme for a small fixed-wing unmanned aerial vehicle. pages 666 –671, aug. 2009.

[4] B. Barshan and H. F. Durrant-Whyte. Inertial navigation systems for mobile robots. 11(3):328–342, June 1995.

[5] L. Ojeda and J. Borenstein. Flexnav: fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain. In *Proc. IEEE International Conference on Robotics and Automation ICRA '02*, volume 1, pages 317–322, May 11–15, 2002.

[6] David H. Titterton and John L. Weston. *Strapdown inertial navigation technology*. The Institution of Electrical Engineers, 2004.

[7] S. Beauregard. Omnidirectional pedestrian navigation for first responders. In *Proc. 4th Workshop on Positioning, Navigation and Communication WPNC '07*, pages 33–36, March 22–22, 2007.

[8] H. J. Luinge and P. H. Veltink. Inclination measurement of human movement using a 3-d accelerometer with autocalibration. 12(1):112–121, March 2004.

[9] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitation–a survey. volume 3, pages 1 – 18, 2008.

[10] E. A. Heinz, K. S. Kunze, M. Gruber, D. Bannach, and P. Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment. In *Proc. IEEE Symposium on Computational Intelligence and Games*, pages 98–102, May 22–24, 2006.

[11] J. E. Bortz. A new mathematical formulation for strapdown inertial navigation. (1):61–66, January 1971.

[12] M B Ignagni. Optimal strapdown attitude integration algorithms. In *Guidance, Control, and Dynamics*, volume 13, pages 363–369, 1990.

[13] N. Yazdi, F. Ayazi, and K. Najafi. Micromachined inertial sensors. 86(8):1640–1659, August 1998.

[14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[15] E. Foxlin. Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter. In *Proc. Virtual Reality Annual International Symposium the IEEE 1996*, pages 185–194,267, March 30–April 3, 1996.

[16] H. J. Luinge, P. H. Veltink, and C. T. M. Baten. Estimation of orientation with gyroscopes and accelerometers. In *Proc. First Joint [Engineering in Medicine and Biology 21st Annual Conf. and the 1999 Annual Fall Meeting of the Biomedical Engineering Soc.] BMES/EMBS Conference*, volume 2, page 844, October 13–16, 1999.

[17] J. L. Marins, Xiaoping Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda. An extended kalman filter for quaternion-based orientation estimation using marg sensors. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 2003–2011, October 29–November 3, 2001.

[18] Xsens Technologies B.V. *MTi and MTx User Manual and Technical Documentation*. Pantheon 6a, 7521 PR Enschede, The Netherlands, May 2009.

[19] MicroStrain Inc. *3DM-GX3 -25 Miniature Attutude Heading Reference Sensor*. 459 Hurricane Lane, Suite 102, Williston, VT 05495 USA, 1.04 edition, 2009.

[20] VectorNav Technologies, LLC. *VN -100 User Manual*. College Station, TX 77840 USA, preliminary edition, 2009.

[21] InterSense, Inc. *InertiaCube2+ Manual*. 36 Crosby Drive, Suite 150, Bedford, MA 01730, USA, 1.0 edition, 2008.

[22] PNI sensor corporation. *Spacepoint Fusion*. 133 Aviation Blvd, Suite 101, Santa Rosa, CA 95403-1084 USA.

[23] Crossbow Technology, Inc. *AHRS400 Series Users Manual*. 4145 N. First Street, San Jose, CA 95134, rev. c edition, February 2007.

[24] A. M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. 53(7):1346–1356, July 2006.

[25] H. J. Luinge and P. H. Veltink. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. *Medical and Biological Engineering and Computing*, 43(2):273–282, April 2006.

[26] David Jurman, Marko Jankovec, Roman Kamnik, and Marko Topic. Calibration and data fusion solution for the miniature attitude and heading reference system. *Sensors and Actuators A: Physical*, 138(2):411–420, August 2007.

[27] Markus Haid and Jan Breitenbach. Low cost inertial orientation tracking with kalman filter. *Applied Mathematics and Computation*, 153(4):567–575, June 2004.

[28] D. Roetenberg, H. J. Luinge, C. T. M. Baten, and P. H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. 13(3):395–405, September 2005.

[29] John L. Crassidis and Landis F. Markley. Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542, 2003.

[30] D. Gebre-Egziabher, R.C. Hayward, and J.D. Powell. Design of multi-sensor attitude determination systems. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(2):627 – 649, april 2004.

[31] N.H.Q. Phuong, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro. A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass. *Journal of Universal Computer Science*, 15(4):859–876, 2009.

[32] Daniel Choukroun. *Novel methods for attitude determination using vector observations*. PhD thesis, Israel Institute of Technology, May 2003.

[33] R. A. Hyde, L. P. Ketteringham, S. A. Neild, and R. J. S. Jones. Estimation of upper-limb orientation based on accelerometer and gyroscope measurements. 55(2):746–754, February 2008.

[34] E. R. Bachmann, I. Duman, U. Y. Usta, R. B. McGhee, X. P. Yun, and M. J. Zyda. Orientation tracking for humans and robots using inertial sensors. In *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA '99*, pages 187–194, November 8–9, 1999.

[35] R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *Automatic Control, IEEE Transactions on*, 53(5):1203 –1218, june 2008.

[36] J. B. Kuipers. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, 1999.

[37] John J. Craig. *Introduction to Robotics Mechanics and Control*. Pearson Education International, 2005.

[38] David R. Pratt Robert B. McGhee Joseph M. Cooke, Michael J. Zyda. Npsnet: Flight simulation dynamic modelling using quaternions. *Presence*, 1:404–420, 1994.

[39] John Arthur Jacobs. *The earth's core*, volume 37 of International geophysics series. Academic Press, 2 edition, 1987.

[40] E. R. Bachmann, Xiaoping Yun, and C. W. Peterson. An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors. In *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, volume 2, pages 1115–1122, April 2004.

[41] C.T.M. Baten F.C.T. van der Helm W.H.K. de Vries, H.E.J. Veeger. Magnetic distortion in motion labs, implications for validating inertial magnetic sensors. *Gait & Posture*, 29(4):535–541, 2009.

[42] Speake & Co Limited. Autocalibration algorithms for FGM type sensors. Application note.

[43] Michael J. Caruso. *Applications of Magnetoresistive Sensors in Navigation Systems*. Honeywell Inc., Solid State Electronics Center, Honeywell Inc. 12001 State Highway 55, Plymouth, MN 55441.

[44] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira. A geometric approach to strapdown magnetometer calibration in sensor frame. In *Navigation, Guidance and Control of Underwater Vehicles*, volume 2, 2008.

[45] Demoz Gebre-Egziabher, Gabriel H. Elkaim, J. David Powell, and Bradford W. Parkinson. Calibration of strapdown magnetometers in magnetic field domain. *Journal of Aerospace Engineering*, 19(2):87–102, 2006.

[46] Vicon Motion Systems Limited. *Vicon MX Hardware*. 5419 McConnell Avenue, Los Angeles, CA 90066, USA, 1.6 edition, 2004.

[47] Vicon Motion Systems Limited. *Vicon Nexus Product Guide - Foundation Notes*. 5419 McConnell Avenue, Los Angeles, CA 90066, USA, 1.2 edition, November 2007.

[48] Itzhack Y Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. *AIAA Journal of Guidance, Control and Dynamics*, 23(6):10851087, Nov.Dec 2000. (Engineering Note).

[49] M. A. Brodie, A. Walmsley, and W. Page. The static accuracy and calibration of inertial measurement units for 3D orientation. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(6):641–648, December 2008.

A IMU filter implementation optimised in C

The following source code is an implementation of the orientation filter for an IMU, in C. The code has been optimised minimise the required number of arithmetic operations at the expense of data memory. Each filter update requires 109 scalar arithmetic operations (18 additions, 20 subtracts, 57 multiplications, 11 divisions and 3 square roots). The implementation requires 40 bytes of data memory for global variables and 100 bytes of data memory for local variables during each filter update function call.

```
// Math library required for 'sqrt'
#include <math.h>

// System constants
#define deltat 0.001f                                          // sampling period in seconds (shown as 1 ms)
#define gyroMeasError 3.14159265358979f * (5.0f / 180.0f)      // gyroscope measurement error in rad/s (shown as 5 deg/s)
#define beta sqrt(3.0f / 4.0f) * gyroMeasError                 // compute beta

// Global system variables
float a_x, a_y, a_z;                                           // accelerometer measurements
float w_x, w_y, w_z;                                           // gyroscope measurements in rad/s
float SEq_1 = 1.0f, SEq_2 = 0.0f, SEq_3 = 0.0f, SEq_4 = 0.0f; // estimated orientation quaternion elements with initial conditions

void filterUpdate(float w_x, float w_y, float w_z, float a_x, float a_y, float a_z)
{
    // Local system variables
    float norm;                                                          // vector norm
    float SEqDot_omega_1, SEqDot_omega_2, SEqDot_omega_3, SEqDot_omega_4; // quaternion derrivative from gyroscopes elements
    float f_1, f_2, f_3;                                                 // objective function elements
    float J_11or24, J_12or23, J_13or22, J_14or21, J_32, J_33;            // objective function Jacobian elements
    float SEqHatDot_1, SEqHatDot_2, SEqHatDot_3, SEqHatDot_4;            // estimated direction of the gyroscope error

    // Axulirary variables to avoid reapeated calcualtions
    float halfSEq_1 = 0.5f * SEq_1;
    float halfSEq_2 = 0.5f * SEq_2;
    float halfSEq_3 = 0.5f * SEq_3;
    float halfSEq_4 = 0.5f * SEq_4;
    float twoSEq_1 = 2.0f * SEq_1;
    float twoSEq_2 = 2.0f * SEq_2;
    float twoSEq_3 = 2.0f * SEq_3;

    // Normalise the accelerometer measurement
    norm = sqrt(a_x * a_x + a_y * a_y + a_z * a_z);
    a_x /= norm;
    a_y /= norm;
    a_z /= norm;

    // Compute the objective function and Jacobian
```

```
    f_1 = twoSEq_2 * SEq_4 - twoSEq_1 * SEq_3 - a_x;
    f_2 = twoSEq_1 * SEq_2 + twoSEq_3 * SEq_4 - a_y;
    f_3 = 1.0f - twoSEq_2 * SEq_2 - twoSEq_3 * SEq_3 - a_z;
    J_11or24 = twoSEq_3;                                                    // J_11 negated in matrix multiplication
    J_12or23 = 2.0f * SEq_4;
    J_13or22 = twoSEq_1;                                                    // J_12 negated in matrix multiplication
    J_14or21 = twoSEq_2;
    J_32 = 2.0f * J_14or21;                                                 // negated in matrix multiplication
    J_33 = 2.0f * J_11or24;                                                 // negated in matrix multiplication

    // Compute the gradient (matrix multiplication)
    SEqHatDot_1 = J_14or21 * f_2 - J_11or24 * f_1;
    SEqHatDot_2 = J_12or23 * f_1 + J_13or22 * f_2 - J_32 * f_3;
    SEqHatDot_3 = J_12or23 * f_2 - J_33 * f_3 - J_13or22 * f_1;
    SEqHatDot_4 = J_14or21 * f_1 + J_11or24 * f_2;

    // Normalise the gradient
    norm = sqrt(SEqHatDot_1 * SEqHatDot_1 + SEqHatDot_2 * SEqHatDot_2 + SEqHatDot_3 * SEqHatDot_3 + SEqHatDot_4 * SEqHatDot_4);
    SEqHatDot_1 /= norm;
    SEqHatDot_2 /= norm;
    SEqHatDot_3 /= norm;
    SEqHatDot_4 /= norm;

    // Compute the quaternion derrivative measured by gyroscopes
    SEqDot_omega_1 = -halfSEq_2 * w_x - halfSEq_3 * w_y - halfSEq_4 * w_z;
    SEqDot_omega_2 = halfSEq_1 * w_x + halfSEq_3 * w_z - halfSEq_4 * w_y;
    SEqDot_omega_3 = halfSEq_1 * w_y - halfSEq_2 * w_z + halfSEq_4 * w_x;
    SEqDot_omega_4 = halfSEq_1 * w_z + halfSEq_2 * w_y - halfSEq_3 * w_x;

    // Compute then integrate the estimated quaternion derrivative
    SEq_1 += (SEqDot_omega_1 - (beta * SEqHatDot_1)) * deltat;
    SEq_2 += (SEqDot_omega_2 - (beta * SEqHatDot_2)) * deltat;
    SEq_3 += (SEqDot_omega_3 - (beta * SEqHatDot_3)) * deltat;
    SEq_4 += (SEqDot_omega_4 - (beta * SEqHatDot_4)) * deltat;

    // Normalise quaternion
    norm = sqrt(SEq_1 * SEq_1 + SEq_2 * SEq_2 + SEq_3 * SEq_3 + SEq_4 * SEq_4);
    SEq_1 /= norm;
    SEq_2 /= norm;
    SEq_3 /= norm;
    SEq_4 /= norm;
}
```

B MARG filter implementation optimised in C

The following source code is an implementation of the orientation filter for a MARG sensor array including magnetic distortion and gyroscope drift compensation, in C. The code has been optimised minimise the required number of arithmetic operations at the expense of data memory. Each filter update requires 277 scalar arithmetic operations (51 additions, 57 subtracts, 155 multiplications, 14 divisions and 5 square roots). The implementation requires 72 bytes of data memory for global variables and 260 bytes of data memory for local variables during each filter update function call.

```
// Math library required for 'sqrt'
#include <math.h>

// System constants
#define deltat 0.001f                                   // sampling period in seconds (shown as 1 ms)
```

```
#define gyroMeasError 3.14159265358979 * (5.0f / 180.0f)  // gyroscope measurement error in rad/s (shown as 5 deg/s)
#define gyroMeasDrift 3.14159265358979 * (0.2f / 180.0f)  // gyroscope measurement error in rad/s/s (shown as 0.2f deg/s/s)
#define beta sqrt(3.0f / 4.0f) * gyroMeasError            // compute beta
#define zeta sqrt(3.0f / 4.0f) * gyroMeasDrift            // compute zeta

// Global system variables
float a_x, a_y, a_z;                                   // accelerometer measurements
float w_x, w_y, w_z;                                   // gyroscope measurements in rad/s
float m_x, m_y, m_z;                                   // magnetometer measurements
float SEq_1 = 1, SEq_2 = 0, SEq_3 = 0, SEq_4 = 0;      // estimated orientation quaternion elements with initial conditions
float b_x = 1, b_z = 0;                                // reference direction of flux in earth frame
float w_bx = 0, w_by = 0, w_bz = 0;                    // estimate gyroscope biases error

// Function to compute one filter iteration
void filterUpdate(float w_x, float w_y, float w_z, float a_x, float a_y, float a_z, float m_x, float m_y, float m_z)
{
    // local system variables
    float norm;                                                          // vector norm
    float SEqDot_omega_1, SEqDot_omega_2, SEqDot_omega_3, SEqDot_omega_4;   // quaternion rate from gyroscopes elements
    float f_1, f_2, f_3, f_4, f_5, f_6;                                  // objective function elements
    float J_11or24, J_12or23, J_13or22, J_14or21, J_32, J_33,            // objective function Jacobian elements
    J_41, J_42, J_43, J_44, J_51, J_52, J_53, J_54, J_61, J_62, J_63, J_64; //
    float SEqHatDot_1, SEqHatDot_2, SEqHatDot_3, SEqHatDot_4;            // estimated direction of the gyroscope error
    float w_err_x, w_err_y, w_err_z;                                     // estimated direction of the gyroscope error (angular)
    float h_x, h_y, h_z;                                                 // computed flux in the earth frame

    // axulirary variables to avoid reapeated calcualtions
    float halfSEq_1 = 0.5f * SEq_1;
    float halfSEq_2 = 0.5f * SEq_2;
    float halfSEq_3 = 0.5f * SEq_3;
    float halfSEq_4 = 0.5f * SEq_4;
    float twoSEq_1 = 2.0f * SEq_1;
    float twoSEq_2 = 2.0f * SEq_2;
    float twoSEq_3 = 2.0f * SEq_3;
    float twoSEq_4 = 2.0f * SEq_4;
    float twob_x = 2.0f * b_x;
    float twob_z = 2.0f * b_z;
    float twob_xSEq_1 = 2.0f * b_x * SEq_1;
    float twob_xSEq_2 = 2.0f * b_x * SEq_2;
    float twob_xSEq_3 = 2.0f * b_x * SEq_3;
    float twob_xSEq_4 = 2.0f * b_x * SEq_4;
    float twob_zSEq_1 = 2.0f * b_z * SEq_1;
    float twob_zSEq_2 = 2.0f * b_z * SEq_2;
    float twob_zSEq_3 = 2.0f * b_z * SEq_3;
    float twob_zSEq_4 = 2.0f * b_z * SEq_4;
    float SEq_1SEq_2;
    float SEq_1SEq_3 = SEq_1 * SEq_3;
    float SEq_1SEq_4;
    float SEq_2SEq_3;
    float SEq_2SEq_4 = SEq_2 * SEq_4;
    float SEq_3SEq_4;
    float twom_x = 2.0f * m_x;
    float twom_y = 2.0f * m_y;
    float twom_z = 2.0f * m_z;

    // normalise the accelerometer measurement
    norm = sqrt(a_x * a_x + a_y * a_y + a_z * a_z);
    a_x /= norm;
    a_y /= norm;
    a_z /= norm;

    // normalise the magnetometer measurement
    norm = sqrt(m_x * m_x + m_y * m_y + m_z * m_z);
    m_x /= norm;
    m_y /= norm;
    m_z /= norm;

    // compute the objective function and Jacobian
    f_1 = twoSEq_2 * SEq_4 - twoSEq_1 * SEq_3 - a_x;
    f_2 = twoSEq_1 * SEq_2 + twoSEq_3 * SEq_4 - a_y;
    f_3 = 1.0f - twoSEq_2 * SEq_2 - twoSEq_3 * SEq_3 - a_z;
    f_4 = twob_x * (0.5f - SEq_3 * SEq_3 - SEq_4 * SEq_4) + twob_z * (SEq_2SEq_4 - SEq_1SEq_3) - m_x;
    f_5 = twob_x * (SEq_2 * SEq_3 - SEq_1 * SEq_4) + twob_z * (SEq_1 * SEq_2 + SEq_3 * SEq_4) - m_y;
    f_6 = twob_x * (SEq_1SEq_3 + SEq_2SEq_4) + twob_z * (0.5f - SEq_2 * SEq_2 - SEq_3 * SEq_3) - m_z;
    J_11or24 = twoSEq_3;                                                 // J_11 negated in matrix multiplication
    J_12or23 = 2.0f * SEq_4;
    J_13or22 = twoSEq_1;                                                 // J_12 negated in matrix multiplication
```

```
    J_14or21 = twoSEq_2;
    J_32 = 2.0f * J_14or21;                                         // negated in matrix multiplication
    J_33 = 2.0f * J_11or24;                                         // negated in matrix multiplication
    J_41 = twob_zSEq_3;                                             // negated in matrix multiplication
    J_42 = twob_zSEq_4;
    J_43 = 2.0f * twob_xSEq_3 + twob_zSEq_1;                        // negated in matrix multiplication
    J_44 = 2.0f * twob_xSEq_4 - twob_zSEq_2;                        // negated in matrix multiplication
    J_51 = twob_xSEq_4 - twob_zSEq_2;                               // negated in matrix multiplication
    J_52 = twob_xSEq_3 + twob_zSEq_1;
    J_53 = twob_xSEq_2 + twob_zSEq_4;
    J_54 = twob_xSEq_1 - twob_zSEq_3;                               // negated in matrix multiplication
    J_61 = twob_xSEq_3;
    J_62 = twob_xSEq_4 - 2.0f * twob_zSEq_2;
    J_63 = twob_xSEq_1 - 2.0f * twob_zSEq_3;
    J_64 = twob_xSEq_2;

    // compute the gradient (matrix multiplication)
    SEqHatDot_1 = J_14or21 * f_2 - J_11or24 * f_1 - J_41 * f_4 - J_51 * f_5 + J_61 * f_6;
    SEqHatDot_2 = J_12or23 * f_1 + J_13or22 * f_2 - J_32 * f_3 + J_42 * f_4 + J_52 * f_5 + J_62 * f_6;
    SEqHatDot_3 = J_12or23 * f_2 - J_33 * f_3 - J_13or22 * f_1 - J_43 * f_4 + J_53 * f_5 + J_63 * f_6;
    SEqHatDot_4 = J_14or21 * f_1 + J_11or24 * f_2 - J_44 * f_4 - J_54 * f_5 + J_64 * f_6;

    // normalise the gradient to estimate direction of the gyroscope error
    norm = sqrt(SEqHatDot_1 * SEqHatDot_1 + SEqHatDot_2 * SEqHatDot_2 + SEqHatDot_3 * SEqHatDot_3 + SEqHatDot_4 * SEqHatDot_4);
    SEqHatDot_1 = SEqHatDot_1 / norm;
    SEqHatDot_2 = SEqHatDot_2 / norm;
    SEqHatDot_3 = SEqHatDot_3 / norm;
    SEqHatDot_4 = SEqHatDot_4 / norm;

    // compute angular estimated direction of the gyroscope error
    w_err_x = twoSEq_1 * SEqHatDot_2 - twoSEq_2 * SEqHatDot_1 - twoSEq_3 * SEqHatDot_4 + twoSEq_4 * SEqHatDot_3;
    w_err_y = twoSEq_1 * SEqHatDot_3 + twoSEq_2 * SEqHatDot_4 - twoSEq_3 * SEqHatDot_1 - twoSEq_4 * SEqHatDot_2;
    w_err_z = twoSEq_1 * SEqHatDot_4 - twoSEq_2 * SEqHatDot_3 + twoSEq_3 * SEqHatDot_2 - twoSEq_4 * SEqHatDot_1;

    // compute and remove the gyroscope baises
    w_bx += w_err_x * deltat * zeta;
    w_by += w_err_y * deltat * zeta;
    w_bz += w_err_z * deltat * zeta;
    w_x -= w_bx;
    w_y -= w_by;
    w_z -= w_bz;

    // compute the quaternion rate measured by gyroscopes
    SEqDot_omega_1 = -halfSEq_2 * w_x - halfSEq_3 * w_y - halfSEq_4 * w_z;
    SEqDot_omega_2 = halfSEq_1 * w_x + halfSEq_3 * w_z - halfSEq_4 * w_y;
    SEqDot_omega_3 = halfSEq_1 * w_y - halfSEq_2 * w_z + halfSEq_4 * w_x;
    SEqDot_omega_4 = halfSEq_1 * w_z + halfSEq_2 * w_y - halfSEq_3 * w_x;

    // compute then integrate the estimated quaternion rate
    SEq_1 += (SEqDot_omega_1 - (beta * SEqHatDot_1)) * deltat;
    SEq_2 += (SEqDot_omega_2 - (beta * SEqHatDot_2)) * deltat;
    SEq_3 += (SEqDot_omega_3 - (beta * SEqHatDot_3)) * deltat;
    SEq_4 += (SEqDot_omega_4 - (beta * SEqHatDot_4)) * deltat;

    // normalise quaternion
    norm = sqrt(SEq_1 * SEq_1 + SEq_2 * SEq_2 + SEq_3 * SEq_3 + SEq_4 * SEq_4);
    SEq_1 /= norm;
    SEq_2 /= norm;
    SEq_3 /= norm;
    SEq_4 /= norm;

    // compute flux in the earth frame
    SEq_1SEq_2 = SEq_1 * SEq_2;                                     // recompute axulirary variables
    SEq_1SEq_3 = SEq_1 * SEq_3;
    SEq_1SEq_4 = SEq_1 * SEq_4;
    SEq_3SEq_4 = SEq_3 * SEq_4;
    SEq_2SEq_3 = SEq_2 * SEq_3;
    SEq_2SEq_4 = SEq_2 * SEq_4;
    h_x = twom_x * (0.5f - SEq_3 * SEq_3 - SEq_4 * SEq_4) + twom_y * (SEq_2SEq_3 - SEq_1SEq_4) + twom_z * (SEq_2SEq_4 + SEq_1SEq_3);
    h_y = twom_x * (SEq_2SEq_3 + SEq_1SEq_4) + twom_y * (0.5f - SEq_2 * SEq_2 - SEq_4 * SEq_4) + twom_z * (SEq_3SEq_4 - SEq_1SEq_2);
    h_z = twom_x * (SEq_2SEq_4 - SEq_1SEq_3) + twom_y * (SEq_3SEq_4 + SEq_1SEq_2) + twom_z * (0.5f - SEq_2 * SEq_2 - SEq_3 * SEq_3);

    // normalise the flux vector to have only components in the x and z
    b_x = sqrt((h_x * h_x) + (h_y * h_y));
    b_z = h_z;
}
```