2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2017)
November 16 - 18, 2017, Daegu, Korea

Improved Formulation of the IMU and MARG Orientation Gradient Descent Algorithm for Motion Tracking in Human-Machine Interfaces

Marcel Admiraal[1], Samuel Wilson[1], Ravi Vaidyanathan[1]

Abstract— **Wearable motion tracking systems are becoming increasingly popular in human-machine interfaces. For inertial measurement, it is vital to efficiently fuse inertial, gyroscopic, and magnetometer data for spatial orientation. We introduce a new algorithm for this fusion based on using gradient descent to correct for the integral error in calculating the orientation quaternion of a rotating body. The algorithm is an improved formulation of the well-known estimation of orientation using a gradient descent algorithm. The new formulation ensures that the gradient descent algorithm uses the steepest descent, resulting in a five order of magnitude increase in the precision of the calculated orientation quaternion. We have also converted the algorithm to use fixed point integers instead of floating point numbers to more than double the speed of the calculations on the types of processors used with Inertial Measurement Units (IMUs) and Magnetic, Angular Rate and Gravity sensors (MARGs). This enables the corrections to not only be faster than the original formulations, but also remain valid for a larger range of inputs. The improved efficiency and accuracy show significant potential for increasing the scope of inertial measurement in applications where low power or greater precision is necessary such as very small wearable or implantable systems.**

I. INTRODUCTION

There is an increasing interest within the machine learning and data fusion communities in creating smoother and more natural human-machine interfaces. A critical component of this vision is the accurate measurement of human movement data for subsequent prediction of user intent for machine interaction [1][2]. Examples of such interfaces include gesture based control of computers or robots [3], movement assessment for physical rehabilitation [4], interfacing with assistive technology [5], gain analysis [6] and sports training.

While all these applications rely on the capacity to track movement, robust measurement systems that can work outside of controlled environments are not widely available today. Controlled settings such as laboratory or pre-prepared environments can provide highly accurate means of motion capture, however such measurements limit the types of activities, as well as the collection time of data, and may not identify problems that occur when moving naturally outside these controlled environments. Therefore, an unobtrusive system capable of logging data from human motion for extended periods of time has broad potential use.

*This research was funded by the Dyson Foundation, the US Office of Naval Research Global (ONR-G) grant N62909-14-1-N221 and the UK-India Educational Research Initiative (UK-IERI) under award IND/CONT/E/14-15/366.

[1]Mechanical Engineering Imperial College, London London, United Kingdom s.wilson14@imperial.ac.uk

Motion analysis in a laboratory setting is often achieved using optical based methods, which rely on cameras and reflective markers on the moving object. Optical systems have high accuracy, but are costly and often non-portable, and do not function well in crowded environments where vision may be occluded. In recent years, motion analysis has also been accomplished with smaller, embedded systems such as Inertial Measurement Units (IMU), which allow portability at the cost of accuracy [7].

There is a very wide body of literature focusing on tracking of motion through inertial measurement. Most inertial based motion tracking consists of individual units, usually relying a Magnetic, Angular Rate and Gravity (MARG) sensor which fuses a triple-axis gyroscope, a triple-axis accelerometer, and a triple-axis magnetometer to estimate rigid body orientation.

Historically, the Kalman filter has been the most common means of fusing MARG data. More recently, gradient descent methods (such as 'Madgwick's algorithm'[8]) have seen very widespread use as a means to alleviate the computational load associated with conventional Kalman-based approaches. The Madgwick et al equations use the measured acceleration and magnetic field to correct for gyroscopic drift. This self contained method for deriving global orientation is extremely useful for monitoring body kinematics, both for activity classification [9][10] and biomedical monitoring [11][12][13], as well as mobile orientation/haptic control [14][15], which is useful in Virtual/Augmented reality applications [16][17], and for navigation [18][19] and robotic control [20][21].

Despite its incalculable impact on inertial measurement systems, the gradient descent filter as implemented today does not use the steepest descent direction in its current form. The result of this is that the algorithm must go through an increased number of iterations to achieve the minimum result, causing an increased error when the device is moving.

In this work, we introduce a reformulated algorithm which uses the steepest descent direction when using gradient descent to solve the simultaneous non-linear equations for updating the rotation quaternions. We present a full derivation of an improved gradient descent algorithm for fusion of MARG data and demonstrate the improved efficiency when compared to existing approaches. Section II presents the mathematical foundations of quaternions mathematics, necessary for the formulations derived in the rest of the paper. Section III then describes our algorithm formulations, while Section IV presents results and discussion of the details of algorithm implementation in comparison to the version used today. Section V then draws conclusions from the work.

978-1-5090-6064-1/17/$31.00 ©2017 IEEE

II. Principles of quaternions

Euler's rotation theorem states that any rotation or sequence of rotations is equivalent to a single rotation about a single axis, and quaternions can be used to represent orientations and rotations of objects in three dimensions. Unlike Euler angles, which depend on the order the angles are provided, quaternions provide a consistent representation and avoid the problem of gimbal lock. Quaternions are also more compact than rotation matrices, and have been shown to be more numerically stable [22].

Quaternions were first described by William Hamilton in 1843 [23]; however their use in computer graphics only dates back to 1985 [24]. Since 1999 they have also been used in embedded IMUs and MARG sensors [25].

Quaternions extend the complex number system and are represented in the form $\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, and $\mathbf{ij} = \mathbf{k}, \mathbf{ji} = -\mathbf{k}, \mathbf{ik} = -\mathbf{j}, \mathbf{ki} = \mathbf{j}, \mathbf{jk} = \mathbf{i}, \mathbf{kj} = -\mathbf{i}$. The product of two quaternions is non-commutative and is defined as:

$$\begin{aligned}\mathbf{q}_1 \cdot \mathbf{q}_2 = {} & w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2 \\ & + (w_1x_2 + x_1w_2 + y_1z_2 - z_1y_2)\mathbf{i} \\ & + (w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2)\mathbf{j} \\ & + (w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2)\mathbf{k}\end{aligned} \tag{1}$$

The conjugate of a quaternion is analogous to the conjugate of a complex number and is defined as:

$$\mathbf{q}^* = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k} \tag{2}$$

The norm of a quaternion is defined as the square root of the product of a quaternion and its conjugate:

$$\|\mathbf{q}\| = \sqrt{\mathbf{q}\mathbf{q}^*} \tag{3}$$

which allows the reciprocal of a quaternion to be defined:

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \tag{4}$$

allowing quaternions to be divided.

A rotation through angle θ around an axis defined by a unit vector $\mathbf{u} = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$ can be represented by a quaternion using an extension of Euler's formula [26]

$$\mathbf{q} = e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = cos\frac{\theta}{2} + (u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})sin\frac{\theta}{2} \tag{5}$$

therefore quaternion $1 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$ represents no rotation.

A three dimensional object can be rotated by converting the points that define it into quaternions with a real coordinate equal to zero $(v_1, \ldots, v_n)$, and using the rotation quaternion ($\mathbf{q}$) to derive their new position. Hence:

$$\mathbf{v} = 0 + u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k} \tag{6}$$

and evaluating the conjugation of $\mathbf{v}$ by $\mathbf{q}$ [27]

$$\mathbf{v}' = \mathbf{q} \cdot \mathbf{v} \cdot \mathbf{q}^{-1} \tag{7}$$

Both $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation[1]. The reverse rotation is achieved by rotating by the reciprocal: $\mathbf{q}^{-1}$. Since a rotation in the local reference frame is the reverse of a rotation in the global reference frame, the transformation from the local reference frame to the global reference frame is achieved by using the reciprocal of the quaternion.

Multiple rotations can be combined by first multiplying the quaternions together and then evaluating the conjugation. If each subsequent rotation is made in the local reference frame, the quaternion multiplications are made in the forward order:

$${}^{\mathbf{L}}\mathbf{q} = \mathbf{q}_1 \cdot \mathbf{q}_2 \tag{8}$$

If each subsequent rotation is made in the global reference frame, the corresponding quaternions need to be multiplied in reverse order:

$${}^{\mathbf{G}}\mathbf{q} = \mathbf{q}_2 \cdot \mathbf{q}_1 \tag{9}$$

Similarly, to describe the rotation of a rigid body rotating at constant angular velocity as a function of time depends on whether the angular velocity is measured in the local reference frame:

$$\mathbf{q}(t) = \mathbf{q}_0 \cdot ({}^{\mathbf{L}}\mathbf{q}_\omega)^t \tag{10}$$[2]

or the global reference frame:

$$\mathbf{q}(t) = ({}^{\mathbf{G}}\mathbf{q}_\omega)^t \cdot \mathbf{q}_0 \tag{11}$$

where $\mathbf{q}_0$ is the rotation of the body at time $t = 0$ and $\mathbf{q}_\omega$ describes the rotation that a body with angular velocity $\boldsymbol{\omega} = \theta\mathbf{u}$ undergoes in one time step. The derivatives of the functions of the rotation of a body with respect to time are:

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2}\mathbf{q}(t){}^{\mathbf{L}}\boldsymbol{\omega} \tag{12}$$[3]

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2}{}^{\mathbf{G}}\boldsymbol{\omega}\mathbf{q}(t) \tag{13}$$[4]

[1] $s\mathbf{q}$ for any s represents the same rotation:
$s\mathbf{q} \cdot \mathbf{v} \cdot (s\mathbf{q})^{-1} = s\mathbf{q} \cdot \mathbf{v} \cdot s^{-1}\mathbf{q}^{-1} = ss^{-1}\mathbf{q} \cdot \mathbf{v} \cdot \mathbf{q}^{-1} = \mathbf{q} \cdot \mathbf{v} \cdot \mathbf{q}^{-1}$.

[2] If the rotation of the body at time $t = 0$, $\mathbf{q}(0) = \mathbf{q}_0$, then the rotation after one time step $\mathbf{q}(1) = \mathbf{q_0} \cdot {}^{\mathbf{L}}\mathbf{q}_{\boldsymbol{\omega}}$ and after two time steps $\mathbf{q}(2) = \mathbf{q_0} \cdot {}^{\mathbf{L}}\mathbf{q}_{\boldsymbol{\omega}} \cdot {}^{\mathbf{L}}\mathbf{q}_{\boldsymbol{\omega}} = \mathbf{q_0} \cdot ({}^{\mathbf{L}}\mathbf{q}_{\boldsymbol{\omega}})^2$, etc.

[3] Using the polar form of the rotation quaternion $\mathbf{q}_\omega = e^{\frac{\theta}{2}u}$,

$$\begin{aligned}(\mathbf{q}_\omega)^t &= e^{t\frac{\theta}{2}u}: \\ \dot{\boldsymbol{q}}(t) &= \frac{d}{dt}\mathbf{q_0}(\mathbf{q}_{\boldsymbol{\omega}})^t \\ &= \frac{d}{dt}\mathbf{q_0}e^{t\frac{\theta}{2}u} \\ &= \mathbf{q_0}e^{t\frac{\theta}{2}u}\mathbf{u}\frac{\theta}{2} \\ &= \mathbf{q}(t)\frac{\theta}{2}\mathbf{u} \\ &= \frac{1}{2}\mathbf{q}(t)\boldsymbol{\omega}\end{aligned}$$

If the angular velocity is not constant, the derivative can be used to quickly estimate the rotation of the body relative to the original orientation by repeatedly sampling the angular velocity and using finite sums:

$$\mathbf{q}(t + \Delta t) \approx \mathbf{q}(t) + \dot{\boldsymbol{q}}(t)\Delta t \tag{14}$$

The error of this estimated rotation quaternion will increase with time; therefore it is necessary to continually correct the estimation.

Another method for estimating the rotation quaternion is to minimise the difference between a reference vector rotated by the estimated rotation quaternion and a measurement of that vector:

$$\min_{\mathbf{q}} f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m) \tag{15}$$

where

$$f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m) = \mathbf{q} \cdot \mathbf{v}_r \cdot \mathbf{q}^{-1} - \mathbf{v}_m \tag{16}$$

and where $\mathbf{q}$ is the estimated rotation quaternion, $\mathbf{v}_r$ is the reference vector and $\mathbf{v}_m$ is the measured vector. Even if only unit quaternions are considered, there are infinite solutions to a rotation of a single vector, therefore two non-parallel reference vectors are required [8].

There are two methods to minimise the difference between two vectors with respect to $\mathbf{q}$: either the length of the difference vector needs to be minimised:

$$\|f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m)\| = 0 \tag{17}$$

or the non-linear equations that make each component of the vector zero need to be solved simultaneously:

$$f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m) = 0 \tag{18}$$

Both the minimisation and the simultaneous non-linear equation approaches can be solved iteratively using gradient descent. When minimising the length:

$$\mathbf{q_{k+1}} = \mathbf{q_k} - \alpha \nabla_{\mathbf{q}} \mathbf{F}(\mathbf{q}) \tag{19}$$

where

$$\mathbf{F}(\mathbf{q}) = \|f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m)\|^2 \tag{20}$$

and α is the step size, and when solving the simultaneous equations:

$$\mathbf{F}(\mathbf{q}) = \frac{1}{2} f^{\mathbf{T}}(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m) f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m) \tag{21}$$

[4]

$$\begin{aligned}
\dot{\boldsymbol{q}}(t) &= \frac{d}{dt}\left(\mathbf{q}_{\omega}^{\mathbf{G}}\right)^t \mathbf{q}_0 \\
&= \frac{d}{dt} e^{t\frac{\theta}{2}u} \mathbf{q}_0 \\
&= \frac{\theta}{2}\mathbf{u} e^{t\frac{\theta}{2}u} \mathbf{q}_0 \\
&= \frac{\theta}{2}\mathbf{u}\mathbf{q}(t) \\
&= \frac{1}{2}\boldsymbol{\omega}\mathbf{q}(t)
\end{aligned}$$

with

$$\nabla_{\mathbf{q}} \mathbf{F}(\mathbf{q}) = \mathbf{J}_{\mathbf{q}}(f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m)) \cdot f(\mathbf{q}, \mathbf{v}_r, \mathbf{v}_m) \tag{22}$$

Both methods for estimating the rotation quaternion can be combined: A single iteration of the difference minimisation gradient descent algorithm can be used to improve the finite sums estimate of the rotation quaternion at each time step, where the rotation at $t = 0$ and $\omega = 0$ is the rotation of the body's reference vectors relative to the measured vectors.

A gyroscope is used to measure a body's angular velocity. Earth's gravity and magnetic field can be used as reference vectors and measured using an accelerometer and a magnetometer respectively.

III. Algorithm Formulation

When calculating the orientation of an IMU or MARG, the reference vectors ($\mathbf{v}_r$) (gravity and magnetic north) are in the global reference frame. Vectors for gravity and magnetic north are measured locally by the rotating body, and each gradient is calculated by applying the reverse rotation to the measured vectors to convert them to the global reference frame, and evaluating them alongside the global reference vectors. Since there are two reference vectors, the algorithm needs to calculate two gradients. From (1) and (6):

$$\begin{aligned}
\mathbf{v}_r \cdot \mathbf{q} = {} & 0q_w - v_{rx}q_x - v_{ry}q_y - v_{rz}q_z \\
& + (0q_x + v_{rx}q_w + v_{ry}q_z - v_{rz}q_y)\mathbf{i} \\
& + (0q_y - v_{rx}q_z + v_{ry}q_w + v_{rz}q_x)\mathbf{j} \\
& + (0q_z + v_{rx}q_y - v_{ry}q_x + v_{rz}q_w)\mathbf{k}
\end{aligned} \tag{23}$$

If we ensure $\mathbf{q}$ is a unit quaternion: $\|\mathbf{q}\|^2 = 1$, the reciprocal can be replaced with the conjugate to simplify calculations; therefore from (1), (2), (4), (7) and (23):

$$\begin{aligned}
\mathbf{q}^{-1} \cdot \mathbf{v}_r \cdot \mathbf{q} = {} & (q_w(-v_{rx}q_x - v_{ry}q_y - v_{rz}q_z) \\
& + q_x(v_{rx}q_w + v_{ry}q_z - v_{rz}q_y) \\
& + q_y(-v_{rx}q_z + v_{ry}q_w + v_{rz}q_x) \\
& + q_z(v_{rx}q_y - v_{ry}q_x + v_{rz}q_w)) + \\
& (q_w(v_{rx}q_w + v_{ry}q_z - v_{rz}q_y) \\
& - q_x(-v_{rx}q_x - v_{ry}q_y - v_{rz}q_z) \\
& - q_y(v_{rx}q_y - v_{ry}q_x + v_{rz}q_w) \\
& + q_z(-v_{rx}q_z + v_{ry}q_w + v_{rz}q_x))\mathbf{i} + \\
& (q_w(-v_{rx}q_z + v_{ry}q_w + v_{rz}q_x) \\
& + q_x(v_{rx}q_y - v_{ry}q_x + v_{rz}q_w) \\
& - q_y(-v_{rx}q_x - v_{ry}q_y - v_{rz}q_z) \\
& - q_z(v_{rx}q_w + v_{ry}q_z - v_{rz}q_y))\mathbf{j} + \\
& (q_w(v_{rx}q_y - v_{ry}q_x + v_{rz}q_w) \\
& - q_x(-v_{rx}q_z + v_{ry}q_w + v_{rz}q_x) \\
& + q_y(v_{rx}q_w + v_{ry}q_z - v_{rz}q_y) \\
& - q_z(-v_{rx}q_x - v_{ry}q_y - v_{rz}q_z))\mathbf{k}
\end{aligned} \tag{24}$$

which when multiplied out:

$$
\begin{aligned}
&= (-v_{rx}q_wq_x - v_{ry}q_wq_y - v_{rz}q_wq_z \\
&\quad + v_{rx}q_wq_x + v_{ry}q_xq_z - v_{rz}q_xq_y \\
&\quad - v_{rx}q_yq_z + v_{ry}q_wq_y + v_{rz}q_xq_y \\
&\quad + v_{rx}q_yq_z - v_{ry}q_xq_z + v_{rz}q_wq_z)+ \\
&\quad (v_{rx}q_wq_w + v_{ry}q_wq_z - v_{rz}q_wq_y \\
&\quad + v_{rx}q_xq_x + v_{ry}q_xq_y + v_{rz}q_xq_z \\
&\quad - v_{rx}q_yq_y + v_{ry}q_xq_y - v_{rz}q_wq_y \\
&\quad - v_{rx}q_zq_z + v_{ry}q_wq_z + v_{rz}q_xq_z)\mathbf{i}+ \\
&\quad (-v_{rx}q_wq_z + v_{ry}q_wq_w + v_{rz}q_wq_x \\
&\quad + v_{rx}q_xq_y - v_{ry}q_xq_x + v_{rz}q_wq_x \\
&\quad + v_{rx}q_xq_y + v_{ry}q_yq_y + v_{rz}q_yq_z \\
&\quad - v_{rx}q_wq_z - v_{ry}q_zq_z + v_{rz}q_yq_z)\mathbf{j}+ \\
&\quad (v_{rx}q_wq_y - v_{ry}q_wq_x + v_{rz}q_wq_w \\
&\quad + v_{rx}q_xq_z - v_{ry}q_wq_x - v_{rz}q_xq_x \\
&\quad + v_{rx}q_wq_y + v_{ry}q_yq_z - v_{rz}q_yq_y \\
&\quad + v_{rx}q_xq_z + v_{ry}q_yq_z + v_{rz}q_zq_z)\mathbf{k}
\end{aligned} \tag{25}
$$

and factored:

$$
\begin{aligned}
&= (v_{rx}(-q_wq_x + q_wq_x - q_yq_z + q_yq_z) \\
&\quad + v_{ry}(-q_wq_y + q_xq_z + q_wq_y - q_xq_z) \\
&\quad + v_{rz}(-q_wq_z - q_xq_y + q_xq_y + q_wq_z))+ \\
&\quad (v_{rx}(q_wq_w + q_xq_x - q_yq_y - q_zq_z) \\
&\quad + v_{ry}(q_wq_z + q_xq_y + q_xq_y + q_wq_z) \\
&\quad + v_{rz}(-q_wq_y + q_xq_z - q_wq_y + q_xq_z))\mathbf{i}+ \\
&\quad (v_{rx}(-q_wq_z + q_xq_y + q_xq_y - q_wq_z) \\
&\quad + v_{ry}(q_wq_w - q_xq_x + q_yq_y - q_zq_z) \\
&\quad + v_{rz}(q_wq_x + q_wq_x + q_yq_z + q_yq_z))\mathbf{j}+ \\
&\quad (v_{rx}(q_wq_y + q_xq_z + q_wq_y + q_xq_z) \\
&\quad + v_{ry}(-q_wq_x - q_wq_x + q_yq_z + q_yq_z) \\
&\quad + v_{rz}(q_wq_w - q_xq_x - q_yq_y + q_zq_z))\mathbf{k}
\end{aligned} \tag{26}
$$

simplifies to:

$$
\begin{aligned}
&=0+ \\
&\quad (v_{rx}(q_wq_w + q_xq_x - q_yq_y - q_zq_z) \\
&\quad + v_{ry}(2q_wq_z + 2q_xq_y) \\
&\quad + v_{rz}(-2q_wq_y + 2q_xq_z))\mathbf{i}+ \\
&\quad (v_{rx}(-2q_wq_z + 2q_xq_y) \\
&\quad + v_{ry}(q_wq_w - q_xq_x + q_yq_y - q_zq_z) \\
&\quad + v_{rz}(2q_wq_x + 2q_yq_z))\mathbf{j}+ \\
&\quad (v_{rx}(2q_wq_y + 2q_xq_z) \\
&\quad + v_{ry}(-2q_wq_x + 2q_yq_z) \\
&\quad + v_{rz}(q_wq_w - q_xq_x - q_yq_y + q_zq_z))\mathbf{k}
\end{aligned} \tag{27}
$$

Using (27) in (16) results in:

$$
\begin{aligned}
&\mathbf{q}^{-1}\cdot\mathbf{v_r}\cdot\mathbf{q} - \mathbf{v_m} = \\
&\quad (v_{rx}(q_wq_w + q_xq_x - q_yq_y - q_zq_z) \\
&\quad + v_{ry}(2q_wq_z + 2q_xq_y) \\
&\quad + v_{rz}(-2q_wq_y + 2q_xq_z) - v_{mx})\mathbf{i}+ \\
&\quad (v_{rx}(-2q_wq_z + 2q_xq_y) \\
&\quad + v_{ry}(q_wq_w - q_xq_x + q_yq_y - q_zq_z) \\
&\quad + v_{rz}(2q_wq_x + 2q_yq_z) - v_{my})\mathbf{j}+ \\
&\quad (v_{rx}(2q_wq_y + 2q_xq_z) \\
&\quad + v_{ry}(-2q_wq_x + 2q_yq_z) \\
&\quad + v_{rz}(q_wq_w - q_xq_x - q_yq_y + q_zq_z) - v_{mz})\mathbf{k}
\end{aligned} \tag{28}
$$

The derivatives of (28) with respect to each component of q are:

$$
\begin{aligned}
&\frac{\delta}{\delta\mathbf{q}_w}(\mathbf{q}^{-1}\cdot\mathbf{v_r}\cdot\mathbf{q} - \mathbf{v_m}) = \\
&\qquad (2v_{rx}q_w + 2v_{ry}q_z - 2v_{rz}q_y)\mathbf{i}+ \\
&\qquad (-2v_{rx}q_z + 2v_{ry}q_w + 2v_{rz}q_x)\mathbf{j}+ \\
&\qquad (2v_{rx}q_y - 2v_{ry}q_x + 2v_{rz}q_w)\mathbf{k}
\end{aligned} \tag{29}
$$

$$
\begin{aligned}
&\frac{\delta}{\delta\mathbf{q}_x}(\mathbf{q}^{-1}\cdot\mathbf{v_r}\cdot\mathbf{q} - \mathbf{v_m}) = \\
&\qquad (2v_{rx}q_x + 2v_{ry}q_y + 2v_{rz}q_z)\mathbf{i}+ \\
&\qquad (2v_{rx}q_y - 2v_{ry}q_x + 2v_{rz}q_w)\mathbf{j}+ \\
&\qquad (2v_{rx}q_z - 2v_{ry}q_w - 2v_{rz}q_x)\mathbf{k}
\end{aligned} \tag{30}
$$

$$
\begin{aligned}
&\frac{\delta}{\delta\mathbf{q}_y}(\mathbf{q}^{-1}\cdot\mathbf{v_r}\cdot\mathbf{q} - \mathbf{v_m}) = \\
&\qquad (-2v_{rx}q_y + 2v_{ry}q_x - 2v_{rz}q_w)\mathbf{i}+ \\
&\qquad (2v_{rx}q_x + 2v_{ry}q_y + 2v_{rz}q_z)\mathbf{j}+ \\
&\qquad (2v_{rx}q_w + 2v_{ry}q_z - 2v_{rz}q_y)\mathbf{k}
\end{aligned} \tag{31}
$$

$$
\begin{aligned}
&\frac{\delta}{\delta\mathbf{q}_z}(\mathbf{q}^{-1}\cdot\mathbf{v_r}\cdot\mathbf{q} - \mathbf{v_m}) = \\
&\qquad (-2v_{rx}q_z + 2v_{ry}q_w + 2v_{rz}q_x)\mathbf{i}+ \\
&\qquad (-2v_{rx}q_w - 2v_{ry}q_z + 2v_{rz}q_y)\mathbf{j}+ \\
&\qquad (2v_{rx}q_x + 2v_{ry}q_y + 2v_{rz}q_z)\mathbf{k}
\end{aligned} \tag{32}
$$

The Jacobian can be constructed using (29) - (32):

$$\boldsymbol{J}_q{}^T\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)=$$

$$\begin{bmatrix} 2v_{rx}q_w+2v_{ry}q_z-2v_{rz}q_y & -2v_{rx}q_z+2v_{ry}q_w+2v_{rz}q_x \\ 2v_{rx}q_x+2v_{ry}q_y+2v_{rz}q_z & 2v_{rx}q_y-2v_{ry}q_x+2v_{rz}q_w \\ -2v_{rx}q_y+2v_{ry}q_x-2v_{rz}q_w & 2v_{rx}q_x+2v_{ry}q_y+2v_{rz}q_z \\ -2v_{rx}q_z+2v_{ry}q_w+2v_{rz}q_x & -2v_{rx}q_w-2v_{ry}q_z+2v_{rz}q_y \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} 2v_{rx}q_y-2v_{ry}q_x+2v_{rz}q_w \\ 2v_{rx}q_z-2v_{ry}q_w-2v_{rz}q_x \\ 2v_{rx}q_w+2v_{ry}q_z-2v_{rz}q_y \\ 2v_{rx}q_x+2v_{ry}q_y+2v_{rz}q_z \end{bmatrix}$$

To get the gradient of the multivariate function used to solve the simultaneous non-linear equations with respect to q, (28) and (33) are substituted into (22):

$$\nabla\mathbf{F}_q=\mathbf{J}_q{}^T\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)\cdot\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)=$$

$$\begin{bmatrix} 2v_{rx}q_w+2v_{ry}q_z-2v_{rz}q_y & -2v_{rx}q_z+2v_{ry}q_w+2v_{rz}q_x \\ 2v_{rx}q_x+2v_{ry}q_y+2v_{rz}q_z & 2v_{rx}q_y-2v_{ry}q_x+2v_{rz}q_w \\ -2v_{rx}q_y+2v_{ry}q_x-2v_{rz}q_w & 2v_{rx}q_x+2v_{ry}q_y+2v_{rz}q_z \\ -2v_{rx}q_z+2v_{ry}q_w+2v_{rz}q_x & -2v_{rx}q_w-2v_{ry}q_z+2v_{rz}q_y \end{bmatrix} \begin{bmatrix} 2v_{rx}q_y-2v_{ry}q_x+2v_{rz}q_w \\ 2v_{rx}q_z-2v_{ry}q_w-2v_{rz}q_x \\ 2v_{rx}q_w+2v_{ry}q_z-2v_{rz}q_y \\ 2v_{rx}q_x+2v_{ry}q_y+2v_{rz}q_z \end{bmatrix} \cdot \tag{34}$$

$$\begin{bmatrix} v_{rx}(q_wq_w+q_xq_x-q_yq_y-q_zq_z) \\ +v_{ry}(2q_wq_z+2q_xq_y) \\ +v_{rz}(-2q_wq_y+2q_xq_z) & -v_{mx} \\ \\ v_{rx}(-2q_wq_z+2q_xq_y) \\ +v_{ry}(q_wq_w-q_xq_x+q_yq_y-q_zq_z) \\ +v_{rz}(2q_wq_x+2q_yq_z) & -v_{my} \\ \\ v_{rx}(2q_wq_y+2q_xq_z) \\ +v_{ry}(-2q_wq_x+2q_yq_z) \\ +v_{rz}(q_wq_w-q_xq_x-q_yq_y+q_zq_z) & -v_{mz} \end{bmatrix}$$

The calculation of the gradient of the difference function is simplified if one, or better two, of the axes of the original reference vector are zero. Therefore the reference vector for earth's gravity is assumed to only operate in the vertical direction and only the magnetic field declination is used [7].

With $\mathbf{v}_r=[0,0,-1]$, (34) simplifies to:

$$\nabla\mathbf{F}_q=\mathbf{J}_q{}^T\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)\cdot\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)=$$

$$\begin{bmatrix} 2q_y & -2q_x & -2q_w \\ -2q_z & -2q_w & 2q_x \\ 2q_w & -2q_z & 2q_y \\ -2q_x & -2q_y & -2q_z \end{bmatrix} \cdot \tag{35}$$

$$\begin{bmatrix} 2q_wq_y-2q_xq_z-v_{mx} \\ -2q_wq_x-2q_yq_z-v_{my} \\ -q_wq_w+q_xq_x+q_yq_y-q_zq_z-v_{mz} \end{bmatrix}$$

With $\mathbf{v}_r=[\mathbf{v}_{rx},0,\mathbf{v}_{rz}]$, (34) simplifies to:

$$\nabla\mathbf{F}_q=\mathbf{J}_q{}^T\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)\cdot\left(\mathbf{q}^{-1}\cdot\mathbf{v}_r\cdot\mathbf{q}-\mathbf{v}_m\right)=$$

$$\begin{bmatrix} 2v_{rx}q_w-2v_{rz}q_y & -2v_{rx}q_z+2v_{rz}q_x \\ 2v_{rx}q_x+2v_{rz}q_z & 2v_{rx}q_y+2v_{rz}q_w \\ -2v_{rx}q_y-2v_{rz}q_w & 2v_{rx}q_x+2v_{rz}q_z \\ -2v_{rx}q_z+2v_{rz}q_x & -2v_{rx}q_w+2v_{rz}q_y \end{bmatrix} \begin{bmatrix} 2v_{rx}q_y+2v_{rz}q_w \\ 2v_{rx}q_z-2v_{rz}q_x \\ 2v_{rx}q_w-2v_{rz}q_y \\ 2v_{rx}q_x+2v_{rz}q_z \end{bmatrix} \cdot \tag{36}$$

$$\begin{bmatrix} v_{rx}(q_wq_w+q_xq_x-q_yq_y-q_zq_z) \\ +v_{rz}(-2q_wq_y+2q_xq_z) & -v_{mx} \\ \\ v_{rx}(-2q_wq_z+2q_xq_y) \\ +v_{rz}(2q_wq_x+2q_yq_z) & -v_{my} \\ \\ v_{rx}(2q_wq_y+2q_xq_z) \\ +v_{rz}(q_wq_w-q_xq_x-q_yq_y+q_zq_z) & -v_{mz} \end{bmatrix}$$

The gradients from (35) and (36) are then used to make two steps in the gradient descent algorithm [8], thereby correcting for the errors introduced in (14).

IV. Results and Discussion

The graphs in Fig. 1. show that our reformulation of the Jacobian and the difference equations (34), (35) and (36), using a single step of the gradient descent algorithm (19) with an α of 0.1, to simultaneously solve the non-linear equations (18) defined by the difference between reference vectors and reversely rotated measured vectors (16) for iteratively updating the rotation unit quaternions provide five orders of magnitude improvement on the Madgwick et al formulation of the estimated rotation quaternion in dynamic scenarios.

The precision of a 32-bit floating point number is limited by the 23 bits allocated to the fraction component of the IEEE 754 single-precision binary floating-point format, which equates to 7.2 decimal digits. As a consequence, most of the errors in our formulation can be attributed to the limits of the number representation.

The speed of the algorithm will limit the minimum time delta in the finite sums algorithm (14). Ultimately the speed of the algorithm is dependent on the processor speed, but within the constraints of a particular processor, for example

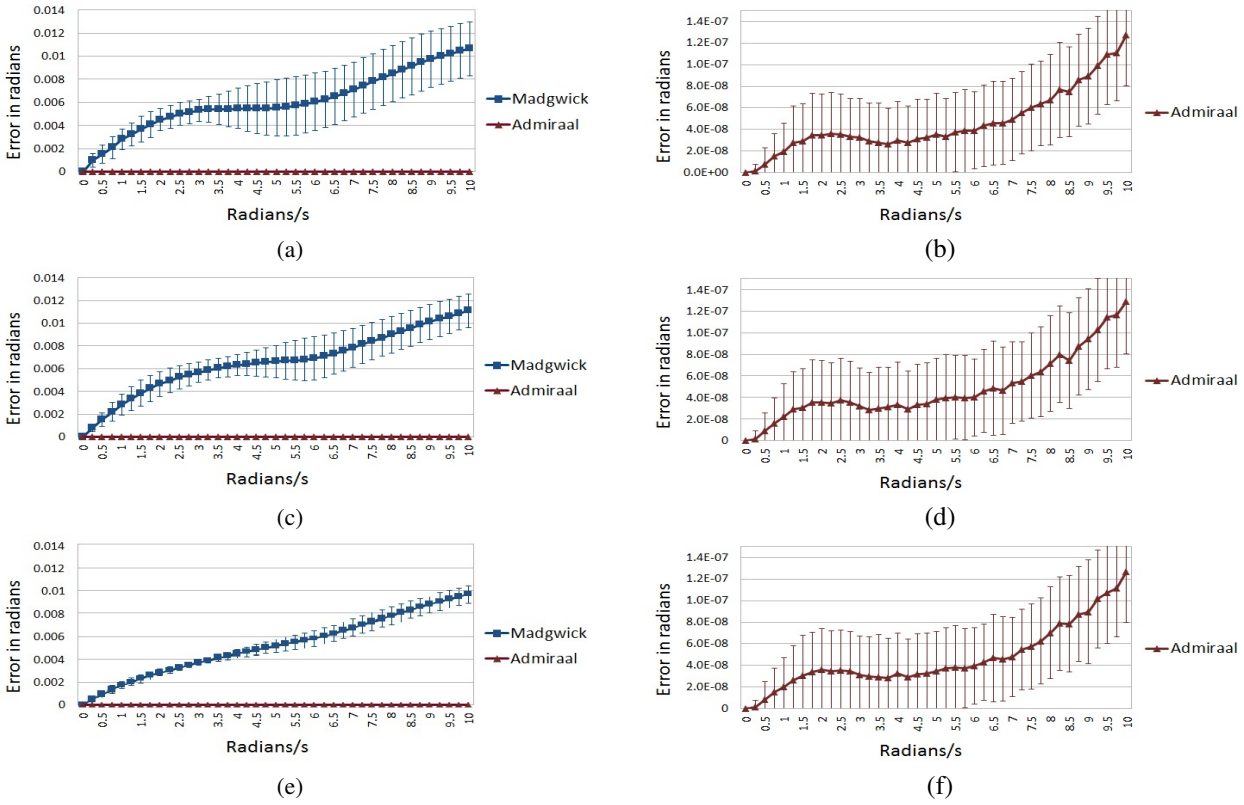

Fig. 1. Comparison between 1,000 iterations using a sample period of 1ms and no magnetic inclination offset of the Madgwick formulation and our reformulation of the estimated rotation quaternion algorithms under dynamic conditions of increasing rotational speed around the x-axis: (a) and (b), the y-axis: (c) and (d) and the z-axis: (e) and (f). The graphs on the left compare the two formulations. The graphs on the right show the errors of my formulation at an increased resolution of five orders of magnitude. The error bars represent one standard deviation.

the processor onboard an IMU, the speed depends on the total number of clock cycles required.

Depending on the processor, using scaled 32-bit integers to represent fixed-point numbers instead of 32-bit floating point numbers can significantly reduce the total number of clock cycles required. On the PIC24FJ running at 30 MHz the floating point version of the algorithm cannot run faster than 320 Hz, but the fixed-point integer scaled by 214 version of the algorithm can be run at over 750 Hz.

The precision of the algorithm is also dependent on the size of α in the gradient descent algorithm (19). Fig. 2. shows the impact of varying α on the precision of the algorithm at different rotational speeds around each axis. The results show that α = 0.25 or α = 0.12 with either algorithm provides the best precision across a range of rotational speeds. The precision available when working with fixed-point integers scaled by 2^{14} is only 4.2 decimal digits or 10^{-4}, which accounts for most of the difference between the precision of the floating-point and fixed-point integer representations.

The precision of the algorithm is less relevant when considering the precision of the gyroscope, accelerometer and magnetometer being used to measure the inputs to the quaternion algorithm. Fig. 3. shows the impact of the precision of the inputs on the precision of the algorithm with varying step sizes α. The precision of the magnetometer has the greatest impact on the precision of the algorithm. The results also suggest that to minimise the impact, a value of α such that $\alpha : 0.016 < \alpha < 0.12$ should be used.

V. Conclusions and Future Work

Machine interface systems are becoming more reliant on the recognition of human motion to generate commands or infer intention. Environments where cameras or other external sensors may be occluded demand wearable systems for recognition of human motion. Of critical import to this vision is the capacity to fuse inertial, gyroscopic, and magnetometer (MARG) data for spatial orientation. While many such systems exist today, there remain issues to be resolved in their operation such as drift with time, computational efficiency, and miniaturization in low-power wearable, or even implantable, packages.

We report the formulation and implementation of a new gradient descent algorithm for efficient fusion of MARG data to provide rigid body motion for human machine interface applications. The primary difference between the new formulation versus existing implementations is that we have ensured that the gradient descent algorithm uses the steepest descent direction to correct for errors resulting from repeated quantised integration, sensor bias and noise. Although the steepest descent direction will not always result in a correction towards the actual minimum, simulations and real-world experiments show that our formulations both correct the integral errors faster, and are effective in a wider

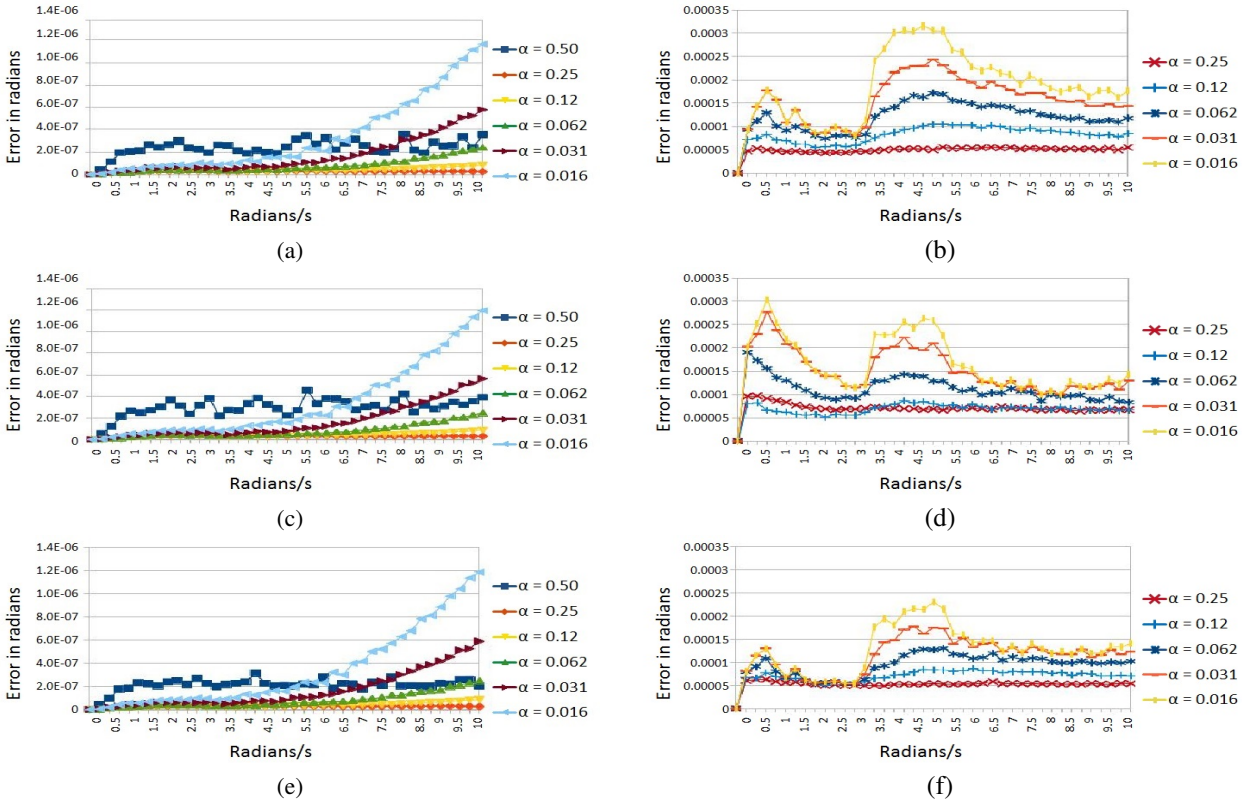

Fig. 2. The impact of α - the step size of the gradient descent algorithm - on the precision of the floating-point (left) and fixed-point integer (right) versions of the algorithm under dynamic conditions of increasing rotational speed around the x-axis: (a) and (b), the y-axis: (c) and (d) and the z-axis: (e) and (f). The precision of the fixed-point integer version of the algorithm with an α value of 0.50 is an order of magnitude worse, therefore it is not shown.

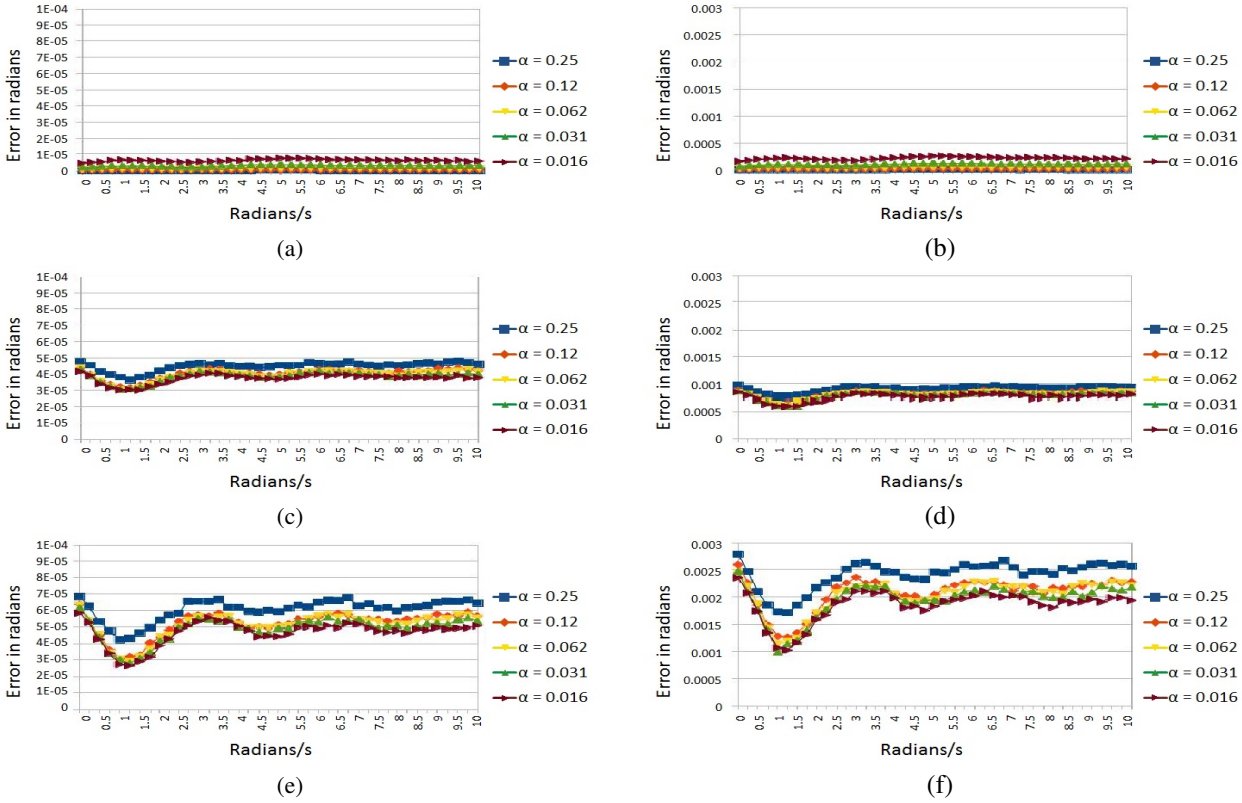

Fig. 3. Impact on the precision of the floating-point version of the algorithm with stated (left) and measured (right) precision on the gyroscope: (a) and (b), the accelerometer: (c) and (d) and the magnetometer: (e) and (f) *with different gradient descent step-sizes:* α.

range of sensor input biases, ranges and time-steps. The improved efficiency and accuracy show significant potential for increasing the scope of inertial measurement systems in applications where low power or greater precision is necessary.

The new formulation of the gradient descent algorithm is presently being implemented in a number of research areas, including ubiquitous computer control and biomedical symptom monitoring in Parkinson's and stroke patients. The reduction in computational load from the conversion to fixed point numbers, along with the reduction in sample rate that is possible with the more efficient formulation will allow for devices which are more power efficient, thereby increasing the length of tests outside controlled environments without increasing the physical properties of the device. We are also initiating work to scope the potential of the algorithm to enable implantable inertial measurement systems to fully exploit its potential for minimization of size and power.

REFERENCES

[1] Vaidyanathan R., Fargues M., Gupta L., Kurcan R., Lin D., Kota S., Quinn R.D., 'A Dual Mode Human-Robot Interface Based On Physiological Signal Capture In The Aural Cavity', International Journal of Robotics Research (IJRR), 26, 11-12, pp 1205-1223, 2007

[2] Gupta L., Kota S., Murali S., Molfese D., Vaidyanathan R., 'A Feature Ranking Strategy To Facilitate Multivariate Signal Classification', IEEE Transactions on Systems, Man, and Cybernetics C (SMC-C), 40, 1, pp 98-108, 2010

[3] Jared Alan Frank, Sai Prasanth Krishnamoorthy, Vikram Kapila, 'Toward Mobile Mixed-Reality Interaction With Multi-Robot Systems', Robotics and Automation Letters IEEE, vol. 2, pp. 1901-1908, 2017

[4] J Burridge, A Lee, R Turk, M Stokes, J Whitall, R Vaidyanathan, P Clatworthy, A M Hughes, C Meagher, E Franco, L Yardley, 'Telehealth, Wearable Sensors, And The Internet: Will They Improve Stroke Outcomes Through Increased Intensity Of Therapy, Motivation, And Adherence To Rehabilitation Programs?' Journal of Neurologic Physical Therapy, 1-25, 2017

[5] S Wilson, R Vaidyanathan, 'Upper-Limb Prosthetic Control Using Wearable Multichannel Mechanomyography', IEEE International Conference on Rehabilitation Robotics (ICORR), 6 pp, London, UK, July 2017

[6] R Woodward, S Shefelbine, R Vaidyanathan, 'Gait Analysis Using Pervasive Motion Tracking And Mechanomyography Fusion' accepted, IEEE Transactions on Mechatronics, 12pp (IF: 3.9, 2016)

[7] S. O.H. Madgwick, A.J.L. Harrison, P. M. Sharkey, R. Vaidyanathan, W. S. Harwin, 'Measuring Motion With Kinematically Redundant Accelerometer Arrays: Theory, Simulation And Implementation', Mechatronics, 23,5, pp 518-529, 2013

[8] S. O. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, 'Estimation Of IMU And MARG Orientation Using A Gradient Descent Algorithm', in Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on, pp. 1-7., 2011

[9] V. Elvira and A. Nazbal-Renteria and A. Arts-Rodrguez,'A Novel Feature Extraction Technique For Human Activity Recognition',2014 IEEE Workshop on Statistical Signal Processing (SSP),pp.177-180,2014

[10] M. O'Reilly and D. Whelan and C. Chanialidis and N. Friel and E. Delahunt and T. Ward and B. Caulfield,'Evaluating Squat Performance With A Single Inertial Measurement Unit',2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN),pp.1-6,2015

[11] M. Giuberti and G. Ferrari and L. Contin and V. Cimolin and C. Azzaro and G. Albani and A. Mauro, 'Assigning UPDRS Scores In The Leg Agility Task Of Parkinsonians: Can It Be Done Through BSN-Based Kinematic Variables?',IEEE Internet of Things Journal,vol.2,no.1,pp.41-51,2015

[12] B. J. Borbly and A. Tihanyi and P. Szolgay,'A Measurement System For Wrist Movements In Biomedical Applications',2015 European Conference on Circuit Theory and Design (ECCTD),pp.1-4,2015

[13] F. Parisi and G. Ferrari and M. Giuberti and L. Contin and V. Cimolin and C. Azzaro and G. Albani and A. Mauro,'Inertial BSN-Based Characterization And Automatic UPDRS Evaluation Of The Gait Task of Parkinsonians',IEEE Transactions on Affective Computing,vol.7,no.3,pp.258-271,2016

[14] T. Lisini Baldi and M. Mohammadi and S. Scheggi and D. Prattichizzo,'Using Inertial and Magnetic Sensors For Hand Tracking And Rendering In Wearable Haptics'2015 IEEE World Haptics Conference (WHC),pp.381-387,2015

[15] H. Debarba and L. Nedel and A. Maciel,'LOP-cursor: Fast And Precise Interaction With Tiled Displays Using One Hand And Levels Of Precision',2012 IEEE Symposium on 3D User Interfaces (3DUI),pp.125-132,2012

[16] J. C. Morgre and J. P. Diguet and J. Laurent,'Electronic Navigational Chart Generator For A Marine Mobile Augmented Reality System',2014 Oceans - St. John's,pp.1-9,2014

[17] Y. Zhang and Y. Fei and L. Xu and G. Sun,'Micro-IMU-based Motion Tracking System For Virtual Training',2015 34th Chinese Control Conference (CCC),pp.7753-7758,2015

[18] N. Strozzi and F. Parisi and G. Ferrari,'A multifloor Hybrid Inertial/Barometric Navigation System',2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN),pp.1-5,2016

[19] N. Strozzi and F. Parisi and G. Ferrari,'On Single Sensor-based Inertial Navigation',2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN),pp.300-305,2016

[20] K. Caluwaerts and J. Bruce and J. M. Friesen and V. SunSpiral,'State Estimation For Tensegrity Robots',2016 IEEE International Conference on Robotics and Automation (ICRA),pp.1860-1865,2016

[21] P. Marantos and Y. Koveos and K. J. Kyriakopoulos,'UAV State Estimation Using Adaptive Complementary Filters',IEEE Transactions on Control Systems Technology,vol.24,no.4,pp.1214-1226,2016

[22] A. Lorusso, D. W. Eggert, A. L. D. Eggert, and R. B. Fisher, 'A Comparison Of Four Algorithms For Estimating 3-D Rigid Transformations', 1995.

[23] W. R. Hamilton, 'On Quaternions; Or On A New System Of Imaginaries In Algebra', Philosophical Magazine, 1850-1844.

[24] K. Shoemake, 'Animating Rotation With Quaternion Curves', in ACM SIGGRAPH computer graphics, vol. 19, pp. 245-254., 1985.

[25] 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', Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on, Monterey, CA, pp. 187-194., 1999.

[26] S. W. Shepperd, 'Quaternion From Rotation Matrix. [Four-Parameter Representation Of Coordinate Transformation Matrix]', J. Guid. Control, Jun. 1978.

[27] J. Diebel, 'Representing Attitude: Euler Angles, Unit Quaternions, And Rotation Vectors', Matrix, vol. 58, no. 15-16, pp. 1-35, 2006.