Mechanical Systems and Signal Processing 130 (2019) 183–200

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Formulation of a new gradient descent MARG orientation algorithm: Case study on robot teleoperation

Samuel Wilson [a,f], Henry Eberle [b], Yoshikatsu Hayashi [b], Sebastian O.H. Madgwick [c], Alison McGregor [d], Xingjian Jing [e], Ravi Vaidyanathan [a,*]

[a] Department of Mechanical Engineering, Imperial College London, UK
[b] School of Biological Sciences, University of Reading, UK
[c] x-io Technologies Limited, Bristol, UK
[d] Department of Surgery and Cancer, Imperial College London, UK
[e] Department of Mechanical Engineering, Hong Kong Polytechnic University, Hong Kong
[f] SERG Technologies Limited, UK

ARTICLE INFO

Article history:
Received 24 December 2018
Received in revised form 27 March 2019
Accepted 30 April 2019

Keywords:
Inertial sensor fusion
mechatronic sensing
Robot teleoperation
Human-machine interface (HMI)
inertial measurement unit (IMU)
Wearable sensors

ABSTRACT

We introduce a novel magnetic angular rate gravity (MARG) sensor fusion algorithm for inertial measurement. The new algorithm improves the popular gradient descent ('Madgwick') algorithm increasing accuracy and robustness while preserving computational efficiency. Analytic and experimental results demonstrate faster convergence for multiple variations of the algorithm through changing magnetic inclination. Furthermore, decoupling of magnetic field variance from roll and pitch estimation is proven for enhanced robustness. The algorithm is validated in a human-machine interface (HMI) case study. The case study involves hardware implementation for wearable robot teleoperation in both Virtual Reality (VR) and in real-time on a 14 degree-of-freedom (DoF) humanoid robot. The experiment fuses inertial (movement) and mechanomyography (MMG) muscle sensing to control robot arm movement and grasp simultaneously, demonstrating algorithm efficacy and capacity to interface with other physiological sensors. To our knowledge, this is the first such formulation and the first fusion of inertial measurement and MMG in HMI. We believe the new algorithm holds the potential to impact a very wide range of inertial measurement applications where full orientation necessary. Physiological sensor synthesis and hardware interface further provides a foundation for robotic teleoperation systems with necessary robustness for use in the field.
© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Applications for MEMS IMU (Inertial Measurement Units)/MARG (Magnetic, Angular Rate, and Gravity) sensor packages capturing world-referenced orientation are proliferating. Recent examples include localisation in real and virtual environments [1–5], autonomous robotics [6–9], aerospace navigation/control [10], medical diagnostics/rehabilitation [11–13], Human Machine Interface (HMI) [14,15] and sports/health tracking [16]. Ideally sensor packages should be lightweight, self-contained, low-cost units with the capacity to determine orientation without external environmental knowledge. While many systems simply use translational or rotational acceleration for assessment, demand is increasing for specific

* Corresponding author.
E-mail address: r.vaidyanathan@imperial.ac.uk (R. Vaidyanathan).

https://doi.org/10.1016/j.ymssp.2019.04.064
0888-3270/© 2019 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

knowledge of limb orientation and kinematic metrics (e.g. joint angle) [17], which necessitates integration of heterogeneous sensor data.

Orientation estimation is fundamentally a problem of multisensor fusion. Gyroscope output can map changes from a known start position through integration [18], which theoretically provides full angular orientation. Practically, however, errors in gyroscopic data are introduced through noise or fast changes in direction; these errors are cumulative, thus reliability drops with every time step, leading to very significant inaccuracy. Gravitational acceleration can be used to correct both pitch and roll orientation estimates by comparing a gravity vector, measured using an accelerometer, to a predefined reference. Similarly magnetometers observing the geomagnetic field can be used to provide an estimation for yaw. Consequently, the most accurate algorithms for determining orientation use a combination of all three modalities through sensor fusion. Fusion can be straightforward in stationary and magnetically clean environments, however accelerometer data is corrupted by linear translation caused by movement and magnetometers are subject to environmental magnetic interference. Both are skewed by bias and drift [19,20]. MARG sensor fusion algorithms combine sensor data to correct these issues.

The complementary filter, Kalman Filter, and gradient descent ('Madgwick') filter have been described as the 'prominent' techniques for MARG sensor fusion today [21]. Arguably the simplest of these is the complementary filter. Complementary filters fuse weighted sums of gyroscope and accelerometer output to estimate orientation. Weights are derived based on sensor reliability (e.g. high frequency noise vs low frequency bias or drift). More advanced complementary filters adapt weights based on the task domain in the sensor workspace [22]. Complementary filters are straightforward to implement and computationally efficient, thus popular in situations where computational resources are low. However, they are slower to converge and less accurate than available alternatives since they only use current data, thus have no direct method of compensating for sensor noise [23].

Noisy data and bias in real-world inertial monitoring systems has motivated implementation of Kalman Filters [24,25] for MARG fusion. Kalman Filters are recursive functions which predict the current state of a linear problem by observing the current input data, the previous input data and the previous state prediction. When the problem is non-linear, such as when observing movement, an Extended Kalman Filter (EKF) can be used [26]. Kalman filter algorithms are particularly strong in situations with significant external noise, such as quadcopter control. Several of methods for extending Kalman filters have also been developed to overcome errors introduced in EKF approximations [27]. Commercial packages such as xsens [28], micro-strain [29], VectorNav [30], Intersense [31] and PNI [32] all rely on a form of the Kalman Filter. Generally, the EKF provides more accurate orientation than complementary filters, but suffers from "system complexity and computation time resulting in high latencies in a real-time environment" [21].

In a previous work, our team introduced a gradient descent fusion algorithm, commonly known as 'Madgwick's Algorithm' [19], as a computationally lean alternative for MARG sensor fusion. That algorithm estimates a vector representing the sensor output at the current orientation and minimizes the difference between predicted and observed sensor output with two gradient descent calculations. This work paralleled that of Mahony [20], who implemented a similar approach using a PI controller. We have open sourced code for this algorithm, which has been very extensively downloaded [33]. The algorithm is widely acknowledged as having had disruptive impact in inertial measurement due to "...benefits such as low computational cost, no need for defining the reference direction of the earth's magnetic field and easy tuning" [34]. However, its accuracy is inhibited by the fact that pitch and roll estimations are not decoupled from magnetic disturbances [35,36], and that magnetic inclination impacts convergence.

1.1. Algorithm trade-offs

MARG sensor fusion algorithm selection is often dictated by the nature of the input and target application. Available algorithms today force compromises between computiational efficiency, robustness, and complexity in tuning and implementaiton. Complementary filters are the most computationally efficient, but compromise accuracy when noise/bias is introduced into the system [37]. The EKF, once tuned, achieves high accuracy for specific conditions, but at the cost of a computational expense [37,38]. The Mahony and Madgwick algorithms are comparable in terms of accuracy and efficiency, and are often used together as a benchmark when evaluating other algorithms [35,39,40,9]. Madgwick's algorithm was very recently reported to be 'the most accurate open source orientation estimation algorithm' [41].

The widespread use of the gradient descent algorithm is a testament to its balance of accuracy, utility, and computational efficiency. However, a challenge in executing two gradient descent calculations consecutively is that the gradients may not be perpendicular in n-dimensional space. In this case, the steepest descent step in one direction may be affected by the step along the subsequent gradient vector, which can impact convergence time. In the gradient descent MARG algorithm, the two steps are defined based on reference vectors for gravity and for local magnetic field. The local magnetic field has a horizontal component, which is used to determine the direction to magnetic north, and a vertical component, resulting in a deviation from the horizontal referred to as magnetic inclination. The inclination varies from 0 at the magnetic equator to $\pi/2$ at the magnetic poles, meaning that it also varies between 0 and $\pi/2$ from the gravity vector. When these two vectors are not perpendicular, interference between the two steps will lead to slower yaw convergence but a faster convergence in the roll direction. This can be beneficial in applications such as quadcopter control, where unobserved changes in pitch and roll result in unwanted lateral movement and slower convergence in yaw is of little consequence. A side effect, however, is that calculation of roll is not decoupled from magnetic interference [34], which is a benefit of Kalman filtering. The time taken for yaw convergence is also dependent on magnetic inclination; as inclination approaches $\pi/2$ the convergence time will

approach infinity. To our knowledge the only other work addressing these issues in the gradient descent was by Yean et al. who proposed synergistic implementation with the Kalman filter [21].

In summary, the gradient descent algorithm is very widely used as a low power solution whose overall accuracy is comparable to far more complex methods. However, slower than optimum convergence and mathematical coupling of non-perpendicular vectors limit its potential use. The algorithm is well suited for implementation on systems with limited processing power, such as wearable robotics for human machine interface, but such applications can include fast movements and changes in yaw angle, which have to potential to amplify these shortcomings. It also requires prior knowledge of the user environment, specifically the inclination of the local geomagnetic field. As such, a new algorithm which maintains the overall accuracy, but is unaffected by these issues holds significant potential to impact inertial orientation measurement.

1.2. Scope of work

In this paper we introduce a new algorithm for MARG sensor fusion for full orientation estimation which addresses the most significant shortcomings in existing alternatives. The algorithm is based on a new formulation in which one gradient descent calculation is formulated such that roll and pitch are decoupled from magnetic interference. We present the mathematical formulation, implementation, and testing of the new algorithm. Results demonstrate convergence several times faster than the original gradient decent algorithm and as well as robustness to changes magnetic inclination. We show improved convergence times for multiple variations of the algorithm specifically in the presence of changing magnetic inclination and simulated magnetic disturbance as well as the decoupling of magnetic interference on roll and pitch. We then present a case-study demonstrating the use of the algorithm in a humam-machine interface (HMI) application. This case study implements the algorithm for a wearable gesture-based interface both in a Virtual Reality (VR) environment and as a control mechanism for the teleoperation of a Baxter, a humanoid robot from Rethink Robotics by fusing inertial (movement) and mechanomyography (MMG, muscle sensing) data.

In summary, new contributions of this investigation include:

1. The complete formulation of a new algorithm for MARG sensor fusion and its implementation for 3D orientation estimation. The algorithm extends the popular gradient descent fusion method by decoupling roll and pitch from magnetic interference and is not impacted by magnetic inclination.
2. Testing of the algorithm in comparison to other implementations of the gradient descent algorithm demonstrating improved performance
3. A case study implementation in robot teleoperation demonstrating algorithm efficacy as well as capacity to interface with other physiological (MMG muscle) sensors.

To our knowledge, this is the first time such an improved algorithm has been both formulated and demonstrated. It is also the first time an IMU and MMG based HMI has been used to control a 14DoF robotic manipulator controlling arm movement and grasp simultaneously. We believe the new algorithm holds the potential to impact a very wider range of inertial measurement applications akin to the original gradient descent algorithm at its introduction.

1.3. Organization of paper

The remainder of this paper is organized as follows: Section 2 outlines the mathematical formulation of the new algorithm and results comparing it to the original gradient descent algorithm as well as a recent improved formulation [42]. Section 3 presents a case study where we demonstrate the algorithm in an HMI application. This application is broken down into two subsections, the first performs a validation experiment using the algorithm in a VR environment and the second demonstrates implementation in a robot teleoperation. Section 4 summarizes the work and draws conclusions as a basis for future work.

2. Algorithm formulation and testing

2.1. Algorithm formulation

The original gradient descent algorithm uses a combination of accelerometer data to provide an estimation for local gravity, magnetometer data to provide an estimation for local magnetic field, and gyroscopic data to track changes in orientation. This raw MARG data is then converted to a world referenced orientation using a gradient descent algorithm [23]. Broadly, a unit quaternion is generated, and at each sample is modified based on two calculations; first, the gyroscope data provides a rotation quaternion through integration. Second, the inverse of the quaternion is applied accelerometer and magnetometer vectors to predict output in the direction of gravity and magnetic north vectors. The difference between predicted sensor and measured sensor output are used to generate two gradients. A corrective step along each is taken, generating the second rotation quaternion. These two rotation quaternions are weighted and combined to provide an estimation for a rotation to align with the world reference orientation. The combination of sensors allows the gyroscope to track orientation during high frequency motion while gyroscopic drift is compensated during low frequency motion using the gradient descent steps.

The new gradient descent algorithm improves the generation of the world referenced 'corrective step'. During this step, world referenced vectors $\mathbf{v}_r$ for gravity $\mathbf{v}_{r(a)}$ and for the earth's magnetic field $\mathbf{v}_{r(m)}$ are defined as:

$$\mathbf{v}_{r(a)} = [0, 0, -1] \tag{1}$$

and

$$\mathbf{v}_{r(m)} = [\mathbf{v}_{rx}, 0, \mathbf{v}_{rz}] \tag{2}$$

respectively. The reference vectors are rotated by the current quaternion as described in [43], and the difference between the measured vector $\mathbf{v}_m$ and the reference vector $\mathbf{v}_r$ after rotation is calculated:

$$f(q, v_r, v_m) = q \cdot v_r \cdot q^{-1} - v_m \tag{3}$$

Since the difference should ideally be zero, $f(q, v_r, v_m)$ must be minimized in terms of q.
This is achieved using gradient descent:

$$q_{k+1} = q_k - \alpha \nabla_q F(q) \tag{4}$$

where α is the step size ($\alpha = 0.1$ for experiments in this investigation), k is the current sample and $\nabla_q F(q)$ is the gradient, given by

$$\nabla F_q = J_q \left(q^{-1} \cdot v_r \cdot q - v_m\right)^T \cdot \left(q^{-1} \cdot v_r \cdot q - v_m\right) \tag{5}$$

One reason for slower than optimum convergence is that, in nearly all cases, the gradients will not be perpendicular, resulting in an element of the second step being in the same plane as the first. This results in a modified step size, leading to a slower convergence. The smaller the angle between the two measured vectors (i.e. the greater the inclination), the larger the effect that the second step has on the first. It has also been noted that the magnetic correction is not decoupled from the pitch and roll of the estimation [34]. To overcome this, a new vector for the magnetic field $(V_{m(e)})$ which is exactly perpendicular to the vector representing the accelerometer $(V_{m(a)})$ is generated. This new reference vector is derived using the cross product as follows:

$$V_{m(e)} = \mid V_{m(a)} \times V_{m(m)} \mid \tag{6}$$

where $V_{m(m)}$ is the measured vector for the magnetic field.

This process is also applied to the reference vectors to create an equivalent reference $(V_{r(e)})$ by substituting (1) and (2) into (6), giving:

$$V_{r(e)} = \mid [0, 0, -1] \times [\mathbf{v}_{rx}, 0, \mathbf{v}_{rz}] \mid \tag{7}$$

$$V_{r(e)} = [0, -1, 0] \tag{8}$$

This new reference vector can be substituted into (5) to create a new gradient to replace the non-perpendicular gradient defined in the original algorithm.

With $\mathbf{v}_r = [0, -1, 0]$, (5) simplifies to:

$$\nabla \mathbf{F}_q = \mathbf{J}_q \left(\mathbf{q}^{-1} \cdot \mathbf{v}_r \cdot \mathbf{q} - \mathbf{v}_m\right)^T \cdot \left(\mathbf{q}^{-1} \cdot \mathbf{v}_r \cdot \mathbf{q} - \mathbf{v}_m\right) = \begin{bmatrix} -2q_z & -2q_w & 2q_x \\ -2q_y & 2q_x & 2q_w \\ -2q_x & -2q_y & -2q_z \\ -2q_w & 2q_z & -2q_y \end{bmatrix} \cdot \begin{bmatrix} -2q_w q_z - 2q_x q_y - v_{mx} \\ -q_w q_w + q_x q_x - q_y q_y + q_z q_z - v_{my} \\ 2q_w q_x - 2q_y q_z - v_{mz} \end{bmatrix} \tag{9}$$

In order to evaluate the success of these modifications, this implementation was used in conjunction with both the original gradient descent algorithm [19], and a recent improved formulation [42]. The three algorithms used to same input data to create a prediction for current orientations. Each algorithm is summarised as follows:

- Original formulation: This algorithm compared the measured vectors to the rotated reference vectors for gravity and the geomagnetic field, and created two functions which could be sequentially minimised using gradient descent. This formulation then took one step in each direction which was approximately towards the function minimum for each.
- Improved formulation: The improved formulation demonstrated a modification to the original formulation which ensured that the step was directly towards the function minima. This resulted in a more predictable rate of convergence.
- The new algorithm generates a new reference vector for the second step, which is perpendicular to both the first and the geomagnetic field. As a result, the two gradient functions are decoupled, which removes interference between the two steps of the algorithm. This leads to higher stability, greater robustness and faster convergence.

2.1.1. Algorithm convergence

Universal convergence of the new algorithm can be proven by demonstrating the gradient exists at all points in the relevant vector space. This is performed by transforming the error function into a frame (l) where v_m is parallel to the X-axis

and the cross product of v_r and v_m is parallel to the Z-axis. At this point, the rotation denoted by q can be described as a rotation of θ around the Z-axis. Since V_m is a unit vector:

$$^{l}V_m = [1, 0, 0] \tag{10}$$

and therefore using θ:

$$^{l}V_r = [cos\theta, sin\theta, 0] \tag{11}$$

The vector describing the error is:

$$^{l}V_e = [cos\theta - 1, sin\theta, 0] \tag{12}$$

Therefore the magnitude of the error is given by:

$$M_e = \sqrt{(cos\theta - 1)^2 + (sin\theta)^2} \tag{13}$$

The value for θ can range from 0 to π, as shown in Fig. 1. From the figure, it is clear the algorithm converges to a single minimum.

2.2. Testing and assessment

2.2.1. Speed of convergence vs existing algorithms

A virtual MARG sensor was used to compare the effect of the extension presented here with both the original gradient descent algorithm and the improved formulation. A random quaternion was generated as a start point for the algorithm. A second was generated to act as a target orientation, and the corresponding vectors for acceleration and magnetic field were generated. The convergence from the initial position to target orientation was then observed under different conditions.

During this period it should be noted that since the accelerometer and magnetometer are now decoupled, if the current quaternion is exactly π from the true orientation, the algorithm does not converge. This is because this point results in the maximum error between the measured and rotated reference vector, therefore the gradient at this point is zero, as shown in Section 2.1.1. This can be compensated for within the algorithm, however for real-world implementations it is not necessary since sensor noise will displace solutions away from this maxima, at which point the algorithm will converge to the minimum. There are several variables in each algorithm that can be used to tune the convergence speed, however for these experiments, the variables were the same for each algorithm. Setting the gyroscope values to zero, the algorithms were then run until all converged to within one degree of the target orientation. This was repeated 1,000 times, and the average times taken to converge was converted to a ratio. An example of the convergence can be seen in Fig. 2. The ratio for Original(Madgwick):Improved(Admiraal):Extension(described here) was found to be approximately 6:7:1.

2.2.2. Robustness to magnetic inclination and disturbance

The effect of the inclination of the magnetic field was also examined. For this experiment, a random quaternion was generated, the simulated magnetic field $V_{m(m)}$ was calculated for different inclinations according to:

$$V_{m(m)} = [cos(inclination), 0, sin(inclination)] \tag{14}$$

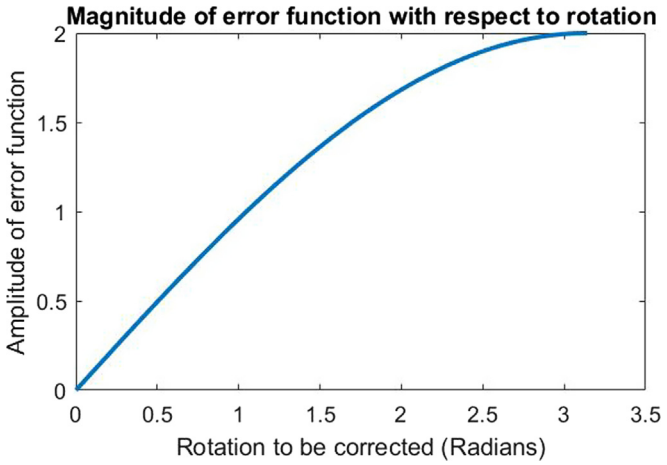


Fig. 1. Algorithm Convergence.

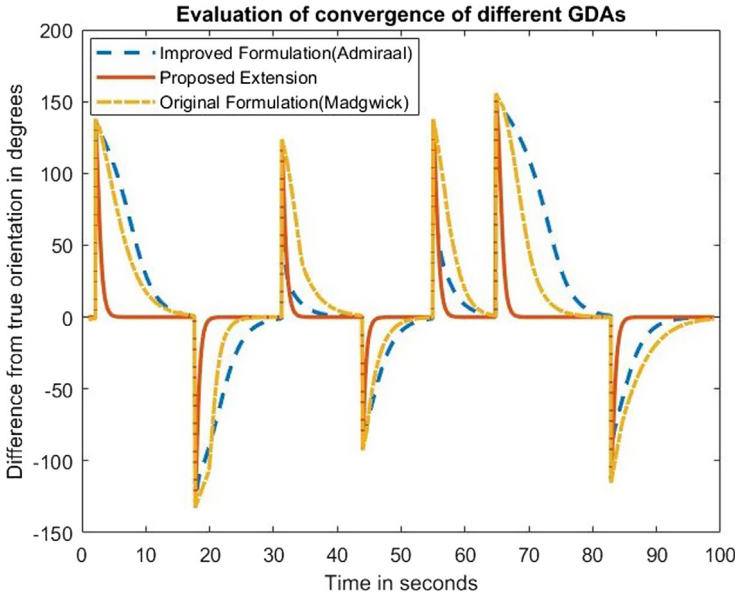


Fig. 2. Convergence of existing GDA algorithms and the new formulation.

and the time taken to converge was recorded. The results from this experiment can be seen in Fig. 3, and demonstrate that the time taken to converge is no longer dependent on the magnetic inclination. This is very important for applications in locations where the magnetic inclination is naturally high, but it also gives the algorithm more robustness in locations where environmental magnetic interference may artificially increase the inclination.

Finally, the random orientation was generated, the vectors were calculated with a fixed inclination, and the algorithm was allowed to converge. The magnetometer vector was then rotated $\pi/2$ around the z-axis, and the convergence was observed in terms of roll, pitch and yaw. The effect of this change of magnetic vector is that the estimation for orientation should rotate $\pi/2$ in yaw, and pitch and roll should remain consistent. When using the algorithm extension outlined in this paper, the convergence moved as expected, demonstrating that the magnetometer calculation is now decoupled from the pitch and roll, addressing the problems raised in [34] with the original gradient descent algorithm. An exemplary instance of this convergence can be seen in Fig. 4.

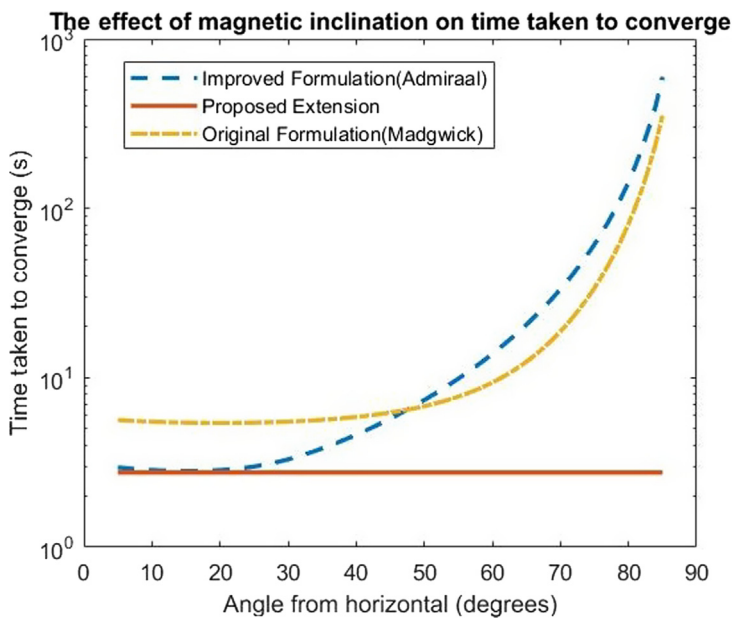


Fig. 3. How inclination affects the convergence time.

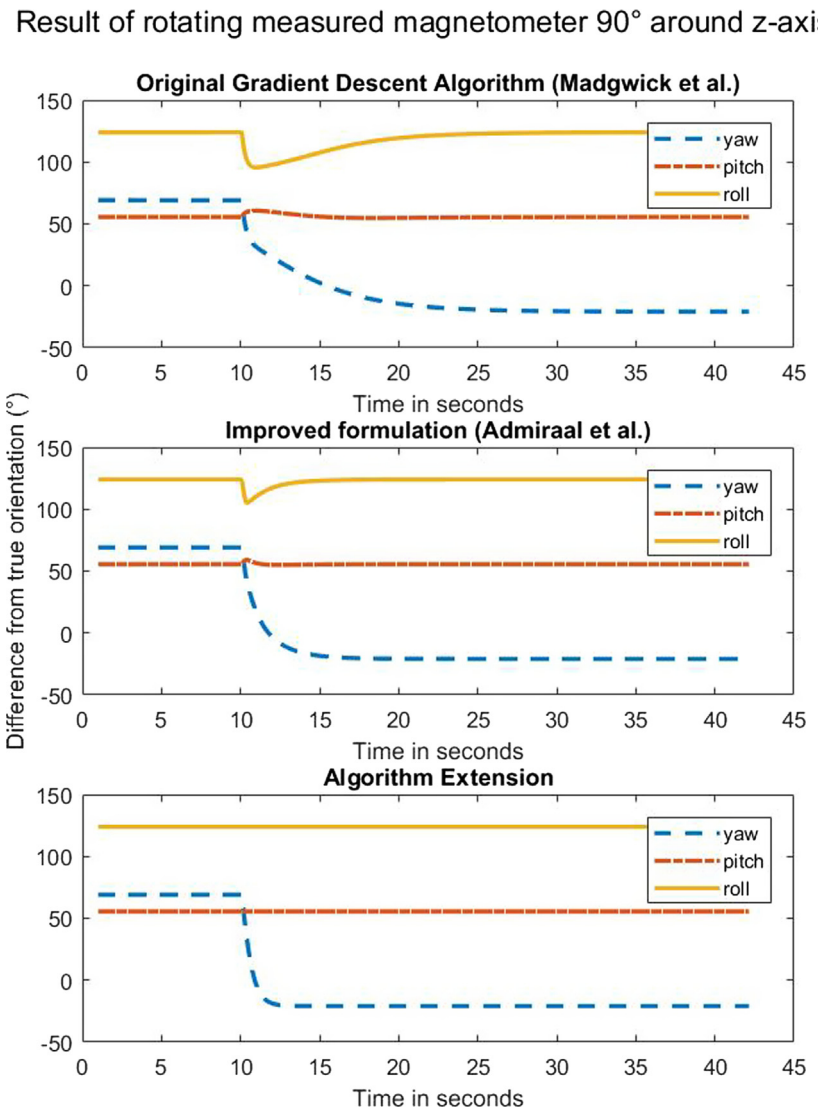


Fig. 4. Effect of changing magnetic field on roll, pitch and yaw.

2.2.3. Robustness to gyro bias

An additional benefit of improving the efficiency of the corrective step is that the algorithm will be able to better compensate for biases introduced by the sensors. Biases in the magnetometer and accelerometer will simply create an offset in the predicted orientation, however since the gyroscopic element of the orientation monitoring uses a finite sums formula, errors in this measurement can have large affects. The virtual IMU was used to observe the algorithms ability to compensate for biases in the gyroscopic sensor data.

To perform this experiment, the IMU was places at a known orientation, and the corresponding accelerometer and magnetometer data were generated. A bias was introduced into one axis of the gyroscopic data, and the algorithms were run until 250,000 steps had been taken. The final thousand steps were analyzed to determine whether the algorithm had converged, and if so what orientation it had converged to. The difference between this orientation and the true orientation of the virtual IMU were recorded. Using this method it was found that the algorithm proposed here improved the range of biases over which the orientation remained stable. This analysis is summarised in Fig. 5.

3. Case study: Robot teleoperation using improved algorithm

We have implemented the algorithm as a part of a human-robot Tele-operative Sensory-Motor Control Interface (TSMCI). The TSMCI consists of bespoke hardware integrating IMUs and physiological (muscle activity) sensors developed as a part of

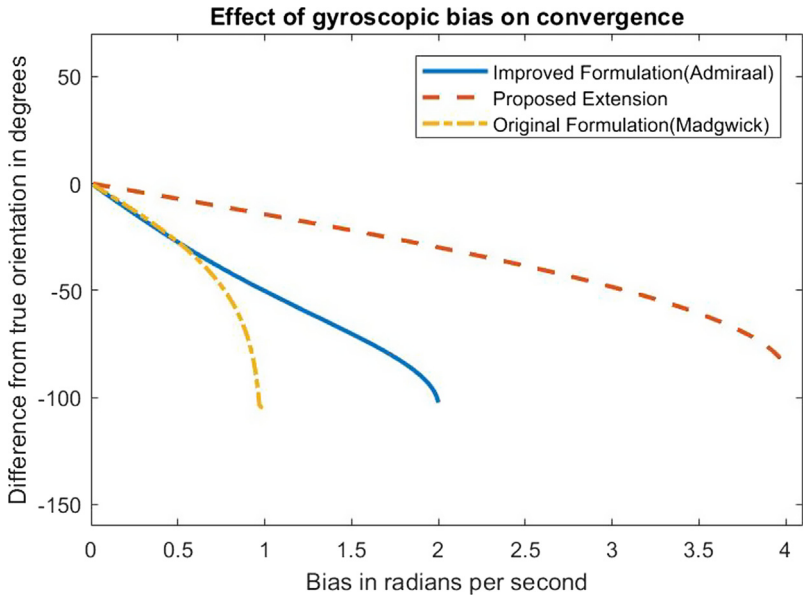


Fig. 5. Analysis of convergence vs gyro bias.

this investigation. Implementation in this application was conducted based on direction from the US Office of Naval Research Global (ONR-G) as basis for demonstration in security and medical robotic arenas. Tests in a virtual environment as well as full physical implementation on a humanoid (Baxter) robot have been executed.

3.1. Background

Sensors, actuators, and onboard intelligence have not yet reached a level where robots may function with complete autonomy; human operation and command is still necessary for utility in unstructured environments, tasks where individual action is undesirable or infeasible, and in situations where a robot must assist and/or interface with people [44]. Efficiency can be increased by providing the operator with an intuitive form of interface, where natural interactions can be captured and used to generate control signals. Hand/Arm gestures provide a successful medium for controlling robotic devices in enumerable applications; recent examples include anthropomorphic robotic arms [45], robotic surgery [46] and control of robots from space [47]. Commonly explored methods to capture hand and arm movement include wearable devices [48,49], exoskeletons [50] and vision-based tracking [45,51]. Wearable sensors can be used in unprepared environments and tend to be self-contained systems, but they can require prior knowledge such as sensor placement, and in some cases are subject to drift. Vision based techniques do not suffer from drift, but are constrained in field of view, suffer from occlusion and need strong ambient light.

3.2. Sensing modalities

There are two categories of data which can be recorded from a pervasive monitoring system: continuous data, such as joint angle, and nominal data, such as the gesture being performed. Recording both allows the user to complete complex tasks such as moving a manipulator into a desired position and triggering a pre-programed action. This is the basis of control for many robotic platforms as it allows the user to control high Degree of Freedom (DoF) robotic platforms. A multi-modal sensor array is often required to achieve this type of control.

Electromyographic sensors (EMGs) placed on the forearm have been used to perform gesture and wrist movement classification [52,53]. Surface EMG has also been used to classify individual finger flexions and extensions [54]. Fusion of EMG and IMU data has been used to control a robotic arm [55,48], as a virtual joystick for control of a robot platform [56,57] and has been shown to be successful within the limitations imposed by the EMG sensors. These limitations include user specificity and issues introduced by transient calibration values, which result in the classification accuracy decreasing over time after calibration [58].

An alternative method of deriving muscular activity is to use Mechanomyography (MMG) sensors. MMG records the mechanical movements induced during muscle contraction, whose component parts include gross lateral movement and volumetric changes of the active motor units [59]. The MMG signal does not require re-calibration and is consistent within un-fatigued muscle. It has been shown to detect both large and small gestures, as defined by [60]. MMG based gesture recognition has been used in prosthetic control applications [61], as well as fused with IMU data for teleoperation of multiple

airborne vehicles [62]. It holds significant potential for actual field use due to ease of use and lack of need for electrical connection to the skin [14].

This case study describes the development and testing of a robot teleoperation control system which uses a combination of MARG and MMG sensors. The MARG sensors output data which is used by the algorithm described in Section 2 to create world referenced quaternions to determine relative arm position. The MMG sensors detect muscular activity and can be used for gesture recognition. This case study was performed using two experimental protocols. First a virtual environment was created to validate the IMUs against the wearers' proprioceptive sense. Second, a task area for a Baxter Robot was constructed, and participants were asked to perform a series of tasks through the robot as fast as possible.

3.3. Experimental hardware

The sensor suite used in this wearable gesture-based interface is made up of two armbands, worn on the upper and lower arm. The upper armband houses one IMU and no MMG sensors, and the lower armband houses one IMU and six MMG sensors. A single example of these armbands has previously been used to provide gesture recognition for upper limb prosthetic control applications [60]. The IMU provides raw measurements for linear acceleration, rotational acceleration and magnetic flux density at up to 500 Hz. This information was then passed through the modified algorithm defined in Section 2 to derive an approximation of the world referenced orientation.

Each MMG sensor consists of a Micro-Electrical-Mechanical (MEMs) micro- phone seated inside a 3D printed case. The case has a conical echo chamber of diameter and height of 7 mm and 5 mm, with the opening covered by a 4 μm Mylar membrane, as described by Posatskiy and Chau [63]. The MMG sensors used in all experiments described here were sampled at a rate of 1 kHz as consistent with past literature in this field [14].

3.4. Hands free control in a virtual environment

Portability and robustness, modality of input, and the need for bodily movement have been reported as major challenges in robot teleoperation [44]. This case study describes the development and testing of a robot teleoperation system using a combination of MARG and MMG sensors designed to address robustness and offer multiple input modalities. To evaluate the IMUs as a sensor for Human-Machine Interface, a virtual test environment was created where individuals were tasked to move from one defined position to another. The experiment used a VR headset (Oculus Rift) to immerse the subjects in test environment, enabling them to observe the position of objects in 3D. This allowed the interface to be evaluated without results being altered by mistakes in depth perception, such as might occur performing 3D tasks on a 2D screen.

A virtual ball ("Test Ball") was placed in the position in virtual space corresponding to the subject's right hand. A second ball ("Target Ball") would then appear in a random position in the subject's field of view, and they were required to move their hand until the Test Ball came into contact with the Target Ball. At this point, the Target Ball would move to a new location. The subjects were instructed to continue to bring them into contact as fast as they could until fifty repetitions had been completed (see Fig. 6).

This exercise was completed twice by each subject. First, each subject held an Oculus Touch Controller in their right hand, which was tracked using two Oculus Sensors. The position of the Test Ball was defined as the position of the Controller. The Oculus Rift tracking system has been shown to be accurate to <1 cm within its working environment [64]. As such, the posi-

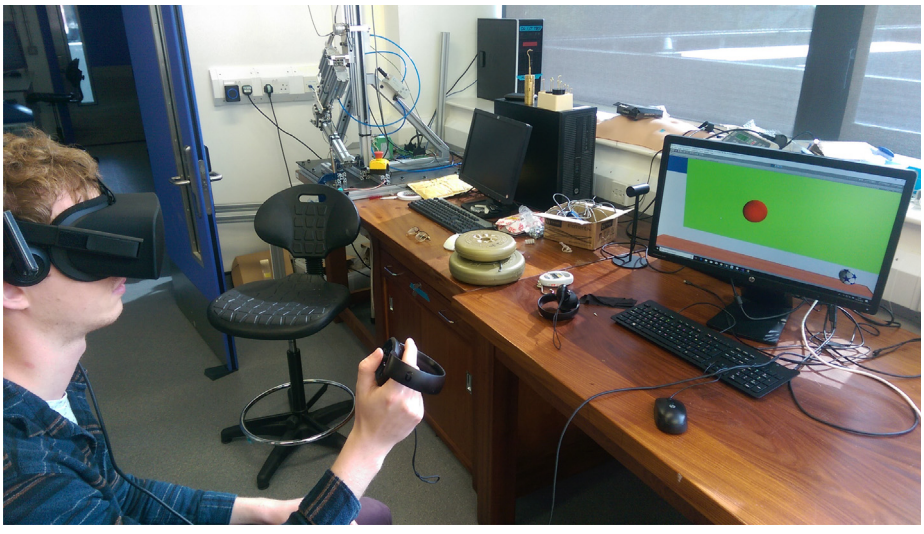


Fig. 6. Using Rift Controller to monitor hand position.

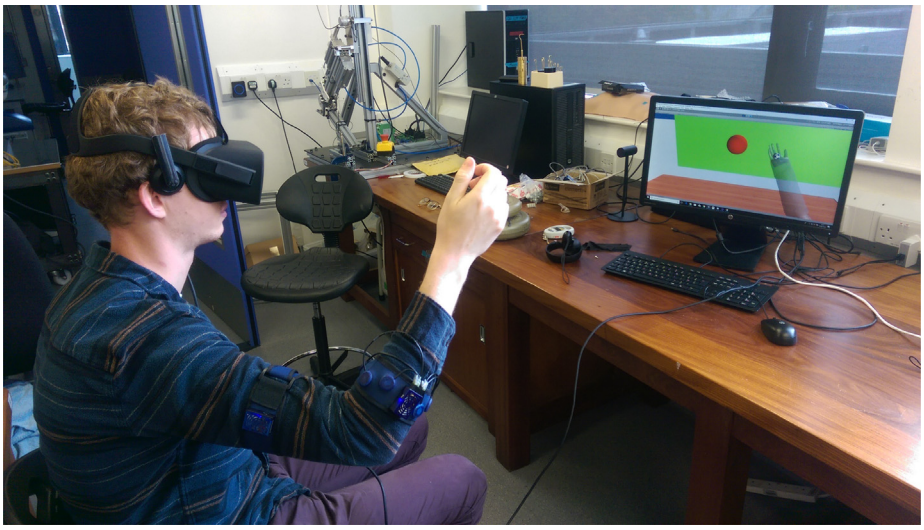


Fig. 7. Using IMUS to derive hand position.

tion given by the Oculus system was assumed to represent the true position. Subjects then repeated test without the controller. Instead, they wore two IMUs, one on the upper arm and one on the lower arm. The subject was instructed to situate their head in a known position in the virtual environment, and this was used to estimate the position of the subject's shoulder (see Fig. 7). The quaternions provided by the new algorithm were then used to calculate the position of the subject's hand using the forward kinematics of the arm. By performing these two parts of the experiment, the difference between the reference motion observed by the Oculus system and the derived position inferred from the algorithm can be quantified.

The efficiency of pointer-based interfaces is often evaluated using Fitts' Law, a model which describes human movement for pointing tasks [65]. Fitts' Law states that the time taken to move from one position to another (MT) is a function of the distance between the positions (A) and the width of the target position (W), according to:

$$MT = a + b\ log_2(2A/W) \tag{15}$$

a and b are constants which are dependent both on the user and the interface. The term $log_2(2A/W)$ describes the task and is referred to as the tasks Index of Difficulty (ID). The average ID for both experiments was equivalent, and the tests were kept short so as to avoid fatigue. As a result of this, the average MT can be said to be inversely proportional to the efficiency of the interface.

A second comparison of the two methods of recording movement was performed by analyzing the path taken between the two targets. A straight line representing the shortest path was constructed between the point at which the Test Ball left the area of the previous Target Ball and the point where it made contact with the new Target Ball. The distance from each point along the actual path to the shortest path was then taken, and defined as the deviation from the path. The subjects path planning was first defined by their proprioceptive sense, and then corrected according to the visual feedback. The deviations when using the controller were defined as the base level for this experiment, and compared to the deviations when using the IMUs.

3.5. Virtual reality results

The average movement time taken to complete a movement from one defined area to another was calculated for five subjects who had little experience using VR systems. On average, subjects took 0.25 s to complete the defined movements with the Oculus Rift controllers, allowing a base time to be established. When using the IMUs to determine hand position, subjects took an average of 0.54 s. From this we can state that using the IMUs as an interface introduced 0.29 s of delay into each movement. The times are summarized in Table 1.

The deviation from a direct path was also recorded. The path lengths were standardized and then an average deviation was calculated. When using the rift controllers, the average deviation was 0.045 m, as opposed to 0.069 m when using the IMUs. The average deviations from a straight path from the point where the hand left the start area to the point where it came into contact with the Target Ball for one user can be seen in Fig. 8.

Table 1
Time taken to complete Validation Experiment.

	Time (s)	
	Controllers	IMU
Average (SD)	0.25 (0.1)	0.54 (0.27)

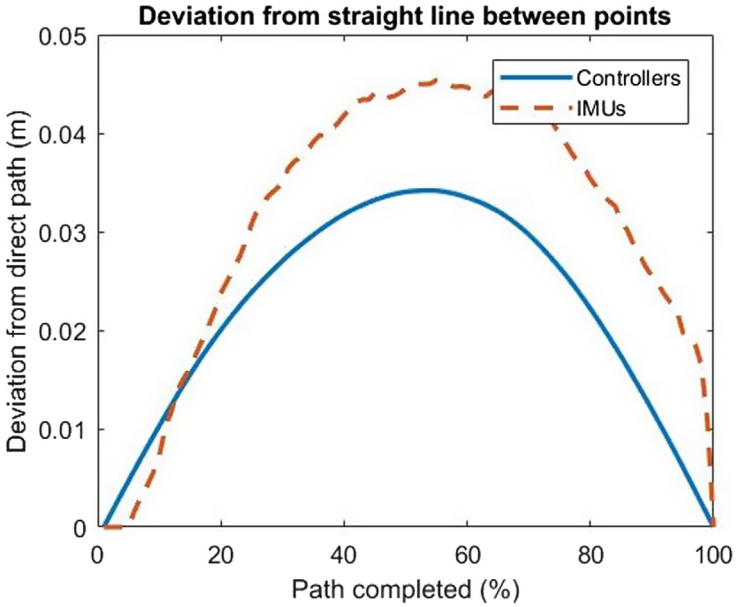


Fig. 8. Average deviation from a direct path.

3.6. Virtual reality discussion

The robot operator was able to complete the tasks both with the Rift Controllers and with the IMUs. While the IMUs did introduce a delay, there are benefits in many situations. The most relevant of these is that they can be using in environments where the Rift controllers would not be suitable. The Rift requires two sensors positioned to observe the test environment, increasing the area required for the system beyond the limits of the area required for the control movement. Since the IMUs are worn, the area required for the control interface is limited to the area required for the control movement, meaning the interface can be significantly more compact. The IMUs also do not require line of sight to a sensor system to work, meaning they can be used in crowded environments. In addition to this, the IMUs provide more information: they allow the entire arm position to be simulated instead of just the hands, giving the user more of an indication of where they are in the virtual environment. The other major benefit of the IMU system is that it leaves the hands of the user free.

In a controlled environment, the IMUs were a less efficient form of interface. While this can be seen in the time taken to complete each task, it can also be seen in the deviation from a straight line. With the rift controllers we observe a smooth path from point to point, whereas the IMUs show three distinct sections, separated by two sharp changes in direction. Initially, the IMUs show a rapid deviation from the straight line, which continues for 0.3 s on average. Since the approximate response time to visual stimuli is 0.25 s, we can hypothesize that this change is due to the misalignment between the predicted position from the IMUs and the subjects predicted position from their own proprioceptive sense. The second phase is the gross movement where the subject has adapted their movement. The final phase is a much slower movement, where the subject has approached the target and is performing the fine corrections needed to reach it. Although the subjects were not able to perceive these repeated correction phases, subjects did report that the IMUs seemed to tire them faster. It can also be seen that throughout the test these phases appeared to become less distinct.

One factor which may have contributed to the difficulty caused by the misalignment of the calculated and user predicted positions is that a generic model was used for the simulated arm. The lengths of the forearm and upper arm were not specified for each individual, and therefore the misalignment was exaggerated.

3.7. Robot teleoperation implementation and testing

3.7.1. Overview

From the validation experiment, it was determined that the IMUs were a suitable input for robot teleoperation, particularly in environments which preclude the use of alternative sensors systems, such as cameras. The IMU system used in the previous experiment was combined with an MMG gesture recognition system previously described in [60] and was expanded to incorporate control signals from both arms. From this, subjects were able to generate 6 analogue control signals corresponding to the 3D Cartesian coordinates from each of their hands, and a discrete control signal by performing gestures with their right hand. These control signals were used to as inputs for the controller of a Baxter Robot.

To demonstrate this system, the positions calculated from the subject's arm orientations were mapped to positions in the Baxter's reference frame, and the Baxter's controller moved the end effectors to these positions at a defined orientation. The discrete control signals were used to direct other behaviour in the robot, such as operating the gripper, but were also used to switch the controller into an autonomous mode to perform actions which required a greater level of accuracy or complexity.

3.7.2. Experimental protocol

Subjects wore one IMU on each upper arm and each forearm for arm motion tracking, and six MMGs on the right forearm for grasp. The MMGs were worn around the circumference of the arm, split evenly between the digit flexor and extensor muscles. The placement of the sensors can be seen in Figs. 9 and 10. Before the experiment began, the users made 30 examples of each gesture used in the experiment. These gestures were used as the training set for the classification algorithms. The experimental protocol was presented to the participants in the form of a game. The objective was to use the robot to diffuse a 'bomb' by performing several tasks on a large circuit board in the correct order and as fast as possible. The decision to present the protocol in this format was taken for three reasons: First, the 'storyline' within the experiment was designed to keep the user engaged; Second, to emphasize that the speed of their progress was dependent on the skill that they showed; Third, because the specific scenario introduced a level of urgency which ensured that the subjects were performing the timed tasks as fast as possible.

The scenario also required the users to use several different 'tools', which were selected by performing different gestures. When a gesture is recognized, the robot performed predefined actions, which include picking up the 'tools' from the tool rack in front of it before returning to the position defined by the user's arms. The overall time to complete the experiment provides an evaluation metric of the interface. A summary of the control loop is shown in Fig. 11. To provide a base completion time, a Microsoft xBox controller was also used to perform the experiment, with the completion time with the controller giving an indication of the difficulty of the protocol.

Two experimental protocols were used to complete the task. The first required the simultaneous control of both arms, and used three gestures to manipulate the gripper, and to pick up and replace tools from the tool rack in autonomous mode. The second protocol required gross movements to be recorded from the left arm alone, with the right providing control of the gripper and triggering one of four autonomous behaviours through the recognition of six distinct gestures. The protocol for these autonomous movements and for the MMG based gesture recognition is described in the following section.

3.7.3. Outline of semi-autonomous control system

Although on operator should have as much control of the robot as possible, certain precise tasks are difficult and inefficient to complete over teleoperation. While the protocol of the task is unknown, the position of the 'tools' in the 'tool rack' is known ahead of time. To create an efficient interface, picking up and replacing the 'tools' was therefore pre-programmed as an autonomous action, with the user choosing when to do this, and what to do with them once they had been picked up. In addition, the nature of the experiment meant that the orientation of the end effector was also kept perpendicular to the board.

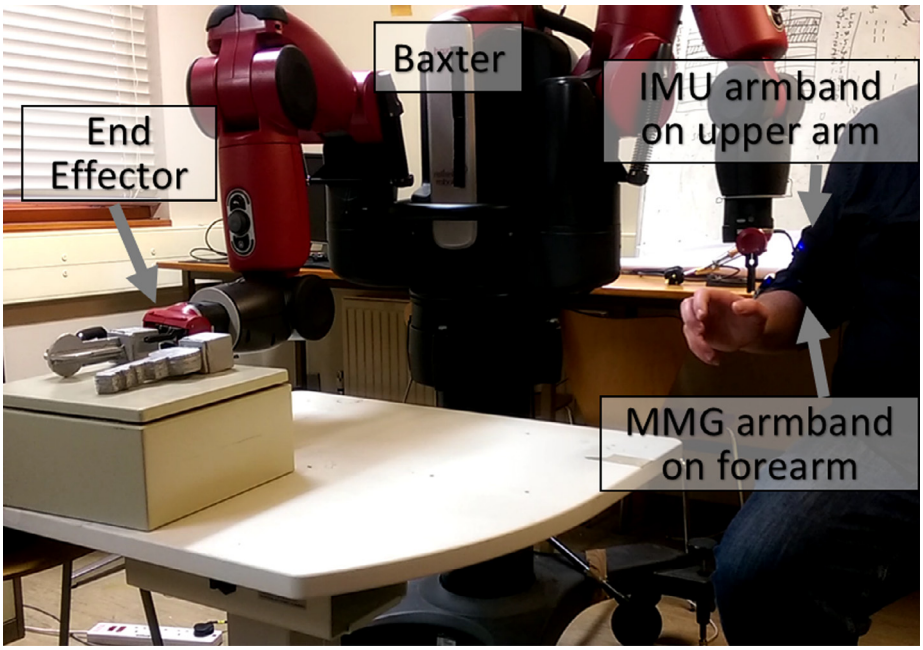


Fig. 9. Experimental setup from front.

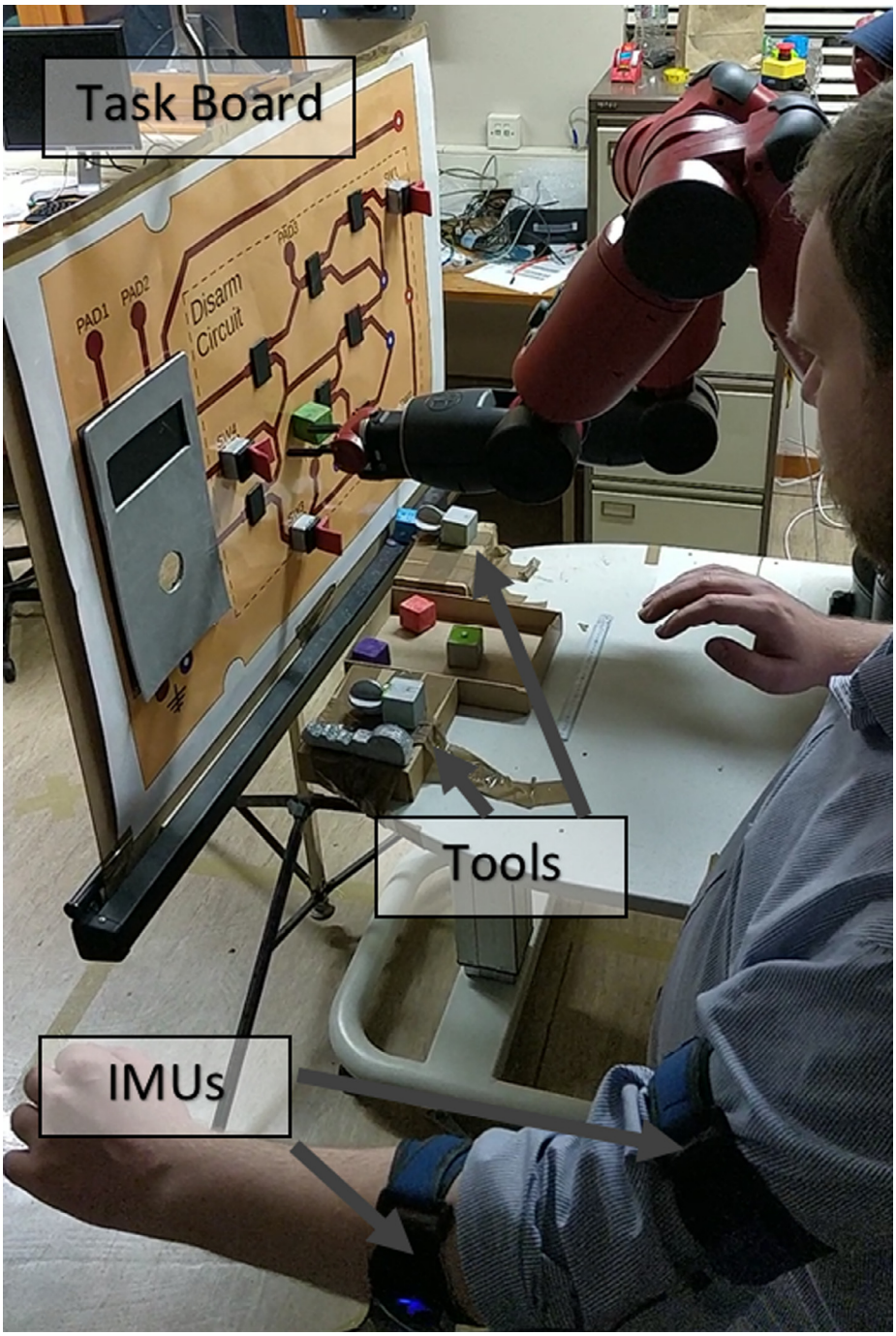


Fig. 10. Experimental setup from side.

Pre-programmed motions were executed by using Baxter's internal position controller, which allowed the robot to converge on a specific joint pose while respecting velocity and torque constraints that are intended to ensure user safety. The robot moved the required end effector to a position directly above the stored location of the desired tool before lowering, closing the gripper around it, and retracting to the prior raised position. If the robot was already holding a tool, this process was first reversed to place it in its correct storage location before selecting a new one.

Direct user control was achieved by a positional control algorithm based on the difference between the spherical coordinates of the human wrist and the robot end effector. This was obtained by first removing the offset from the origin of the human's shoulders and the robot's base joints and adding a displacement of 0.8 m and 0.5 m to the y and z axes of

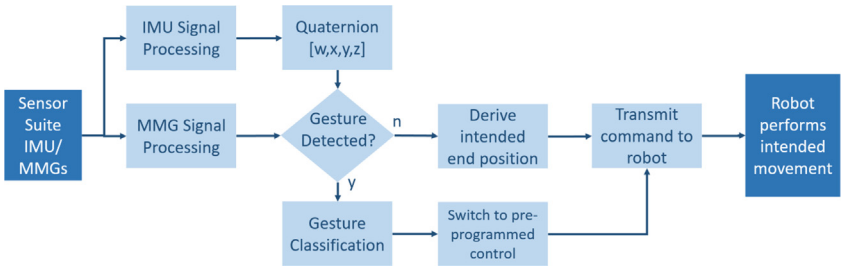


Fig. 11. Outline of Teleoperation control loop.

the hand coordinates in order to account for Baxter's inability to reach very close to, or below its base. The transformation was then used to convert both sets of coordinates into radius (r), polar angle (θ) and azimuthal angle (ϕ). The arctangent is calculated using numpy's atan2 function in order to ensure that the azimuthal (ϕ) coordinate remains continuous:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arccos z/r \\ \phi &= \arctan y/x \end{aligned} \tag{16}$$

The spherical error term is used to control only the first four joints (counted from the base) in the robot's arm, which ensures that the response to the user's movements will remain predictable and consistent. The orientation of the end effector is held constant (perpendicular to the work surface) by an additional feedback term which controls the three terminal joints of the arm. The sum of the two terms gives the full control signal for the robot:

$$u(t) = g_p M_p\big(h_p(t) - r_p(t)\big) + g_o M_o(O - r_o(t))M_p = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{17}$$

where g_o and g_p are position gain terms, M_p is a static matrix determining the relationship between the position of the end effector and the robot's joints, M_o is a static matrix determining the relationship between the orientation of the end effector and the robot's joints, h_p and r_p are the vectors of the human and robot's wrist/end effector positions and r_o is the orientation of the robot's end effector. O is a constant vector representing a fixed orientation perpendicular to the work surface.

The control signal $u(t)$ is added to the current joint positions and passed to the Baxter's position controller, treating it as a delta term. Because the control signal is updated multiple times per second, the robot's movement approximates continuous tracking of the user's hand while remaining strictly within Baxter's velocity and torque restrictions.

This protocol for semi-autonomous control system provided control 14 joints within the robot. On each arm, three joints were directly controlled by the movements of the user, and three others were used to maintain the orientation of the end effector at all times. The one DoF end effector was also controlled by the user.

3.7.4. MMG based gesture recognition

Arm position was derived inertially, however muscle activity for hand grasp was necessary for simultaneous hand and arm telemanipulation. A novel template-based approach was implemented for real-time classification of grasp intent using MMG signals. The method has two distinct phases, Gesture Segmentation and Gesture Recognition. In this implementation, a user creates signal templates by making 30 instances of each of the gestures used in the experiment before it started. Each of these gestures was segmented, and labelled based on the gesture the subject had been instructed to make.

To perform segmentation, the data for each MMG was band-pass filtered between 1 Hz and 50 Hz as captured in real-time. At each sample, this filtered data from each of the N MMGs was compiled into a vector m^*. The power of the signal p was then defined as:

$$p = \sum_{j=1}^{N} m_j^{*2} \tag{18}$$

Each value for p was then compared to a threshold T_p, and when $p > T_p$, this indicated that a gesture may have been made. Since MMGs detect low frequency vibrations, the signal power can also increase during motion. To ensure that this methods of segmentation excluded motion induced artefacts, the movement of the sensor suite was also monitored. Motions were

accompanied by an increase in rotational acceleration, and therefore a second threshold T_g using the energy of the gyroscopic data g was used to reject instances where increased power in the MMG signals were likely due to movements. A gesture therefore had to satisfy two conditions simultaneously: $p > T_p$ and $g < T_g$. The time this occurs is noted as time i. Finally, to ensure one value of i per gesture, a new value of i must be the equivalent of 2 s after the previous one.

Offline analysis demonstrated the first 0.2 s of a gesture is sufficient for classification. It also demonstrated that the gesture begins approximately 0.05 s before the conditions are met. As a result, the previous 0.05 s of data is stored, and a new value of i triggers the recording of an additional 0.15 s of data which is then appended to create a new matrix S containing the segmented gesture.

To create the templates (S), each channel in 30 training instances for each gesture were averaged. This created one template gesture where common features were retained, but random or inconsistent features were removed. An alternative to this approach would be to consider each data point as a feature, and perform a ranking based on minimising intra-class variance while maximising interclass variance. This has been shown to be a successful method of dimensionality reduction which could reduce the computational expense for real-time implementations [66].

During the experimental phase, classification was performed by correlating each channel of a new segmented gesture against the corresponding channel for each of the templates using the Pearson Product-Moment Correlation Coefficient ρ. This is calculated using the standard deviation σ of each vector in the data to be classified and the template matrix, as well as the average of both ($\overline{s_n}$) and ($\overline{S_n}$) respectively). For a system which uses N MMGs and records b samples during a gesture, the correlation coefficient can be summarised as:

$$\rho_{sS} = \frac{1}{N}\sum_{n=1}^{N}\frac{\sum_{j=1}^{b}\left[s_{n_j} - \overline{s_n}\right]\left[S_{n_j} - \overline{S_n}\right]}{\sigma_{s_n}\sigma_{S_n}} \tag{19}$$

The label of the template with the highest overall Correlation Coefficient was presented as the systems prediction for the class of the new gesture, and triggered the autonomous functionality of the robot.

The gestures used are summarised in Table 2:

3.8. Robot teleoperation – Results

Subjects were first timed performing the experiments using the xBox controller to establish a measure of the difficulty of the task using a known industry-standard interface. They then completed both tasks using the IMU/MMG sensor suite, and the times taken appeared proportional to the derived difficulty of the task. The time taken using the IMU/MMG system was longer when compared to the Xbox, but was completed without specific manipulation of the fingers.

One possible reason for this increase time is the large number of gesture misclassifications which one subject recorded during this protocol, each of which will have increased the overall experimental time. The average times taken to complete each of the protocols can be seen in Table 3. A simple robot experiment with the hardware and algorithm implemented is posted at:https://youtu.be/ncm49Z1CDko.

Table 2
Gestures used for teleoperation experiments.

Hand Gesture	Robot Behaviour	Protocol
Open hand	Open robot gripper	1&2
Close hand	Close robot gripper	1&2
Point index finger	Retrieve probe from tool rack	1&2
Pinch	Retrieve key from tool rack	2
Rotate hand anti-clockwise	Rotate robot gripper anti-clockwise	2
Rotate hand clockwise	Rotate robot gripper clockwise	2

Table 3
Time taken to complete teleoperation demonstration.

Subject	Time (m:s)			
	Protocol 1		Protocol 2	
	Controllers	IMU/MMG	Controllers	IMU/MMG
Average	5:02	6:40	3:55	7:02

3.9. Robot teleoperation – Discussion

The IMUs functioned well, allowing subjects to finish experiments quickly and efficiently. The use of a generic model for the forward kinematics of the arm was not reported as problematic by any subjects in this application. This reinforces the efficacy of the teleoperation system, as users were not focused on the position of their own arms, but were making adjustments according to their desired robotic movements. In this application, the user's visual feedback and proprioceptive sense were not providing conflicting information, making the system easier to use.

The IMU system also allowed the robot to make large movements much faster than the controllers were able to. Smaller adjustments were difficult to make with both forms of control. One explanation could be the compliance built into the robot's joint, which resulted in very fine movements being ignored.

To remove the issues introduced in providing visual feedback, the user was able to observe the movement of the robot from the same room as the robot. With more advanced visual feedback methods, the work demonstrated here could form part of the user interface for a complete telepresence control system.

4. Conclusion

In this paper we have formulated, tested, and implemented a new sensor fusion algorithm taking in MARG data and outputting full orientation. When dealing with 9DoF IMUs, the sensor most commonly susceptible to interference from the environment is the magnetometer. This interference can change the direction of the field as well as significantly impact the world-referenced inclination. Implementation of the new algorithm demonstrates inclination no longer affects convergence, meaning that it will reliably converge regardless of the magnetic interference in the environment. The algorithm benefits from significant computational efficiency as well, demonstrating a factor of 7 improvement in convergence time over the other gradient descent formulations. This is critical for embedded systems and real-time accuracy.

The algorithm has been implemented in a human-machine interface (HMI) case study for robotic manipulator teleoperation. We have validated the algorithm in this context using a virtual environment, where subjects were tasked to move their hands to specified positions as fast as possible. Through this, we demonstrated that humans are able to direct robot action mimicking their limb with a delay of less than 0.3s compared to camera-based systems, even when forward kinematics provided through the IMUs did not match the proprioceptive sense of hand position. This enables control of manipulators that do not exactly match the degrees of freedom for a human arm to be directed in a manner transparent to the operator.

Finally, we have developed bespoke hardware enabling the fusion of the inertial sensor system with a novel physiological sensor we have introduced for hand grasping. The entire system has been implemented in hardware for teleoperation of a 14 DoF Baxter robot to perform complex manipulation tasks. The interface fuses seamlessly with the semi-autonomous robot control system, via a server located outside the local network. Individuals have transparently controlled the robot using their arms and gestures as if it was their natural arm, even though its configuration did not match the user's arm. Experiments were executed with the user was in the room with the robot, however if combined with existing immersive visual feedback, the system can be used for telepresence. Results are comprable to industry controllers requiring additional equipment such as cameras that are dependent on preprepared environments.

Removal of the need for cameras for kinematic monitoring-based HMIs divorces geographic constraints from the interface. With this system, the user is no longer concerned with remaining in the field of view of the interface, is not dependent on ambient light and has the freedom to move around their environment, and interact with their devices in a natural way. While the response is slower than industrial standard controllers, this system does not require a prepared environment, and does not require the user to hold any hardware. This is a critical enabling factor for use in the field which has severly limited vision-based teleoperation.

The algorithm introduced in this paper has applications far beyond those described here. Within the field of pervasive monitoring, accurate limb orientation is useful for observing body kinematics both in rehabilitation, sporting applications and in the gaming industry. Accurate orientation is also useful for autonomous robot navigation and drone applications. Additionally, it is of critical important in Virtual Reality (VR) applications to track full orientation of the head, and systems running this algorithm can be used to bring more of the body into VR spaces. The improved formulation offers significant improvement in computational expense compared to Kalman filtering approaches and greater accuracy and robustness compared to gradient descent and complementary filter alternatives. We believe the algorithm holds the potential to impact a very wide range of inertial sensor fusion applications. Commercial translation is being pursued in control of artificial limbs, rehabilitation/physiotherapy, sports/athletic performance and human-robot interface [67].

Acknowledgments

This work was supported by the US Office of Naval Research Global (N62909-14-1-N221-P00002), the UK EPSRC Centre for Doctoral Training in Neurotechnology, the UK-India Educational Research Initiative (UKIERI-DST-2014-15-08), the Royal British Legion Centre for Blast Injury Studies at Imperial College London, the UK Medtech Superconnector fund and x-io Technologies Limited.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.ymssp.2019.04.064.

References

[1] A. Kallapur, I. Petersen, S. Anavatti, A robust gyroless attitude estimation scheme for a small fixed-wing unmanned aerial vehicle, in: 2009 7th Asian Control Conference, 2009, pp. 666–671..
[2] S. Wang, H. Wen, R. Clark, N. Trigoni, Keyframe based large-scale indoor localisation using geomagnetic field and motion pattern, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 1910–1917. doi:10.1109/IROS.2016.7759302..
[3] S.M. LaValle, A. Yershova, M. Katsev, M. Antonov, Head tracking for the oculus rift, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 187–194. doi:10.1109/ICRA.2014.6906608.
[4] T. Liu, C. Fan, Visible-light wearable eye gaze tracking by gradients-based eye center location and head movement compensation with imu, in: 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018, pp. 1–2. doi:10.1109/ICCE.2018.8326076..
[5] L. Zhao, D. Guan, R. Jr Landry, J. Cheng, K. Sydorenko, et al, An accurate and fault-tolerant target positioning system for buildings using laser rangefinders and low-cost mems-based marg sensors, Sensors 15 (10) (2015) 27060–27086.
[6] Y. Liu, X. Fan, C. Lv, J. Wu, L. Li, D. Ding, An innovative information fusion method with adaptive kalman filter for integrated ins/gps navigation of autonomous vehicles, Mech. Syst. Signal Processing 100 (2018) 605–616, http://www.sciencedirect.com/science/article/pii/S0888327017304211.
[7] B. Barshan, H.F. Durrant-Whyte, Inertial navigation systems for mobile robots, IEEE Trans. Rob. Autom. 11 (3) (1995) 328–342, https://doi.org/10.1109/70.388775.
[8] T. Fauser, S. Bruder, A. El-Osery, A comparison of inertial-based navigation algorithms for a low-cost indoor mobile robot, in: 2017 12th International Conference on Computer Science and Education (ICCSE), 2017, pp. 101–106, https://doi.org/10.1109/ICCSE.2017.8085471.
[9] kos Odry, R. Fullr, I.J. Rudas, P. Odry, Kalman filter for mobile-robot attitude estimation: Novel optimized and adaptive solutions, Mech. Syst. Signal Processing 110 (2018) 569–589, http://www.sciencedirect.com/science/article/pii/S088832701830181X.
[10] C. Faria, Sensor fusion and rotational motion reconstruction via nonlinear state-observers, Mech. Syst. Signal Processing 114 (2019) 571–578, http://www.sciencedirect.com/science/article/pii/S0888327018302747.
[11] H. Zhou, H. Hu, Human motion tracking for rehabilitation-a survey, Biomed. Signal Process. Control 3 (1) (2008) 1–18, https://doi.org/10.1016/j.bspc.2007.09.001, http://www.sciencedirect.com/science/article/pii/S1746809407000778.
[12] R. Abbasi-Kesbi, A. Nikfarjam, H. Memarzadeh-Tehran, A patient-centric sensory system for in-home rehabilitation, IEEE Sens. J. 17 (2) (2017) 524–533, https://doi.org/10.1109/JSEN.2016.2631464.
[13] P. Angeles, Y. Tai, N. Pavese, S. Wilson, R. Vaidyanathan, Automated assessment of symptom severity changes during deep brain stimulation (dbs) therapy for parkinson's disease, in: 2017 International Conference on Rehabilitation Robotics (ICORR), 2017, pp. 1512–1517. doi:10.1109/ICORR.2017.8009462..
[14] R.B. Woodward, S.J. Shefelbine, R. Vaidyanathan, Pervasive monitoring of motion and muscle activation: Inertial and mechanomyography fusion, IEEE/ASME Trans. Mechatron. 22 (5) (2017) 2022–2033, https://doi.org/10.1109/TMECH.2017.2715163.
[15] M. Masters, L. Osborn, N. Thakor, A. Soares, Real-time arm tracking for hmi applications, in: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2015, pp. 1–4. doi:10.1109/BSN.2015.7299391..
[16] E.A. Heinz, K.S. Kunze, M. Gruber, D. Bannach, P. Lukowicz, Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment, in: 2006 IEEE Symposium on Computational Intelligence and Games, 2006, pp. 98–102. doi:10.1109/CIG.2006.311687..
[17] J.H. Burridge, A.C.W. Lee, R. Turk, M. Stokes, J. Whitall, R. Vaidyanathan, P. Clatworthy, A.-M. Hughes, C. Meagher, E. Franco, et al, Telehealth, wearable sensors, and the internet: will they improve stroke outcomes through increased intensity of therapy, motivation, and adherence to rehabilitation programs?, J Neurologic Phys. Therapy 41 (2017) S32–S38.
[18] J.E. Bortz, A new mathematical formulation for strapdown inertial navigation, IEEE Trans. Aerosp. Electron. Syst. AES-7 (1) (1971) 61–66, https://doi.org/10.1109/TAES.1971.310252.
[19] S.O.H. Madgwick, A.J.L. Harrison, R. Vaidyanathan, Estimation of imu and marg orientation using a gradient descent algorithm, in: 2011 IEEE International Conference on Rehabilitation Robotics, 2011, pp. 1–7. doi:10.1109/ICORR.2011.5975346..
[20] R. Mahony, T. Hamel, J. Pflimlin, Complementary filter design on the special orthogonal group so(3), in: Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 1477–1484. doi:10.1109/CDC.2005.1582367..
[21] S. Yean, B.S. Lee, C.K. Yeo, C.H. Vun, H.L. Oh, Smartphone orientation estimation algorithm combining kalman filter with gradient descent, IEEE J. Biomed. Health Inform. 22 (5) (2018) 1421–1433.
[22] T.S. Yoo, S.K. Hong, H.M. Yoon, S. Park, Gain-scheduled complementary filter design for a mems based attitude and heading reference system, Sensors 11 (4) (2011) 3816–3830, https://doi.org/10.3390/s110403816. http://www.mdpi.com/1424-8220/11/4/3816.
[23] M. Nowicki, J. Wietrzykowski, P. Skrzypczyski, Simplicity or flexibility? complementary filter vs. ekf for orientation estimation on mobile devices, in: 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), 2015, pp. 166–171. doi:10.1109/CYBConf. 2015.7175926..
[24] R. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME – J. Basic Eng. 82 (1960) 35–45.
[25] A.M. Sabatini, Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing, IEEE Trans. Biomed. Eng. 53 (7) (2006) 1346–1356, https://doi.org/10.1109/TBME.2006.875664.
[26] J.L. Marins, X. Yun, E.R. Bachmann, R.B. McGhee, M.J. Zyda, An extended kalman filter for quaternion-based orientation estimation using marg sensors, in: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No.01CH37180), Vol. 4, 2001, pp. 2003–2011 vol 4. doi:10.1109/IROS.2001.976367..
[27] M. Filiashkin, M. Novik, Combined complementary filter for inertial navigation system, in: 2012 2nd International Conference Methods and Systems of Navigation and Motion Control (MSNMC), 2012, pp. 59–62. doi:10.1109/MSNMC.2012.6475087..
[28] Mti user manual, Xsens Technologies B.V. Pantheon 6a The Netherlands..
[29] 3dm-gx5-35 attitude and heading reference system (ahrs) with gnss, MicroStrain Sensing Systems Williston VT United States of America..
[30] Vn-100 user manual, VectorNav Technologies Dallas TX USA..
[31] Product manual for use with inertiacube4 serial and usb interfaces, InterSense Billerica MA..
[32] User manual trax, PNI Sensor Corporation Santa Rosa CA USA..
[33] S. Madgwick, x-io technologies. http://x-io.co.uk/x-imu/..
[34] B. Fan, Q. Li, T. Liu, How magnetic disturbance influences the attitude and heading in magnetic and inertial sensor-based orientation estimation, Sensors 18 (1) (2017), https://doi.org/10.3390/s18010076, http://www.mdpi.com/1424-8220/18/1/76.
[35] R.G. Valenti, I. Dryanovski, J. Xiao, Keeping a good attitude: a quaternion-based orientation filter for imus and margs, Sensors 15 (8) (2015) 19302–19330, https://doi.org/10.3390/s150819302, http://www.mdpi.com/1424-8220/15/8/19302.
[36] J. Wu, Z. Zhou, J. Chen, H. Fourati, R. Li, Fast complementary filter for attitude estimation using low-cost marg sensors, IEEE Sens. J. 16 (18) (2016) 6997–7007, https://doi.org/10.1109/JSEN.2016.2589660.
[37] S. Madgwick, An efficient orientation filter for inertial and inertial/magnetic sensor arrays, Report x-io and University of Bristol (UK) 25 (2010) 113–118.

[38] W.T. Higgins, A comparison of complementary and kalman filtering, IEEE Trans. Aerosp. Electron. Syst. AES-11 (3) (1975) 321–325, https://doi.org/10.1109/TAES.1975.308081.
[39] A. Cavallo, A. Cirillo, P. Cirillo, G.D. Maria, P. Falco, C. Natale, S. Pirozzi, Experimental comparison of sensor fusion algorithms for attitude estimation, IFAC Proceedings Volumes 47 (3) (2014) 7585–7591, 19th IFAC World Congress. doi: 10.3182/20140824-6-ZA-1003.01173. http://www.sciencedirect.com/science/article/pii/S1474667016428089..
[40] Q. Mourcou, A. Fleury, C. Franco, F. Klopcic, N. Vuillerme, Performance evaluation of smartphone inertial sensors measurement for range of motion, Sensors 15 (9) (2015) 23168–23187, https://doi.org/10.3390/s150923168. http://www.mdpi.com/1424-8220/15/9/23168.
[41] C. Yi, J. Ma, H. Guo, J. Han, H. Gao, F. Jiang, C. Yang, Estimating three-dimensional body orientation based on an improved complementary filter for human motion tracking, Sensors 18 (11) (2018), https://doi.org/10.3390/s18113765, http://www.mdpi.com/1424-8220/18/11/3765.
[42] M. Admiraal, S. Wilson, R. Vaidyanathan, Improved formulation of the imu and marg orientation gradient descent algorithm for motion tracking in human-machine interfaces, in: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2017, pp. 403–410. doi:10.1109/MFI.2017.8170354..
[43] J. Diebel, Representing attitude: Euler angles, unit quaternions, and rotation vectors, Matrix 58 (15–16) (2006) 1–35.
[44] R. Vaidyanathan, M. Fargues, D. Lin, S. Kota, L. Gupta, R. Quinn, A dual mode human-robot interface based on physiological signal capture within aural cavity, Int. J. Robotics Res. 26 (11–12) (2007) 1205–1223.
[45] J. Grasshoff, L. Hansen, I. Kuhlemann, K. Ehlers, 7dof hand and arm tracking for teleoperation of anthropomorphic robots, in: Proceedings of ISR 2016: 47st International Symposium on Robotics, 2016, pp. 1–8.
[46] M.L. Lorentziadis, A short history of the invasion of robots in surgery, Hellenic J. Surgery 86 (3) (2014) 117–121, https://doi.org/10.1007/s13126-014-0112-2.
[47] A. Schiele, G. Visentin, The esa human arm exoskeleton for space robotics telepresence, 2003..
[48] O. Fukuda, T. Tsuji, M. Kaneko, A. Otsuka, A human-assisting manipulator teleoperated by emg signals and arm motions, IEEE Trans. Robotics Automation 19 (2) (2003) 210–222, https://doi.org/10.1109/TRA.2003.808873.
[49] E.R. Bachmann, R.B. McGhee, X. Yun, M.J. Zyda, Inertial and magnetic posture tracking for inserting humans into networked virtual environments, in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '01, ACM, New York, NY, USA, 2001, pp. 9–16. doi:10.1145/505008.505011. http://doi.acm.org/10.1145/505008.505011..
[50] S. Chang, J. Kim, I. Kim, J.H. Borm, C. Lee, J.O. Park, Kist teleoperation system for humanoid robot, in: Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289), Vol. 2, 1999, pp. 1198–1203 vol 2. doi:10.1109/IROS.1999.812842..
[51] H. Reddivari, C. Yang, Z. Ju, P. Liang, Z. Li, B. Xu, Teleoperation control of baxter robot using body motion tracking, in: 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014, pp. 1–6. doi:10.1109/MFI.2014.6997722..
[52] A. Boyali, N. Hashimoto, O. Matsumoto, Hand posture and gesture recognition using myo armband and spectral collaborative representation based classification, in: 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), 2015, pp. 200–201. doi:10.1109/GCCE.2015.7398619..
[53] S.N. Sidek, A.J.H. Mohideen, Measurement system to study the relationship between forearm emg signals and wrist position at varied hand grip force, in: 2012 International Conference on Biomedical Engineering (ICoBE), 2012, pp. 169–174. doi:10.1109/ICoBE.2012.6178999..
[54] A.H. Al-Timemy, G. Bugmann, J. Escudero, N. Outram, Classification of finger movements for the dexterous hand prosthesis control with surface electromyography, IEEE J. Biomed. Health Inform. 17 (3) (2013) 608–618, https://doi.org/10.1109/JBHI.2013.2249590.
[55] C. Yang, S. Chang, P. Liang, Z. Li, C. Su, Teleoperated robot writing using emg signals, in: 2015 IEEE International Conference on Information and Automation, 2015, pp. 2264–2269. doi:10.1109/ICInfA.2015.7279663..
[56] M.T. Wolf, C. Assad, M.T. Vernacchia, J. Fromm, H.L. Jethani, Gesture-based robot control with variable autonomy from the jpl biosleeve, in: 2013 IEEE International Conference on Robotics and Automation, 2013, pp. 1160–1165. doi:10.1109/ICRA.2013.6630718..
[57] G. Luh, H. Lin, Y. Ma, C.J. Yen, Intuitive muscle-gesture based robot navigation control using wearable gesture armband, in: 2015 International Conference on Machine Learning and Cybernetics (ICMLC), Vol. 1, 2015, pp. 389–395. doi:10.1109/ICMLC.2015.7340953..
[58] K. Trker, Electromyography: Some methodological problems and issues, Phys. Therapy 73 (1993) 698–710, https://doi.org/10.1093/ptj/73.10.698.
[59] C. Orizio, Muscle sound: Bases for the introduction of a mechanomyographic signal in muscle studies, Critical Rev. Biomed. Eng. 21 (1993) 201–243.
[60] S. Wilson, R. Vaidyanathan, Upper-limb prosthetic control using wearable multichannel mechanomyography, in: 2017 International Conference on Rehabilitation Robotics (ICORR), 2017, pp. 1293–1298. doi:10.1109/ICORR.2017.8009427..
[61] S. Wilson, R. Vaidyanathan, Gesture recognition through classification of acoustic muscle sensing for prosthetic control, in: M. Mangan, M. Cutkosky, A. Mura, P.F. Verschure, T. Prescott, N. Lepora (Eds.), Biomimetic and Biohybrid Systems, Springer International Publishing, Cham, 2017, pp. 637–642.
[62] Y. Ma, Y. Liu, R. Jin, X. Yuan, R. Sekha, S. Wilson, R. Vaidyanathan, Hand gesture recognition with convolutional neural networks for the multimodal uav control, in: 2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017, pp. 198–203. doi:10.1109/RED-UAS.2017.8101666..
[63] A. Posatskiy, T. Chau, Design and evaluation of a novel microphone-based mechanomyography sensor with cylindrical and conical acoustic chambers, Med. Eng. Phys. 34 (2012) 1184–1190, https://doi.org/10.1016/j.medengphy.2011.12.007.
[64] A. Borrego, J. Latorre, M. Alcaiz, R. Llorens, Comparison of oculus rift and htc vive: Feasibility for virtual reality-based exploration, navigation, exergaming, and rehabilitation, Games for Health J. 7 (3) (2018) 151–156, pMID: 29293369. arXiv:https://doi.org/10.1089/g4h.2017.0114, doi:10.1089/g4h.2017.0114.
[65] J. Park, H. Kim, W. Chung, S. Park, Comparison of myocontrol and force control based on fitts' law model, Int. J. Precision Eng. Manuf. 12 (2) (2011) 211–217, https://doi.org/10.1007/s12541-011-0029-4.
[66] L. Gupta, S. Kota, S. Murali, D.L. Molfese, R. Vaidyanathan, A feature ranking strategy to facilitate multivariate signal classification, IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev..) 40 (1) (2010) 98–108, https://doi.org/10.1109/TSMCC.2009.2024648.
[67] R. Vaidyanathan, N. Nowlan, R. Woodward, S. Shefelbine, Biomechanical activity monitoring, uS Patent App. 15/034,165 (Sep. 15 2016)..