CORDIC Framework for Quaternion-based Joint Angle Computation to Classify Arm Movements

Dwaipayan Biswas*, Zixuan Ye†, Evangelos B. Mazomenos¢, Michael Jöbges‡, Koushik Maharatna†
Email: Dwaipayan.Biswas@imec.be zy2u14@soton.ac.uk e.mazomenos@ucl.ac.uk joebges@brandenburgklinik.de km3@ecs.soton.ac.uk
*IMEC, Heverlee, Belgium
†Electronics and Computer Science, University of Southampton, Hampshire, UK
¢Centre for Medical Image Computing, UCL, London, UK
‡Brandenburg Klinik, Berlin, Germany

***Abstract*— We present a novel architecture for arm movement classification based on kinematic properties (joint angle and position), computed from MARG sensors, using a quaternion-based gradient-descent method and a 2-link model of the upper limb. The design based on Coordinate Rotation Digital Computer framework was validated on stroke survivors and healthy subjects performing three elementary arm movements (reach and retrieve, lift arm, rotate arm), involved in '*making-a-cup-of-tea*', an archetypal daily activity, achieved an overall accuracy of 78% and 85% respectively. The design coded in System Verilog, was synthesized using STMicroelectronics 130 nm technology, occupies 340K NAND2 equivalent area and consumes 292 nW @ 150 Hz, besides being functionally verified up to 25 MHz making it suitable for real-time high speed operations. The orientation, arm position and the joint angle, are computed on-the-fly, with the classification performed at the end of movement duration.**

***Keywords*—MARG sensor, activity recognition, quaternion, CORDIC, classification.**

I. INTRODUCTION

Neurorehabilitation for post-stroke survivors require a long period for regaining impaired motor functionalities. Post initial period of rehabilitation within the clinic, it is important to ascertain patient mobility at home involving daily living activities. Activity recognition in natural settings for remote health monitoring systems has been aided by the advent of mobile and ubiquitous computing facilities using low-cost inertial sensors [1], radio-frequency identification (RFID) [2] and fusion of vision and inertial sensors [3]. MARG (magnetic, angular rate-gyroscope and gravity-accelerometer) sensors are non-intrusive and do not require a clear line of sight compared to the more accurate and expensive marker-based camera/optical systems. Sensor-based activity recognition generally involves intensive computations in feature engineering/classification using learning algorithms [4] or Deep Learning [5]. Research into Wireless Sensor Networks (WSN) have shown that for real-time continuous operations using battery-powered wearable systems, the data analysis primarily needs to be carried out at the sensor node to yield energy efficient solutions as compared to systems transmitting data continuously to a remote station [6]. This necessitates a low-complexity architecture and its optimized implementation for real-time operation.

This work was supported by the European Union under the Seventh Framework Programme, grant agreement #288692, StrokeBack.

On this backdrop, an algorithm was proposed in [7] to precisely detect specific movements of the stroke-impaired arm to enumerate the occurrences during daily activities which would help quantify movement quality on a longitudinal scale. The algorithm uses a 2-link upper limb model, in conjunction with two MARG sensors attached proximal to the wrist and elbow, for estimating the upper limb kinematics in terms of joint angles and position of arm segments, through data fusion using a quaternion-based gradient descent (GD) method. These kinematic metrics act as discriminating features to detect three investigated arm movements (reach and retrieve, lift arm and rotate arm) in a dynamic environment. The algorithm selection was influenced by its novelty in using upper limb kinematics to classify arm movements, besides negating the requirement of a large training dataset as common with majority of the supervised learning techniques. Furthermore, it generalizes and produces consistent outputs when tested upon a wide variety of movement profiles including both healthy and stroke survivors.

In this paper, an optimised algorithm-to-architecture mapping has been proposed using the various transcendental functions realizable using Coordinate Rotation Digital Computer (CORDIC) [8] algorithm. We estimate the arm position and angles on-the-fly from the sensor data, with the classification achieved at the end of movement duration. The design was coded in System Verilog and synthesized using STMicroelectronics 130 nm technology, which occupies 340K NAND2 equivalent cell area and consumes 292 nW, at 150 Hz, deemed sufficient for capturing the kinematic information. The design was functionally verified at higher frequencies up to 25 MHz making it suitable for real-time high-speed operations. The evaluation of experimental data collected during a semi-naturalistic experiment of '*making-cup-of-tea*' with 4 stroke survivors and 4 healthy subjects yielded an overall detection accuracy of 78% and 85% for the three movements. The architecture although developed for upper limb, could be utilized for orientation/position estimation and joint angle computation of different body parts using MARG sensors, an active research area in biomechanics. The remainder of this paper is organized as follows. The algorithm-to-architecture mapping in conjunction with a CORDIC framework is presented in Section II whilst Section III presents the implementation details along with the validation results. Finally, a discussion is presented in Section IV.

II. Algorithm To Architecture Mapping

In this work, we aim to identify/recognize three arm movements – (1) *Action A* – reach and retrieve, (2) *Action B* - lift arm and (3) *Action C* - rotate arm, which constitute majority of the upper limb movements performed in daily life, besides resembling a subset of wolf motor function test (WMFT) activities, an established clinical test battery for stroke rehabilitation. A 2-link limb model has been employed to represent the upper limb (cf. Fig. 1), where the links 1-2/2-3 model the upper/forearm respectively. The shoulder (joint1) is fixed at origin of the global coordinate frame and the elbow is modeled by joint2 connecting the two links. The orientation and position vectors of the upper limb joints in space for dynamic movements can be determined by continuously calculating the MARG sensor orientation w.r.t the 2-link model.

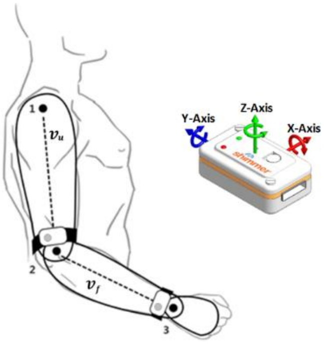

Fig.1 The 2-link limb model and corresponding sensors position [7].

The body coordinate frame (viz. local) is used to obtain the initial orientation of each limb using MARG data and correspondingly are mapped w.r.t a static global coordinate frame (origin on shoulder), to obtain the joint angles and position of the respective links. The body coordinate is considered aligned with the attached sensor coordinate frame, therefore the latter's orientation in effect represents the orientation of the body segment. This helps to estimate the flexion/extension angles from shoulder and elbow [7], used by a rule-based engine to classify the investigated arm movements.

The architecture is based on CORDIC, an iterative 2D vector rotation algorithm for realizing various mathematical operations, implemented through shift-add mechanism making it a low-complexity design choice. For an input vector $[x_0\ \ y_0]^T$, CORDIC can operate in - (a) vectoring mode ($y_0 \rightarrow 0$), the magnitude of the vector and angle between the initial vector and the principal coordinate axis are computed; (b) rotation mode ($z_0 \rightarrow 0$), for a given angle of rotation, the final vector is computed, in three coordinate systems – circular, linear and hyperbolic. We use vectoring mode functions in these three coordinates, denoted as Vec_C, Vec_L, Vec_H, for computing arctangent operations and vector normalization (square root, division), as shown in Table I [9].

TABLE I. CORDIC Vectoring Modes.

Circular (Vec_C)	**Linear (Vec_L)**	**Hyperbolic (Vec_H)**
$x_n = K_c\sqrt{{x_0}^2 + {y_0}^2}$	$x_n = x_0$	$x_n = K_h\sqrt{{x_0}^2 - {y_0}^2}$
$y_n = 0$	$y_n = 0$	$y_n = 0$
$z_n = z_0 + arctan(\frac{y_0}{x_0})$	$z_n = z_0 + \frac{y_0}{x_0}$	$z_n = z_0 + arctan(\frac{y_0}{x_0})$

An overview of our proposed methodology has been illustrated in Fig. 2, showing the datapath widths and the modules - (1) orientation estimation (*OE*), (2) position vector estimation (*PV*), (3) joint angle estimation (*JA*) and lastly, (4) classification (*CLASS*), which have been described in detail. The architecture for each associated module is shown in Fig. 3, where the CORDIC operations have been highlighted in 'blue'.

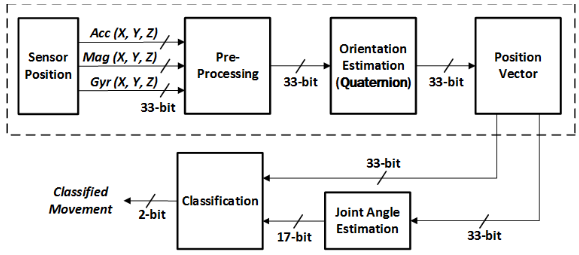

Fig. 2 Architectural overview of the algorithm.

A) *Pre-Processing* – The raw sensor data were calibrated and de-noised using FIR filter. The accelerometer and magnetometer data were low-pass filtered having respective cut-offs 12 Hz and 10 Hz and the gyroscope data was band-passed between 0.5-25 Hz. For each MARG sensor position, we have nine 33-bit signals which are further processed. The filtering architecture has not been discussed since FIR filter implementation is a well-researched topic and here the primary focus was on classification framework.

B) *OE* – We use quaternions to mathematically represent the 3-D orientation of a rigid body (i.e. upper limb), which is devoid of the singularity issues often associated with Euler angle representations. The initial orientation of the upper arm and forearm is obtained from the 3-D orientation of the attached MARG sensors, by fusing data from three sensors and applying a quaternion GD algorithm proposed in [10] during the arm movements. The gyroscope output (s_ω) can be used for deriving the rate of change of orientation (${}^S_E\dot{q}_{\omega,t}$) of the static global reference frame against the dynamic sensor frame, which can be represented in quaternion as:

$$\begin{cases} s_\omega = [0 \quad \omega_x \quad \omega_y \quad \omega_z] \\ {}^S_E\dot{q}_{\omega,t} = \frac{1}{2}\,{}^S_E\hat{q}_{est,t-1} \otimes s_\omega \end{cases} \tag{1}$$

The orientation (${}^S_E\hat{q}_{est,t-1}$) can be estimated by integrating the quaternion derivative ${}^S_E\dot{q}_{\omega,t}$ over time, given an initial condition and sampling frequency of the system. These independent quaternion estimations suffer from inherent sensor limitations, namely accumulation of gyroscope error during integration resulting in distorted orientation and addition of linear acceleration and magnetic interference. The final orientation is achieved through the gradient descent method whereby the integration of the rate of change of orientation (obtained from gyroscope) after subtracting the magnitude of the gyroscope error (β) along the direction specified by the accelerometer/magnetometer readings.

$${}^S_E q_{est,t} = {}^S_E\hat{q}_{est,t-1} + {}^S_E\dot{q}_{est,t} * \Delta_t \tag{2}$$

$${}^S_E\dot{q}_{est,t} = {}^S_E\dot{q}_{\omega,t} - \beta * \frac{\nabla f}{\|\nabla f\|} \tag{3}$$

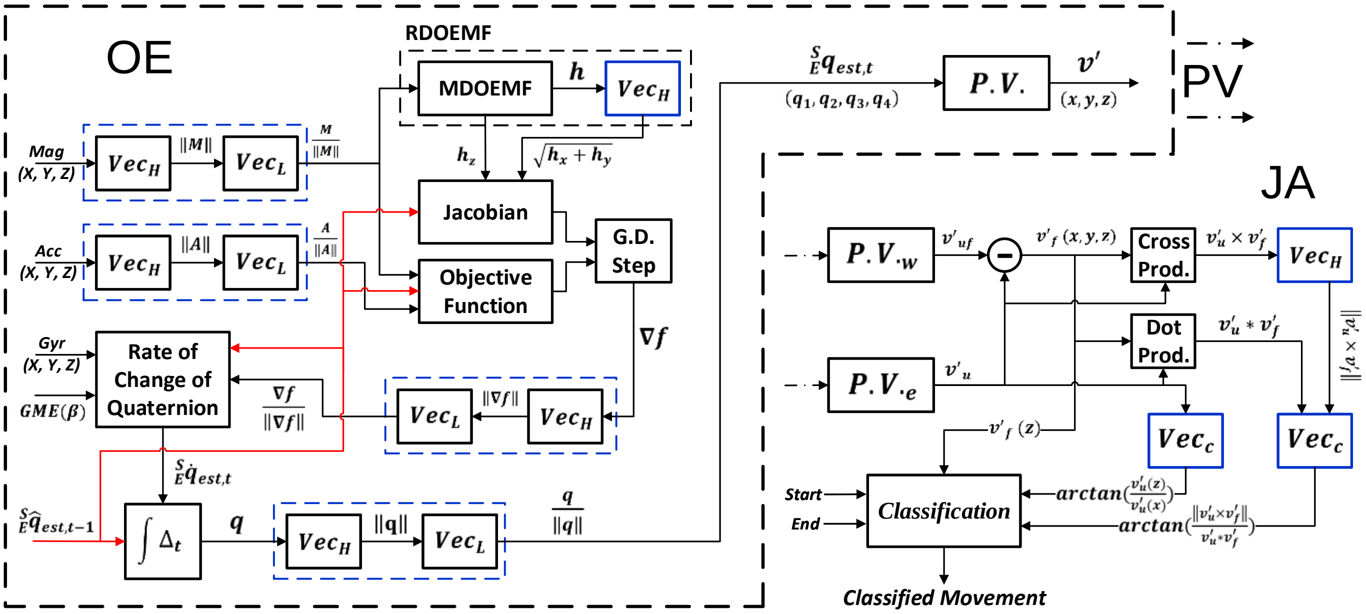

Fig. 3 CORDIC framework for orientation estimation (*OE*), position vector (*PV*), joint angle (*JA*) estimation and arm movement classification (*CLASS*). The *OE* module is replicated for computing orientation on data from both MARG sensor positions – wrist and elbow.

OE module involves the following mathematical operations – (a) 3D vector normalization for accelerometer and magnetometer data, realized with Vec_H and Vec_L, requiring 4 CORDIC operations for 1 square root and 3 divisions respectively; (b) quaternion normalization for the Gradient Descent (GD) computation (∇f), achieved similarly through Vec_H and Vec_L, requiring 5 operations (1 square root and 4 divisions for 4D quaternions); (c) quaternion multiplication (1).

Quaternion multiplication has been computed with 4D CORDIC thereby avoiding the 16 multiplications and 12 addition operations [11]. However, in our design, we have used 2D CORDIC primarily to achieve uniformity and resource re-utilization since it is also used for vector normalization and arctangent computation (in *JA* module). Furthermore, using 4D CORDIC prevents us from tailoring the iteration for known elements with zero value, which for example is the case since our multiplicand (data) is in 3D (cf. 1). Hence, we require 12 multiplications and 8 additions for achieving a Quaternion product. An example normalization can be illustrated with a vector V (x, y, z) or quaternion Q (q_1, q_2, q_3, q_4), where $\|V\|$, $\|Q\|$ (cf. Fig. 3) can be expressed as $\sqrt{x^2+y^2+z^2}$ and $\sqrt{{q_1}^2+{q_2}^2+{q_3}^2+{q_4}^2}$, thus the *norm* of V ($\frac{V}{\|V\|}$) and Q ($\frac{Q}{\|Q\|}$) are computed through Vec_H and Vec_L. In Fig. 3, the *OE* module takes in tri-axial input from *Mag*, *Acc* and *Gyr*; ${}^S_E\hat{q}_{est,t-1}$ represents the initial or previous orientation estimation. The input *Mag*, *Acc* are first normalized, where $\|M\|$ and $\|A\|$ refer to the modulus of each input vector, the normalized products $\frac{M}{\|M\|}$, $\frac{A}{\|A\|}$ are passed on to the Objective Function and Jacobian computation for the GD Algorithm. According to [10], Jacobian requires the reference direction of earth's magnetic field (RDOEMF), which can be represented by a vector $(0, \sqrt{{h_x}^2+{h_y}^2}, 0, h_z)$, where, $h\,(0, h_x, h_y, h_z)$ refers to the measured direction of earth's magnetic field (MDOEMF), obtained by $\frac{M}{\|M\|}$. The normalized GD output ($\frac{\nabla f}{\|\nabla f\|}$) is used to estimate the rate of change of quaternion ${}^S_E\dot{q}_{est,t}$ in accordance to (1).

C) PV & JA – Two position vectors (v'_u and v'_f) w.r.t the body frame as illustrated in Fig.1 are used to locate the position of the upper arm and the forearm during movements. The x-axis of the body frame is aligned with the upper/forearm while it is lying prone against the body and thus the relative positions could be expressed as (4), where l_f and l_u are the lengths of forearm and upper arm respectively. The elbow joint is dynamic while the shoulder is fixed w.r.t the body coordinate frame. During movement, the sensor placed at wrist measures the vector of whole arm v'_{uf} (upper limb + forearm). The relative location of upper limb and forearm in the global coordinate frame is determined according to (5), where q_e and q_w represent the quaternions for the elbow and wrist respectively.

$$\begin{cases} v_u = [-l_u \quad 0 \quad 0] \\ v_f = [-l_f \quad 0 \quad 0] \end{cases} \tag{4}$$

$$\begin{cases} v'_u = q_e \otimes v_u \otimes q_e^* \\ v'_{uf} = q_w \otimes (v_f + v_u) \otimes q_w^* \\ v'_f = v'_{uf} - v'_u \end{cases} \tag{5}$$

Here the 4D quaternion (q_e) multiplication with its conjugate (q_e^*) in (5) is avoided by using a Rotation matrix notation (6) following the scheme proposed in [12].

$$R = \begin{bmatrix} 2({q_0}^2+{q_1}^2)-1 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 2({q_0}^2+{q_2}^2)-1 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 2({q_0}^2+{q_3}^2)-1 \end{bmatrix} \tag{6}$$

The matrix is multiplied with the vectors (4) to achieve the final position vectors. This helps us to exploit the two zero elements in the upper/forearm vectors (4). Hence, we require 8 multiplications and 4 additions in total to achieve each position vector (for upper/forearm), however, these resources can be re-used from *OE*. Using the position vectors (v'_u, v'_f) 2 joint angles pertaining to the shoulder and elbow are computed as in (7). As shown in Fig. 3, the cross product, $\|v'_u \times v'_f\|$ is implemented using Vec_H, while $v'_u * v'_f$ is achieved using a dot product. Finally, the angle calculation using *arctan* is achieved by Vec_C.

$$\begin{cases} sh\_fle\_ext = 90^{\circ} + \arctan(\frac{v'_u(z)}{v'_u(x)}) \\ el\_fle\_ext = \arctan(\frac{\|v'_u \times v'_f\|}{v'_u * v'_f}) \end{cases} \quad (7)$$

D) *CLASS* – The rule-based classification, based on extension/flexion angles of the shoulder and elbow are summarized in Table II. For a detailed understanding of the formulated rules, please refer to [7]. We use a FIFO in association with the classification module whereby the joint angles and position vector computed from each tri-axial MARG sample are stored till the end of duration of the arm movement. A FIFO depth of 1024 is selected, based on the longest duration of an action (approximately 20 seconds). This module works on a combinatorial logic to infer the performed movement at the end of task duration, using the information in the FIFO.

TABLE II. TASK DEFINITION.

Action	Features
A	*Reach and retrieve* $-el\_fle\_ext$ will have a minimum value while sh_fle_ext has a maximum value that are nearly coincident in time.
B	*Lift object to mouth* $-$ el_fle_ext remains near constant and the movement distance of $v_f(z)$ will be a positive value at the middle of task period time.
C	*Rotate an object* $-$ el_fle_ext remains constant and the movement distance of $v_f(z)$ will be a negative value at the middle of task period time.

A detailed list of arithmetic operations for each module is presented in Table III.

TABLE III. SUMMARY OF ARITHMETIC OPERATIONS

FEATURES	CORDIC	MULT	ADD/SUB	ACC	COMP
OE	19	93	88	0	0
PV	0	16	8	0	1
JA	3	9	8	0	0
CLASS	0	0	1	1	3

III. IMPLEMENTATION AND VALIDATION

The design was coded in System Verilog and verified against a corresponding Matlab version. In order to achieve the desired 16-bit accuracy, a 22-bit word-length should be selected [13], according to the formulation (N + Log2N + 2) and have at least 16 CORDIC iterations. In this design to obtain a high accuracy, especially for angle calculation, we use a 33-bit wide datapath (cf. Fig. 2), with the final output using 2-bits representation for the three performed movements. The design was validated on data collected during a bespoke experiment protocol of '*making-a-cup-of-tea*' performed in semi-naturalistic settings to incorporate maximal kinematic variability in the performed movements. The experiment comprised of 20 arm movements with interleaved representations of *Actions A*, *B* and *C* having 10, 5 and 5 occurrences respectively. *Action A* being a common activity in daily life finds more number of occurrences in the protocol [7]. The simulation results (accuracy) presented in Table IV (*A* - 79.5%, *B* - 90.5%, *C* - 82.5%, overall - 83.00%), demonstrate an average drop of 5% compared to software implementation [7], primarily caused due to truncation effects in fixed-point implementation. Here, we have only presented validation results from the stroke survivors as they present a challenge owing to the inherent variability. The overall accuracy for healthy subjects was 85% (software accuracy of 89% [7]).

TABLE IV. HARDWARE RESULTS OF DECTCTION FOR THE STROKE SURVIVORS.

SUBJECT NO	TASK ACCURACY(%) A(#/100)	B(#/50)	C(#/50)	OVERALL (#/200)(%)
SUBJECT 1	82(82%)	43(86%)	42(84%)	167(83.5%)
SUBJECT 2	78(78%)	42(84%)	41(82%)	161(80.5%)
SUBJECT 3	57(57%)	41(82%)	38(76%)	136(68%)
SUBJECT 4	82(82%)	43(86%)	36(72%)	161(80.5%)
TOTALS	299/400 74.75%	169/200 84.5%	157/200 78.5%	625/800 78.13%

The design was synthesized in ST130 nm technology library and consumes a dynamic power of 292 nW, with a NAND2 equivalent area of 340K at 150 Hz. The design was functionally verified up to 25 MhZ making it amenable for high-speed operations. The *OE* module takes 1 clock cycle for the quaternion computation from each sensor module, whereas the *PV* and *JA* modules together require 1 cycle to compute the joint angle and position vector, therefore requiring 3 clock cycles in total (2 for *OE*; 1 for *PV* & *JA*) for a corresponding set of MARG sample inputs. The complexity and area are traded-off with respect to the high speed which has been the focus of this design, given its requirement for real-time joint angle computation and classification. The design can be further optimized by re-utilizing one CORDIC module in various coordinate systems depending on the required computation, however, that would lead to *OE* requiring 20 cycles for each sensor module, along with associated control logic and memory requirements for data-storage, negating on-the-fly computation.

IV. DISCUSSION

In this paper, we have made a novel 'proof-of-concept' attempt to design the architecture for joint angle/body segment position estimation based on quaternion computation for achieving movement classification. The design choices have been primarily guided by high-speed real-time computation requirements, achieved through a uniform CORDIC framework. Our design achieves 85% and 78% prediction accuracy on three movements performed in a semi-naturalistic environment having inherent variability. Future dissemination will focus on further optimisation based on detailed analysis of design choice, resource utilisation and timing/performance of the implementation.

REFERENCES

[1] B. Najafi *et al*., "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 6, pp. 711–723, 2003.

[2] F.E. Martınez-Pérez *et al.*, "Activity inference for ambient intelligence through handling artifacts in a healthcare environment," *Sensors*, vol. 12, no. 1, pp. 1072–1099, 2012.

[3] C. Zhu and W. Sheng, "Motion-and location-based online human daily activity recognition," *Pervasive and Mobile Computing*, vol. 7, no. 2, pp. 256–269, 2011.

[4] D. Biswas *et al.*, "Recognizing upper limb movements with wrist worn inertial sensors using k-means clustering classification," *Hum. Mov. Sci.*, vol. 40, pp. 59–76, Apr 2015.

[5] F.J. Ordóñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," *Sensors*, vol. 16, no. 1, pp. 115, 2016.

[6] K. Maharatna *et al.*, "Towards the development of next-generation remote healthcare system: Some practical considerations," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pp. 1–4, 2012.

[7] E.B. Mazomenos *et al.*, "Detecting Elementary Arm Movements by Tracking Upper Limb Joint Angles with MARG Sensors," *IEEE Journal of Biomedical & Health Informatics*, vol. 20, no. 4, pp.1088-1099, 2016.

[8] J.E. Volder, "The CORDIC trigonometric computing technique," IRE Transactions on *Electronic Computers*, EC-8, no. 3, pp.330-334, 1959.

[9] P.K. Meher *et al.*, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp.1893-1907, 2009.

[10] S.O. Madgwick, A.J. Harrison and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1-7, Jun 2011.

[11] M. Parfieniuk and S. Y. Park. "Sparse-iteration 4D CORDIC algorithms for multiplying Quaternions." IEEE Transactions on Computers 65.9 (2016): 2859-2871.

[12] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors.," *Matrix*, vol. 58, no. 15-16, pp.1-35, 2006.

[13] L. Vachhani *et al.*, "Efficient CORDIC algorithms and architectures for low area and high throughput implementation," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 56, no. 1, pp. 61–65, 2009.