

EXHIBIT 1052

CORDIC-Based Inclination Sensing Algorithm Using Three-Axis Accelerometer-Based Inertial Sensors

Wei-Ting Liao, W.-C. Chou, and Wen-Yen Lin*

Dept. of Electrical Engineering
Healthy Aging Research Center, Chang Gung University
Tao-Yuan, Taiwan

Kin Fong Lei, Wen-Wei Tsai and Ming-Yih Lee

Graduate Institute of Medical Mechatronics
Healthy Aging Research Center, Chang Gung University
Tao-Yuan, Taiwan

Abstract— In this paper, we present a low complexity and memory-less algorithm for angle calculations of a system using 3-axis accelerometers. Accelerometers have been widely used to determine the inclination or tilt of a system where the net acceleration or force on a system over time is gravity. However, the angle calculations are complicated and most of the work either implementing with look-up tables or simply transmit the data to systems with more powerful computation resources, such as a PC, to calculate the angles. To reduce the computational complexity and release the memory burden, we propose a 2D CORDIC-based algorithm for the inclination or tilt angle calculations so that it is feasible on a low-cost and low power microcontroller in an embedded system.

Keywords- accelerometer; inclination sensing; CORDIC; algorithm

I. INTRODUCTION

Nowadays, low-g MEMS accelerometers have been widely used for inclination or tilt sensing of a system. In some applications, where the net acceleration or force on the system over time is gravity, the projections of the gravity vector on the axes of the accelerometer can determine the static inclination or tilt angles of the system. Such applications include some in consumer electronics, such as gaming, horizon detection in digital cameras, screen rotation in PDA phones and tablet PCs, some in industrial applications, such as detecting the heading of a device, and some in medical applications, such as human posture monitoring, post surgery patient care, etc. It has been brought into much attention for the medical applications due to the aging of the societies and the increasing demands for the tele-care as well as the diagnosis and early detections of the syndromes in many elderly related chronic diseases, such as Parkinson's disease, etc. Other studies, as in [1][2][3], used accelerometer in patient's head for head's angle position monitoring after vitreoretinal surgery. Keeping the head in an optimum position during the post-operation period is crucial for the whole recovery process to proceed well. In [4], it used accelerometers to monitor spinal posture during activities of daily living. The measurement was accomplished by positioning accelerometers along the vertical axis of the spine in the mid-sagittal plane and collecting data indicating the angles of spinal movement.

When a 3-axis accelerometer is used for inclination sensing, the orientation of the sensor can be determined in a complete

sphere using trigonometric functions from the projected gravity vector on the 3 axes of the accelerometer. However, the mathematical calculations of these trigonometric functions are usually too complicated to implement on a microcontroller. Hence, most previous research work either store the raw accelerometer data on storage media and using PC for offline calculations, as in [4], or transmit the raw data to PC or PDA and have the orientation information calculated there, as in [5][6][7]. In [8], it used tables to look up to the inclination/tilt angles with accelerometer raw data to avoid the complex computation. However, to reach the maximum resolution and sensitivity of the inclination/tilt angles available on a 3-axis accelerometer, 3-dimensional table will be required. For example, the ADXL345, a 3-Axis digital accelerometer from Analog Device, Inc., has 10-bit fixed resolution for $\pm 2g$ range. It would require 9-bit for $\pm 1g$ sensing range on each axis. Even if the angle is represented in a byte, it will require a table size of 128 Mbytes ($2^9 \times 2^9 \times 2^9$ bytes) to reach the maximum resolution and sensitivity. This is not feasible on typical microcontrollers which usually only have few hundred Kbytes or even less of flash or ROM on chip.

In this paper, we propose a memory-less and low complexity algorithm for angle calculation to supplant the complex multiplication, division, square-root operations required in the original trigonometric functions. This algorithm is based on the 2D CORDIC algorithm [9] that can use simple addition/subtraction and shifting operations to obtain inclination/tilt angles. With this algorithm, it is feasible to have microcontroller obtaining orientation information from accelerometer in real time, and the microcontroller can also make further detection and post early alarm on some critical posture behaviors from an embedded system. This can also avoid the excessively wireless data transmission to remote PC/PDA side for angle calculation there.

II. CORDIC

The CORDIC (COordinate Rotation Digital Computer) is an algorithm developed in 1959, introduced by Jack E. Volder [10]. It performs two-dimensional (2D) rotations using simple hardware components, essentially adders and shifters. It implements the rotation of a 2D vector in Euclidean or pseudo-Euclidean space as a sequence of simple elementary rotations. The CORDIC algorithm has been used for many years to provide the calculation in hardware of essentially the

trigonometric functions. So far the CORDIC algorithm has been widely used in the applications of digital communications and graphics, etc.

The basic idea of CORDIC is to transform 2D-rotation into an operation that does not require complex operations or large lookup tables. There are two major operation modes in CORDIC, i.e. vectoring and rotation modes. Vectoring mode (or known as angle accumulation mode), as shown in Figure 1, is to rotate the given initial vector $V(0)$ ($[x_0, y_0]$) back to the x-axis so that the angle θ between them can be accrued, and the y-coordinate will be eliminated. The rotation mode, as shown in Figure 2 is to compute the final vector $V(f)$ ($[x_f, y_f]$) with the given desired angle θ rotated from the initial vector $V(0)$ ($[x_0, y_0]$).

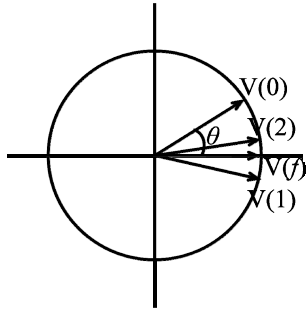


Figure 1. CORDIC vectoring mode operation

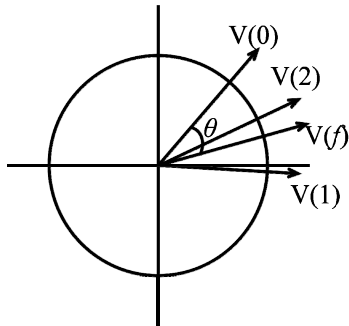


Figure 2. CORDIC rotation mode operation

The basic principle of a 2D-CORDIC vectoring operations is to perform a series of elementary rotations until the y-coordination approaching zero. After that, the total angle accumulated from the series of rotation would be the angle θ . The 2D-rotation for each elementary rotation for vector $V(i)$ ($[x_i, y_i]$) rotated with angle θ_i can be calculated as in (1).

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \cdot \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} = \cos\theta_i \cdot \begin{bmatrix} 1 & -\tan\theta_i \\ \tan\theta_i & 1 \end{bmatrix} \cdot \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (1)$$

The rotation angle θ_i in i -th elementary rotation is predefined as $\tan(\theta_i) = 2^{-i}$, i.e. $\theta_i = \arctan(2^{-i})$, where $i = 0, 1, 2, \dots$, and the rotation direction, d_i , would depend on the current value of $y(i)$. If $y(i) < 0$, then $d_i = 1$, the i -th rotation would be counter clockwise, otherwise $d_i = -1$, the i -th rotation would be clockwise. Therefore, the angle θ can be accrued as in (2).

$$\theta = \sum_{i=0} d_i \theta_i = \sum_{i=0} d_i \arctan(2^{-i}) \quad (2)$$

Equation (3) shows the iterative 2D-rotation derived from (1) and (2).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \prod_{i=0} \cos\theta_i \begin{bmatrix} 1 - d_i 2^{-i} \\ d_i 2^{-i} + 1 \end{bmatrix} \cdot \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{K} \cdot \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \Rightarrow K = \frac{1}{\prod_{i=0} \cos\theta_i}$$

The un-scaled rotation equation can be easily implemented iteratively in a program as shown in (4).

$$\begin{cases} x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i} \\ z_{i+1} = z_i + d_i \cdot \tan^{-1}(2^{-i}) \end{cases} \text{ where } \begin{cases} d_i = 1, y_i < 0 \\ d_i = -1, \text{otherwise} \end{cases} \quad (4)$$

Thus, they can be realized by purely shift and add operations. The final scaling factor $1/K$ (about 0.6073) is then multiplied to x_f (last iteration of x_i) to get the final scaled vector length. Since the scaling factor is a constant, so the multiplication operation can be decomposed into few shift and add/sub operations.

The rotation mode operation can be done similarly with z_0 set to the desired rotation angle θ . Through similar iteration operations but trying to eliminate the final z_i to zero, then the scaled x_i and y_i would be the final vector after the original vector rotated with the desired angle θ .

III. PRINCIPLES OF INCLINATION/TILT ANGLE ON 3-AXIS TILT CALCULATION

Inclination sensing uses the gravity vector and its projection on the axes of the accelerometer to determine the tilt angles. This section discusses the basic principles for converting the output of an accelerometer to the angles of inclination or tilt. Depending on the number of axes available in the accelerometer, different inclination or tilt angles could be defined and calculated. However, using single-axis or dual-axis sensing will either suffer from the resolution issues and sensitivity issues. Also, limitation on using single-axis or dual-axis measurement data will not be able to provide the orientation of the sensor in a complete sphere [11][12].

Therefore, in the section, we will be only focus on the principles of inclination/tilt sensing using 3-axis accelerometers.

Three-Axis Tilt Calculation

In a 3-axis accelerometer, when the net acceleration or force on the system over time is gravity, the measured data from the 3-axis accelerometer are the projections of the gravity

vector on the three axes, X, Y, and Z. Naturally, the measurement data are the rectangular coordination (A_x, A_y, A_z) of gravity vector in the accelerometer coordinate space. To understand the orientation of the accelerometer, we can also represent the gravity vector in the accelerometer coordinate space as in spherical coordinate system, (ρ, θ, ϕ). Figure 3 illustrates the representation, where ρ is the length of the gravity vector, i.e. $1g$, θ is the angle between x-axis and the projected gravity vector on the x-y plane, where ϕ is the angle between z-axis and the gravity vector.

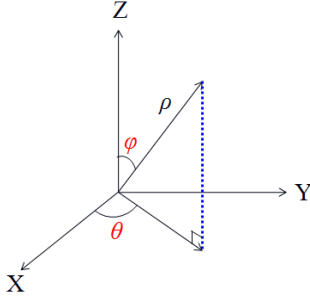


Figure 3. Angles of spherical coordinate system

The transformation equations between these two coordinate systems can be expressed in (5).

$$\begin{aligned} A_x &= \rho \sin \phi \cdot \cos \theta \\ A_y &= \rho \sin \phi \cdot \sin \theta \\ A_z &= \rho \cos \phi \end{aligned} \quad (5)$$

where, $0 \leq \phi \leq \pi$ and $-\pi \leq \theta \leq \pi$.

So given a reading from 3D accelerometer data (A_x, A_y, A_z), we can calculate the two angles ϕ and θ as in (6).

$$\begin{aligned} \theta &= \arctan(A_y / A_x) + \sigma \cdot \lambda \cdot \pi \\ \phi &= \pi / 2 - \arctan(A_z / \sqrt{A_x^2 + A_y^2}) \end{aligned} \quad (6)$$

The term of $\sigma \cdot \lambda \cdot \pi$ added in θ is to have it cover the whole 360 degree range, where $\sigma = 0$, if $A_x \geq 0$, and otherwise, $\sigma = 1$. $\lambda = 1$, if $A_y \geq 0$, and otherwise, $\lambda = -1$.

IV. PROPOSED CORDIC-BASED INCLINATION SENSING ALGORITHM

To calculate the inclination/tilt angles as defined in section III, we have hereby proposed a memory-less and low complexity algorithm which is based on the 2D CORDIC operations. Using 2D CORDIC operations supplants the complex multiplication, division, square-root operations required in the original trigonometric functions. Hence, it is feasible to implement the inclination sensing in microcontroller to obtain orientation information from accelerometer in real time without the need of huge memory space for table look up or excessive computation power required. The proposed

inclination sensing algorithm is separated into two phases as described in the following subsections.

A. Phase I - Calculation of θ and $\sqrt{A_x^2 + A_y^2}$

The first phase is to calculate the angle θ and the length of the projected gravity vector on x-y plane, i.e. $\sqrt{A_x^2 + A_y^2}$. We can apply 2D CORDIC operations on this calculation which is shown in Figure 4

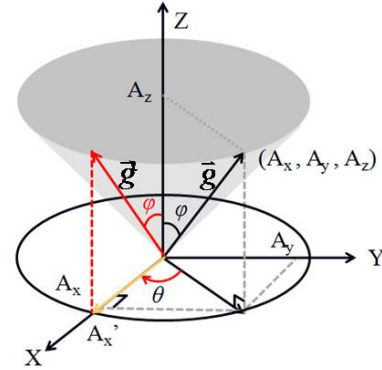


Figure 4. Phase I - calculation of θ and $\sqrt{A_x^2 + A_y^2}$

The idea is to rotate the $\bar{g}$ vector around Z-axis (along the edge of the cone) so that the projected vector on X-Y plane from $\bar{g}$ vector will be aligned to X-axis. We can then apply 2D CORDIC vectoring mode operations on x-y plane with given initial 2D vector (A_x, A_y). After couples of iterations, the angle θ is accrued and the final vector, ($A_x', 0$) is generated, where A_x' is exactly the length of the projected gravity vector on X-Y plane, $\sqrt{A_x^2 + A_y^2}$. The 2D CORDIC operation equations are shown in (7). Notice that since the original gravity vector is rotated around the Z-axis (along the edge of the cone in gray color in Figure 4), the angle ϕ remains the same and hence the projected vector on z-axis, A_z , also remains unchanged.

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ \theta_{i+1} &= \theta_i + d_i \cdot \arctan(2^{-i}) \end{aligned} \quad (7)$$

Initially, $x_0 = A_x$, $y_0 = A_y$ and $\theta_0 = 0$. After j iterations to reach the desired accuracy, we can obtain $A_y' \approx 0$ and $A_x' = \sqrt{A_x^2 + A_y^2} \approx 0.6037 \cdot x_j$.

The calculated θ_j is still within the range of $\pm (\pi/2)$, and we can apply the conversion term as describe in (6) to convert it into $\pm \pi$ range accordingly. After finishing of this phase, we have obtained θ and $\sqrt{A_x^2 + A_y^2}$.

B. Phase II – Calculation of φ

After Phase I, we have obtained angle θ and most importantly, $\sqrt{A_x^2 + A_y^2}$, to be used in Phase II to calculate angle φ .

Instead of calculating angle φ directly, we proposed to find angle ψ instead for better sensitivity, where ψ is the angle between the gravity vector and x-y plane, as illustrated in Figure 5. Then, $\varphi = \pi/2 - \psi$. Since ψ could be calculated as in (8), with the range of $-\pi/2 \leq \psi \leq \pi/2$, then, we can obtain φ as in our desired range $0 \leq \varphi \leq \pi$.

$$\psi = \arctan\left(A_z / \sqrt{A_x^2 + A_y^2}\right) \quad (8)$$

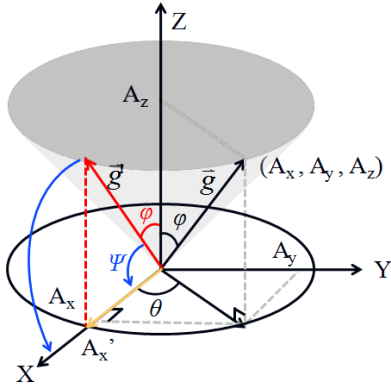


Figure 5. Phase II – calculation of φ

As we already rotated the original gravity vector $\bar{g}$ to $\bar{g}'$ which aligned its projected vector on x-y plane to x-axis in phase I, we can then rotate the $\bar{g}'$ vector on x-z plane and to have it aligned on x-axis so that the angle ψ can be accrued. This can be implemented by applying a 2D CORDIC vectoring mode operations on x-z plane with given initial 2D vector (A_x', A_z) . The iterative equations are described in (9).

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ \psi_{i+1} &= \psi_i + d_i \cdot \arctan(2^{-i}) \end{aligned} \quad (9)$$

Initially, $x_0=A_x'$, $y_0=A_z$ and $\psi_0=0$. After j iterations to reach the desired accuracy, we can obtain $\varphi = \pi/2 - \psi_j$. Since the only task in this phase is to find the angle φ , the scaling factor multiplication is not needed in this phase.

After conducting these two phases of algorithm, the inclination/tilt angles, θ and φ , are obtained with simple shifting and addition operations only by this proposed inclination sensing algorithm.

V. ACCURACY ANALYSIS AND PERFORMANCE SIMULATION

The accuracy of the algorithm would be one of the primary concerns. There are two factors that would impact the accuracy of the results from CORDIC operations: number of iterations and input conditions. As a result, we name the errors due to these two factors, iteration error and input error.

A. Analysis of Iteration Errors

Assume that the CORDIC vectoring mode operations perform only i iterations, then the angles which are not accumulated in the following iterations would cause the maximum iteration error. This is described in (10). The maximum iteration error on the angle will set to the limit of the resolution that it could reach in our algorithm.

$$\theta_{\text{iteration - errors}} = \pm \sum_{n=i}^{\infty} \arctan(2^{-n}) \quad (10)$$

Assume 8 iterations are performed in a CORDIC operation, from (10), we can calculate the angle resolutions from our algorithm as in (11), and that is ± 0.4476 degree.

$$\theta_{\text{resolution}} \approx \pm 0.22381 \times \sum_{n=0}^{\infty} (2^{-n}) \approx \pm 0.4476^\circ \quad (11)$$

From this analysis, our proposed algorithm can reach very fine angle resolution (± 0.4476 degree) with only 8 iterations with a CORDIC operation if the input errors are not considered.

B. Phenomena of Input Errors

From the iterative CORDIC operation equations in (4), when the initial values of x_0 and y_0 are relatively small, the input errors start to cut in. In such cases, after a few iterations, the x and y values will remain unchanged through the rest of iterations due to $x_i \cdot 2^{-i}$ and $y_i \cdot 2^{-i}$ becoming zero; however, the angle z_i continues to be accumulated until the number of iterations is reached. Hence, the input errors will impact the angle resolution of the proposed algorithm when the input vectors for the CORDIC operations have relative small length.

C. Simulation and Performance measurement

To evaluate our proposed CORDIC-based inclination sensing algorithm, we have implemented simulation program under Visual C++ 2008 environment with a 16-bit CORDIC engine in order to simulate the accuracy of the proposed algorithm implemented in a 16-bit microcontroller. The data input are assumed to be from ADXL345 3-axis accelerometer which has fixed 12-bit resolution for $\pm 8g$ range. Therefore, we adopt 12.4 two's complement fixed-point format (12 bits for integer part and 4 bits for fractional) for the accelerometer reading data. The CORDIC engine will deal with angles within ± 180 degree range, so we adopt 9.7 two's complement fixed-point format for angle representation.

We have thoroughly generated the 3-axis accelerometer data of gravity vectors in a complete sphere, that is all ranges

of A_x , A_y , and A_z , that fulfill the equation of $A_x^2 + A_y^2 + A_z^2 = 255^2$. The angles calculated from the proposed algorithm are then compared with the calculation results from using trigonometric functions. We have simulated with different number of iterations in CORDIC operations from 4 to 12 iterations and plot the error ranges, $\Delta\theta$ and $\Delta\phi$, as in Figure 6.

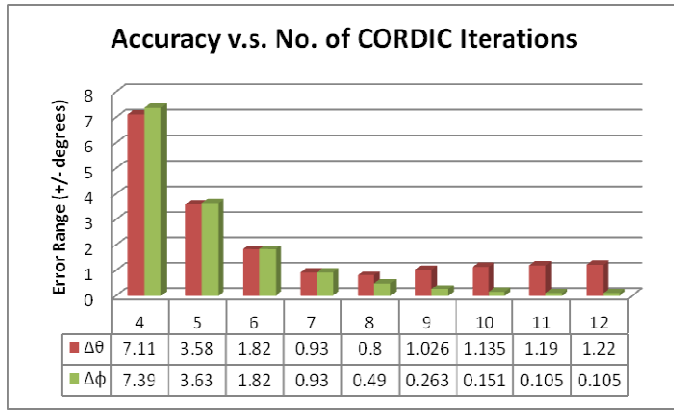


Figure 6. Accuracy Comparison of θ and ϕ with varied number of iterations

From this comparison, we observe that we could reach optimum angle resolution for θ at 8 iterations. Surprisingly, the error range of angle θ does not always decrease while the number of iterations increased. From the simulation results, we can observe that the error range of angle θ also start to increase slightly with the number of iterations increased after 8 iterations. Actually, the iteration errors dominate the maximum error ranges of angle θ with lower number of iterations (< 8) and the input errors dominate the maximum error range with higher number of iterations (> 8). From equation (6), there are few cases that both A_x and A_y are small and will incur input errors.

However, the maximum error ranges of angle ϕ always decreases as the number of iterations increased. This is because that the input errors do not exist in the calculation of angle ϕ . As we can see from equation (6) and our test cases always have the vector lengths equal to 255 (1g), therefore, no input errors exist in angle ϕ calculations. Even without the input errors while calculating angle ϕ , the maximum error range of it still exceeds the iteration errors. For example, the maximum error range of angle ϕ is ± 0.49 which is larger than the iteration errors, ± 0.4476 in the case of 8 iterations. This is because that the error of the calculation of $\sqrt{A_x^2 + A_y^2}$ in phase I is accumulated into phase II calculation as it uses $\sqrt{A_x^2 + A_y^2}$ as the input for phase II calculation.

VI. FUTURE WORK AND CONCLUSIONS

In summary, the proposed CORDIC-based inclination sensing algorithm can deliver ± 0.8 degree of resolution for angle θ and ± 0.5 degree of resolution for angle ϕ in 8 iterations of CORDIC operations without the need of huge memory space and excessive computation power. Using 2D CORDIC operations supplants the complex multiplication,

division, square-root operations required in the original trigonometric functions with simple shifter and adder. As a result, the proposed algorithm can be easily implemented in a 16-bit microcontroller for inclination sensing in real time.

We are in the process of designing an embedded system for real time inclination sensing using a low power microcontroller, ADCU7024, from Analog Devices, Inc. to implement with the proposed CORDIC-based algorithm. We will also looking into some variances of CORDIC algorithm to further improve the resolution.

ACKNOWLEDGMENT

This work was supported in part by National Science Council, Taiwan, R.O.C. under Grant NSC100-2221-E-182-039 and funding from Chang Gung University (CGURP) under Grant No. UERPD290031. The authors would also like to thank the support of High Speed Intelligent Communication (HSIC) Research Center in Chang Gung University for the required resources of this work.

REFERENCES

- [1] Dlouhy, J.; Cizek, M.; Rozman, J., "Patients head position monitoring," *17th International Conference on Radioelektronika*, 2007., April. 2007
- [2] Cech, P.; Dlouhy, J.; Cizek, M.; Vicha, I.; Rozman, J., "Head position monitoring system," *Applied Electronics*, 2009, Sept. 2009
- [3] Cech, P.; Dlouhy, J.; Cizek, M.; Vicha, I.; Rozman, J., "Head position monitoring system design," *2010 20th International Conference on Radioelektronika*, April. 2010
- [4] Nevins, R.J.; Durdle, N.G.; Raso, V.J., "A posture monitoring system using accelerometers," *IEEE Canadian Conference on Electrical and Computer Engineering*, 2002 (CCECE 2002), vol.2, pp.1087- 1092, 2002
- [5] E. Farella, A. Pieracci, L. Benini, and A. Acquaviva, "A wireless body area sensor network for posture detection," in *Proceedings of 11th IEEE Symposium on Computers and Communications (ISCC '06)*, pp. 454-459, June 2006
- [6] W.T. Huang, C.N. Hung, C.C. Tuan, C.C. Chen, Y.J. Chang, and Y.Y. Chen, "Novel design of interactive caption and recognition postures within the wearable and non-vision system over wireless sensor network," *2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pp.752-757, Feb. 2010
- [7] H.R. Silva, J.A. Afonso, P.C. Morim, P.M. Oliveira, J.H. Correia, and L.A. Rocha, "Wireless hydrotherapy smart-suit network for posture monitoring," *IEEE International Symposium on Industrial Electronics (ISIE 2007)*, pp.2713-2717, June 2007
- [8] Cech, P.; Dlouhy, J.; Cizek, M.; Vicha, I.; Rozman, J., "Software for automatic head position monitoring," *19th International Conference on Radioelektronika*, 2009. (RADIOELEKTRONIKA '09), April. 2009
- [9] Y.H. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Process. Magaz.*, vol.9, no.3, pp.16-35, Jul 1992
- [10] Volder, Jack E.; , "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol.EC-8, no.3, pp.330-334, Sept. 1959
- [11] Christopher J. Fisher, "Using an accelerometer for inclination sensing," *Application Note, AN-1057*, Rev. 0, Analog Devices, Inc.
- [12] "Tile measurement using a low-g 3-axis accelerometer," *Application Note, AN3182*, Doc ID 17289, Rev. 1, STMicroelectronis, April 2010