

# An IMU-based Wearable Presentation Pointing Device

Dimitrios Sikeridis and Theodore A. Antonakopoulos  
 Department of Electrical and Computer Engineering  
 University of Patras  
 Patras 26504, Greece  
 Email: d.sikeridis@upnet.gr, antonako@upatras.gr

**Abstract**—Wearable motion capture systems based on MEMS IMUs are considered a promising technology that evolves rapidly over the last few years. In this paper we present a wearable pointing solution developed with a pair of inertia sensors equipped with Bluetooth connectivity. Each IMU consists of acceleration and angular velocity sensors and the pair is mounted on the user’s arm. Our system can be used for pointing on screen by capturing the arm’s rotational movement. We present the system’s functionality and a training method for estimating the screen’s distance and relative position. Experimental results show promising pointing accuracy and precision in a variety of different display sizes and distances making our system an effective solution for on-screen pointing.

## I. INTRODUCTION

During the last few years screen displays are increasing both in size and resolution and often conventional pointing methods are not an efficient option in a variety of cases. In scenarios where users are required to point to a screen during a presentation or a teaching session there is often a need for a wearable system able to facilitate this option. Especially in places like conference halls where the size of the display and the distance between the speaker and the screen can vary and be of a significant proportion, the use of a portable, lightweight and comfortable solution becomes desirable.

Since MEMS (Micro-Electro-Mechanical-Systems) sensors such as gyroscopes and accelerometers have become greatly available, human movement can be instantly and easily measured. Therefore such wearable devices spark great interest as input devices interfacing human motion and computer environments and are suitable for applications concerning pointing tasks. Various works deploy MEMS sensors to mainly introduce mouse functionality. These sensors are either mounted on the human body (head [1], [2] or hand [3]) or a part of other devices [4].

In this work, we present a wearable pointing system consisting of two inertial measurement units (IMUs) mounted on the user’s arm as shown in Fig. 1(a). By fusing data from the inertial sensors and applying calibration and filtering techniques we manage to estimate the screen positions pointed by the user. The algorithm used outputs two angles,  $\theta$  and  $\psi$  shown in Figures 1(b),1(c). These angles are used to estimate the projected on screen position. To evaluate our design, we created an experimental process using a PTU-D46 Pan/Tilt Unit [5] and we studied the accuracy of the estimation when

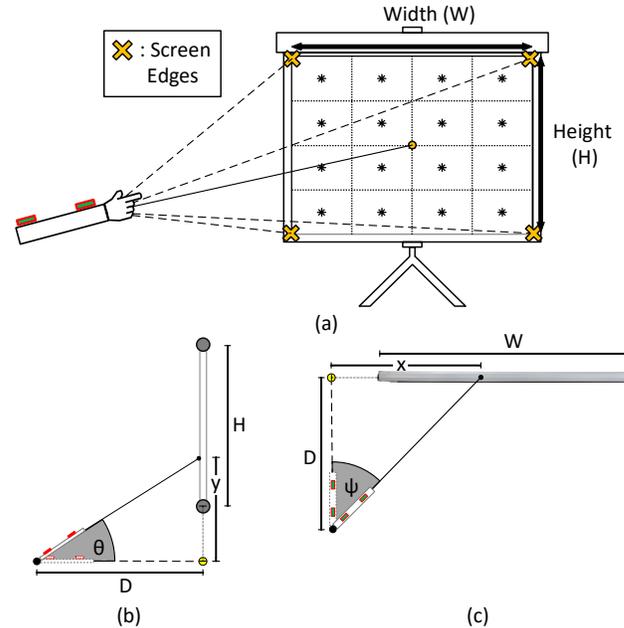


Fig. 1. (a) Arm pointing on screen. Angular orientation translated to pointer coordinates: (b) Rear view, (c) Top View

parameters such as the screen-user distance and screen size change.

Section II presents the basic methods used by the algorithm while in Section III we present the experimental setup used for the performance analysis. Finally, the experimental results are presented in Section IV and discussed in Section V.

## II. PROJECTION ESTIMATION METHODS

### A. Problem Formulation

A MEMS accelerometer measures linear accelerations caused by motion and gravity while a gyroscope measures angular velocity. The aim is to extract sensor attitude estimation by combining these two signals. Various methods have been proposed to accommodate this functionality including Kalman filters [6] or gradient descent based algorithms [7]. Our approach is based on Mahony’s IMU algorithm [8] implemented using quaternion representation. Quaternions are four-dimensional complex numbers used to represent the

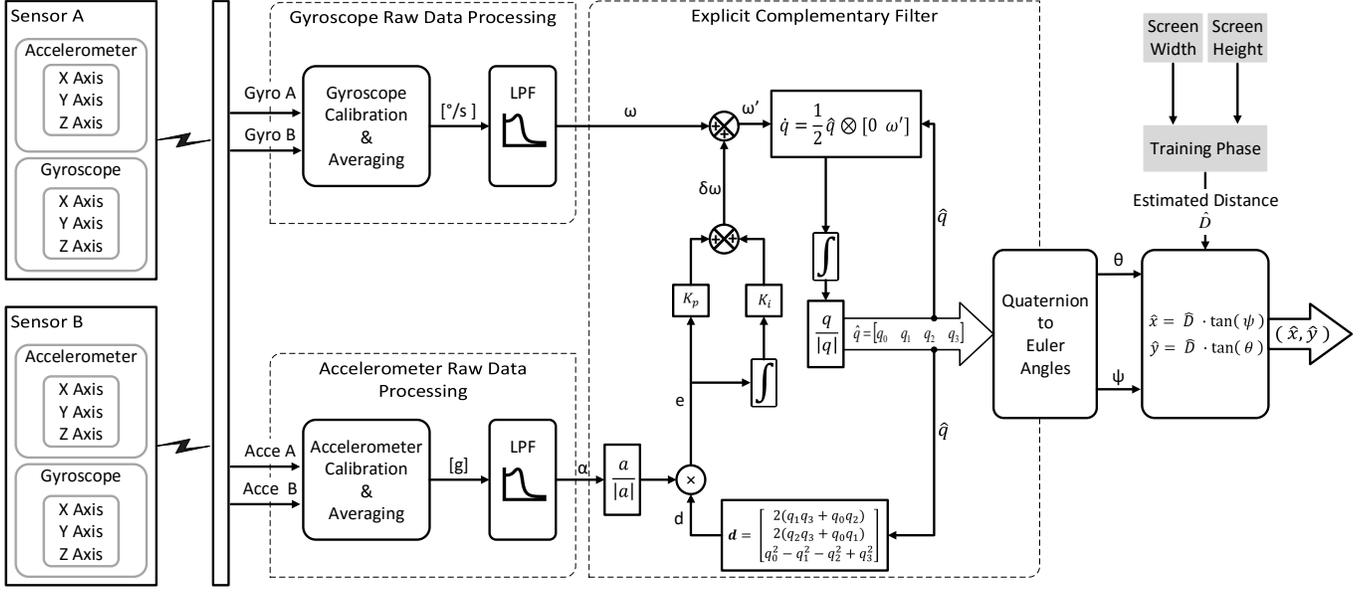


Fig. 2. System block diagram.

orientation of a rigid body -here the arm- in three-dimensional space. Other ways to represent rotation are Direction Cosine Matrices and Euler angles. Euler angles represent a rotation about each axis of the 3D space with an angle -X axis: "Roll" angle  $\phi$ , Y axis: "Pitch" angle  $\theta$ , Z axis: "Yaw" angle  $\psi$ -. More about rotations can be found in [9].

### B. Raw Data Processing

The raw digital signals of the IMU pair go through an initial processing that consists of three stages: Calibration, Averaging and Filtering.

1) *Calibration and Averaging*: The two sensors initially undergo a calibration procedure mentioned in Section III. Raw data transmitted from each individual MEMS sensor are corrected using a gain matrix and a bias vector which are produced by the calibration procedure. This correction removes bias, gain and misalignment errors that exist in this type of sensors.

As both IMUs are mounted on the same part of the arm they describe the same motion. Therefore, an averaging strategy is used to combine the accelerometer and gyroscope measurements of the IMU pair. As the sensors are identical, this technique is a method for approximating the true value. The objective is to combine the measured values linearly so that the resulting signal has a minimum variance noise error component.

2) *Low-Pass Filtering*: In order to further eliminate high-frequency noise and make the signals more smooth, a low pass IIR filter was used to filter the accelerometer and gyroscope readings. The filter has been chosen according to a characterization of human activity where it is found that the average frequency region of human motion is about 1 Hz and almost total signal energy is below 4 Hz [10], [11]. Therefore, a 6th

order Butterworth IIR low-pass filter was implemented with a 4 Hz cutoff frequency.

### C. Explicit Complementary Filter

As mentioned previously, in order to estimate the arm's attitude, in our prototype we adopted the strategy proposed by Madwick in [8]. This method is referred to as *Explicit Complementary Filter* (ECF) and it is shown in Fig. 2 in its quaternion form. The attitude acquisition filter uses as inputs the three components of accelerometer and gyroscope measurements respectively:

$$\omega = [\omega_x \quad \omega_y \quad \omega_z]^T \quad (1)$$

$$\alpha = [\alpha_x \quad \alpha_y \quad \alpha_z]^T \quad (2)$$

All measurements provided by the IMUs are performed in the sensor (body-fixed) frame with respect to the Earth-fixed frame which is tangent to the Earth's surface. The sensor frame aligns with the arm frame as the sensors are mounted to it.

The ECF combines the two sources of data in a complementary manner. The filter mixes the static low-frequency information provided by accelerometers and the dynamic high-frequency information provided by the angular rate sensors. The aim is to balance the short-term integration of the gyroscope and the long-term measurements obtained by the accelerometer.

The algorithm is based on the quaternion kinematic differential equation:

$$\dot{q} = \frac{1}{2} \hat{q} \otimes \omega_G \quad (3)$$

where:

- $\hat{q} = [q_0 \quad q_{vect}^T]^T$  is a unit quaternion that represents a rigid body attitude between two frames, the body-fixed frame B and the Earth-fixed frame N

- $q_{vect} = [q_1 \ q_2 \ q_3]^T$  represents the vector part of  $q$ .
- $\omega_G = [0 \ \omega_g^T]^T$  is a pure vector quaternion with a zero scalar part and a vector part  $\omega_g = [\omega_x \ \omega_y \ \omega_z]^T$  equal to the corrected measurements of angular rate.
- $\hat{q} \otimes \omega_G$  is a quaternion multiplication
- $\dot{q}$  is quaternion rate

More about quaternions can be found in [12].

The method starts by initializing the quaternion value at  $[1 \ 0 \ 0 \ 0]$  as no rotation is present at this point. After that, the vectors of gyroscope and accelerometer data (eqs. 1 and 2) are entered and the error of the measured inertial direction in proportion to the estimated gravitational direction is computed by cross multiplying them:

$$e = a \times d \quad (4)$$

The measured inertial direction  $a$  is calculated by normalizing the new accelerometer readings, while the estimated gravitational direction is computed as:

$$d = \begin{bmatrix} 2(q_1 q_3 + q_0 q_2) \\ 2(q_2 q_3 + q_0 q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (5)$$

where the quaternion of the previous iteration:  $\hat{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$  is used.

When the error  $e$  has been calculated, it is applied as a feedback term with two coefficients: the proportional gain  $K_p$  and the integral gain  $K_i$ , forming a PI controller. The corrected angular rate  $\omega'$  is used to compute the aforementioned quaternion rate of change  $\dot{q} = \frac{1}{2} \hat{q} \otimes [0 \ \omega']$ , where  $\hat{q}$  is the quaternion of the previous iteration. Finally, the corrected quaternion rate is integrated and the result is normalized to yield the new estimated quaternion.

After this final step the updated attitude expressed in quaternion form is transformed into Euler Angles representation according to the following formulas:

$$\begin{aligned} \psi &= \text{Atan2}(2q_2 q_3 - 2q_1 q_4, 2q_1^2 + 2q_2^2 - 1) \\ \theta &= -\sin^{-1}(2q_2 q_4 + 2q_1 q_3) \\ \phi &= \text{Atan2}(2q_3 q_4 - 2q_1 q_2, 2q_1^2 + 2q_4^2 - 1) \end{aligned} \quad (6)$$

#### D. Projected Position Estimation

Our system aims to direct a virtual ray originated at a predefined on-body location using the rotational data acquired from the IMU pair, which is mounted in the user's arm. In our case this on-body origin is the user's shoulder. Assuming that the projection of the on-body origin on the screen's plane is the origin of our 2D coordinate system, the location of the pointer is described by absolute positioning. A specific angular orientation always corresponds to the same pointer coordinates in this case.

For the present application, only the yaw angle  $\psi$  and the pitch angle  $\theta$  are exploited. The pitch angle describes the vertical movement and the yaw angle describes the horizontal movement of the pointer as shown in Figs. 1(b) and 1(c). The vertical and horizontal coordinates of the pointer are referred

as  $y$  and  $x$  respectively. As the system uses only the arm orientation to define pointing, there is a need to initially define this constant location of on-body origin as well as to obtain the relative position of the screen. In order to achieve that, our system uses a training phase that is described in Section III. The screen dimensions are passed as inputs to the training procedure, which outputs an estimation  $\hat{D}$  of the distance between the user and the display. When training has been completed, the system is able to compute the 2D projected on screen position, where the arm is pointing at. The pointer position coordinates are calculated by the expression:

$$\text{Position}(\psi, \theta) = [\hat{x}, \hat{y}] = [\tan(\psi) \cdot \hat{D}, \tan(\theta) \cdot \hat{D}] \quad (7)$$

### III. EXPERIMENTAL METHODS

#### A. Hardware

In order to test the system proposed in this work, two independent sensor modules were deployed and compared. IMU-1 is an off-the-shelf sensor platform, while IMU-2 is a custom design implemented in our lab. The two sensors transmit data via the Bluetooth protocol and the characteristics of each individual MEMS sensor within these two solutions are summarized in Table I.

TABLE I  
CHARACTERISTICS OF INDIVIDUAL SENSORS

|                      | IMU-1                 |                       | IMU-2                 |                       |
|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|                      | Accelerometer         | Gyroscope             | Accelerometer         | Gyroscope             |
| <b>Dynamic Range</b> | $\pm 2 \text{ g}$     | $\pm 250 \text{ dps}$ | $\pm 2 \text{ g}$     | $\pm 125 \text{ dps}$ |
| <b>Sampling Rate</b> | $50 \text{ Hz}$       | $50 \text{ Hz}$       | $50 \text{ Hz}$       | $50 \text{ Hz}$       |
| <b>RMS Noise</b>     | $0.005 \text{ m/s}^2$ | $0.048 \text{ dps}$   | $0.081 \text{ m/s}^2$ | $0.100 \text{ dps}$   |

#### B. Calibration

MEMS accelerometers and gyroscopes usually suffer from biases, misalignments and gain errors. To be able to identify and reduce the effect of these errors, it is important to calibrate each sensor to the reference readings before feeding the data into the next stages. For that purpose and prior to data acquisition, we implement the calibration procedures proposed in [13] and [14]. The three-axis accelerometers were placed in six different positions and held stationary during each calibration measurement. Gyroscopes' calibration is conducted by comparing measured rotations to a reference  $180^\circ$  rotation. After completing the calibration measurements, the data are used in order to generate the gain matrices and biases vectors of each MEMS sensor used. For the calibration measurements, a Pan/Tilt PTU-D46 Unit was used to ensure the accuracy of the process.

#### C. Experimental Setup

The experimental setup used to evaluate our system is shown in Fig. 3. A 60" presentation screen was used along with a PTU-D46 Pan/Tilt Unit. The IMU pair was mounted on

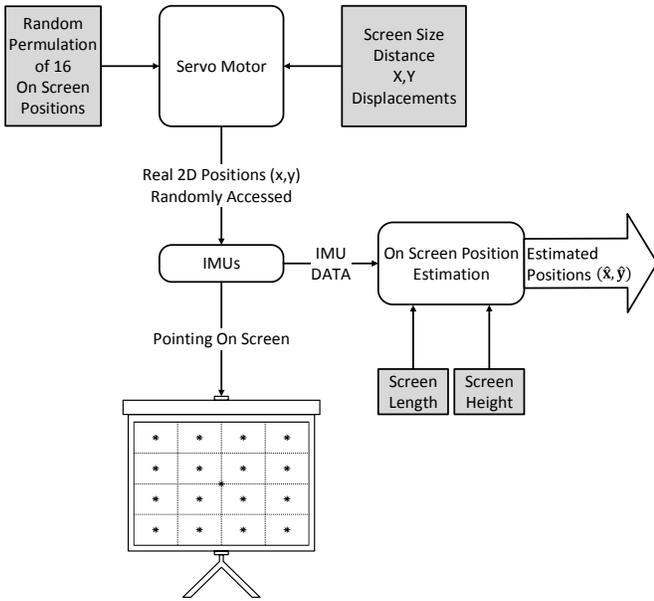


Fig. 3. Block diagram of the experimental setup.

the Pan/Tilt Unit in a way that resembles mounting on the human arm. Initially the configuration is oriented so that its longitudinal axis is perpendicular to the screen (Fig. 4). The program that handles the Pan/Tilt Unit movements receives as inputs the geometric characteristics of the setup -screen width and height, distance from the screen, displacement of the point of origin in regard to the center of the screen- and moves the configuration to point to sixteen on screen positions randomly accessed as shown in Figs 1(a) and 5. At the end of this task, the exact on screen positions pointed  $(x, y)$  are known and can be compared with the ones estimated  $(\hat{x}, \hat{y})$ . The experiment is taking place at six different distances of 1.5, 2.0, 2.5, 3.0, 3.5 and 4.0 meters and the results are shown in Section IV.

#### D. System Training Phase

As suggested in Subsection II-C, the estimation of the pointed position requires absolute knowledge of the distance between the user and the plane he is pointing at. This information is not initially provided to the system. Therefore, a training phase was designed to precede the measurements' procedure in order to provide the system with the distance information as well as the relative position of the screen. During the training phase the user has to point straight to the screen's plane in order to initialize the system and define the origin of the 2D coordinate system on this perpendicular plane. After that, he is required to point at the four edges of the display (Fig. 1(a)). As Fig. 4 shows such movement supplies the system with four angle measurements  $\theta_1, \theta_2, \psi_1$  and  $\psi_2$ , where  $\psi_1$  and  $\theta_1$  always satisfy that:

$$|\psi_1| > |\psi_2| \quad |\theta_1| > |\theta_2|$$

The training method assumes that the screen dimensions, width (W) and height (H), are known and provides an estima-

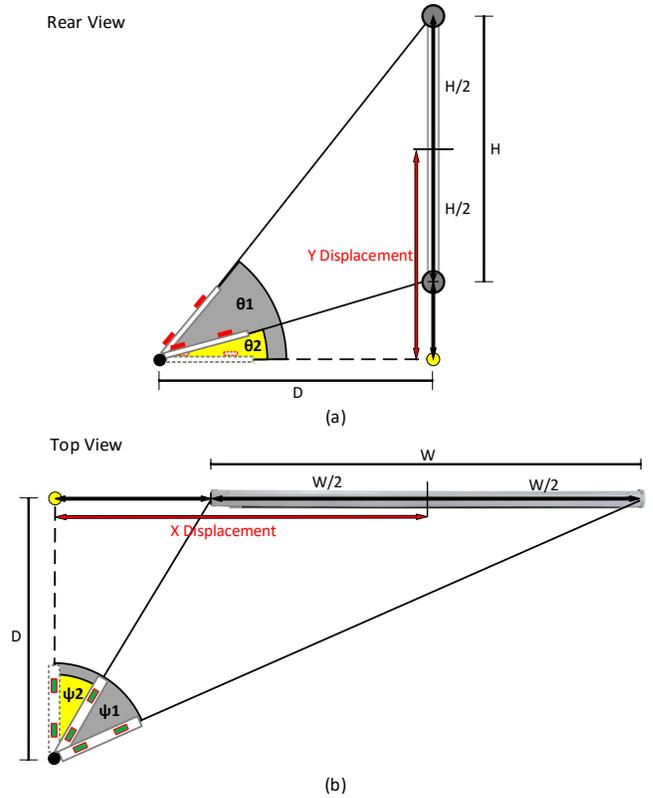


Fig. 4. System training phase: (a) Rear View (b) Top View

tion for the distance between the user and the screen. In addition, this process estimates the displacement of the screen's center regarding the point of origin of the 2D coordinate system as shown in Fig. 4. These X and Y displacements are distances from the display's center, therefore positive values. The sign of the four training angles can determine whether these displacements are positive or negative in respect to the coordinate system's origin. As the user is lying across the display the expected  $\psi$  and  $\theta$  angles are inside the limits of  $[-\frac{\pi}{2}, \frac{\pi}{2}]$ , where the function  $\tan()$  is increasing. Therefore, the aforementioned distance  $\hat{D}$  and displacements  $X, Y Displacement$  are calculated as following:

$$\hat{D}_\psi = \left| \frac{W}{\tan\psi_1 - \tan\psi_2} \right|, \quad \hat{D}_\theta = \left| \frac{H}{\tan\theta_1 - \tan\theta_2} \right| \quad (8)$$

$$\hat{D} = \frac{\hat{D}_\psi + \hat{D}_\theta}{2} \quad (9)$$

$$X Displacement = \frac{W}{2} \cdot \frac{\tan\psi_1 + \tan\psi_2}{\tan\psi_1 - \tan\psi_2} \quad (10)$$

$$Y Displacement = \frac{H}{2} \cdot \frac{\tan\theta_1 + \tan\theta_2}{\tan\theta_1 - \tan\theta_2} \quad (11)$$

A limitation that emerges from this approach is that a new training is required when significant change in the user's position occurs.

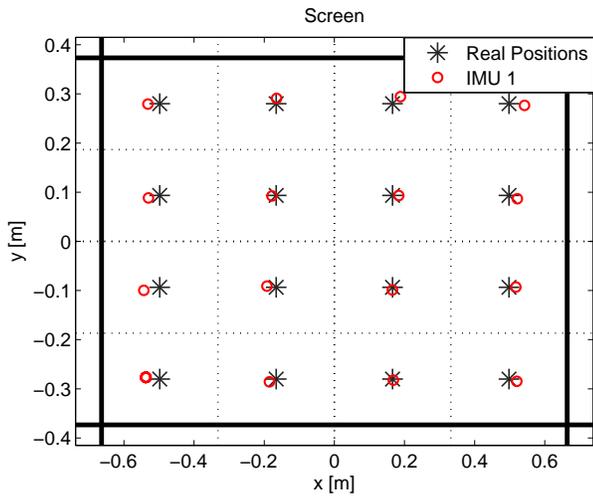


Fig. 5. Estimated On Screen Positions

#### IV. EXPERIMENTAL RESULTS

Previously we explained how the experiments were carried out. For each experiment the origin of the 2D coordinate system coincides with the center of the display. Fig. 5 shows the real on screen positions  $(x, y)$  and their estimations  $(\hat{x}, \hat{y})$  when the experiment was conducted at the distance of 1.5m.

For each position the euclidean distance error  $e_p$  as well as the x and y axis errors were computed as performance metrics:

$$e_p = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \quad (12)$$

$$e_x = |x - \hat{x}| \quad e_y = |y - \hat{y}| \quad (13)$$

In order to evaluate the system's performance in each distance we extract the mean position error of the sixteen accessed positions. The following results refer to readings acquired from IMU1. Fig. 6 shows these results when both the real and the estimated distances are used in equation (7). Therefore, it also presents an evaluation of the training procedure's distance estimation.

Fig. 7 shows the axis errors in respect to specific angles  $\psi$  and  $\theta$  that describe the same x and y coordinate respectively at all experiment distances. The angle changes observed in this figure is caused by the inverse changes of the distance occurring at different experiments. Fig. 7 clearly shows that the x axis error has the main contribution to the position error.

Using the experiment results depicted above we performed a second order curve fitting to extract the estimated position error function and the axis error functions (Fig. 6, 7). Utilizing these error functions we computed the estimated position errors in two cases. Table II presents the error estimations acquired from Fig. 6 when the user-display distance and the screen diameter increase by the same factor  $k$ . Table III presents the error estimations when when the user-display distance and the screen area increase by the same factor  $k$ . In this case, the angles  $(\psi, \theta)$  are different for every area-distance pair so the axis error functions of Fig. 7 are used to compute the estimated errors.

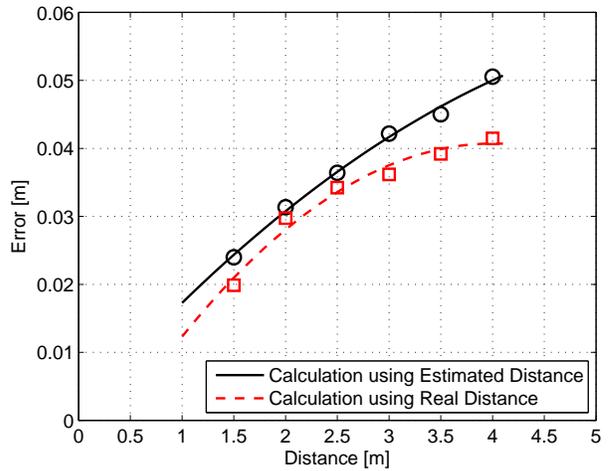


Fig. 6. Mean euclidean distance errors versus distance from screen

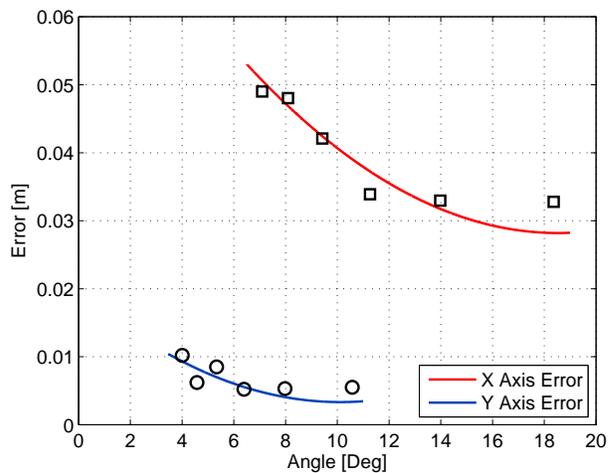


Fig. 7. Axis errors versus angles  $\psi, \theta$

Finally, Table IV shows a performance comparison of the two different IMU pairs used in our experiments.

#### V. DISCUSSION AND FUTURE WORK

Our system evaluation showed that it is capable of pointing at a specific on-screen target with a small position error that varies according to the distance from the screen. It was also shown that the training's accuracy in distance estimation has a small effect in the system's performance. The main contribution of this error is found in the estimation of the x-coordinate and to the estimation of the  $\psi$  angle. This observation can be explained by the fact that the attitude estimation filter in practice uses both gyroscope and accelerometer to sense the rotation around y-axis while regarding the rotations around z-axis the filter relies merely on gyroscope readings. The accelerometer can not provide corrections to this rotation as the Earth's gravitational field gives no information about it. As a future work, we consider adding a magnetic sensor to

TABLE II  
ESTIMATED ERRORS WHEN DISTANCE AND SCREEN DIAGONAL SIZE INCREASE PROPORTIONALLY

| k | Diagonal | Area [ $m^2$ ] | Distance [m] | Position Error [cm] | Relative Error [%] |
|---|----------|----------------|--------------|---------------------|--------------------|
| 1 | 60"      | 1              | 1.5          | 2.43                | 1.62               |
| 2 | 120"     | 4              | 3.0          | 4.17                | 1.39               |
| 3 | 180"     | 9              | 4.5          | 5.32                | 1.18               |
| 4 | 240"     | 16             | 6.0          | 5.89                | 0.98               |

TABLE III  
ESTIMATED ERRORS WHEN DISTANCE AND SCREEN AREA INCREASE PROPORTIONALLY

| k | Area [ $m^2$ ] | Distance [m] | X Axis Error [cm] | Y Axis Error [cm] | Euclidean Distance Error [cm] |
|---|----------------|--------------|-------------------|-------------------|-------------------------------|
| 1 | 1              | 1.5          | 2.82              | 0.34              | 2.84                          |
| 2 | 2              | 3.0          | 3.31              | 0.44              | 3.34                          |
| 3 | 3              | 4.5          | 3.83              | 0.58              | 3.87                          |
| 4 | 4              | 6.0          | 4.24              | 0.70              | 4.30                          |
| 5 | 5              | 7.5          | 4.57              | 0.79              | 4.64                          |

TABLE IV  
IMUS' PERFORMANCE COMPARISON ON A 60" SCREEN

| Distance [m] | IMU 1               |               | IMU 2               |               |
|--------------|---------------------|---------------|---------------------|---------------|
|              | Position Error [cm] | $\sigma$ [cm] | Position Error [cm] | $\sigma$ [cm] |
| 1.5          | 2.40                | 1.21          | 2.22                | 1.51          |
| 2.0          | 3.13                | 1.43          | 2.50                | 1.74          |
| 2.5          | 3.64                | 1.55          | 2.73                | 1.92          |
| 3.0          | 4.22                | 1.93          | 3.87                | 2.86          |
| 3.5          | 4.50                | 2.44          | 4.21                | 3.13          |

our system. This will provide an extra distinct vector related the magnetic north pole which is an absolute frame of heading reference.

## VI. CONCLUSIONS

In this paper we presented a wearable screen pointing system composed of MEMS accelerometers and gyroscopes.

We also presented a training process able to identify the relative position of the display and its distance from the user. The analysis was based on a custom experimental set-up and the results show that the proposed system results to a reliable pointing solution.

## REFERENCES

- [1] N. Sim, C. Gavriel, W. W. Abbott, and A. A. Faisal, "The head mousehead gaze estimation" in-the-wild" with low-cost inertial sensors for bmi use," in *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*. IEEE, 2013, pp. 735–738.
- [2] R. Raya, J. Roa, E. Rocon, R. Ceres, and J. Pons, "Wearable inertial mouse for children with physical and cognitive impairments," *Sensors and Actuators A: Physical*, vol. 162, no. 2, pp. 248–259, 2010.
- [3] H. Han *et al.*, "Novel multiple-functional imu-based wearable air mouse for the simultaneous operation with (air) keyboards," in *SENSORS, 2015 IEEE*. IEEE, 2015, pp. 1–4.
- [4] H. Debarba, L. Nedel, and A. Maciel, "Lop-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*. IEEE, 2012, pp. 125–132.
- [5] *Computer Controlled Pan-Tilt Unit Model PTU-D46*, Directed Perception, 2006.
- [6] H. Hyyti and A. Visala, "A dcm based attitude estimation algorithm for low-cost mems imus," *International Journal of Navigation and Observation*, vol. 2015, 2015.
- [7] S. O. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, 2010.
- [8] R. Mahony, T. Hamel, and J.-M. Pfimlin, "Nonlinear complementary filters on the special orthogonal group," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [9] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.
- [10] W.-z. Wang, Y.-w. Guo, B.-Y. Huang, G.-r. Zhao, B.-q. Liu, and L. Wang, "Analysis of filtering methods for 3d acceleration signals in body sensor network," in *Bioelectronics and Bioinformatics (ISBB), 2011 International Symposium on*. IEEE, 2011, pp. 263–266.
- [11] K. A. Mann, F. W. Wernere, and A. K. Palmer, "Frequency spectrum analysis of wrist motion for activities of daily living," *Journal of Orthopaedic research*, vol. 7, no. 2, pp. 304–306, 1989.
- [12] J. B. Kuipers *et al.*, *Quaternions and rotation sequences*. Princeton university press Princeton, 1999, vol. 66.
- [13] S.-h. P. Won and F. Golnaraghi, "A triaxial accelerometer calibration method using a mathematical model," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 8, pp. 2144–2153, 2010.
- [14] J. Rohac, M. Sipos, and J. Simanek, "Calibration of low-cost triaxial inertial sensors," *Instrumentation & Measurement Magazine, IEEE*, vol. 18, no. 6, pp. 32–38, 2015.