Design and Implementation of a Computer Mouse Using a 5-DOF Inertial Measurement Unit and A Sensor Fusion Algorithm

Daniel Ignacio Barbosa Tarazona
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
dibarbosat@correo.udistrital.edu.co

David Andrés Diaz Gómez
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
dadiazg@correo.udistrital.edu.co

Santiago Arévalo González
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
sarevalog@correo.udistrital.edu.co

***Abstract* – This paper presents the development of a computer peripheral device. This device implements the conventional mouse functions by sensing the user's hand movement in the air. To achieve this purpose, the device implements a modified Kalman Filter to calculate the position of the mouse in the Y-axis through the fusion of a 3-axis accelerometer and a 2-axis gyroscope. For the X-axis an integration algorithm of the horizontal movement gyroscope signal was implemented. A USB HID stack in the microcontroller is used to perform the communication with the host computer.**

Keywords- *Fusion sensor with Kalman Filter; Mice; gyroscope; accelerometer, user interfaces.*

I. INTRODUCTION

Mice are an integral part of the common usage of computer work. The technological advance of this input device has just presented changes in its electronic and mechanical concept: it introduced the scrolling concept and modified the computer connection according to the wire and wireless technologies, however its functionality has limited the user to a support surface to produce the movement of the pointer. Such way of utilization of the mouse provokes, according to medical research, diverse ergonomic problems; intensive mouse use has been associated with increased risk of upper extremity musculoskeletal disorders, including carpal tunnel syndrome [1], [2]. Moreover, medical research indicated that jobs with long periods of intensive mouse use may be at an increased risk of median mononeuropathy [3]. On account of the above, this research proposes a change of the mouse handle concept: As a final result the user is able to move the mouse pointer by maneuvering the device in the air. This solution could be applied in multiple applications such as: presentations, lectures, multi-monitor environment, among others. It allows the user to find an interesting and versatile experience with the device.

The recent technological development in electronic engineering area has reached breakthroughs in motion sensing. For the measurement of variables that imply the movement of a body: the variables referred to the cinematic and dynamic of an object, the electronic industry has developed the inertial measurement units, also known as IMU's (Inertial Measurement Unit). These sensors are able to measure variables such as acceleration, angular velocity and the orientation of a body in the space, these sensors are the accelerometers, the gyroscopes and the magnetometers, respectively. Applications of these sensors are very easy to implement due to the small size of the devices – they belong to the category of MEMS (Micro Electro-Mechanical Systems) –, these sensors facilitate the implementation of motion sensing characteristics in a device; features that before were measured by mechanical systems.

Numerous researchers have implemented different kind of human-machine interface using MEMS, Anala Pandit et al. [4] implemented an interface using accelerometers which have some drawbacks that will be mentioned below and developed a specific application to interact with the device. In addition, different approaches using different kind of IMU's to improve the performance of the interaction have been developed [5]. However, these developments have specialized on specific applications and they present lack of versatility.

The aim of this paper is the processing of the signals from a 5 DOF (Degrees of Freedom) IMU, composed by a three axis accelerometer and a two axis gyroscope, in order to emulate a typical PC mouse. Due to the hardware resources of the microcontroller's family 18FXX5X of Microchip, it is used an USB communication with the PC using this module with the HID stack.

The overall architecture of the proposed device is shown in Figure 1.

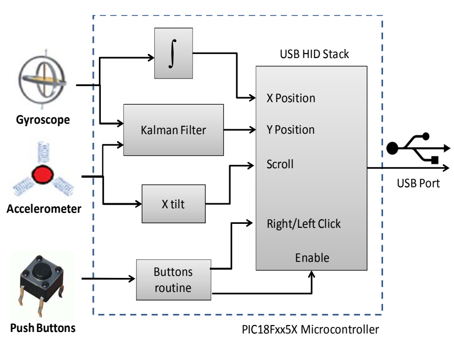

Figure 1 Block diagram of the designed device.

Section II details the sensor-fusion technique which is implemented to determine the Y position, taking as input the accelerometer and gyroscope data. The tilt of the device is determined by measuring the angular position with the gyroscope and the accelerometer obtains the tilt value of the device using a trigonometric function. Through the implementation of a Kalman filter it is estimated the angular position correcting the gyroscope data using the observations realized by the accelerometer in a similar way as done in [5], [6] and [7]. Section III exposes the solution of the calculation of the X position, there is proposed a gyroscope horizontal movement signal integration algorithm to compute the angular position. In section IV is presented the design and implementation of the USB controller, for the mouse wheel state it is used the accelerometer tilt data used for the Y axis. Finally, section V presents the final results of the device.

II. MEASUREMENT OF THE ANGULAR POSITION IN THE Y-AXIS

A. Sensor fusion

Since accelerometers cannot distinguish between acceleration due to movement or due to gravity, their outputs need to be filtered when are being used as a tilt sensor. However, filtering causes a sluggish response. On the other hand, if it is desired to get the angular position with the gyroscope signal, it is necessary to perform integration on the gyroscope signal. Nevertheless, the BIAS error (the output of the gyro when rotation is zero) leads to an error that increases with integration time [8], [9]. Therefore, it is necessary the sensor fusion of the accelerometer and gyroscope signal in order to: achieve better accuracy, improve motion sensing response and realize the gyro Bias compensation which avoid the drift effect due to the time and the sensor temperature [10].

The sensor fusion technique enables an appropriate fusion of the signals from these devices: the acceleration measured by the accelerometer and the angular velocity measured by the gyroscope, which allows to make use of the advantages of each device to estimate the parameters required by the final application: orientation and tilt measured by the angular position. The developed application implements a Modified Kalman filter that performs the sensor fusion technique.

B. Modified Kalman filter

The proposed Kalman Filter uses the gyroscope signal to obtain *a priori* angular position, it means a prediction of this value, and through the accelerometer signal the filter acquires the observation values in order to obtain an *a posteriori* value or estimation of the angular position as is shown in Figure 2.

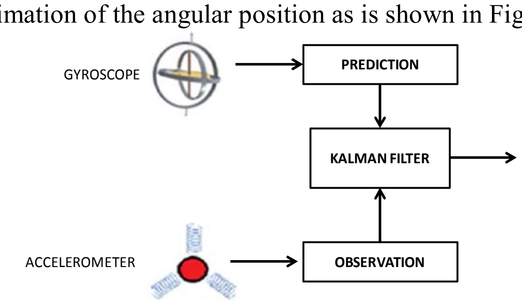

Figure 2 Sensor-fusion developed for the motion estimation in the application.

1) Filter initialization

Regardless of the space orientation model to use: Euler angles, quaternions or Direct Cosine Matrix, the Kalman filter provides the prediction of the gyroscope angular velocity and calculates the variation throughout every sampled time step.

The output of the Kalman Filter is given by the integration of the prediction and the observations done by the accelerometer. The state vector is:

$$x_k = [\theta, g_{bias}] \tag{1}$$

Where:
θ: is the angular position computed by the gyroscope, as follows:

$$\theta = q * dt \tag{2}$$

$$q = gyroscope_{data} - g_{bias} \tag{3}$$

Where q is the difference between the current angular velocity and the estimation in the last state.
$gyroscope_{data}$: is the angular velocity received by the gyroscope in the present state.
g_{bias}: is the estimation of the angular velocity in the last sampled time.
Therefore, the estimated state vector is:

$$\dot{X_k} = [\dot{\theta}, \dot{g}_{bias}] \tag{4}$$

Where:
$\dot{\theta}$: is the estimated angle after the Kalman filter process.
$\dot{g}_{bias}$: is the estimation of the angular velocity in the last sampled time after the Kalman filter process.
The noise process covariance matrix, Q , and the noise observations matrix, R, were estimated through lab analysis. This analysis was done using MathWorks Real-Time Windows Target Simulink Toolbox. The data computer acquisition was done with a serial 57,6kbps transmission to the computer. The tests results demonstrated that the expected noise, or jitter in the accelerometer measure is 0.3rad (approximately 17.18°), therefore:

$$R = 0.3 \tag{5}$$

$$Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.003 \end{bmatrix} \tag{6}$$

Q, represents the relation between the accelerometer data and the gyroscope data, so it means how much can the system trusts in accelerometer in contrast with the gyroscope.
The covariance matrix,P, is also initialized with:

$$P0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

2) Prediction

The state transition matrix F is obtained by the Jacobian of the state vector $X_k = [\theta, g_{bias}]$:

$$F = \begin{bmatrix} \frac{\partial(\dot{\theta})}{\partial(\theta)} & \frac{\partial(\dot{\theta})}{\partial(g_{bias})} \\ \frac{\partial(\dot{g}_{bias})}{\partial(\theta)} & \frac{\partial(\dot{g}_{bias})}{\partial(g_{bias})} \end{bmatrix} \quad (8)$$

This matrix represents the evolution of the system through the time. Given the relation between the variables in the state vector, this matrix could be simplified by solving the partial derivates of the Jacobian, as follows:

$$F = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \quad (9)$$

According to the Kalman Filter theory, the covariance matrix is given by:

$$P_K^- = F_{K-1} P_{K-1} F_{K-1}^T + Q_{K-1} \quad (10)$$

The system updates the estimated angle and the covariance matrix, as follows:

$$\dot{\theta} = \dot{\theta} + q * dt \quad (11)$$

Where $\dot{\theta}$ is initialized to zero.

3) Observation

Once the prediction is done, the proposed system acquires the observations by the accelerometer in order to determine the error estimation: the difference between the computed angles by the accelerometer and the gyroscope signal, which is given by:

$$\theta_{err} = \theta_{measured} - \dot{\theta} \quad (12)$$

Where:

$$\theta_{measured} = atan2\left(-\frac{Az}{Ax}\right) \quad (13)$$

Az = acceleration measured by the accelerometer in the Z-axis
Ax = acceleration measured by the accelerometer in the X-axis

Then the system must calculate the observation matrix, this matrix is given by the Jacobian of the observations done by the accelerometer respect to the state vector, so that the observation matrix H is:

$$H = \left[\frac{\partial(\theta_{measured})}{\partial(\theta)} \; \frac{\partial(\theta_{measured})}{\partial(g_{bias})}\right] \quad (14)$$

Since the relation between the measurements given by the accelerometer and the state vector, the observation matrix is:

$$H = [1 \;\; 0] \quad (15)$$

The final step of the algorithm is the Kalman filter gain calculation. The Kalman gain is calculated by parts:

$$K = PH^{'}E^{-1} \quad (16)$$

Where:

$$E = HP_K^- H^T + R \quad (17)$$

Is the estimated error computed by the Kalman filter algorithm [11]. It is computed based on the accelerometer observations and the expected noise in its signal.

Finally, the covariance matrix is updated, based on the Kalman gain, as follows:

$$P_k = (I - KH)P_k^- = P_k^- - KHP_k^- \quad (18)$$

And the estimated state is updated:

$$\dot{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (19)$$
$$\dot{x}_k^+ = \dot{x}_k^+ + \theta_{err} * K \quad (20)$$

Thus,

$$\dot{\theta} = \dot{\theta} + K0 * \theta_{err} \quad (21)$$
$$\dot{g}_{bias} = \dot{g}_{bias} + K1 * \theta_{err} \quad (22)$$

III. INTEGRATION ALGORITHM FOR THE X-AXIS POSITION OF THE DEVICE.

The Kalman Filter method is successful to obtain Y axis tilt, unfortunately it can not be used for the X axis due to the fact that there is not a change in gravity vector, as a natural reference, when a horizontal movement is done. Therefore there is no change in the tilt of the device. Consequently, in order to sense the horizontal movement the data integration from the GY-axis gyroscope is made.

Data filtering of the gyroscope signal is done by a moving average filter, which is explained as follows:

$$mov_{gy} = [a\ b\ c\ d\ e] \quad (23)$$

The designed moving average filter takes 5 samples, $[a\ b\ c\ d\ e]$ – which are continuously moving on the time –, of the signal and then divides the addition of these samples by 5. So that, it gets a low pass filtered value of the gyroscope signal. Therefore, sum saves the average of the taken samples, as follows:

$$sum = (a + b + c + d + e)/5 \quad (24)$$

The gyroscope BIAS generates a drift effect, this BIAS depends on the temperature of the sensor and produces a reference loss and, as a result, a meaningless angle information data. According to the IDG500 gyroscope technical datasheet, after 5 minutes of turned on, this value become constant [12]. To resolve this problem it is performed a calibration routine, where the first 5000 samples are averaged when the device starts, the average result is stored in a variable that is continuously subtracted from the current gyroscope signal, in the following way:

$$sum = \frac{a+b+c+d+e}{5} - calibrate \quad (25)$$

As previously mentioned, the BIAS depends on the temperature, which means that eventually, the BIAS value will differ from the value stored and it will produce a drift as in figure 3. In this case it is necessary to restart the device to repeat the routine calibration.

sum is given in angular rate units (°/s), the conversion to angle is done multiplying the filtered value by the sensitivity, as follows:

$$sum = sum * (sensitivity) \tag{26}$$

where: $$sensitivity = \left(\frac{scale}{resolution}\right) \tag{27}$$

$$\theta_x = \theta_x + sum * dt \tag{28}$$

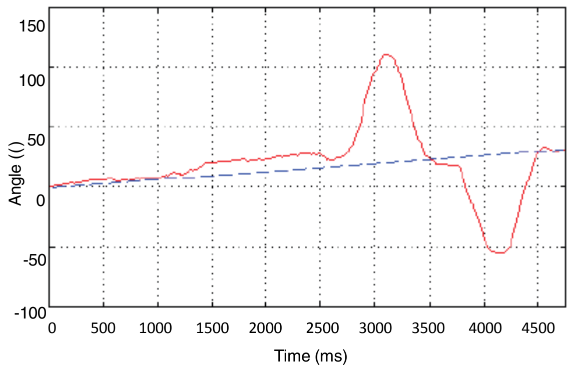

Figure 3. Drift error produced by the gyroscope BIAS signal

In the figure 4 the blocks diagram is shown.

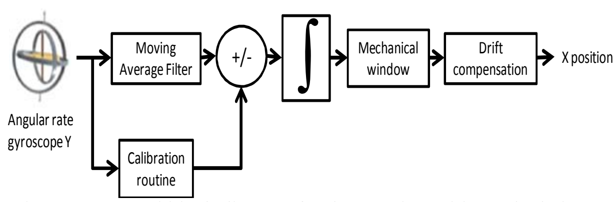

Figure 4. General block diagram for the X-axis position calculation.

IV. USB

As previously mentioned, the device communication with the computer is done using USB. Taking into account the connectivity options of the USB stack, it was decided that the more appropriate option for the application was the utilization of a microcontroller with embedded USB controllers. This choice was made because of the use of USB-serial and USB-parallel converters would require the design of a specific *HID* controller, which already exists in the embedded device: the mouse controller. Moreover, it should be mentioned that the other USB configurations are not totally independents, so that they require a microcontroller for the protocol management.

Using the CCS C Compiler, it is implemented the function *usb_put_packet()*. This function sends a package to the computer USB port as follows: *usb_put_packet(1, out_data, 4, USB_DTS_TOGGLE)*. Where: *out_data*: Is a four-element array: In the first part sends the right and left click state buttons. In the second one, it is sent the X-axis movement. In the third part, it is sent the Y-axis movement. And finally it is sent the wheel mouse state.

A. Polling Rate

In Microsoft Windows, keyboards, mouse and joysticks, plugged via USB are commonly configured with a sample frequency of 125Hz [13]. In order to follow this reference, the designed application uses the same transmission frequency.

B. Dots Per Degree (DPD)

A conventional mouse measures the resolution of the user's movement in DPI (Dots Per Inch). It gives a relation between the user's movement and the movement of the pointer on the screen. In the performed device, due to the angular changes produced by the user when produces motion, it is required to change the reference for that resolution. Hence, this device uses DPD (Dots Per Degree). This measurement unit relates the movement on the screen – in pixels – with the angular movement produced by the user. The lab tests gave the result that 15DPD offers the best user experience with the designed mouse.

C. Scroll design

Normally, a mouse scroll is an electromechanical device which converts angular position or an axial movement into a digital codification. This codification allows the mouse to determine the direction and velocity of any rotation axis [14].

The performed device replaces the conventional mouse scrolling with the angle information provided by the accelerometer, as in equation (13).

TABLE 1. ENCODER CODIFICATION FOR THE DESIGNED SCROLL.

ANGLE BETWEEN AXIS X AND Y	INFORMATION FROM ENCODER	MEANING
80<ANGLE<100	[1 0 1]	Slow movement upwards
100<ANGLE<150	[0 1 0]	Rapid movement upwards
-100<ANGLE<-80	[-1 0 -1]	Slow movement downward
-150<ANGLE<-100	[0 -1 0]	Rapid movement downward

For the application, these two speeds were smooth and intuitive for the user.

V. RESULTS

A. Kalman filter implementation results

All the tests were performed with an IMU Analog Combo Board- 5 Degrees of Freedom IDG500/ADXL335 distributed by Sparkfun [12]. All tests showed a 275ms initialization time.

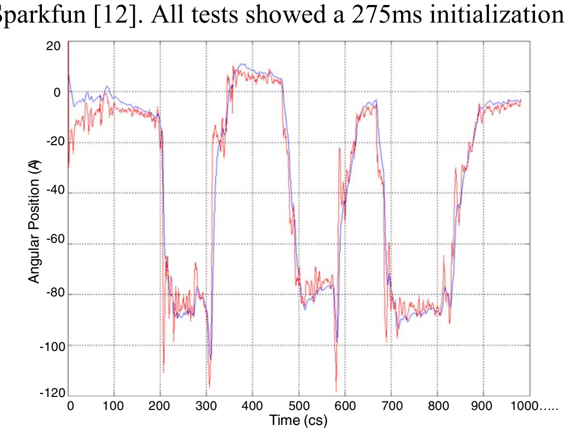

Figure 5. Angular position behavior with and without the Kalman filter. The red signal shows the angular position without the Kalman filter, it represent

the angular position computed by the accelerometer as given in equation (13). The blue signal shows estimation of the angular position with the Kalman filter.

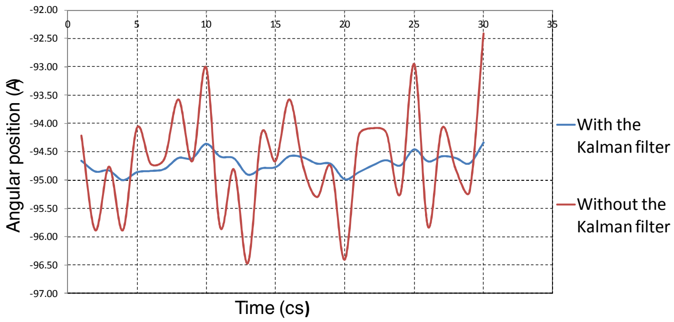

Figure 6. Angular position behavior in a fixed position of -90° with and without the Kalman Filter implementation.

As a consequence of the results of the figure 6, it stands the following parameters:

TABLE 2. COMPARISON OF PARAMETERS BETWEEN THE NON-ESTIMATED AND THE ESTIMATED ANGULAR POSITION WITH THE PROPOSED KALMAN FILTER. (*) THE NEGATIVE VALUE IS GIVEN BECAUSE OF THE SIGN OF THE ANGLE.

With the Kalman Filter			
Standard deviation	**Mean**	**Variance**	**S/N(*)**
0,161335328	-94,71	0,02602908	-587,007
W/O the Kalman Filter			
Standard deviation	**Mean**	**Variance**	**S/N(*)**
0,983753187	-94,64	0,9677703	-96,2023

B. *Integration algorithm results*

In Figure 7 the output (blue line) from gyroscope is shown, it presents the results of a movement sensed and measured by a 10-bit ADC (Analog – digital converter) from a PIC18F2550 microcontroller, in this way the measurement range is between 0 and 1024. Each bit is corresponds to 3.22mV.

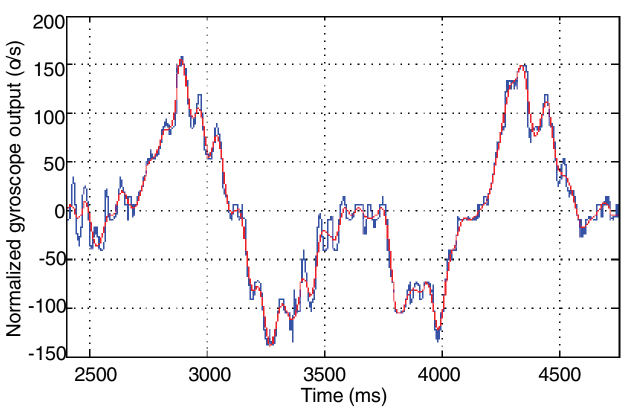

Figure 7. Gyroscope signal (blue) original (red) Filtered gyroscope signal with a moving average filter.

The red line in the figure 7 is the result of a digital low pass moving average filter designed to smooth the signal.

When a non-movement condition is presented, gyroscope signal noise that was not filtered by the moving average produces a wrong reading of the position due to the integration of these non-zero values. In an ideal case of a non-movement condition all samples must be zero. Therefore a mechanical-windows filter was applied. It converts to zero all the angular rate values that are between -0.3°/s and 0.3°/s. These values were found through lab test, looking for the highest stability of the pointer in a non-movement condition.

Due to the fact that the final application requires a difference between two measurements: angles from the previous sample and the present sample, it is not necessary a highly accurate measurement of the angle. However, it is important that the drift effect does not be perceptible by the final user, to achieve this, the positive and negative data are multiplied by a different scalar number to minimize the derive effect. Even though it is not the better choice, it works well and allows an intuitive and smooth performance for the user.

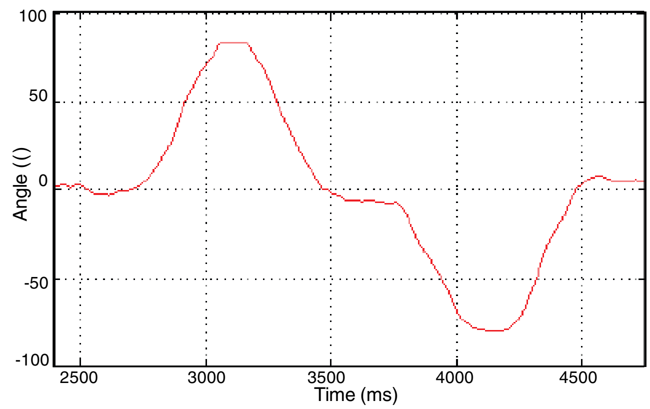

Figure 8. Calculation of angular position by the integration

C. *Implementation*

This section shows the physical handle of the device by the user in order to produce movements in the mouse pointer.

The figure 9 shows the way as the user moves the designed mouse without causing forced and uncomfortable positions in the hand's user.

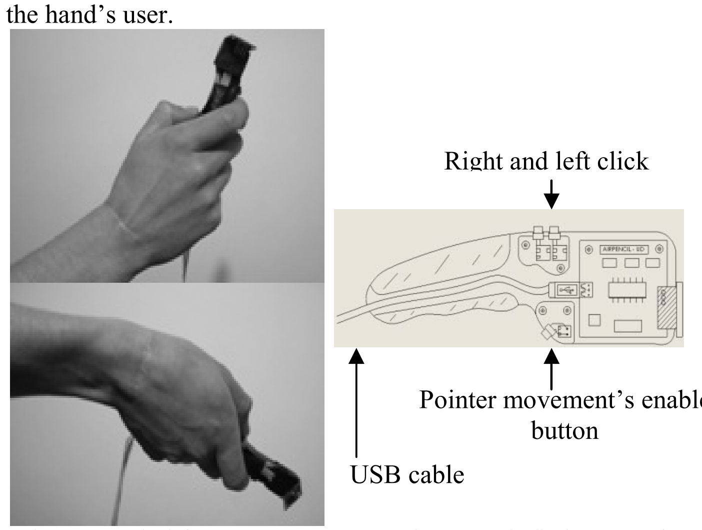

Figure 9. On the left, user's movement to produce a Y axis displacement of the pointer. On the right, it is shown the proposed mechanical design of the device (conceptual).

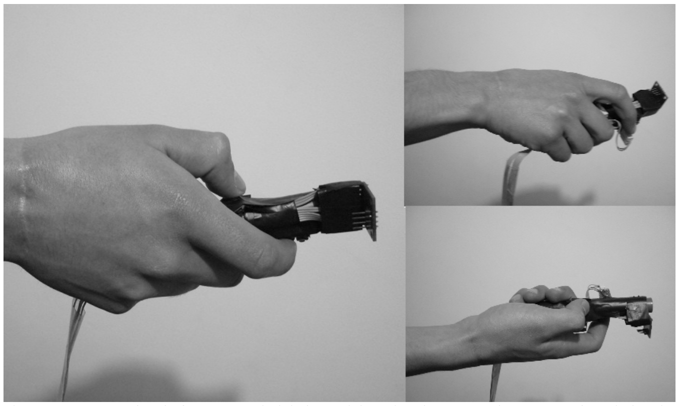

Figure 10. On the left, user's initial position is shown. Above on the right, is presented the rotation of the device to perform scroll upwards movement. Below on the right, rotation of the device to perform scroll downwards movement is illustrated.

VI. CONCLUSIONS

The gyroscope and accelerometers integration in devices aimed to permanently capture movements with high accuracy in the variables measurement (orientation, tilt, position, gravity, among others) has became indispensable. This is due to the fact that accelerometers cannot distinguish between acceleration due to the movement or due to gravity, and, in the gyroscopes the integration of a rotational rate with a miscalibrated BIAS results in a derive effect that increases the error in the measurement trough time. Moreover, the positive and negative angular rate values are not symetric, this increseas the problem. However, if it is obtained angular position from gyroscope and is corrected with a calculated tilt value from accelerometer, it can be obtained a correct answer, this answer is meets the device requirements.Therefore, it is used the advantage that the gyroscope is able to measure rotational movements independently form the gravity and the accelerometer allows to measure tilt using a trigonometric function.

In this paper, the Kalman filter allows to execute the sensors fusion, because its adaptive nature allows the system to estimate the angular position. The angular position is corrected with the accelerometer observations.

When comparing the Y-axis behavior (based in modified Kalman filter) and X-axis position (based in integration algorithm), Y-axis shows an stable response and highly accurate, according to pixels per degree (DPD) behavior.

It is possible to think in the computational intelligence tools for to resolve the following and estimation problem[15].

Aditionally, a future version could have a BIAS compensation through the temperature, in similar way than in [16]. In view of the IDG500 gyroscope has a temperature pin, and this implementation is directly.

In addition, the user's experience could be improved, if it was implemented a wireless technology.

REFERENCES

[1] *Effects of computer mouse design and task on carpal tunnel pressure,* Peter J. Keir, Joel M. Bach & David Rempe [ed]. Ergonomics The Official Journal of the Institute for Ergonomics and Human Factors. Volume 42, Issue 10, 1999 pages 1350-1360. 0014-0139 (Print), 1366-5847 (Online).

[2] *Risk factors for musculoskeletal disorders among computer users.* Tittiranonda P, Burastero S, Rempel D. Occup Med 1999, 14: 17-38.

[3] *Computer use increases the risk of musculoskeletal disorders among newspaper office workers.* Ortiz Hernandez, Luis, Tamez-González, Silvia, Martínez-Alcántara, Susana and Méndez-Ramírez, Ignacio. Archives of Medical Research. Volume 31, Issue 4, July-August 2003, Pages 331-342. 0188-4409.

[4] *A simple wearable hand gesture recognition device using iMEMS* Anala Pandit et al. [ed.] 2009 International Conference of Soft Computing and Pattern Recognition 978-0-7695-3879-2/09.

[5] Comotti Daniele y Michele Ermidoro. Progetto di Microelettronica Sviluppo di algoritmi per la stima dell'orientamento di un sensore inerziale. (2011)

[6] Sachs, David Matthew. A Forearm Controller and Tactile Display: Submitted to the Program in Media Arts and Sciences for the degree of Master of Science in Media Arts and Sciences. MASSACHUSETTS INSTITUTE OF TECHNOLOGY (Septiembre 2005).

[7] *A sensor fusion algorihm for an integrated angular position estimation with inertial measurement unit.* Sabatelli, Simone, y otros. [ed.] University of Pisa and SensorDynamics AG. Pisa : IEEE Xplore, 14 de March de 2011, Design, Automation & Test in Europe Conference & Exhibition, pages. 1-4. 978-1-61284-208-0.

[8] Vargha, Dough and Michael Maia. An Overview of Motion Processing Solution for Consumer Products. InvenSense, Inc.

[9] Retana, David and Aldo Vargas. Proyecto para obtener el título de Ingeniero mecatrónico: Estabilización de un helicóptero a escala mediante sistemas neuro-difusos. UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO Pp 23 (2010)

[10] ST Microelectronics. TA0343 Technical article: Everything about ST Microelectronics' 3-axis digital MEMS gyroscopes. July 2011.

[11] Kalman, R.E., A new approach to linear filtering and prediction problems, Transactions of the ASME, Ser. D, Journal of Basic Engineering, 82, 34–45(1960).

[12] http://www.sparkfun.com/products/9268

[13] Mellinger, Jürgen. *A Practical Guide to Brain-Computer Interfacing with BCI2000.* s.l. : Springer-Verlag. pág. 152. 978-1-84996-092-2.

[14] TechTools. PIC Application Notes 4:Reading Rotary Encoders. p. 23.

[15] *Comparación de la Red Neuronal y del Filtro de Kalman en la Estimación de Velocidad del Motor de Inducción.* González, Jaime. s.l. : IER CONGRESO IBEROAMERICANO DE ESTUDIANTES DE INGENIERÍA ELECTRICA, 2004.

[16] *Compensation of temperature drift of MEMS gyroscope using BP neuronal network.* Zhang, Qintuo, Tan, Zhenfan and Guo, Lidong. [ed.] College of Automation Harbin Engineering university. Nantong Street 145, Harbin, Heilongjiang,China : IEEE Xplore, 19 de December de 2009, Information Engineering and Computer Science, 2009. ICIECS 2009., pages. 1-4. 978-1-4244-4994-1.