A NOVEL SOFTWARE FRAMEWORK FOR 3D MOTION ESTIMATION WITH A 6-DOF MEMS IMU FOR A VIRTUAL REALITY ENVIRONMENT

Cem Küççük, Ersin Özkan, D. Turgay Altılar
Department of Computer Science and Institute of Informatics,
Istanbul Technical University, 34469, Maslak, Istanbul, Turkey
{kuccuk, ozkaner, altilar}@itu.edu.tr

ABSTRACT

The design of a novel framework for estimation and tracking the motion of a handheld object in 3D space for virtual reality environments is explained in this paper. Low-cost MEMS (Micro-Electro-Mechanical Systems) IMU (Inertial Measurement Unit) have been used in the system design. The system hardware includes a three axis accelerometer and a three axis angular rate sensor. By using simply two sensors, a 6 DOF (Degrees of freedom) dynamic system is designed. Although estimation and tracking the motion of a gamepad share the same principles with estimation and tracking the motion of an aerial/land vehicle essentially, the significant differences rely on their specific motion types. In the first application area the change in the orientation and velocity are bigger than the ones in the second application area. In this paper, an environment for a gamepad application is designed based on a fundamental approach to 3D motion sensing that covers both application areas. To complete the system design, a low-pass filter module, a transformation matrix based orientation calculation module, a discrete-time wiener process acceleration model based velocity and position estimation by using a Kalman filter module and a motion ending process for dynamic system module are designed as software modules. This system is developed as a part and parcel of the main project called MODEMOF (*MO*tion *D*etection *E*stimation and *MO*deling *F*ramework) which cooperates with another hardware module developed for the same project.

Keywords: *3D Orientation, 3D Position, 6-DOF, MEMS, IMU, Motion Estimation, Kalman Filter, Virtual Reality*

1. INTRODUCTION

Motion detection hardware has become very popular in many research areas because of the recent developments in MEMS technology. On the other hand detection, estimation and modeling of a 3D motion of an object using such MEMS technology yield innovative applications in many areas spanning from industry-automotive to entertainment, defense to medical/health-care. MEMS technology has been providing user with a wider area of application by producing smaller sized and lower-cost sensors. Since in order to make a project successful the cost of the systems has to be affordable [8], the approach of using low-cost sensor to design a high-end product is preferred. MEMS-IMU sensors have widely used in guidance researches; especially unmanned aerial/land vehicles. Unfortunately, such low-cost sensors pose the problem of the stability of outputs. Thus the integration of the IMU outputs must be specially designed to improve the performance of the system. Once sensors have been used in unmanned guidance systems MEMS technology is applied to different research areas. Motion sensing units have become a very popular hardware for entertainment and virtual reality fields. Virtual reality is a term that applies to computer-simulated environments that can simulate places in the real world as well as in imaginary worlds. Human-computer interaction technology, which is closely related with the virtual reality, aims to improve interaction between users and computers. In this paper, the framework for a low-cost handheld device with high accuracy for virtual reality applications is explained. Besides some specialized applications with different approaches such as pattern recognition, determination of 3D orientation and position of the object can be considered as the common problem. The framework is developed to sort out this common problem.

2. 3D MOTION ESTIMATION

An accelerometer measures proper acceleration. As explained in the relativity theory, proper acceleration is the physical acceleration experienced by an object relative to freefall [4]. The accelerometer data is a combination of static acceleration (acceleration of gravity) and dynamic acceleration (acceleration of sensor motion). Such

ISTEC 2010

accelerations are popularly measured in terms of g-force. Drift, rolling, vibration, linear velocity and position of an object can be estimated by using a multiple axis accelerometer sensor.

A gyroscope is a device for measuring or maintaining orientation, based on the principles of conservation of angular momentum. The gyroscope data is the rate of change of angular displacement for a specific axis [7]. Orientation of an object can be estimated by using a three-axis gyroscope sensor.

There are two types of IMU modules according to its layout characteristic; gimbaled and strapdown systems. In a strapdown system, sensors are entirely fixed to the body in motion, so IMU axes chance during the motion and conversion of body coordinates to navigational coordinates is needed. In a gimbaled system IMU axes stay same by using a specific mechanical design. A gimbaled system's primary advantage is it's inherently lower error. Since its three orthogonal accelerometers are held in a fixed inertial orientation, only the vertically oriented one will be measuring gravity. The strapdown inertial navigation system's main advantage over the gimbaled system is the simplicity of its mechanical design [9].

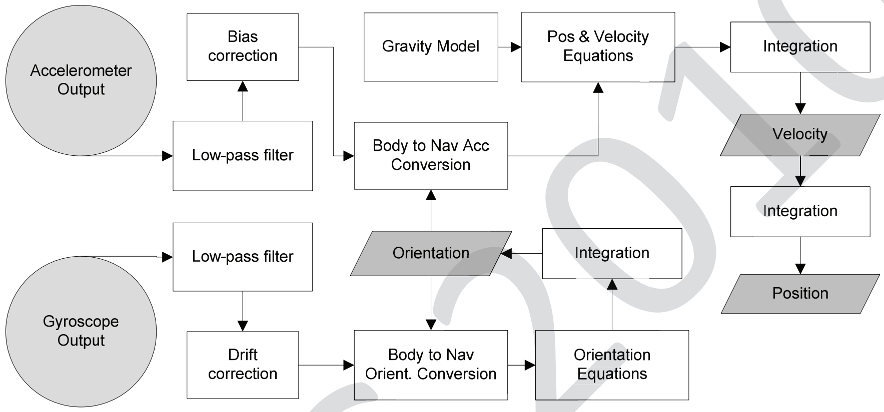

Figure 1: 3D Motion Estimation Structure

The developed general approach of 3D motion estimation of an object with a 6 DOF strapdown IMU is given in Figure 1. (Shaded circles are sensor outputs and system inputs, shaded parallelograms are system outputs) The first step for both system inputs is filtering the high frequency noise of raw accelerometer and gyroscope data individually. Then, orientation of the object must be estimated using gyroscope data. The system use previous orientation to convert angular rate outputs to navigational coordinates. The integration of these converted velocities produces the new orientation. After that, acceleration axes must be converted to navigational axes by using the orientation of object and acceleration of gravity must be extracted. For all axes, first integration of the acceleration gives the velocity and second integration of calculated velocity gives the position of object. Kalman filtering is used for the estimation of the explained complex dynamic system. The motion estimation approach is similar to inertial navigation system of an unmanned aerial vehicle, but has important differences according to application area. A motion based 3D game controller is aimed to develop and the sensor specifications change for this purpose. The sensors have to measure rapid movements and there is a tradeoff between higher measurement ranges and stable/accurate outputs.

2.1. Filtering High Frequency Noise

Noise filtering module is important for low-cost MEMS motion sensors; especially accelerometers. The essential issue for this filter module is to reduce high frequency noise without losing the real motion. Averaging filters can be used as low-pass filters for sensor outputs. A filter that averages the last five values of sensor output for each axis is used as a low-pass filter. The result of the low-pass filter on a shifting (by rolling) state is given in Figure 2 (Dotted line is low pass filtered signal).

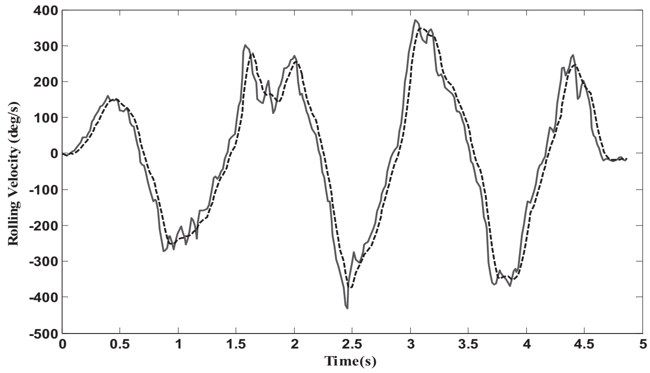

Figure 2: Low-pass Filter

2.2. Orientation Estimation

For a strapdown system, gyroscope sensors measure the angular velocities in the body frame. Even after a small orientation change, body frame of the object changes and new angular velocities need to be converted to navigational frame by using the current orientation.

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} nav = B \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} body \tag{2.1}$$

($\omega_x, \omega_y, \omega_z$: angular rates around axes, nav : navigational coordinate axes, $body$: body(object) coordinate axes)

The sensor system outputs are obtained in body coordinates and the 3D motion of the object is defined in navigational coordinates.

"Small Angle Approximation" can be used for the kinematic relationship between angular velocities and the rates of changes of the Euler angles [1];

$$B = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{2.2}$$

(θ : pitch angle, ϕ : roll angle)

Orientation of an object is represented as pitch-roll-yaw angles. For the angle based approach above, firstly these angles are calculated then coordinate transformation matrix is generated by using these angles.

The coordinate transformation matrix C_b^u is a 3×3 matrix. It is used to transform a vector from one set of resolving axes to another, the lower index represents the body frame and the upper index indicates the navigational frame [5].

$$C_b^u = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\omega\sin\theta\sin\phi + \cos\psi\cos\theta & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2.3}$$

(ψ : yaw angle, θ : pitch angle, ϕ : roll angle)

Instead of the angle based orientation representation, transformation matrix based approach can be used. The matrix is representing the rotation from the body (object) frame to the reference frame. Pitch-roll-yaw angles can be obtained by reverse transformation [5];

$$\begin{aligned} \phi &= \arctan 2(C^u_{b3,2}, C^u_{b3,3}) \\ \theta &= -\arcsin(C^u_{b3,1}) \\ \psi &= \arctan 2(C^u_{b2,1}, C^u_{b1,1}) \end{aligned} \tag{2.4}$$

($C^u_{b1,3}$: first row - third column value of the transformation matrix)

ISTEC 2010

For this orientation representation, firstly the coordinate transformation matrix is calculated. A relation between the angular rate vector and coordinate transformation matrix is needed for this operation. The skew-symmetric matrix of the angular rate vector is also commonly used [5];

$$\eta = \begin{pmatrix} 0 & -\omega_z & -\omega_y \\ \omega_z & 0 & -\omega_x \\ \omega_y & \omega_x & 0 \end{pmatrix} \tag{2.5}$$

There is such relation between C_b^u and the angular rate vector [6];

$$C_b^u(t+1) = C_b^u(t)(I_3 + \eta\Delta t) \tag{2.6}$$

($C_b^u(t)$ and $C_b^u(t+1)$: the consecutive coord. trans. matrixes, I_3 : 3x3 identity matrix, Δt : elapsed time)

$(I_3 + \eta t)$ part of the equation (2.6) can be written as;

$$(I_3 + \eta t) = \begin{pmatrix} 1 & -w_z\Delta t & -w_y\Delta t \\ w_z\Delta t & 1 & -w_x\Delta t \\ w_y\Delta t & w_x\Delta t & 1 \end{pmatrix} \tag{2.7}$$

2.3. Velocity and Position Estimation

In this part, state of the system and necessary inputs for the Kalman Filter algorithm in estimation process are explained. Welch and Bishop state that Kalman filter is a set of mathematical equations that provides an efficient computational means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown. [10]. Details of the algorithm is not explained; only necessary definitions for the filter is given.

For a strapdown system; after orientation estimation finished, the accelerometer outputs need to be converted from body frame to navigational coordinates by coordinate transformation matrix [3].

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} nav = C_b^u \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} body \tag{2.8}$$

(a_x, a_y, a_z : accelerations on axes, nav : navigational coordinate axes, $body$: body(object) coordinate axes)

After this conversion, acceleration of gravity is extracted from z axis output and three axes accelerometer output on navigational coordinates are ready for estimation of velocity and position. The state of system for one axis can be written as;

$$x : \{x_p, x_v, x_a\} \tag{2.9}$$

(x_p, x_v, x_a : position, velocity and acceleration)

Discrete-time Wiener Process Acceleration Model is used for system design;

$$x_k = Ax_{k-1} + Lw_{k-1} \tag{2.10}$$

(x_{k-1}, x_k : consecutive system states, w : process noise)

$$A = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} L = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix} \tag{2.11}$$

In this model, process noise (w_k) is the accelerometer error on kth sampling time and considered as zero-mean white noise. The covariance of the process noise is;

$$Q = L\sigma_w^2 L' = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^3 & T^2 & T \\ \frac{1}{2}T^2 & T & 1 \end{bmatrix} \sigma_w^2 \tag{2.12}$$

For the Q matrix, σ_w value is decided according to deviation of sensor data. It is practically determined as half of the largest possible deviation [2].
For the measurement equations of kalman filter;

$$z_k = Hx_k + Gv_k \tag{2.13}$$

(z : measurements of system, v : measurement noise)

$$H = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.14}$$

The only measurable output of the system state is acceleration data, so the z vector includes only acceleration value. In this model, measurement noise (v_k) is the possible difference between the real acceleration value and measured acceleration value on kth sampling time and considered as zero-mean white noise. The covariance of the process noise is;

$$R = G\sigma_v^2 G' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \sigma_v^2 \tag{2.15}$$

σ_w and σ_v values are decisive factors for process and measurement noise and decided by considering off-line sensor data observations and on-line filter performance.
This one-axis approach is implemented for all axes and 3D velocity and position estimation is obtained.

2.4. Motion Ending

Requirement of motion ending operation is related with objective usage of the sensor. When the motion sensor that is not designed for rapid orientation and velocity changes is used in a 3D game application, users' rapid movements generally force system to malfunction. This can be experimented with most of the inertial navigation systems. After user finished the action, errors of the system; especially linear velocity values are not exactly returned back to zero. Thus the system assumes that the motion is going on. That is why the motion ending process is required for this system.
The accelerometer output is used for the motion ending process. When the IMU is in motionless state, magnitude of the acceleration vector is nearly 1g. Also after converting body acceleration vector to navigational acceleration vector, 1g magnitude effects only z axis. The motion ending operation basically consists of checking these two conditions with practically pre-determined threshold values;

ISTEC 2010

$$abs\left(Acc_X^2 + Acc_Y^2 + Acc_Z^2 - 1\right) < \lambda \quad (2.16)$$

$$abs\left(Acc_{navZ}^2 - 1\right) < \lambda \quad (2.17)$$

Although the motionless state provides these conditions, they are not only valid for motionless state. For example the Figure 3 shows the accelerometer outputs for z axis during a vertical motion through z+ axis. Error point provides the conditions too. Because of that, checking current instant values would not be sufficient for the solution. Sequence of last values with a user defined size must be checked for determining the motionless state of IMU.

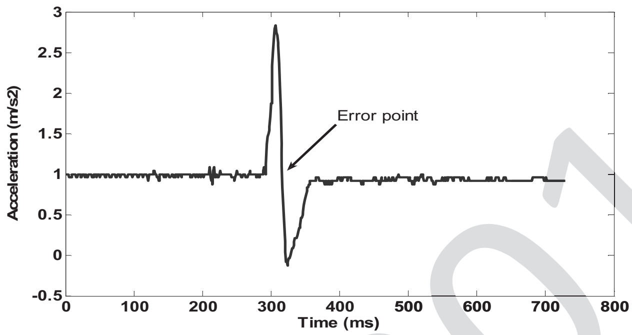

Figure 3: Error Point in Motion Detection

3. SOFTWARE AND USER INTERFACE DESIGN

The software system is designed in such a manner that it could be used with different types of hardware. The structure of the design is given in Figure 4. Connection Adaptor is an interface for different hardware connection libraries and the 3D Motion Sensing Library is structurally independent of hardware. The relation between the hardware specifications and the motion sensing module is created in application layer.

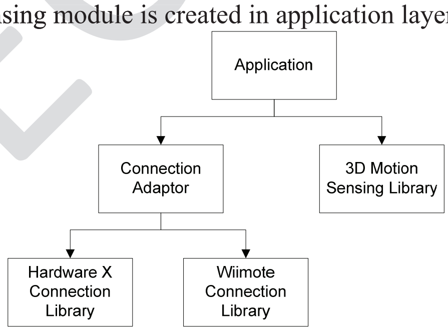

Figure 4: Structural Design of System

3D Motion Sensing Library which uses design pattern techniques is implemented as a modular, reusable and platform independent motion estimation software framework.

A simple 3D user interface application is developed for putting the pieces together and visualizing the motion. This application uses the connection adaptor for Wiimote and 3D motion sensing library. Some hardware specific calibration parameters are passed to motion sensing library. Hardware outputs are received with a specific method. All motion estimation operations are made by calling the motion estimation library and the algorithm results are obtained in this method. The results are passed on the visualizing object in the application. Some samples about usage of the hardware and the visualization of the motions by this application can be seen in Figure 5.

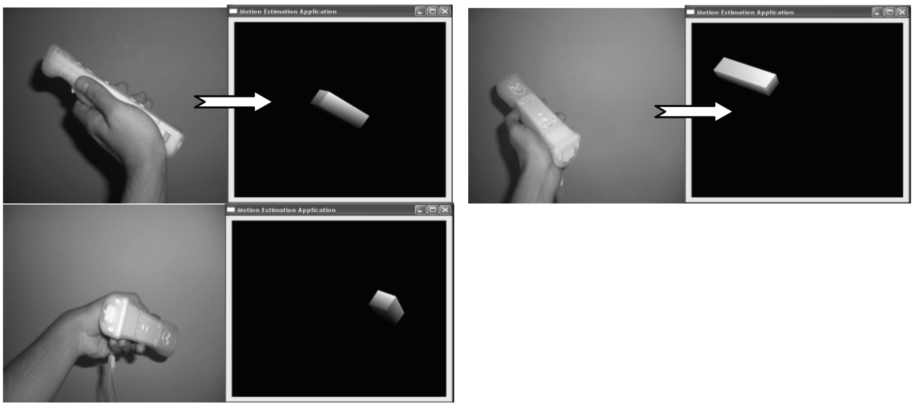

Figure 5: Hardware Motion and User Interface Application

4. CONCULUSION AND FUTURE WORK

Firstly, specifications and error characteristics of accelerometer and gyroscope sensor are examined for every axis by isolated experiments. Before getting through the estimation process, different low-pass filters are tried for the pre-noise filtering operation and averaging filter is selected. Reasonable smoother signals are obtained by using an averaging filter that uses last 5-10 values. Estimation process starts with orientation determination. Drift rates of angular rate sensors cause an orientation error for an instant update. Because of using previous orientation for determining new orientation, previous errors of orientation calculation accumulatively increase. Then, instant accelerometer values are converted from body to navigational coordinates. Both accelerometer biases and previous errors from orientation calculation combine and total system error continue increasing. After these steps, velocity and position estimation is done and this loop repeats for each update, so the total system error enduringly increases during the motion.

Therefore after a user made a specific movement using the object, it would not be suggested to continue with a proceeding movement by using the final, which is also faulty, system parameters. Thus, a motion ending operation is required to reset faulty system parameters prior to further operations. Therefore by using a low-cost MEMS IMU we were designed a high-end system for handheld devices. We have tested the system for the accuracy over various movements and the results are sufficiently accurate for virtual reality applications.

For future work, error-state approach for the dynamic system will be researched. This approach contains estimation of the errors of the system; namely the system state will be the errors that will be occurred in the system. Furthermore, the more effective improvement can be obtained by using additional sensors like magnetometers. Error correction improves by extra inputs and a more stable system can be developed.

5. REFERENCES

[1] Arkadani, H. A., Bridges, T. J. (2010). *"Review of the 3-2-1 Euler Angles: a yaw-pitch-roll sequence"*. pp.7-8.

[2] Bar-Shalom, Y., Li, X.R. and Kirubarajan, T., (2001): Coordinate Transformation in Estimation with Applications to Tracking and Navigation, p. 515, A Wiley-Interscience Publication, New York, USA

[3] Bar-Shalom, Y., Li, X.R. and Kirubarajan, T., (2001): Estimation For Kinematic Models in *Estimation with Applications to Tracking and Navigation*, pp. 72-77, A Wiley-Interscience Publication, New York, USA

[4] Einstein, Albert (1920). "20". *Relativity: The Special and General Theory*. New York: Henry Holt. p. 168. ISBN 1-58734-092-5.

[5] Groves, P. D. (2008*)*. Coordinate Frames, Kinematics, and the Earth in *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. pp. 26-28. ISBN-13: 978-1-58053-255-6.

[6] Groves, P. D. (2008*)*.Inertial Navigation in *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. pp. 123-125. ISBN-13: 978-1-58053-255-6.

[7] "Gyroscope" by Sándor Kabai, Wolfram Demonstrations Project.

ISTEC 2010

[8] Huang Z. J., Fang J. C., (2005). Integration of MEMS Inertial Sensor-Based GNC of a UAV. International Journal of Information Technology Vol.11, No.10, 2005

[9] Verplaetse, Chris (1995). Advantages and Disadvantages of Gimbaled vs. Strapdown. Retrieved from <*http://xenia.media.mit.edu/~verp/projects/smartpen/node9.html*>

[10] Welch, G. and Bishop, G., 2001: An Introduction to the Kalman Filter, Retrieved from <*http://www.cs.unc.edu/~welch/kalman/*>

ISTEC 2010