Third International Conference
Modelling and Development of Intelligent Systems
October 10 - 12, 2013
"Lucian Blaga" University
Sibiu - Romania

Inertial Measurement Unit Simulator

Kiril Alexiev

Abstract

During the last few years microminiaturized inertial sensors were introduced in many applications. Their small size, low power consumption, rugged construction open doors to many areas of implementation. The main drawback of these sensors is the influence of different type of errors, leading to an unavoidable wrong position and orientation estimation. In the paper a simulator of Inertial Measurement Unit is proposed. The simulator is a tool for assistance of trajectory set up and on the base of input data it generates IMU output according given error/noise parameters. It allows us to simulate different types of IMUs based on prior knowledge of the IMU error's properties. One of the main goals in developing of the simulator is to validate new methods involving inertial technology. Something more, the simulator is an excellent tool for tuning complex filtering procedures and enhancing navigation accuracy. The simulation of different scenarios gives more information to receive better understanding of the weight of different sensor noises and errors on the final results.

1 Introduction

Inertial Measurement Unit (IMU) consists from one or more sensors, measuring the change of kinematic energy of a moving body. The sensors are divided in two groups: gyro sensors and accelerometers. Gyro sensor measures rotation rate of the body. Accelerometer provides information about linear acceleration of the body. Usually description of 3D motion of a body is given by 3 orthogonally placed accelerometers giving transition dynamic of the body and 3 orthogonally placed gyro sensors determining the orientation of the body. The axes of the both types of sensors normally coincide – e.g. in a 3D orthogonal coordinate system there are sensors to measure linear accelerations on each of the axes and rotation rate of the same axes. Thus the calculation process is also simplified. Two type of IMU were realized in the years. The first one is built on the scheme of the classical gyroscope and it preserves one and the same (initial) position, remaining independent of body rotation. In this case the body orientation is measured as a difference between gyroscopes axes orientation and the present orientation of the body - its roll, pitch and yaw. The second one, called also strapdown gyro sensor, is fixed tightly on the body and provides measurement of rate of rotation of the body. For this class of sensors, the body orientation is received through the integration of gyro measurements in respect to a priori known body orientation. Usually the strapdown sensors are produced as a MEM device with extremely high robustness and low power consumption. In this paper such a type of devices will be considered. The Inertial Navigation System (INS) is a system that relies entirely on inertial measurements for determination of dynamical body position and orientation. Today a wide range of strapdown INS is available on the market. The simulator, presented in this paper, emulates the behaviour of standard MEM realization of an INS with three linear accelerometers and three angular rate sensors. It generates inertial sensor measurements in accordance with the precision and accuracy specifications of particular sensor sample. The improvement in computers' capability allows the simulation to become instrumental in technology development [5]. The tool, presented here, will be used to:

- Enhance our understanding of inertial technology;
- Simulate different types of IMUs based on prior knowledge of their specifications;
- Simulate a wide range of scenarios, even unrealistic ones;
- Test and validate new navigation algorithms;
- Study of different error propagation and estimation of the error influence over system precision and accuracy;
- Estimate the required hardware/sensor characteristics for a given application;
- Laboratory test of installed systems to assure that they are working properly before real test and to verify system performance in critical/rare situations.

A modular architecture is used in design of the proposed simulator that allows you to modify, improve and replace the individual modules without changing the overall architecture. Simulator gives also flexability in designing and research work and dramatically reduce time and money consuming field experiments. The system under test can be examined on different motion and vibration probations through the computer generation.

The paper is organized as follows. In the next chapter the mathematical background for inertial sensor modeling and simulation in navigation is revealed. Third chapter is devoted on error propagation for accelerometers and gyro sensors and a short overview of different error types is given. The fourth chapter describes the structure of IMU simulator. Some results are described in the next chapter. The concluding remarks are given the last chapter.

2 IMU based navigation (mechanization equations)

The body motion in an inertial frame of reference can be described as a result of simultaneous action of two forces - gravitational F_g and specific F_{sp}:

$$a_i = \frac{F_{sp}}{m_b} - \frac{F_g}{m_b} = a_{sp} - g\,, \tag{1}$$

where g is acceleration,caused by gravitational force and a_{sp} is the acceleration caused by specific force. Gravitational force is a function, depending on the distance between body and the Earth:

$$F_g = G\frac{M_e m_b}{r^2}\,, \tag{2}$$

where G is the gravitational constant $G = 6.6742*10^{-11}$, r is the distance between the interacting bodies, M_e is the mass of the Earth and $K = GM_e = 398600\,.44*10^9$.

To explain the specific force we introduce three frames of reference - one assosiated with the moving body, denoted by subscript $_b$, the second one is a geocentric frame, rotating with the rate of rotation of the Earth - it is associated with the subscribt $_e$ and the last one is also geocetric, but it is inertial and it is marked by subscribt $_i$. Let now denote the rate of the Earth rotation by ω. The last introduction note concerns the differential of a vector in absolute reference frame if it is presented in rotating system:

$$\frac{de_a}{dt} = \frac{de_r}{dt} + \omega \times e_a\,, \tag{3}$$

Let now express the velocity in inertial reference frame, applying expression from (3):

$$v_i = \frac{dr_e}{dt} + \omega \times r_e = v_e + \omega \times r_e , \qquad (4)$$

The next step is to express acceleration, applying twice (3):

$$\frac{dv_i}{dt} = \frac{d(v_e)}{dt} + \frac{d(\omega \times r_e)}{dt} = \underbrace{\frac{dv_e}{dt} + \omega \times v_e}_{\text{first term}} + \underbrace{\omega \times \frac{dr_e}{dt} + \omega \times \omega \times r_e}_{\text{second term}} , \qquad (5)$$

$$\frac{dv_i}{dt} = a_e + 2\omega \times v_e + \omega \times \omega \times r_e , \qquad (6)$$

Regarding the received result as equal to specific acceleration and substituting in (2) we receive:

$$a_i = a_e + 2\omega \times v_e + \omega \times \omega \times r_e - g , \qquad (7)$$

The acceleration $2\omega \times v_e$ is result of Coriolis force, and the term $\omega \times \omega \times r_e$ corresponds to centrifugal acceleration. Usually the last two terms of (7) are grouped together and replaced by so called local gravitational acceleration or simply gravity:

$$a_i = a_e + 2\omega \times v_e - g_l(h) , \qquad (8)$$

where h is the height of the body above the Earth surface. The equation (8) is regarded as fundamental navigational equation.
It is worth to estimate the significance of all terms. Let consider a motion with velocity of 36 km/h on the Earth surface near to Equator. The applied force creates acceleration equal to 1 m/s^2. For this example $a_e = 0.1g$, $2\omega \times v_e = 1.46 * 10^{-3}$, $\omega \times \omega \times r_e = 3.4 * 10^{-2}$.
For calculation of g_l the Gelmert formula is applied:

$$g_l(h) = 9.7803\,(1 + 0.005302\ \sin^2 \varphi - 0.000007\ \sin^2 2\varphi) - 0.00014\ - 2\omega_0^2 h , \qquad (9)$$

where φ is northern latitude and $\omega_0 = 1.2383 * 10^{-3}$ is the Schuler frequency.
The goal of navigation is to find coordinates of a body and its orientation. In the case of IMU sensor the task is solved based on IMU measurements and integral equation (10) and (11):

$$r_i = r_0 + \int_0^t \int_0^t \left(g_l + a_i(t) \right) dt dt \qquad (10)$$

where r_0 is the initial body attitude in inertial coordinate system at time t=0.
The body space orientation can be described accordingly:

$$\alpha = \alpha_0 + \int_0^t \omega_i dt \qquad (11)$$

where α_0 denotes the initial body orientation in inertial coordinate system at time t, ω_i is the vector 3D rate turn, adjusted to inertial coordinate system.
A simple algorithm for coordinate determination is presented below. The calculation scheme is based on Euler angles. Let us denote the rotation matrix, transforming a vector from the moving body to inertial coordinate system by $C_b^i(t)$. Then an acceleration vector $a_b(t)$ in the body coordinate system will be transformed to inertial coordinate system by (3):

$$a_i(t) = C_b^i(t) a_b(t) \qquad (12)$$

Now the rotation matrix $C_b^i(t)$ will be represented by Euler angles [2]:

$$C_b^i(t) = C_z^{'}(t) C_y^{'}(t) C_x^{'}(t), \qquad (13)$$

where

$$C_x(t)=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos(\varphi(t)) & \sin(\varphi(t))\\ 0 & -\sin(\varphi(t)) & \cos(\varphi(t))\end{pmatrix},\ C_y(t)=\begin{pmatrix}\cos(\theta(t)) & 0 & -\sin(\theta(t))\\ 0 & 1 & 0\\ \sin(\theta(t)) & 0 & \cos(\theta(t))\end{pmatrix},$$

$$C_z(t)=\begin{pmatrix}\cos(\psi(t)) & \sin(\psi(t)) & 0\\ -\sin(\psi(t)) & \cos(\psi(t)) & 0\\ 0 & 0 & 1\end{pmatrix}$$

are the rotation matrixes that rotate vectors on angles $\varphi(t)$, $\theta(t)$, $\psi(t)$ on axes x, y and z. It is important to mention that the order of rotation is important. If the angles of rotation are sufficiently small:

$$\delta t\to\begin{cases}\delta\varphi\to 0\\ \delta\theta\to 0\\ \delta\psi\to 0\end{cases},$$

or the measurement sampling rate is sufficiently high (in other words satisfies Nyquist sampling rate, which guarantees that you capture a signal properly because you sample it at least twice per cycle of the highest frequency component it contains) the following substitutions for an angle α may be applied: $\cos\delta\alpha\approx 1$ and $\sin\delta\alpha\approx\delta\alpha$. The product of small angles can be also approximated by zero: $\delta\alpha*\delta\alpha\approx 0$. The final expression for the change in rotation matrix will be:

$$C_b^i(\delta t)\Big|_{\delta t\to 0}=\begin{pmatrix}1 & -\delta\psi & \delta\theta\\ \delta\psi & 1 & -\delta\varphi\\ -\delta\theta & \delta\varphi & 1\end{pmatrix}==I+\Delta_\upsilon \tag{14}$$

Finally, the rotation matrix is presented as a product of the rotation matrix at t and calculated above rotation matrix $C_b^i(\delta t)$, corresponding to small additional rotations, committed in time interval δt:

$$C_b^i(t+\delta t)=C_b^i(t)C_b^i(\delta t)=C_b^i(t)(I+\Delta_\upsilon) \tag{15}$$

Let now express the derivative of rotation matrix:

$$\dot{C}_b^i(t)=\lim_{\delta t\to 0}\frac{C_b^i(t+\delta t)-C_b^i(t)}{\delta t}=\lim_{\delta t\to 0}\frac{C_b^i(t)(I+\Delta_\upsilon)-C_b^i(t)}{\delta t}=C_b^i(t)\lim_{\delta t\to 0}\frac{\Delta_\upsilon}{\delta t}=C_b^i(t)\dot{\Delta}_\upsilon, \tag{16}$$

where $\dot{\Delta}_\upsilon=\begin{pmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{pmatrix}$ and $\omega_x,\omega_y,\omega_z$ are the lastly received measurements from gyro sensors on corresponding axis.

The solution of (16) is $C_b^i(t)=C_b^i(0)\ell^{\dot{\Delta}_\upsilon}$.

The matrix exponent in solution can be presented as an infinite sum:

$$\ell^{\dot{\Delta}_\upsilon}=\sum_{k=0}\frac{\dot{\Delta}_\upsilon}{k!}=I+\frac{\dot{\Delta}_\upsilon}{1!}+\frac{\dot{\Delta}_\upsilon^2}{2!}+\cdots+\frac{\dot{\Delta}_\upsilon^k}{k!}+\cdots$$

Taking into account only the first two terms (linear approximation) we receive an approximate formula for recurrent calculation of rotation matrix:

$$C_b^i(t+\delta t)=C_b^i(t)(I+\Delta_\upsilon) \tag{17}$$

Let now calculate the exact expressions for angle derivatives. In the differential equation (16) we substitute the rotation matrix taking expression in explicit form from (13). The matrix equation will be resolved for matrix element (3,1) (3-rd row, 1-st column). The corresponding equation looks like:

$$\frac{d(-\sin\theta)}{dt} = \begin{pmatrix} -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ \omega_z \\ -\omega_y \end{pmatrix} \tag{18}$$

Therefore:

$$\dot{\theta} = \cos\varphi\,\omega_y - \sin\varphi\,\omega_z \tag{19}$$

For matrix element (3,2) in a similar way we receive:

$$\dot{\varphi} = \omega_x + tg\theta\left(\sin\varphi\,\omega_y + \cos\varphi\,\omega_z\right) \tag{20}$$

To find the expression for $\dot{\psi}$ the equations that contain ψ have to be used. For example, if the element (1,1) is used:

$$\dot{\psi} = \frac{1}{\cos\theta}\left(\sin\varphi\,\omega_y + \cos\varphi\,\omega_z\right) \tag{21}$$

The equation (19), (20) and (21) are most often used for calculation of rotation angles between two successive gyro measurements with a linear approximation only.
The explained above mathematical model is implemented in the simulator.

3 IMU errors

The body attitude is calculated using simultaneously the measurements of 6 sensors - 3 gyros and 3 accelerometers. Body orientation is given by integration of gyro sensors measurements. Transition of the body is calculated by double integration of accelerometers readings, according current body orientation. The integration process quickly accumulates errors. Due to existence of almost constant gravitational acceleration even small errors in the estimates of orientation of the body cause big deviation in the decomposition of gravitational acceleration on the axes, leading to large scale of attitude errors. Due to the quality of sensors IMU are divided in four groups of class of accuracy [1]:

Table 1: Accumulated Error due to Accelerometer Bias Error

Grade	Accel. Bias Error [mg]	Horizontal Position Error [m]			
		1 s	10 s	60 s	1 hr
Navigation	0.025	0.00013	0.012	0.44	1600
Tactical	0.3	0.0015	0.15	5.3	19000
Industrial	3	0.015	1.5	53	190000
Automotive	125	0.62	60	2200	7900000

Table 2: Accumulated Error due to Accelerometer Misalignment

Accelerometer Misalignment [deg]	Horizontal Position Error [m]			
	1 s	10 s	60 s	1 hr
0.050	0.0043	0.43	15	57000
0.10	0.0086	0.86	31	110000
0.50	0.043	4.3	150	570000
10	0.086	8.6	310	1100000

Table 3: Accumulated Error due to Gyro Angle Random Walk

Grade	Gyro Angle Random Walk [deg/√hr]	Horizontal Position Error [m]			
		1 s	10 s	60 s	1 hr
Navigation	0.002	0.00001	0.0001	0.0013	620
Tactical	0.07	0.0001	0.0032	0.046	22000
Industrial	3	0.01	0.23	3.3	1500000
Automotive	5	0.02	0.45	6.6	3100000

As it can be seen from Table 1, Table 2 and Table 3, even small errors in gyro angle estimation may discredit navigation.
The sensors are subject to different types of errors due to sensor imperfectness, model inaccuracy or computational errors.
The main errors influencing on the attitude estimation accuracy may be grouped into three categories [3, 4]:
A. *Sensors do not provide perfect and complete data.*

- Bias errors produce constant or almost constant shift of sensor values from the true ones.
- The scale factor errors cause lack of correspondence between real turn velocities and real straight linear accelerations and output sensors readings (gyro and accelerometer correspondingly).
- Errors due to manufacturing imperfections in IMU. Usually they are caused by non-orthogonally placed accelerometer or gyro sensors on the chip or by lack of coincidence between axes of corresponding accelerometer and gyro sensors. The last error more often is initiated by the first one, but sometimes can exist alone.
- The sensors readings are also contaminated by additive Gaussian noise.
- Temperature dependent errors. Temperature deviation affects output readings.
- There is time synchronization problem. Sensors readings do not belong to one and the same moment of time.
- Dynamic error (lag of sensor reaction/response to force implementation).

B. *Imperfectness of the used models and computational arithmetic*

- Usually the model inaccuracy is caused by inexact sensor approximation, incorrect gravitational acceleration estimate.
- The computational errors are caused by limitations of computer arithmetic, iterative procedures for optimization, calculations of trigonometric functions, loss of orthonormality of matrices, etc.

C. External sources of disturbances (uncontrolled, unpredictable even unknown sources of different type disturbances)

- Platform vibration. The vibration counteracts to sensor accuracy. It depends of different random factors, platform dynamics, mass distribution, switching on/off of different devices, and etc.
- Others

The Fig. 1 below displays the influence of different types of errors on quality of attitude estimation.
Let consider now errors in sensor measurements.
The error propagation for acceleration sensors only looks like:

$$r_i = r_0 + \frac{g_l t^2}{2} + \int_0^t \int_0^t \left(a_i(t) + \varepsilon_a\right) dt dt = r_0 + \frac{g_l t^2}{2} + \frac{\varepsilon_a t^2}{2} + \int_0^t \int_0^t a_i(t) dt dt \tag{22}$$

Here ε_a denotes the error vector of acceleration sensors.

The error propagation for gyro sensors only looks like:

$$\alpha = \alpha_0 + \int_0^t \left(\omega_i(t) + \vec{\varepsilon}_\omega\right)dt = \alpha_0 + \varepsilon_\omega t + \int_0^t \omega_i(t)dt \qquad (23)$$

Here ε_ω denotes the error vector of gyro sensors.

The equations (22) and (23) give error propagation in the simplest case of independent errors. In practice there are many types of errors, influencing one to others. The influence of rotation rate error measurements on angle determination is obvious from (19), (20) and (21). As a consequence the error propagation in (22), for example, generates/induces nonlinear errors in estimation of accelerations, leading to quickly growing errors in estimated system position. That is why (22) and (23) are used only to approximate the order of generated errors and these equations are not of practical use.

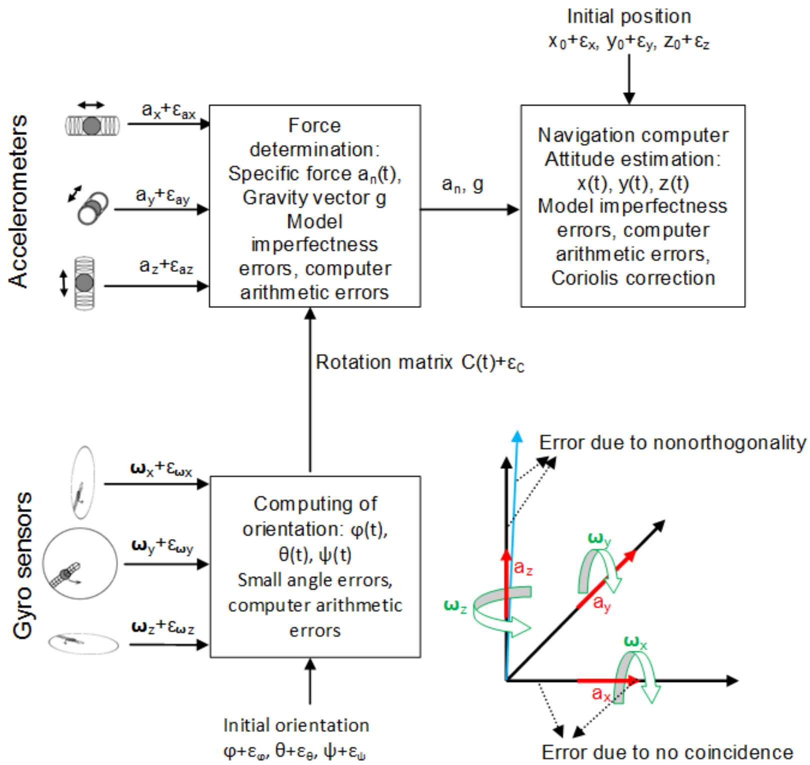

Fig. 1 Errors in an IMU

In order to minimize different type of errors we have to estimate their influence on the position estimate.

There are many well established methods for self-consistency check and normalization. One of them concerns the rows/columns of the rotation matrix. The rotation matrix is direction cosine matrix, which row/columns are projections of unity vector onto orthogonal axes. That means, that the sum of squares of values in each row/column have to be equal to 1 and due to their orthogonality, their scalar products have to be zero. In the cases of using quaternions the normalization means that the sum of squares of quarternion elements has to be equal to 1. This normalization usually doesn't correct errors. Even if optimization procedure is started, the best received result does not guarantee the error compensation. Moreover, the normalization algorithm propagates the error over correct terms. That is why the precise error expression is not of practical use.

4 The structure of IMU simulator

The simulator has modular structure, presented on fig. 2.
Input Data Interface is an interactive module with functionality to insert, edit, save and search user data. The problem of choice what kind of editor to be used for trajectory parameterization (graphical editor or text editor) was resolved in favor of the text editor, which, although being unfriendly and more cumbersome, allows exact parameterization of the trajectories.
Trajectory Generator uses kinematic equations to generate body trajectory. The module has direct output to graphical interface to check generated trajectories and correct them in the case of wrong input data.
Noise Generator adds different type of noises and inaccuracies. This module underwent several adjustments due to change of authors understanding of the influence of different errors on final result.
Inertial sensor model simulates inaccuracy and imperfectness of the sensors like sensor axes non-orthogonality, bias instability and scale errors, lag in sensor data, and others.
Navigation model consists of suit of tested algorithms. There are several classical realizations of navigational algorithms and their modifications for implementation in mobile devices.
Graphical output gives 3D presentation of generated trajectories, noised data and results of navigation algorithms' data processing. A special form of presentation of 3D body orientation is introduced.

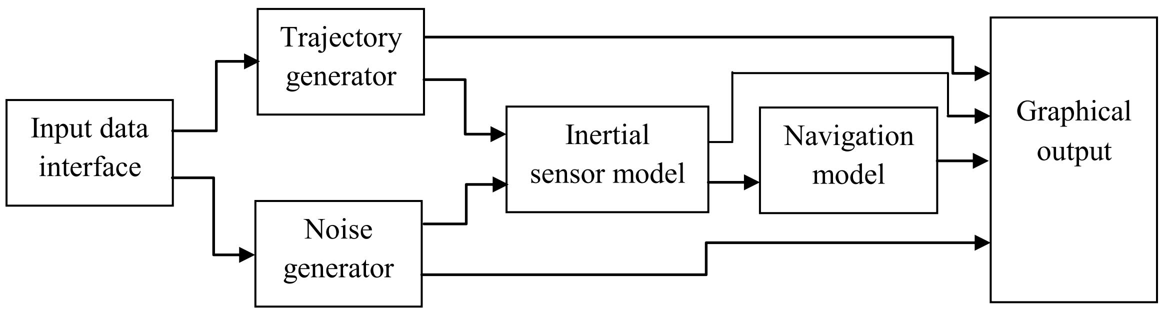

Fig. 2 The simulator structure

5 Results

The simulator was tested in an example for both: simulated data and real hardware generated data (a platform with MPU-6050 strapdown inertial sensors). The experiment on the fig 3 includes a simple body move following contour of a quadrate in horizontal plane. The data flow from simulator and sensors (3 gyros and 3 accelerometers) were saved and different types of navigation algorithms were applied. The hardware gyro and accelerometer signals are shown on Fig. 4. The calculated platform trajectory received by data processing from a navigation algorithm is shown on Fig. 5.

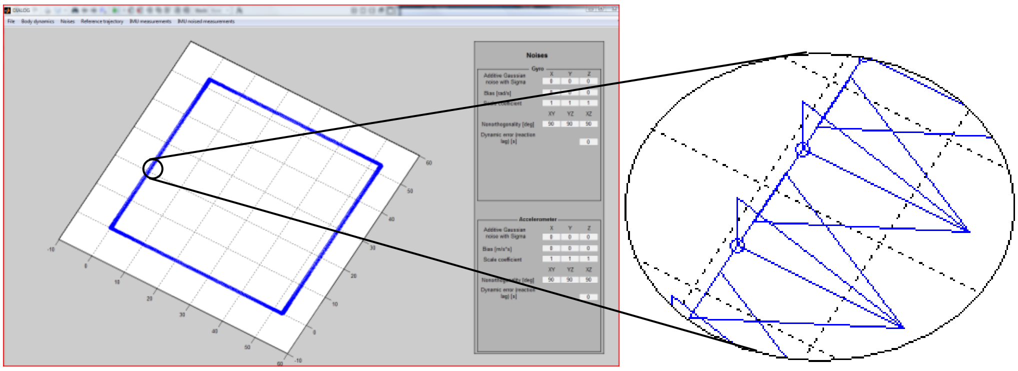

Fig. 3 Simulator with graphical output of the reference trajectory. In the circle on the right side the orientation of the moving body is presented.

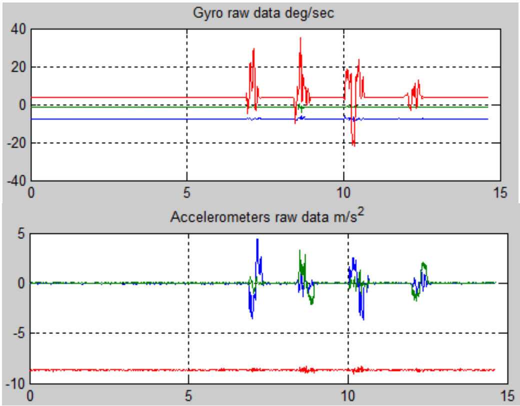

Fig. 4 Gyro and accelerometer sensors raw signals (from hardware platform)

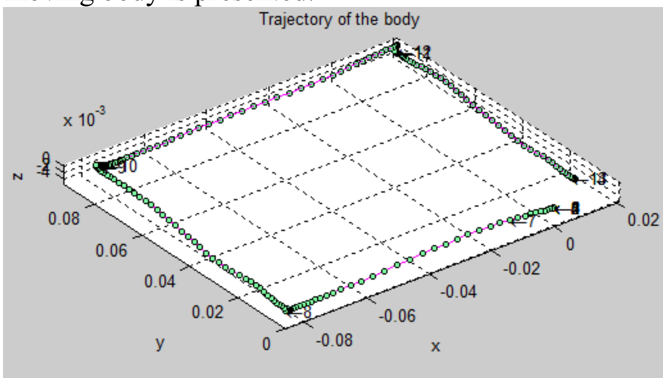

Fig. 5 The output results of navigation algorithm

6 Conclusion

The contemporary strapdown inertial MEMs are far behind in accuracy from the precise, very heavy and costly navigation platforms. In spite of this a lot of applications are waiting for more precise inertial sensors. The proposed in this article simulator of IMU shortened the road from idea generation to design of real application. It improves design by executing comprehensive and exhaustive simulations in the lab, minimising field testing. Something more, the simulator allows optimization of the choice of inertial sensors for a particular application based on published sensors datasheets only, materializing “software in loop” simulation approach. The modular structure of simulator allows further enhancement and enrichment of suit of algorithms. One of the most interesting directions for further development of the simulation tool is realization of “hardware in loop” simulation [6] through appropriate hardware interface and software drivers.

Acknowledgement: The research work reported in the paper is partly supported by the project AComIn "Advanced Computing for Innovation", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions) and by the project No DFNI – I01/8 funded by the Bulgarian Science Fund. All data, laboratory equipment were supplied by “MM Solutions” in the framework of the project “Industrial research for development of technology for image enhancement and video stabilization using inertial sensors”, Contract BG161PO003-1.1.06-0037-C0001, Operational Program "Development of the Competitiveness of the Bulgarian Economy".

References

[1] http://www.vectornav.com/index.php?&id=76
[2] David H. Titterton, John L. Weston Navigation Technology - 2nd Edition, The Institution of Electrical Engineers, 2004, ISBN 0 86341 358 7
[3] Grewal, M.S., Weill L.R., Andrews A.P., Global Positioning Systems, Inertial Navigation, and Integration, John Wiley & Sons, 2001, ISBN 0-471-20071-9.
[4] Oliver J. Woodman, An introduction to inertial navigation, Technical Report UCAM-CL-TR-696, ISSN 1476-2986, 2007.
[5] http://www.sbir.gov/sbirsearch/detail/87111
[6] http://www.americangnc.com/products/rtgis.htm

KIRIL ALEXIEV
Institute of Information and Communication Technologies
Mathematical Methods for Sensor Information Processing
Sofia, 25A Acad. G. Bonchev Str.
BULGARIA
E-mail: alexiev@bas.bg