Online ISSN: 2367-5357
DOI: 10.7546/itc-2019-0012

Algorithms for IMU Navigation – A Review

K. Alexiev

Key Words: *Inertial measurement unit; navigation; attitude calculation.*

Abstract. *This article discusses the four main approaches to implementing navigation (attitude and location estimation) using inertial sensors. Two ways to evaluate the accuracy of the algorithms are proposed - using synthesized data and using real one. Due to the fact that all algorithms belong to the "dead reckoning" algorithms and have the unlikeable property to accumulate errors, some options have been proposed to create reference points for estimating the accumulated error without availability of other sensors. The article helps to resolve the complex compromise between the complexity of the algorithm and the accuracy of work, which would support the work on the practical implementation of this type of algorithms.*

1. Introduction

When the inertial sensors were created about a century ago, their penetration in the markets others than aviation and military weaponry was not economically viable. Until recently the inertial sensors were regarded as too expensive, too heavy, too energy intensive to be applied. But since 1964, when the first MEMS device was patented [1], the situation changes abruptly. The MEMS technology was rapidly improved and today different MEMS devices flood the market and nearly every sector is influenced by them. The price of MEMS performed inertial sensors fall down and now they are available for several dollars only. Their small size, low power consumption and rugged construction open doors to many areas of implementation. Their numerous applications are realized in transportation, telecommunication, healthcare, smart homes, etc. Today the microminiaturized inertial sensors are embedded in billions electronic devices. The number of people involved in design arises, too. Usually the developers have to make a compromise between goals to reach, means to use, algorithm complexity, sensor quality, etc. In the past 40 or more years a large amount of articles and monographs on the strapdown inertial navigation were published. Due to importance for military purposes the leading specialists are mostly from USA and Russia. The theoretical founder of strapdown navigation in Russian school is considered M. Zakharin, who proved the consistency of the idea of strapdown system and developed its mathematical description [2]. In that time the term "strapdown" was not known and instead of it Zakharin called these system "navigation without stabilized platform". In USA the works of Savage [3], Bortz [4] and Mayback [5] have to be outlined. Today the monographs of Titterton [6], Britting [7], Collins [8], Grewall at al. [9], Salichev [10] and Meleshko [11], to mention few of many, are used as foundation for navigation systems design. The most of cited authors described how to design highly accurate navigation system for back up guidance of the Apollo Lunar Module [12], for example, without any limitation of resources.

In this paper an approach is suggested for algorithm verification, estimation of achieved accuracy and minimization of resources. The paper will be useful for developers of applications using low accuracy inertial sensors in the smart phone, toys, home automation, human activity monitoring and estimation, etc. The class of high accuracy systems for navigation, guidance and control is not commented here.

The careful analysis of error propagation in the inertial navigation systems shows that the errors in gyros and respectively in determining the attitude influence much more the final result of the navigation in comparison with the errors in accelerometers. This fact was the main argument for further analysis and study of these particular errors only. Two scenarios were considered. The first one uses real sensors data and the second one is based on simulated data. The computer generated data emulate inertial sensor measurements in accordance with the technical specifications of particular sensor sample and without considering sensor errors. The real sensor data were received for a number of specially designed scenarios, assuring existence of reference points. The most widely spread algorithms for attitude calculation were programmed and examined.

The paper is organized as follows. In the next section the problem under consideration is described. The third section contains mathematical formulation of realized algorithms for attitude calculation. The experimental setting is given in the fourth section. The fifth section describes the experimental results and conclusion with a summary of main outcomes finishes the paper.

2. Problem description

A complete inertial navigation system usually consists of 3 accelerometers and 3 gyros. Here the most complicated case of measuring of position and attitude of a body in 3D space is considered. The accelerometers provide information about linear acceleration of the body (exerted on the body forces including gravity).The gyro sensors measure the rotation rate of the body. The accelerometers and gyros are placed on the axis of an orthogonal coordinate system. Usually the axes of accelerometers and gyros coordinate systems coincide. Our further considerations are based on these assumptions. The functional diagram of the strapdown navigation system is shown on *figure 1*. The measurements, received by gyros are denoted by $\vec{\omega}^b$. Here ω is rate of rotation of the body, on which the gyros are mounted. The vector of body accelerations, measured by accelerometers is denoted by $\vec{a}^b$. The corresponding vector of accelerations in navigation coordinate system is $\vec{a}^n$. The transformation matrix for transition from body coordinate system to a chosen navigation system is C_b^n. The output information of navigation system includes the attitude of the body, its position, velocity and acceleration.

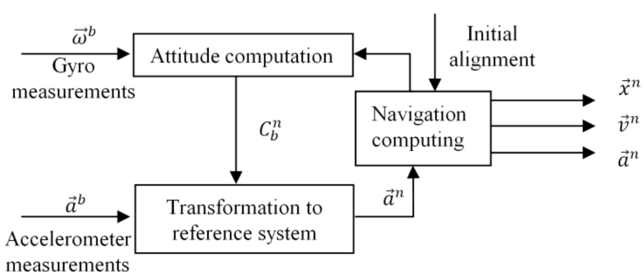

Figure 1. Functional diagram of strapdown navigation system

The inertial navigation does not need any other sensor information to detect the attitude and position of the body. It is often called dead reckoning system, due to the fact that the next attitude and position are calculated on the basis of attitude and position on the previous step and the measurements of accelerometers and gyros.

The main idea of an inertial navigation system is to calculate the velocity and position of the body through integration of accelerations. The first integral yields the velocity, the second one – the position. Due to the fact, that body may have random attitude in time, the acceleration vectors have to be transformed in a constant (inertial) coordinate system in order to calculate the body position.

If the accelerometer measures body acceleration with error denoted by δa, the corresponding error in distance calculated after two integrations will be $\delta a \cdot t^2/2$, e.g., it is proportional to the square of the time. If the gyro measures turn rate with error equal to $\delta\varphi$, after integration the error in calculated attitude will be proportional to $\delta\varphi \cdot t$. Since the attitude is applied further for computation of the orientation of the acceleration vectors, the error in its calculation is propagated over distance calculation (double integration) and the final error is equal to $\delta\varphi \cdot t^3/6$. The analyses show that the attitude computation plays more important role and it has to be precisely determined.

For evaluation of a designed navigation system usually high quality measuring equipment is used. Sometime especially designed laboratory apparatus realizes repeatedly scenarios with infinitesimal deviation as a pattern or benchmark for quality estimation. This paper considers design of applications, most of all on the basis of inertial sensors mass production. The low cost sensors usually are with limited accuracy. The goals here are more modest and concern only the repeatability of algorithms, convergence to real attitude, algorithm efficiency, etc.

The accuracy of a measuring system usually refers to how close is the measured (and calculated on the base of measurements in a more complex system) value to the true one. The true value is called reference point. There are several approaches to create reference points. The first one considers the case when the exact mathematical model of observed process exists. Evaluation of this model in time will give us reference points. Usually, in order to achieve a sufficiently accurate model, it is necessary to greatly complicate its mathematical description. The second approach is to have measuring sensors with higher accuracy, used in parallel with estimated ones. The accuracy of reference sensors have to be at least one order of magnitude higher than the accuracy of the estimated ones. The third approach considers systems, which realize 3D translations and rotation with known trajectory and attitude with very high accuracy. The most serious flaw of the second and third approaches is the price of the necessary equipment.

As a compromise between the latter two approaches in this article the following approach was proposed: realize free motion of the body in 3D space, measuring periodically its position and orientation with as high as possible accuracy. For example, if a smart phone with its inertial sensors is our measuring device (body), we could rotate it, leaving periodically it on fixed places (with measured accurately in advance positions and attitudes). Using this approach allows us to obtain an assessment of the quality of the entire navigation system. We get the opportunity to determine the impact of both types of errors – due to the sensor's inaccuracy and due to the imperfection of the algorithm. In order to estimate and tune the algorithms only a simplified generator of gyros measurements is created.

It models pure rotations (measurements without any disturbances) on one, two and three axes, generates measurements with additive noise, biased measurements and measurements with trend.

3. Attitude computation

The most widely used algorithms for attitude computation are programmed. Because of the fact, that the low accuracy gyro sensors are considered, the expressions for world rotation and the influence of linear velocity onto the measured turn rate are excluded.

3.1 Naive integration approach for attitude calculation

The simplest way to calculate attitude is to integrate the received turn rate measurements directly independent on each axis:

$$(1)\quad \begin{vmatrix}\varphi\\ \theta\\ \psi\end{vmatrix} = \begin{vmatrix}\varphi_0\\ \theta_0\\ \psi_0\end{vmatrix} + \int_0^t \omega dt.$$

Here, $\varphi_0, \theta_0, \psi_0$ are the angles of initial body attitude, which is changed after body rotation with rotation rate ω for time interval, equal to t. In discrete presentation, receiving measurements from the 3D gyroscope sensors at $k-1$ and k -th moment, Eq. (1) takes the following form:

$$(2)\quad \begin{vmatrix}\varphi_k\\ \theta_k\\ \psi_k\end{vmatrix} =$$

$$\begin{vmatrix}\varphi_0\\ \theta_0\\ \psi_0\end{vmatrix} + \sum_k \begin{vmatrix}(\omega_x(k)+\omega_x(k-1))/2\\ (\omega_y(k)+\omega_y(k-1))/2\\ (\omega_z(k)+\omega_z(k-1))/2\end{vmatrix}(t_k - t_{k-1}).$$

3.2 Attitude calculation through Euler-Krylov algorithm

Usually the computation of attitude is presented in Euler (Euler-Krylov in Russia) form. The transformation matrix C_b^n rotates vectors from one coordinate system to another (from body coordinate system to navigation coordinate system in this particular case). It is described as non-commutative product of three matrices, every one of which realizes rotation around corresponding axis of the coordinate system:

$$(3)\quad C_b^n(t) = C_z(t)C_y(t)C_x(t),$$

where

$$(4)\quad C_x(t) = \begin{pmatrix}1 & 0 & 0\\ 0 & \cos(\varphi(t)) & \sin(\varphi(t))\\ 0 & -\sin(\varphi(t)) & \cos(\varphi(t))\end{pmatrix},$$

$$(5)\quad C_y(t) = \begin{pmatrix}\cos(\Theta(t)) & 0 & -\sin(\Theta(t))\\ 0 & 1 & 0\\ \sin(\Theta(t)) & 0 & \cos(\Theta(t))\end{pmatrix},$$

$$(6)\quad C_z(t) = \begin{pmatrix}\cos(\psi(t)) & \sin(\psi(t)) & 0\\ -\sin(\psi(t)) & \cos(\psi(t)) & 0\\ 0 & 0 & 1\end{pmatrix}.$$

Here $\varphi(t)$, $\Theta(t)$and $\psi(t)$ are the angles of rotation. The rate of change of transformation matrix C_b^n (direction cosine matrix) looks like [6]:

$$(7)\quad \dot{C}_b^n = C_b^n \Omega \times.$$

In this matrix differential equation $\Omega\times$ is skew symmetric matrix of rotation vector $\vec{\omega}$:

$$\Omega \times = \begin{vmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{vmatrix}.$$

Let write in detail this equation for elements (3, 1), (3, 2) and (1, 1) of transformation matrix C_b^n. After trivial transformations we receive:

$$(8)\quad \dot{\varphi} = \omega_x + tg\theta(sin\varphi * \omega_y + cos\varphi * \omega_z),$$

$$(9)\quad \dot{\theta} = \cos\varphi * \omega_y - \sin\varphi * \omega_z,$$

$$(10)\quad \dot{\psi} = \frac{1}{cos\theta}\left(sin\varphi * \omega_y + cos\varphi * \omega_z\right).$$

The differential Eqs. (8)-(10) are the most frequently cited for calculation of Euler rotations. When the angle $\theta \to \pm 90^0$, $tg\theta$ in (8) becomes infinite and $cos\theta$ in Eq. (10) goes to 0. In this case the equations become undeterminated.

3.3 Using Poison differential equation for attitude calculation

Let now consider again the Poisson Eq. (7). The solution for $t = t_{k+1}$ is:

$$(11)\quad C_{k+1} = C_k \exp\int_{t_k}^{t_{k+1}} \Omega \mathrm{dt} = C_k \exp(\sigma\times).$$

Here $\sigma\times$ is skew symmetric matrix of vector $\vec{\sigma} = \begin{vmatrix}\varphi_{k+1}-\varphi_k\\ \theta_{k+1}-\theta_k\\ \psi_{k+1}-\psi_k\end{vmatrix}$. Let denote the norm of the vector by σ_n.

The exponent may be expanded as:

$$\exp(\sigma\times) = I + \sigma\times + \frac{(\sigma\times)^2}{2!} + \frac{(\sigma\times)^3}{3!} + \cdots$$

After some transformations [6]:

$$(12)\quad \exp(\sigma\times) = I + \frac{\sin\sigma_n}{\sigma_n}\sigma\times + \frac{1-\cos\sigma_n}{\sigma_n^2}(\sigma\times)^2.$$

On the basis of (11) and (12) the final expression for C_{k+1} is received:

(13) $C_{k+1} = C_k \left(I + \frac{\sin \sigma_n}{\sigma_n} \sigma \times + \frac{1 - \cos \sigma_n}{\sigma_n^2} (\sigma \times)^2 \right).$

3.4 Quaternion approach

The quaternion can be regarded as alternative form of rotation vector presentation. It is four component vector $q = [q_0, q_1, q_2, q_3]^{\mathrm{T}}$. It is suggested that the transformation from one coordinate system to another can be presented by a single rotation around a vector. Let denote by $\vec{\mu}$ the vector of rotation and by μ_n its norm. The components of the quaternion will be as follow:

(14) $q_0 = \cos(\mu_n/2),$

(15) $q_1 = \frac{\mu_x}{\mu_n} \sin(\mu_n/2),$

(16) $q_2 = \frac{\mu_y}{\mu_n} \sin(\mu_n/2),$

(17) $q_3 = \frac{\mu_z}{\mu_n} \sin(\mu_n/2).$

The component q_0 is real, while the components q_1, q_2, q_3 are imaginary. The complex presentation of quaternion looks like: $q = q_0 + q_1 i + q_2 j + q_3 k$.

The rotation of a vector $\vec{a} = [a_x, a_y, a_z]$ from body frame to navigation frame may be expressed though presentation of vector $\vec{a}$ in quaternion form

$$a_q = 0 + a_x i + a_y j + a_z k$$

and the multiplication of the quaternions:

$$a_q^n = q a_q^b q^*.$$

Here $q^* = q_0 - q_1 i - q_2 j - q_3 k$ is the complex conjugate of q.

The quaternion variant of Poisson differential equation is:

(18) $\dot{q} = 0.5 q \omega_q =$
$$0.5 \begin{vmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{vmatrix} \begin{vmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{vmatrix}.$$

The more useful form of quaternion differential equation is:

(19) $\dot{q} = 0.5 q \omega_q =$
$$0.5 \begin{vmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{vmatrix} \begin{vmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{vmatrix}.$$

For $t = t_{k+1}$ and in analogy with Eq. (11)-(13) for vectors of size 4 (quaternions) the following solution may be found [6]:

(20) $q_{k+1} = q_k \left(A \otimes \sigma_q \right).$

The operator $\otimes$ means component-wise multiplication of quaternions $A = \begin{vmatrix} c \\ s \\ s \\ s \end{vmatrix}$ and σ_q, received on the base of vector $\vec{\sigma}$, defined in previous section. The components of A have to be calculated up to a given truncation point from the series:

(21) $c = 1 - \frac{(0.5\sigma)^2}{2!} + \frac{(0.5\sigma)^4}{4!} - \cdots,$

(22) $s = 0.5 \left(1 - \frac{(0.5\sigma)^2}{3!} + \frac{(0.5\sigma)^4}{5!} - \cdots \right).$

4. Experimental setting

A number of input data sets were generated through modelling or received from smart phone sensors, measuring phone turn rate for specially planned scenarios.

The generated measurement data model the most interesting cases of 1D rotation around each of the axis, 3D rotations with non-overlapping in time rotations around each of the axis and with overlapping rotations, highly intensive rotations, etc. All rotations were generated with known end point (attitude). The sensor measurement generator may create data without any errors, with white noise with chosen amplitude, with bias, trend, etc. Some of the used scenarios are presented on *figure 2*.

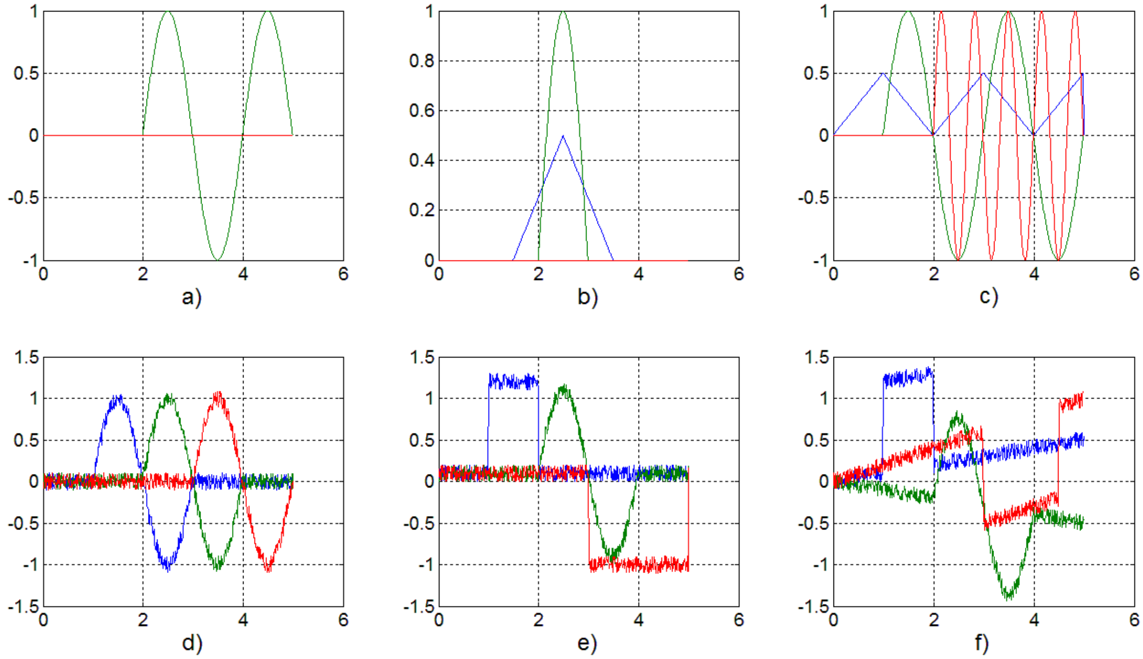

Figure 2. Generated gyros data (turn rate versus time): a) simple 1D rotation; b) overlapping rotations around two axes; c) overlapping rotation around all axes with different intensity; d) noised 3D overlapping rotations; e) noised 2D overlapping rotations with bias; f) noised 2D overlapping rotations with different trends

The real measurement sets received from gyros were taken from a smart phone. Contemporary smart phones are equipped with complete set of sensors for inertial navigation. The sensor set usually includes gyro sensors, accelerometers and magnetometers. There are many ways to receive sensor data on a remote computer even in real time. We used the simplest one – during testing the sensor data were recorded in phone memory and after experiment the data were transferred through USB cable into a desktop computer. The freeware application "Sensorstream IMU+GPS" (in Google Play) was used to assure access to the gyro sensors in Google Android operating system.

The scenarios were especially designed to allow error estimation and convergence of the algorithms without additional measurement equipment. On *figure 3* three possible scenarios are visualized. To estimate the accuracy every experiment contains at least one reference point with known attitude (and position) of the phone. These points are marked with red circles at the cited figures. Only one experiment with no fixation of starting and end position of the phone was presented on *figure 3c*.

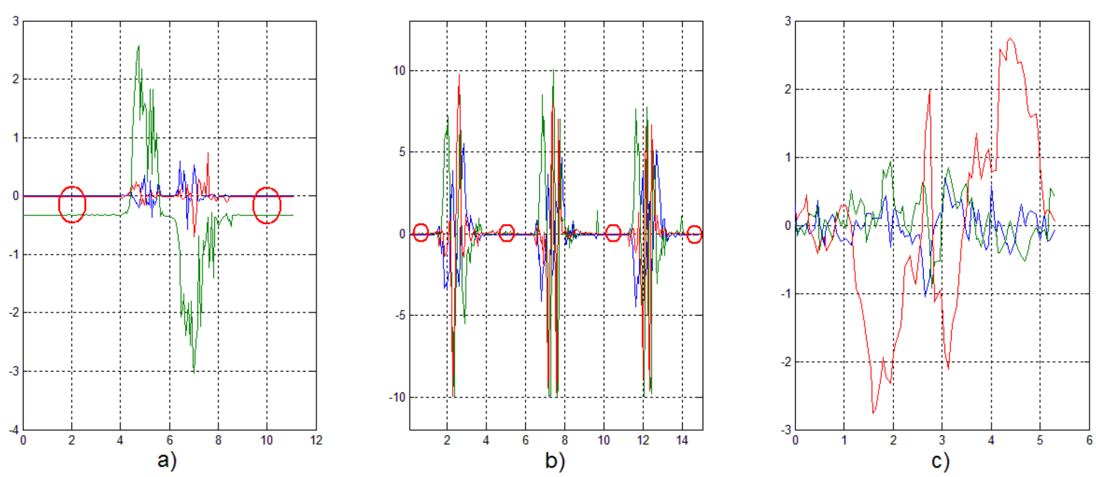

Figure 3. Real gyro data: a) 1D real data; b) 3D free rotations in cycle; c) 3D free rotations

5. Experimental results

Numerous scenarios were experimented with simulated and real gyro data in order to compare programmed algorithms. The first one of algorithms – naive integration approach for attitude calculation, in spite of its simplicity, showed good results especially for high measurement rate. The attitude calculation through Euler-Krylov algorithm again proved its instability around 90^0. DCM differentiation and quaternion algorithms showed their excellent properties. Along with the no surprising results some interesting fact were found which will be discussed below. To describe them better three especially designed scenarios were generated.

The first scenario concerns simulated gyro data without any noise. The x-axis gyro (denoted by blue line on *figure 4a*) measures 0.1 rad/s starting at 2 s till 3 s. The y-axis gyro at the same time measures 2 rad/s turn rate on this axis. The value of turn rate is especially chosen to cross the sensitive value of 90^0 in $1\,s$. The result of differentiation using Euler-Krylov algorithm is incorrect (*figure 4b*). This result was expected. The surprise was that the errors in estimation of angles of rotation around axes x and z start to accumulate away before reaching the angle of 90^0 degrees (*figure 4b*). On *figure 4c* the correct result of differentiation DCM is shown as a reference.

In the second scenario simulated gyro data without noise are used again. The input signal corresponded to overlapping rotation around all axes with different intensity (*figures 5a* and *5d*). Two cases of gyro measurements were considered 1) with measurement rate equal to 5 Hz (*figure 5a*); 2) with measurement rate equal to 100 Hz (*figure 5d*). Two of described algorithms were compared on this data set – naive integration and superior DCM differentiation. In contrary to our expectations the simplest cumulative computation gives very good results for the case of higher measurement rate (*figure 5e* and *figure 5f*). The difference of quality was determined in lower measurement rate (*figure 5b* and *figure 5c*).

The last scenario considers data received by smart phone gyro sensors with 20 Hz measurement rate. The rotation rate was chosen maximally intensive. Two reference points were created – at 5 s and 9 s. At this moments of time the smart phone took one and the same position and orientation on the table. The results of experiments are shown on *figure 6*. The initial state of the phone was very stable at first 5 seconds without any accumulation of error. In phase of intensive rotations, however, in spite of superiority of applied algorithm of differentiation DCM a substantial error was accumulated, which cannot be explained with low sensor quality or algorithmic inaccuracies. The reason for this crucial error is the low measurement rate, nonconforming with rotation intensities.

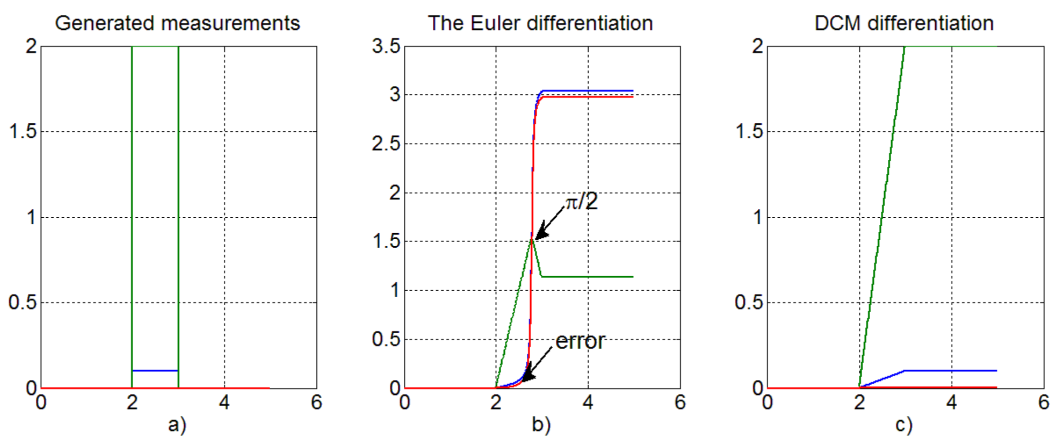

Figure 4. The errors in Euler differentiation approach: a) generated gyro data; b) result of Euler differentiation algorithm; c) result of DCM differentiation as a reference

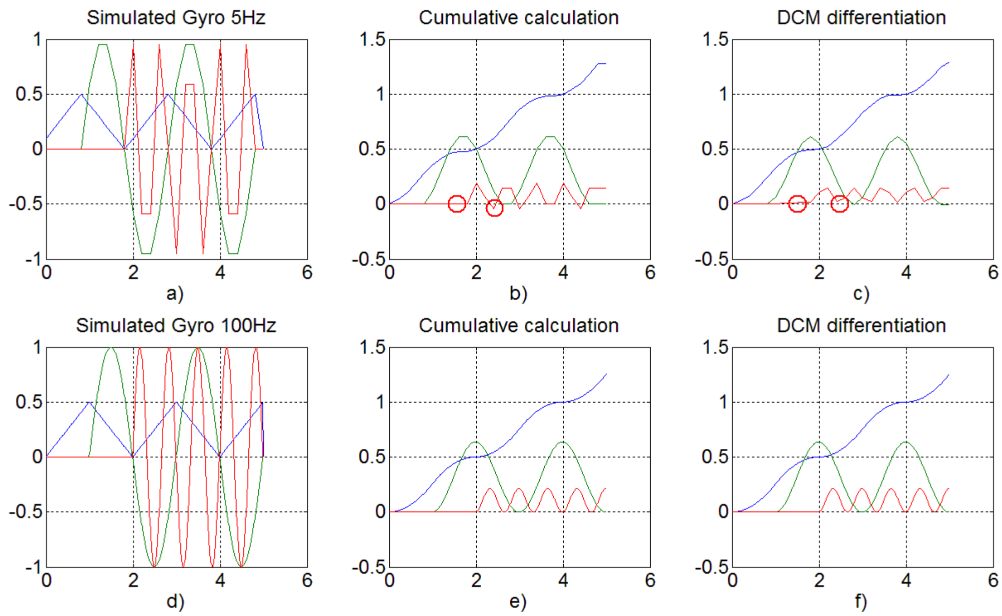

Figure 5. Comparison between naive integration and DCM differentiation algorithms for attitude calculation: a)simulated gyro data with 5 Hz measurement rate for overlapping rotation around all axes with different intensity; b) cumulative calculation with denoted with red circles errors; c) result of DCM differentiation as a reference; d) simulated gyro data with 100 Hz measurement rate for overlapping rotation around all axes with different intensity; e) cumulative calculation; c) result of DCM differentiation as a reference.

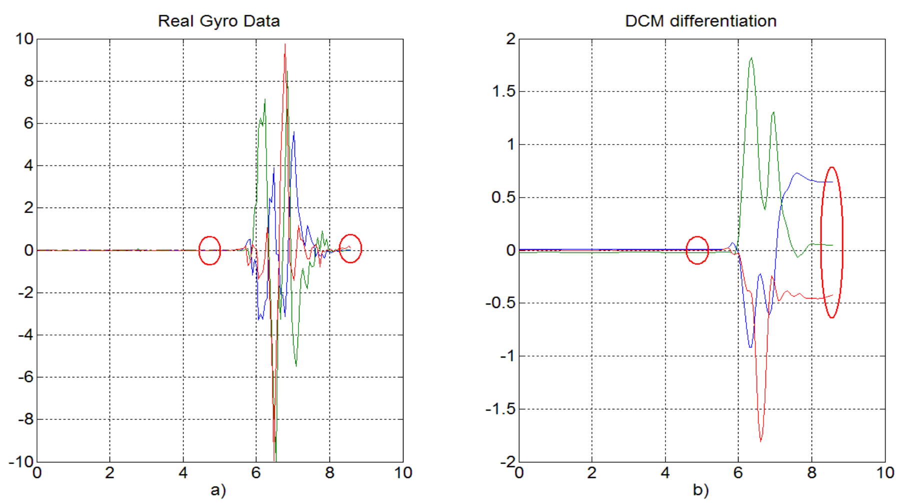

Figure 6. Application of DCM differentiation algorithm for real gyro data with two reference points

6. Conclusion

In this paper four contemporary algorithms for attitude estimation of strapdown inertial navigation system are examined on real and synthetic data. The real sensor data are used to estimate sensor errors through artificially created reference points. The simulated sensor data serve for algorithm estimation, debugging, applicability testing and tuning. In numerous tests the superior algorithms using DCM differentiation and quaternions proved their properties, but the simplest integration for attitude calculation also received good estimates and it can be applied in many practical cases.

The received results are useful for suitable choice of the lowest complexity navigation algorithm with a sufficient accuracy for given application. Something more, the paper gives an idea how to estimate or orientate in applicability of existing big variety of algorithms without complex and very expensive test devices.

References

1. Nathanson, H. C., R. A. Wickstrom. Microelectronic Frequency Selective Apparatus with Vibratory Member and Means Responsive Thereto. US Patent 3413573, Issued 1968-11-26.
2. Zakharin, M. I., F. M. Zakharin. Kinematics of Inertial Navigation Systems. Mashinostroyeziye Press Moscow, 1968, NASA Technical Translation 1970.
3. Savage, P. G. Strapdown Analytics. Strapdown Associates, Inc., Maple Plain, Minnesota, 1997.
4. Bortz, J. E. A New Concept in Strapdown Inertial Navigation. NASA Technical Report, TR R-329.
5. Maybeck, P. S. Wander. Azimuth Implementation Algorithm for a Atrapdown Inertial System. Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, 1973.
6. Titterton, D. H., J. L. Weston Navigation Technology, 2nd Edition, The Institution of Electrical Engineers, 2004.
7. Britting, K. R. Error Analysis of Strapdown and Local Level Inertial Systems Which Compute in Geographic Coordinates, Measurement Systems Laboratory, MIT, 1969.
8. Collins, R. P. G. Introduction to Avionics Systems. Springer, 2011.
9. Grewal, M. S., L. R. Weill, A. P. Andrews. Global Positioning Systems. Inertial Navigation and Integration, Wiley-Interscience, 2007.
10. Salychev, O. Inertial Systems in Navigation and Geophysics. Bauman MSTU Press, Moscow, 1998.
11. Meleshko, V. V., O. I. Nestorenko. Non-platform Inertial Navigation Systems. Textbook, Kirovograd: Polimed-Service, 2011, p. 171 (in Russian).
12. Miller, J. E. Space Navigation, Guidance and Control, Technivision Limited, Maidenhead, England, 1964.

Manuscript received on 12.02.2019

Kiril Alexiev *received a M.Sc. degree in Cybernetics from National Technical University of Ukraine (“Igor Sikorsky Kyiv Polytechnic Institute”) in 1984. He is currently Assoc. Prof. at the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences. His scientific research interests include sensors and sensor networks, computer networks, track initiation and target tracking, identification, digital image processing, 3D scene restoration, sensor data processing.*

Contacts:
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 25A
1113 Sofia, Bulgaria
e-mail: alexiev@bas.bg