Modelling an Accelerometer for Robot Position Estimation

Zdzisław Kowalczuk
Faculty of Electronics, Telecomm. and Informatics
Gdańsk University of Technology
Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: kova@pg.gda.pl

Tomasz Merta
Dept. of Robotics and Decision Systems (ETI)
Gdańsk University of Technology
Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: tommerta@gmail.com

***Abstract*—The article describes a new model of a MEMS accelerometer for usage in inertial measurement units (IMU). Such units allow to measure orientation and location of the sensor/system and therefore can be applied for systems positioning. The main purpose of the paper is to model pertinent accelerometer functions substantial in determination of the location of the sensor by means of double integration of acceleration. The model takes into account static and dynamic working conditions. Based on this model an estimator is constructed that allows us to test the developed system in terms of the effects of rotational dynamics. Computer simulations are applied to illustrate the performance of the estimator for simulated measurement signals. The results obtained show what conditions must be met in order to properly determine the linear acceleration with accelerometer measurements.**

I. Introduction

The MEMS (ang. *Micro Electro-Mechanical System*) sensors are in a widespread use for the last years. The low cost of sensor chips is the main reason for using them in many applications. MEMS accelerometers are frequently used in cell phones (smartphones), hard disks, photo cameras, console games etc. [1]. In most cases, however, accelerometers are used only to find the direction of the vector of gravity. In advanced applications, such as determining the orientation of air crafts, and surveillance of human health, there is a need for sensors of high fidelity and high precision.

In robotic applications accelerometers are applied to specific single tasks, like detection of free fall or high g-force, or measuring vibrations. There are also complex tasks, like automotive crash detection, adjusting surgical trajectory [2], or other dedicated solutions for micro robotics [3]. Most commonly used systems are focused generally on the determination of orientation for static conditions or on other inertial measurements [4], [5].

If you are using a video camera with inertial measurements, appropriate image processing can be improved in many ways. For instance, to improve the quality of the results of processing, especially when acquisition conditions are poor, an image shift can be reduced using inertial measurements [6]. Tasks related to robot navigation can also give better effects if one combines a MEMS sensor output with vision feature detection algorithms [7], [8]. Similarly, SLAM (*Simultaneous Localization an Mapping*) solutions often use various sensory data. Thus the design and the use of a proper sensor data fusion algorithm appears to be a crucial task.

Our previous research in this context has been associated with the problem of SLAM and image processing in mobile robotics. We have focused our attention on designing a Vis-Robot system for the purpose of creating a 3D map in a limited environment. The principal part of mapping has been based on the idea of relocating systemic robots for the purpose of producing different stereo images.

In this system the orientation data rely on inertial sensors. Magnetometer, accelerometer and gyroscope placed on a single board are used to determine the actual rotation of an object. Actually, there are two algorithms of sensor fusion tested on our robotic platform. Localization at this stage of research is determined using accelerometers. Although the system orientation does not appear to be problematic, location based on acceleration data is not an easy task. To ensure precise results of the integration of accelerometer readings, we have to analyse accelerometer parameters and data to create a dynamical model of the applied accelerometer. Due to the well known problems related to data integration, there are certain restrictions for this approach. Therefore, we assume that the accelerometer data have to be integrated only for a short period of time, and the robot has to stop after this time.

To discuss our approach to double integration, let us first assume that all sensors are mounted on a stable carrier or platform. Moreover, the sensors are isolated from the external rotational motion resulting from movements of the applied gimbals and the application of rolling robotics or pedometers. As results from tests with long integration time [9]: ”Double integration of an acceleration error of 0.lg would mean a position error of more than 350 m at the end of the test”. There are, however, solutions (under development) used in pedometry and based on double integration of accelerometer data that do not indicate large position errors [10].

There are other problems with accelerometers which should be taken into consideration. Readings from accelerometers suffer from the loss of data precision, arising from the MEMS technology. There are descriptions of some problematic features of MEMS detailed in [11], [12]. In practice though, such features are to be modelled, since it is difficult to omit their impact on data degradation.

Typical issues associated with MEMS are their scale factor and bias. The scale factor, sometimes referred to as sensitivity, is the ratio of the change at the sensor output to the change at its input that is intended to be measured. Bias of an accelerometer can be identified as a non-zero offset of the sensor output signal from the expected true value, in m/s^2. Bias has no correlation with the input. It is measured over a certain time at specified operating conditions. Both characteristics are affected by factors related to material and construction, e.g. nonlinearity, hysteresis, cross-axis effects, etc. There are also factors related to the conditions of operation, such as temperature and pressure [12]. The bias and sensitivity (scale factor) can be improved using a proper calibration procedure.

Flicker noise occurs due to certain thermo-mechanical noise, fluctuations at a rate greater than the sampling rate of the sensor. As a consequence, the bias can wander over time. Such fluctuations can be modelled as random walk. Flicker noise creates a second-order random walk in velocity (uncertainty grows proportionally to $t^{3/2}$), and a third-order random walk in position (uncertainty grows proportionally to $t^{5/2}$). Thus double integration of a signal with fluctuating bias causes an error in position, which grows quadratically with time [13].

There are works which deeply analyse the nonlinearities of MEMS [14]. They consider internal resonators, oscillators, internal vibrations, etc. All of them affect the noise bias in MEMS. In some approaches the analysis of noise in MEMS is realized using an Allan Variance [1], [14]. The latter idea means a time domain analysis designed to characterize noise and stability in clock systems. The time of such analysis amounts to few hours. There are though no perfect solutions to eliminate the flicker noise.

The above mentioned problems with noise in MEMS as well as the difficulties of double integration are taken into account in the construction of a suitable model of a MEMS accelerometer and in the synthesis of appropriate transformation equations. For this purposes, we need to characterize the forces that have impact on the accelerometer.

II. Rotations and accelerometer data

In robotic applications of accelerometers one should calculate linear acceleration of an object (a robot) or some part of this object. However, accelerometer measurements include accelerations associated with various forces. Most important factors are linear movements, rotations, and the impact of gravity. Linear movement appears when the object changes its location. Acceleration connected with rotations is a consequence of the object movement related to its coordinates (external forces are neglected). On the contrary, gravity is caused by an external force, which can be easily described in global coordinates. Different coordinate systems require to define proper transformation matrices.

Assuming the Cartesian coordinate system the transformation between the object and global coordinates can be expressed as

$$\begin{bmatrix} a_x^w \\ a_y^w \\ a_z^w \end{bmatrix} = T_o^w \begin{bmatrix} a_x^o \\ a_y^o \\ a_z^o \end{bmatrix} \tag{1}$$

To ensure the clarity of used symbols, the global coordinates will be expressed as a_x, a_y, a_z and the objects coordinates as a_l, a_s, a_t (longitudinal, side, transversal component respectively). The relationship between the object and global coordinates can be expressed with rotation matrices and a translation vector. All transformations are written assuming that we deal with the right-handed system. Rotation matrices express rotations around the object *x* axis (roll), *y* axis (pitch), *z* axis (yaw) as shown in (2), (3), and (4), respectively. Due to the resulting size of following rotation matrices, the trigonometric functions are given in an abbreviated notation ($s\alpha = \sin\alpha, c\alpha = \cos\alpha$).

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \tag{2}$$

$$R_y(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \tag{3}$$

$$R_z(\psi) = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Combining all the 3 rotations R_x, R_y, R_z, the resulting rotation matrix can be expressed as follows:

$$\begin{aligned} R_{xyz} &= R_x(\psi)R_y(\theta)R_z(\phi) = \\ &= \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\phi s\psi + s\phi s\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix} \end{aligned} \tag{5}$$

As all the rotation matrices are unitary, it is easy to find their inverses. Note that due to inversion the sequence of the component matrices is reverted [15]:

$$R_{xyz}(\phi, \theta, \psi)^{-1} = R_z(\psi)^{-1} R_y(\theta)^{-1} R_x(\phi)^{-1} \tag{6}$$

Based on the rotation matrix, we can show how gravity affects acceleration in an actual object orientation. The gravity vector is constant in the global coordinates and in the opposite direction to the *z* axis. We assume that R_{xyz} means a transformation from the object to the global coordinate system. In a steady situation, without any movement of the object, the influence of gravity can be (statically) expressed as

$$\begin{bmatrix} a_l \\ a_s \\ a_t \end{bmatrix} = R_{xyz}^{-1} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = g \begin{bmatrix} s\phi s\psi + c\phi s\theta c\psi \\ -s\phi c\psi + c\phi s\theta s\psi \\ c\phi c\theta \end{bmatrix} \tag{7}$$

For a static situation, when an accelerometer measures only gravity, it is easy to calculate the actual rotation. Unfortunately, not all rotations cause changes in gravity measurements. With solely a yaw rotation there are no changes in the measured gravity (Fig. 4). There is also a gimbal lock problem which can occur while (further) working with rotations [16]. If the object is not only affected by gravity, but also by linear acceleration, the gravity substantially interferes with the measurements of linear acceleration. Therefore, gravity is the most crucial element in modelling an accelerometer.

Fig. 1. Tested sensor Pololu IMU9.

III. Accelerometer Model

In order to distinguish between acceleration measurements (accelerometer data) and accelerations which affect the accelerometer model, we shall refer to a measured longitudinal acceleration denoted as $\tilde{a}_l$. Considering a static model of an accelerometer, measurements are affected by gravity and linear acceleration. Such a model can be expressed by (8), which is a result of joining linear acceleration with (7),

$$\begin{bmatrix}\tilde{a}_l\\ \tilde{a}_s\\ \tilde{a}_t\end{bmatrix} = \begin{bmatrix}a_l\\ a_s\\ a_t\end{bmatrix} + g\begin{bmatrix}s\phi s\psi + c\phi s\theta c\psi\\ -s\phi c\psi + c\phi s\theta s\psi\\ c\phi c\theta\end{bmatrix} \tag{8}$$

The description is suitable for applications which use a gravity vector like bubble level, pinball etc. If linear acceleration of the object a_l occurs, measurements may indicate a small additional rotation or changes in gravity. The effect depends on the angle between a_l and g as well as on a_l/g.

If we can acquire data about the rotation angles ϕ, θ, ψ, e.g., by using sensor data fusion, it is possible to determine the linear acceleration of the sensor (before the object moves, and after this movement stops). However, to calculate the linear acceleration during the movement of the object, we should analyse rotation dynamics.

Let us assume that during a movement the object can invoke rapid rotations for a short period of time. This is a typical situation, e.g. when the robot travels on a rough ground. In real applications of IMUs not all sensors are implemented in a single chip. Typically, accelerometers and magnetometers are in one box, and a gyroscope is in a separate box. The distance between the chips in classical low-cost solutions (e.g. Pololu IMU-9 v.1 1) is about 1 cm. Moreover, it is difficult or impossible in real implementations to mount the sensor precisely at a point where all the 3 rotation axes of the object intersect. The rotation axis of the object is translocated as compared to the rotation axis of the sensor by a vector r (having r_l, r_s, and r_t as components). Thus, in practice, the impact of dynamic rotations on the accelerometer readings is sure to appear.

As the result of rotation dynamics the centrifugal and tangential accelerations should be taken into consideration. In Figs. 2, 3, and 4 the centrifugal acceleration a_{cf} is normal to the rotation direction and nonzero when a respective angular velocity, $\dot{\phi}, \dot{\theta}, \dot{\psi}$, is nonzero. The tangential acceleration a_{tg} is parallel to the rotation direction and takes nonzero value when the angular acceleration $(\ddot{\phi}, \ddot{\theta}, \ddot{\psi})$ is nonzero.

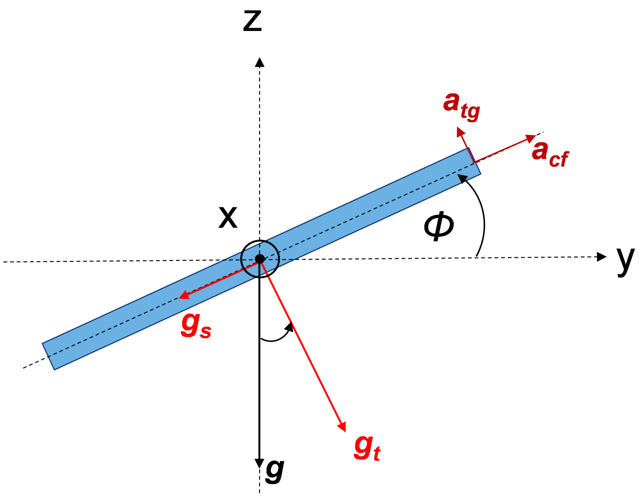


Fig. 2. Roll, rotation around the x axis.

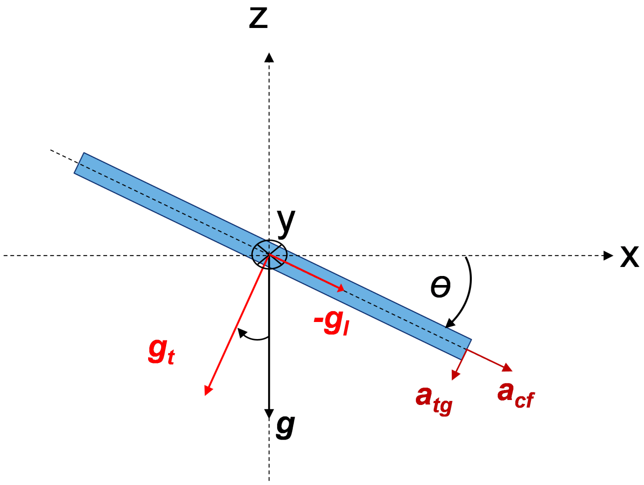


Fig. 3. Pitch, rotation around the y axis.

For roll, i.e. the rotation around the x axis in the right-handed coordinate system shown in Fig. 2, the centrifugal acceleration $a_{cf} = \dot{\phi}^2$ appears in the component $\tilde{a}_s$, and the tangential acceleration $a_{tg} = \ddot{\phi}$ is present in $\tilde{a}_t$. There is no influence of the roll dynamic rotation on the longitudinal component $\tilde{a}_l$.

Based on Figs. 3 and 4, we can describe the influence of the centrifugal and tangential accelerations for the other two rotations. For pitch rotation, $a_{cf} = \dot{\theta}^2$ is a component of $\tilde{a}_l$, and $a_{tg} = -\ddot{\theta}$ is the one of $\tilde{a}_t$. For yaw rotation, $a_{cf} = \dot{\psi}^2$ enriches $\tilde{a}_s$, and $a_{tg} = \ddot{\psi}$ appears in $\tilde{a}_t$. These results enable us to formulate a composed dynamic model of acceleration by incrementing 8 with a correction term as follows

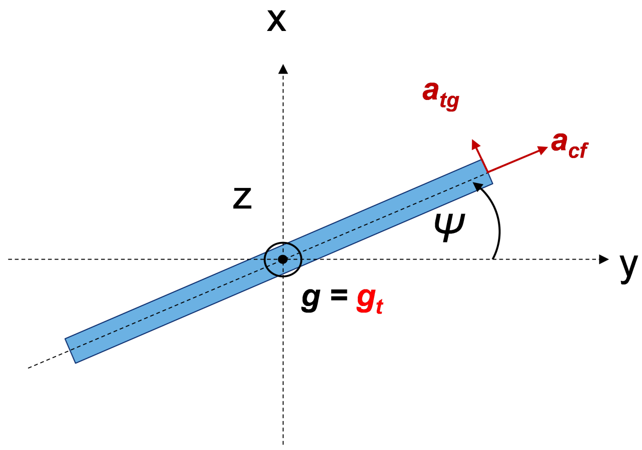


Fig. 4. Yaw, rotation around the z axis.

$$\begin{bmatrix}\tilde a_l\\ \tilde a_s\\ \tilde a_t\end{bmatrix} = \begin{bmatrix}a_l\\ a_s\\ a_t\end{bmatrix} + g\begin{bmatrix} s\phi s\psi + c\phi s\theta c\psi \\ -s\phi c\psi + c\phi s\theta s\psi \\ c\phi c\theta\end{bmatrix} + \begin{bmatrix}(\dot\theta^2+\ddot\psi)r_l\\ (\dot\phi^2+\dot\psi^2)r_s \\ (\ddot\phi-\ddot\theta)r_t\end{bmatrix} \tag{9}$$

This model includes the influence of dynamic movements of the object. There are also other forces, like precession and Coriolis forces, which are neglected at this stage of research. Nevertheless, certain issues related to the MEMS features, such as the previously mentioned scale factor, noise characteristics, and bias should be taken into account. Therefore, a proper calibration of the sensor appears to be mandatory to reduce the impact of these effects.

Based on the provided dynamic model of the object coordinates, it is now easy to express acceleration measurements in the global coordinates. Using the rotation matrix (5) the measured acceleration can be written in the following form:

$$\begin{bmatrix}\tilde a_x\\ \tilde a_y\\ \tilde a_z\end{bmatrix} = R_{xyz}\begin{bmatrix}a_l\\ a_s\\ a_t\end{bmatrix} + \begin{bmatrix}0\\ 0\\ g\end{bmatrix} + R_{xyz}\begin{bmatrix}(\dot\theta^2+\ddot\psi)r_l\\ (\dot\phi^2+\dot\psi^2)r_s \\ (\ddot\phi-\ddot\theta)r_t\end{bmatrix} \tag{10}$$

It is worth noticing that the gravity vector in the global coordinates is constant. However, the angles estimated with a certain accuracy will cause a gravity error in this equation.

IV. Estimation of Acceleration Data

Based on the described model of accelerometers, we can now derive an appropriate estimator. We assume that all the rotation angels ϕ, θ, ψ and rotation radius r are known in this model. Moreover, a Gaussian noise sequence with zero mean contaminates the measured data. Hence the following measurement model with noise (η) have to be considered:

$$\begin{bmatrix}\tilde a_l\\ \tilde a_s\\ \tilde a_t\end{bmatrix} = \begin{bmatrix}a_l\\ a_s\\ a_t\end{bmatrix} + g\begin{bmatrix} s\phi s\psi + c\phi s\theta c\psi \\ -s\phi c\psi + c\phi s\theta s\psi \\ c\phi c\theta\end{bmatrix} + \begin{bmatrix}(\dot\theta^2+\ddot\psi)r_l\\ (\dot\phi^2+\dot\psi^2)r_s \\ (\ddot\phi-\ddot\theta)r_t\end{bmatrix} + \begin{bmatrix}\eta_l\\ \eta_s\\ \eta_t\end{bmatrix} \tag{11}$$

In real applications the actual rotations are known only to a certain degree. To determine the linear movement of the object, we propose using an estimator described as follows

$$\begin{bmatrix}\hat a_l\\ \hat a_s\\ \hat a_t\end{bmatrix} = \begin{bmatrix}\tilde a_l\\ \tilde a_s\\ \tilde a_t\end{bmatrix} - g\begin{bmatrix} s\hat\phi s\hat\psi + c\hat\phi s\hat\theta c\hat\psi \\ -s\hat\phi c\hat\psi + c\hat\phi s\hat\theta s\hat\psi \\ c\hat\phi c\hat\theta\end{bmatrix} - \begin{bmatrix}(\dot{\hat\theta}^2+\ddot{\hat\psi})r_l\\ (\dot{\hat\phi}^2+\dot{\hat\psi}^2)r_s \\ (\ddot{\hat\phi}-\ddot{\hat\theta})r_t\end{bmatrix} \tag{12}$$

where the estimated value of linear acceleration is denoted as $\hat a$ and the estimated angles are represented by $\hat\phi = \phi + \Delta\phi$ (an ideal value of the angle plus an error). The error values should be set adequately to the precision of determining the angle values (by using the sensor fusion) and the sensor noise which is characteristic for MEMS. Furthermore, we need to calculate the first and second derivatives of the angles. As the angle error $\Delta\phi$ generates noise, the differentiated noise can produce undesirable rapid changes in the estimated linear acceleration. Thus the estimated angles should be processed by a low pass filter to reduce spikes in estimated acceleration.

V. Experiments and Simulations

The described model and the estimator were examined in a simulation study. Tests were performed in the Matlab environment using generated angle signals and linear acceleration signals. The following 3 variants of test signals were applied:

- angle step in the x axis rotation (roll)
- move which simulates the behaviour of a boat when a large wave arise
- movement of a robot on a manoeuvring square.

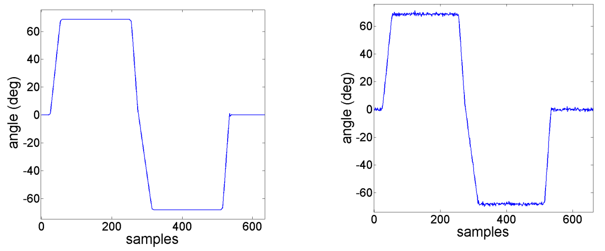
Generated signals (angle and acceleration) have rapid amplitude changes (an unlimited spectrum). To ensure that these signals are similar to real accelerometer data they were processed with the use of a low-pass FIR filter of order 10, attenuation 30 dB, normalized cut-off frequency 0.6 (the associated delay caused by filtration was removed).

Moreover, the generated angle signals (ϕ, θ, ψ) were modified with the Gaussian white noise with the zero mean and the variance selected according to the typical precision of sensor fusion algorithms [10].

Similarly, the generated acceleration signals (a_l, a_s, a_t) were modified with the Gaussian white noise of the zero mean, and the variance 0.03 g. The variance value was chosen according to noise density (220), and zero-g level offset accuracy, as specified in the Pololu accelerometer documentation [17].

A. Correction term for roll step signals

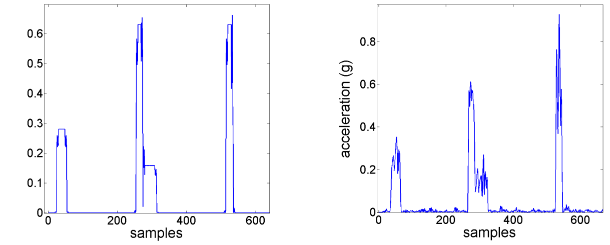
The simulation was intended to check how the rotation dynamics influences the measured acceleration according to the derived model, in particular, the dynamical correction term



(a) Roll step signal.

(b) Roll step signal with noise.

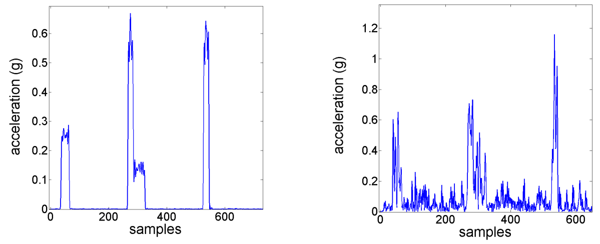
Fig. 5. Test signal: roll step.



(a) Correction term for roll step.

(b) Correction term for noise.

Fig. 6. Rotation dynamics (correction terms) for roll step.



(a) The term for low noise.

(b) The term for high noise.

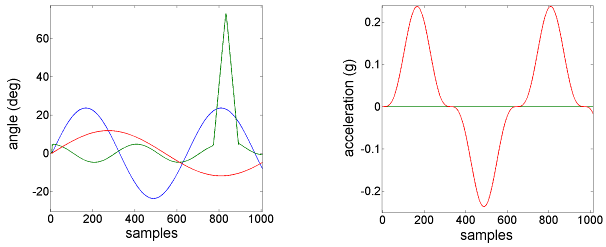
Fig. 7. Rotation dynamics for roll step (2 other noisy variants).

in 9 and 11 and the roll step testing signals (note that the influence of other angles is neglected/nullified).

The angle signal for the roll step after low pass filtering and noise contamination (with the variance 0.7°), are shown in Fig. 5, subfigures (a) and (b), respectively.

The resulting dynamical correction term (impact of rotation on acceleration) for a displacement of the sensor $r = 5cm$ in the roll step test of Fig. 5, is illustrated in Fig. 6 for the second coordinate of the term ($\dot{\hat{\phi}}^2$). Note that the term can exceed the acceleration of 0.5 g. Thus the dynamical correction represents an essential factor in acceleration estimation. According to other observations, peaks existent in the correction term (see Fig. 6) grow linearly with the effective sensor displacement r (projected on the x axis).

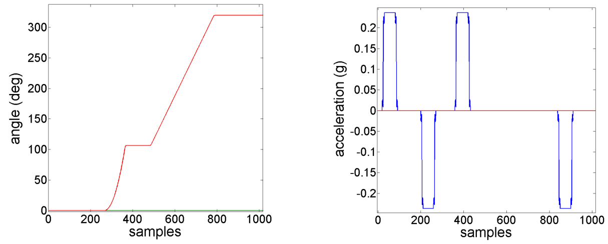
The correction term in the presence of lower measurement noise (with the variance 0.2°) and higher noise (with the variance 2.0°) are illustrated in Fig. 7. As can be seen from the Fig. 7b, the rotation dynamics can be destructive for highly contaminating noise (approximately, starting from 2.0°). Probably, this effect can be improved by suitable filtration.



(a) Angular signals (ϕ - blue, θ - green, ψ - red).

(b) Linear acceleration (a_l - blue, a_s - green, a_t - red).

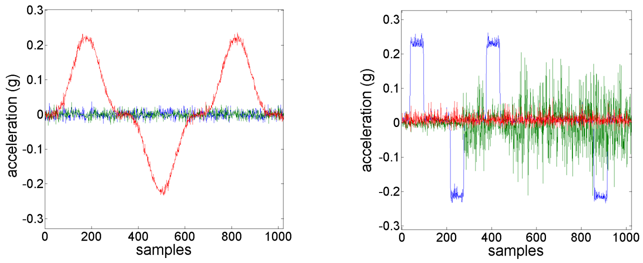
Fig. 8. Simulated input signals of boat movement.



(a) Angular signals (ϕ - blue, θ - green, ψ - red).

(b) Linear acceleration (a_l - blue, a_s - green, a_t - red).

Fig. 9. Simulated input signals of manoeuvring robot.



(a) Linear acceleration estimation of boat movement (ϕ - blue, θ - green, ψ - red).

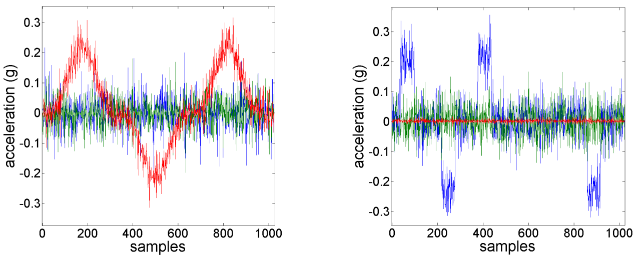
(b) Linear acceleration estimation of manoeuvring robot (a_l - blue, a_s - green, a_t - red).

Fig. 10. Linear acceleration estimation for two test and low noise.

B. Acceleration estimation using generated measurements

The low-pass filtered signals (generated according to the typical precision of accelerometers [17]) were used to compute the locally measured, linear acceleration 9, denoted with , and next applied in the estimator ((12)), using the knowledge about the angles gained from separate gauges. The angle information from such gauges, denoted with ˆ, was generated according to the above described rules (LPF plus Gauss). The obtained 3D linear acceleration and angular (rotation) signals corresponding to the simulated boat movement and the manoeuvring robot, are shown in Fig. 8 and Fig. 9, respectively.

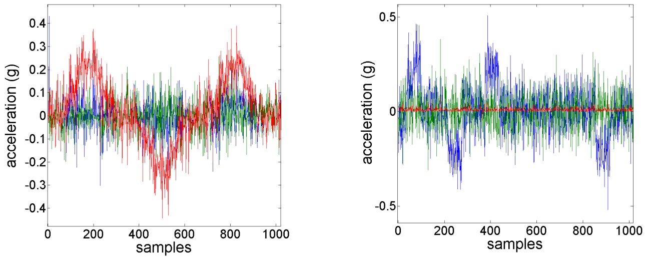
For the considered stage of preliminary estimation the noise level (0.7°) corresponding to the sensor fusion precision proved to be too high. Therefore, we found experimentally that relatively good signal estimation can be obtained for the noise variance lower than 0.02°. The estimated linear acceleration signals for both tests signals are shown in Fig. 10.



(a) Linear acceleration estimation of boat movement (ϕ - blue, θ - green, ψ - red).

(b) Linear acceleration estimation of manoeuvring robot (a_l - blue, a_s - green, a_t - red).

Fig. 11. Linear acceleration estimation for two test and medium noise.



(a) Linear acceleration estimation of boat movement (ϕ - blue, θ - green, ψ - red).

(b) Linear acceleration estimation of manoeuvring robot (a_l - blue, a_s - green, a_t - red).

Fig. 12. Linear acceleration estimation for two test and high noise.

The estimator output for larger noise levels are presented in Fig. (11) and Fig. (12). We set the variance to 0.1° and 0.3°, respectively. Higher values than 0.3° resulted in improper estimation. Perhaps, additional noise suppression or high frequency signal filtration may enhance the results.

In the presence of low noise in the angular data we found that the rotation dynamics (correction term) is important at rapid movements (at the beginning or/and the end of them). While the noise increases, the rotation dynamics deteriorate the estimation precision. Taking into account the detrimental derivatives, in case of very high noise the dynamical correction term in (12) should rather be disregarded. Nevertheless, to find linear acceleration using a low-cost accelerometer much attention should be paid to noise filtration.

VI. Conclusion

The derived dynamic model of a MEMS accelerometer operates properly for dynamic signals tested in our simulation study. The influence of the dynamic rotations is significant for small displacements of the sensors. For instance, the measured acceleration increases by 0.8 g even for a displacement of $r = 5cm$.

Moreover, the estimation tests performed proved that the modelled rotation is sensitive to the angle error. The angle precision of 0.05° is sufficient for computing the value of linear acceleration from accelerometer measurements. It appears that this type of errors can be suppressed via proper filtration.

In future research we plan to enhance the system performance using additional signal processing. The anticipated tests should include both simulated signals and real data from accelerometers. The expected results should also enhance the functionality and fidelity of the VisRobot system in terms of 3D mapping. Another type of strategy for robot positioning is expected for the case of robots moving on large distances.

References

[1] Y. Wang, S. Adams, J. Thorp, N. MacDonald, P. Hartwell, and F. Bertsch, "Chaos in MEMS, parameter estimation and its potential application," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 45, 1998, pp. 1013 – 1020.

[2] R. Leniowski and L. Leniowska, "MEMS motion tracking system for surgical robot ROCH-1," in *International Conference on Methods and Models in Automation and Robotics*, 2013, pp. 508–513.

[3] Y.-A. Chapuis, L. Zhou, Y. Fukuta, Y. Mita, and H. Fujita, "FPGA-Based Decentralized Control of Arrayed MEMS for Microrobotic Application," in *IEEE Transactions on Industrial Electronics*, vol. 54, 2007, pp. 1926 – 1936.

[4] W. H. Ko, "Trends and frontiers of MEMS," *Sensors and Actuators, A: Physical*, vol. 136, no. 1, pp. 62–67, 2007.

[5] M. Pedley, "Tilt Sensing Using a Three-Axis Accelerometer," in *Freescale Semiconductor Application Note*, 2013.

[6] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration," in *International Conference on Intelligent Robots and Systems*, 2011, pp. 3297–3304.

[7] Y. Maki, S. Kagami, and K. Hashimoto, "Localization and tracking of an accelerometer in a camera view based on feature point motion analysis," in *Proceedings of SICE Annual Conference*, 2012, pp. 293 – 294.

[8] R. Szeliski, *Computer Vision*, ser. Algorithms and Applications. Springier, 2010, vol. 5.

[9] S. Y. Khoo, P. K. Khosla, and C. N. Riviere, "Physical model of a MEMS accelerometer for low-g motion tracking applications," in *International Conference on Robotics and Automation*, vol. 2, no. April, 2004, pp. 1345–1351.

[10] S. O. H. Madgwick, A. J. L. Harrison, and A. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *International Conference on Rehabilitation Robotics*. IEEE, Jan. 2011, p. 5975346.

[11] F. Mohd-Yasin, D. J. Nagel, and C. E. Korman, "Noise in MEMS," *Measurement Science and Technology*, vol. 21, no. 1, pp. 12–21, 2009.

[12] A. Albarbar, A. Badri, J. K. Sinha, and A. Starr, "Performance evaluation of MEMS accelerometers," *Measurement: Journal of the International Measurement Confederation*, vol. 42, no. 5, pp. 790–795, 2009.

[13] G. Pang and H. Liu, "Evaluation of a Low-cost MEMS Accelerometer for Distance Measurement," *Journal of Intelligent and Robotic Systems*, vol. 30, no. 3, pp. 249 – 265, 2001.

[14] D. K. Agrawal, J. Woodhouse, and A. A. Seshia, "Modeling nonlinearities in MEMS oscillators," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 60, no. 8, pp. 1646–1659, Aug. 2013.

[15] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Reston, USA: The Institution of Electronical Engineers, 2004, vol. 17.

[16] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Cambridge, UK, Tech. Rep. 696, 2007.

[17] STMicroelectronics, "Sensor module LSM303 datasheet," STMicroelectronics, Tech. Rep. Doc ID 023312 Rev 1, 2011.

[18] G. A. Aydemir and A. Saranl, "Characterization and calibration of MEMS inertial sensors for state and parameter estimation applications," *Measurement*, vol. 45, no. 5, pp. 1210–1225, Jun. 2012.

[19] Z. Kowalczuk and T. Merta, "Stereo image visualization for VISROBOT system," in *International Conference on Methods and Models in Automation and Robotics*, 2013, pp. 794–799.