

# Indoor mobile robot attitude estimation with MEMS gyroscope

Radostaw Cechowicz<sup>1,\*</sup>

<sup>1</sup>Lublin University of Technology, Mechanical Engineering Faculty, Department of Automation, Nadbystrzycka 36, 20-618 Lublin, Poland

**Abstract.** A method of attitude estimation with a low-cost, strap-on MEMS sensor was proposed in the article. The method relies on dynamic gyroscope bias change estimation and updates during detected stops of the robot. The algorithm has been tested in laboratory with iRobot Roomba robot and should be improved to be useful in an industrial environment. The measurement of attitude of a mobile platform is necessary for correct navigation, especially in autonomous vehicles, which use dead reckoning between position updates from external sources. Since the system is intended to operate indoors, in industrial halls and shops, it cannot avail of GPS and Earth magnetic field sensors because of anomalies, which are common inside the steel constructions. Therefore, the accuracy of the gyroscope-based attitude estimation is significant. The proposed method aims to address the common problem of gyroscope bias drift, by dynamic update of sensor bias and simultaneous use of all gyroscope axes, to improve the quality of the measurements. A popular 3-axial gyroscope and 3-axial accelerometer sensors were used during the test runs. Obtained results suggest that it is possible to improve short-term accuracy of inertial dead reckoning, to get a system that could be of practical use in industrial AGV systems or intelligent vehicles.

## 1 Introduction

One of the impedimenta to the development of mobile platforms, designed to work in industrial halls and public buildings, are the difficulties in identifying the position of the platform (in relation to other elements of the environment/building). In industrial AGV systems, as well as those operating in warehouses, this problem is usually solved by installing fixed route markings or way-points. This can be done by using contrasting paint, magnetic tape, inductive circuits, or surface modifications (guiding canals, rails, kerbs, etc.). Precise position of the mobile unit can also be determined by reading or counting route markings — for example bar codes, QR codes, RFID tokens, etc. In some systems, the position of the moving object, in relation to fixed beacons, can be established by triangulation [1,2].

According to [3], industrial navigation systems share a common problem of limited flexibility. The mobile units in such systems are forced to follow fixed routes (or go over fixed way-points). In consequence, the mobile units can only follow a limited number of routes. Additionally, in some solutions, the space used by the mobile units (communication paths) cannot be shared or allocated to other tasks (such as temporary storage). Transport routes can only be changed by moving or installing new infrastructure (route markings) and reserving new space for the AGV system [4].

If the autonomy of mobile platforms was increased by

detaching them (partially or completely) from permanently marked routes, it would allow [3]:

- to increase the flexibility of the system, allowing an adjustment of transport routes to the dynamic change of the environment (e.g. the emergence of obstacles, temporary route closure, etc.).
- to reduce the number of fixed way-points, decreasing the cost of the infrastructure.

Detaching the mobile units from fixed paths would be possible, if their local control systems are able to navigate autonomously over a (limited) time. In the areas where fixed markings are not available, the system could determine its position from dead reckoning (reading onboard sensors), and react to events nearby.

Orientation of mobile platform can be established from odometry, inertial sensors or a combination of these two. The possible solutions have been documented, for instance in [5–7].

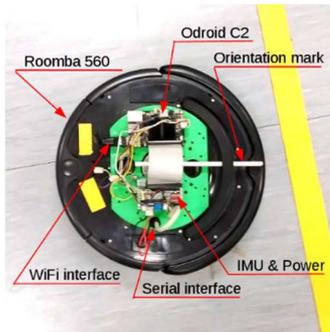
## 2 The experiment setup

The purpose of the experiment was to determine the accuracy of the relative orientation angle measurement of a mobile platform with a low-cost, strap-on inertial gyroscope sensor.

The measurements were taken in the conditions similar to those of industrial warehouse mobile platforms – the sensor was mounted onto the robot (iRobot Roomba was used for this purpose), which was moving on an even surface, similar to those in clean warehouses (PVC cover).

\* Corresponding author: [r.cechowicz@pollub.pl](mailto:r.cechowicz@pollub.pl)

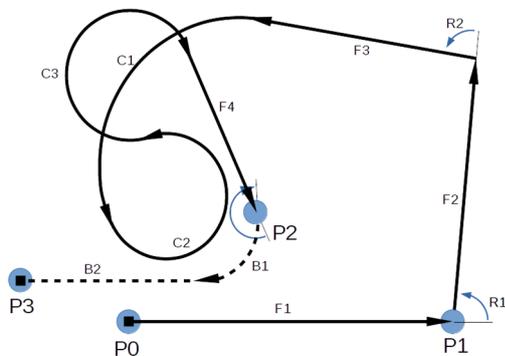
The 6-axis IMU unit consisting of MEMS gyroscope (L3GD20H) and accelerometer (LSM303D) was used during the experiment. The sensor was mounted onto the modified iRobot Roomba robot (fig. 1). The sensor was attached to the Odroid C2 single board computer (control unit) via the I2C interface. The computer was also used to control the robot and to read sensor data via the Roomba serial interface. All data collected during the experiment was stored on-board and then sent to central system for processing.



**Fig. 1.** Modified iRobot Roomba used in the experiment.

During the test runs, the robot was controlled by the single-board computer in the open-loop mode (without position feedback). The pre-programmed and timed sequence of movements was being sent to the Roomba, and the inertial sensor and odometry data were read.

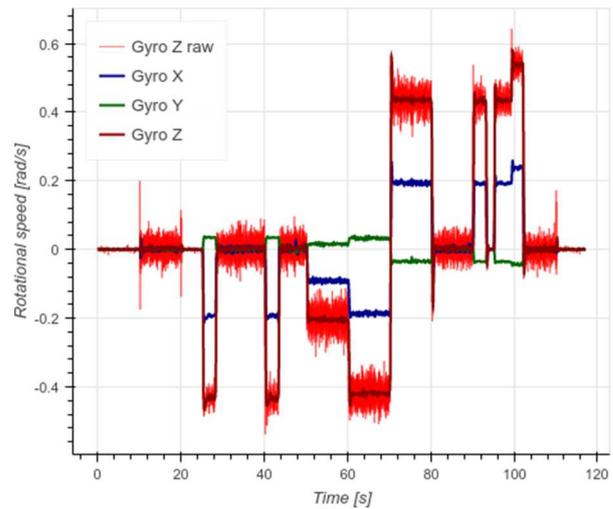
Since the main purpose of the experiment was to test the gyroscope, a custom sequence of movements (fig. 2) was chosen, instead of standard square path (proposed by [8]), to simulate various working conditions of the warehouse platform, such as various turn radii and forward and backward movements. The real orientation angle of the robot was measured optically, in defined stop points along the path, from the picture taken by a camera above the robot. The optical measurements were taken by relating the position of the robot to the markings on the floor (vertical and parallel lines, markers). The accuracy of optical measurement was about  $\pm 1$  deg. The area covered by the robot path fitted within the rectangle of 3 by 4 m. Time of a single round (including control stops) was 250 s.



**Fig. 2.** Robot path during the tests (approximate): P0 - starting point, P1 - first control point, P2 - second control point, P3 - end point, Rx - in-place rotation, Fx, Cx - straight and curved path forward movement (continuous line), Bx - backward movement (dashed line).

### 3 Measurements

The IMU sensor sampling rate was 100 Hz. During the test runs, the IMU was fixed to the Roomba in a position where none of the axes were perpendicular to the floor. The final rotation in the floor plane was then calculated from vector (quaternion) transformation. A sample of recorded gyroscope signal is shown in fig. 3. In the figure, there are filtered (lowpass Butterworth  $f_{cut}=0.05f_{sample}$ ) signals shown for each axis, and a raw signal from axis Z to illustrate the gyro noise. The noise level in the remaining two axes was similar. All gyroscopes were calibrated during the first 10s of the test run.



**Fig. 3.** Gyro signal recorded during test run.

Normalised vector  $A_v$  perpendicular to the floor was derived from accelerometer:

$$A_v = [A_x, A_y, A_z] / \|[A_x, A_y, A_z]\| \quad (1)$$

where:  $A_x, A_y, A_z$  – stand for values of accelerometer readings while the robot was idle (not moving).

Rotation vector at the moment  $t$  can be obtained from gyroscope readings (scaled in rad/s), using quaternion mathematics, commonly applied to rotation, orientation and tracking problems in navigational systems [9–12].

For any sufficiently small  $dt$ , a rotation vector can be derived directly from 3-axis gyroscope reading:

$$r_v = [g_x, g_y, g_z] \quad (2)$$

where:  $r_v$  – rotation vector,  $g_x, g_y, g_z$  – gyroscope readings (axes X, Y and Z) scaled to rad/s. Rotation vector can be converted to quaternion  $Q$  (or angle-vector representation) with the following:

$$Q = q(\|r_v\|dt, r_v/\|r_v\|) \quad (3)$$

The (normalised) unit quaternions can be used for tracking the orientation of an object in space, especially, when the rotation axis changes frequently, such as in the case of wearable sensors [11]. For the purpose of the experiment, an assumption was made that the floor plane is horizontal (perpendicular to the Earth gravity vector) and does not change during the run time. The vertical

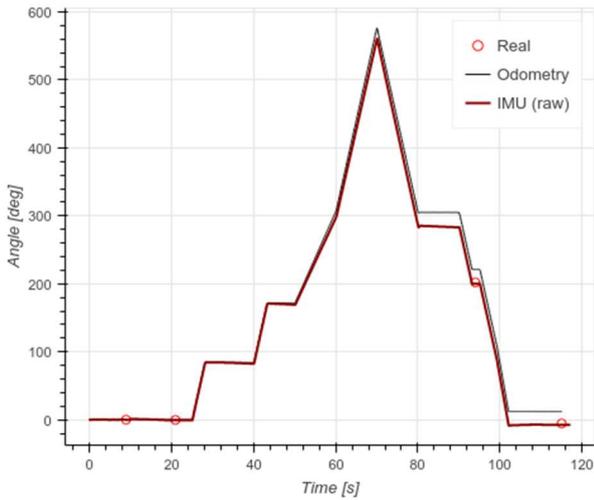
direction, obtained from (1) was assumed to be constant. The rotation about the axis  $A_v$  is then a dot product of:

$$R_v = \text{dot}(r_v, A_v) \quad (4)$$

The relative orientation angle is then a sum of  $R_v$  over time:

$$O = \int_0^t R_v dt \quad (5)$$

For a short time periods, orientation angle calculated with (5) gives good estimation of robot attitude. The obtained results (fig. 4) are similar to those documented in [5,6]. The attitude error from gyroscope signal is smaller than the odometry error.



**Fig. 4.** Relative orientation angle (IMU) compared to orientation from Odometry and optical measurements in control points (Real)

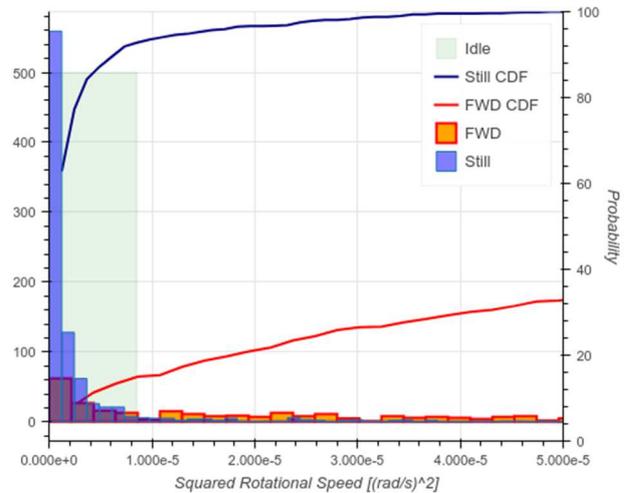
The result presented in fig. 4 is valid only for short periods of time, for measurements taken immediately after gyroscope calibration, and for the sensor working in stable thermal conditions. For any longer integration periods, the attitude error would grow because of gyro bias fluctuations.

#### 4 Sensor movement detection

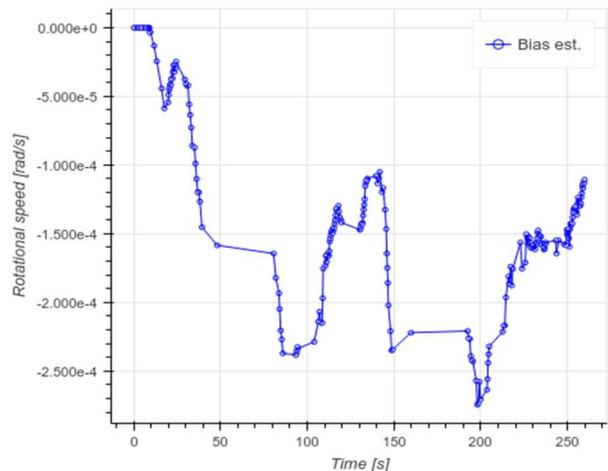
It was shown in [3] that even in a static MEMS gyroscope, the bias changes (drifts) constantly over time. It is possible to compensate the integration error from bias drift using a filtering [13] or modelling the process [14] but this could be difficult to perform in an industrial system, consisting of a number of gyros or for a low-cost or mass produced system.

Periodic, runtime calibration could be used to follow the drift of the gyroscopic sensors and to keep the attitude error low. This can be done on a hardware level, as proposed by [15], or by updating sensor bias (as in [3,16]). Bias value can be updated while the sensor remains stationary. Therefore, in such system a movement detection procedure is required. In [3], a statistical filter was proposed for this purpose. For a mobile robot, a simple classification algorithm could be used, since the state of moving can be determined by measuring vibration noise.

In fig. 4, movement can be easily noticed by the increased noise in the sensor output. Following this observation, a simple measure of sample range could be probably used for detection. However, it provided only rough estimation of movement. It turned out that better and more exact rotational movement classification can be done with measure proportional to signal energy. In fig. 5, there are two histograms shown: one for the still robot (Still), the other for the robot moving forward (FWD, only leftmost part of the histogram shown) and their cumulative distribution functions (CDF) calculated from data. It can be noted that slightly less than 20% of movement samples are in the overlap with still area. It is assumed that robot is not turning at these points. The remaining samples are classified as rotating and included into attitude estimation.



**Fig. 5.** Rotational speed histograms of the idle and moving robot (section F1) and cumulative distribution functions (CDF) from data. Idle area marked in light blue.

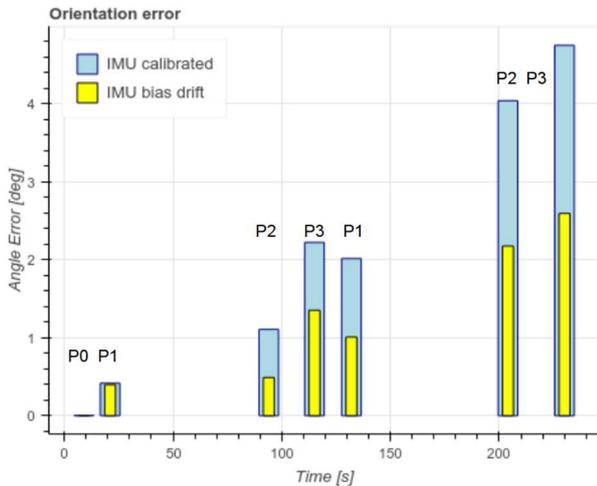


**Fig. 6.** Estimated sensor bias change during test run.

The robot was slightly turning during all the straight-line segments, due to unequal effective diameter of the wheels. The average rotational speed of the robot during the straight-line segments, was about 0.0005 rad/s. The classification method was then used to update the sensor bias value during the test runs. In the points classified as still (no rotational movement) that bias of

the sensor was updated, using a complementary filter with parameters determined from experiment. The resulting bias change after first calibration is shown in fig. 6. The plot was made from a test run composed of two rounds along defined path (fig. 2), one immediately after the other. It can be noted that the general shape of bias changes is similar for both rounds, which can suggest a correlation of bias and movement history.

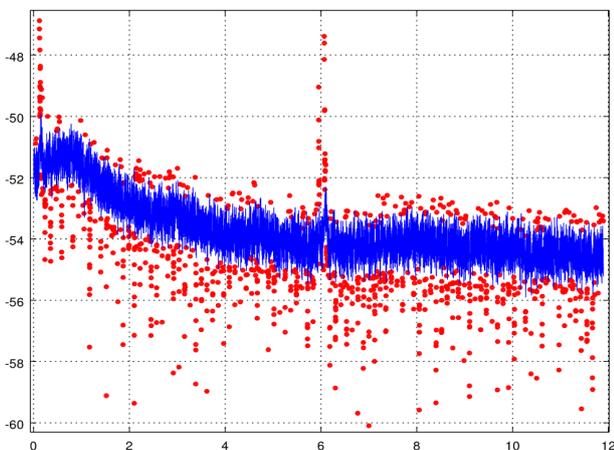
The bias drift estimation allowed to significantly decrease the attitude error. The comparison of the obtained results is shown in fig. 7.



**Fig. 7.** Comparison of attitude estimation results for calibrated IMU and IMU with bias drift estimation, against real orientation of the robot in defined control points.

## 5 Conclusion

The proposed method of sensor bias change estimation allows to decrease the attitude error by about 40% in comparison to calibrated-only approach. Since the gyroscopic signal must be integrated to obtain orientation angle, bias change brings major contribution to attitude error. Bias changes registered during the 12 hours-long still test with the sensor used in the experiment are shown in fig. 8.



**Fig. 8.** Sensor bias change estimate for 12 hours still test [3].

The bias does not change linearly – it is influenced by the temperature (of the sensor itself and of the environment), fluctuations in power supply, reading procedure and other, less known factors.

The use of quaternions or rotation vectors (in simplified case) simplifies the calculations and makes possible to include measurements from more than one axis, which results in better statistical properties of the signal, or allows to easily incorporate inertial measurement in tracking applications like robot arm steering [17].

The research will be continued, the movement detection algorithm should be improved to increase its selectivity. The error distribution in the control points suggest that the slight drifting of the robot during the forward movement is not fully measured.

## References

1. F. Ijaz, H. K. Yang, A. W. Ahmad, C. Lee, 15th International Conference on Advanced Communications Technology (ICACT), 1146–1150 (IEEE, 2013)
2. R. Mautz, *Geod. Kartogr.* **35**, 18 (2009)
3. R. Cechowicz, *Acta Mechanica et Automatica.* **11**, 2, 104–110 (2017)
4. D. Herrero-Pérez, J.J. Alcaraz-Jiménez, H. Martínez-Barberá, *Int. J. Adv. Rob. Syst.* **10**, 292 (2013)
5. M. Guráň, T. Fico, A. Chovancová, F. Duchoň, P. Hubinský, D. Dúbravský, 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), 1–7 (2014)
6. N. Houshangi, F. Azizi, *World Automation Congress*, 1–8 (2006)
7. B. Gersdorf, U. Freese, *ICINCO*, **1**, 388–395 (2013)
8. J. Borenstein, L. Feng, *IEEE Trans. Rob. Autom.* **12**, 869 (1996)
9. A. Janota, V. Šimák, D. Nemeč, J. Hrbček, *Sensors.* **15**, 7016–7039 (2015)
10. R. G. Valenti, I. Dryanovski, J. Xiao, *Sensors.* **15**, 19302–19330 (2015)
11. V. Renaudin, C. Combettes, and F. Peyret, *IEEE/ION Position, Location and Navigation Symposium*, 645–656 (PLANS 2014)
12. A. Kim and M. F. Golnaraghi, in *Position Location and Navigation Symposium*, 268–272 (IEEE, 2004)
13. X. Yuan, S. Yu, S. Zhang, G. Wang, S. Liu, *Sensors.* **15**, 10872–10890 (2015)
14. L. Xue, C. Jiang, L. Wang, J. Liu, W. Yuan, *Micromachines.* **6**, 266–280 (2015)
15. I. P. Prikhodko, C. Merritt, J. A. Gregory, J. A. Geen, J. Chang, J. Bergeron, W. Clark, M. W. Judy, *Transducers – 2015, 18th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*, 35–38 (IEEE, 2015)
16. S. K. Hong, S. Park, *Sensors.* **8**, 7287–7299 (2008)
17. M. M. Plechawska-Wójcik, P. K. Wolszczak, R. J. Cechowicz, K. Łygas, 9th International Conference on Human System Interactions (HSI), 348–354 (IEEE, New York, 2016)