Integration of Inertial Sensor Data into Control of the Mobile Platform

Rastislav Pirník[1], Marián Hruboš[1](✉), Dušan Nemec[1],
Tomáš Mravec[1], and Pavol Božek[2]

[1] Faculty of Electrical Engineering, University of Žilina, Žilina, Slovak Republic
{rastislav.pirnik,marian.hrubos,dusan.nemec,
tomas.mravec}@fel.uniza.sk

[2] Faculty of Materials Science and Technology, Institute of Applied Informatics,
Automation and Mechatronics, Slovak University of Technology,
Trnava, Slovak Republic
pavol.bozek@stuba.sk

Abstract. The paper presents the designed algorithm, which is able to integrate of inertial sensor data into control algorithm. Autonomous operation of the mobile system requires reliable measurement of its position. Sources of such data are various; most commonly used is global satellite navigational system. However, this technique can be used only outdoors. For navigation inside building, under metal roof or underground only inertial or contact methods are available. This article analyzes possibilities of deployment of the inertial navigation in the control of the wheeled mobile platform. Experimental platform uses inertial measurement unit x-IMU manufactured by x-IO Technologies. According to our experiments inertial navigation can be reliably used only in fusion with other absolute sensors (odometers, magnetometers).

1 Inertial Navigation Principle

Inertial navigation uses sensors, in this case accelerometers and gyroscopes and is being used to locate objects e.g. inside buildings, in air and road traffic etc. Today, the inertial navigation systems is not used alone for short time because with the increasing time the error increases, too. Recent progress in the development is a fusion of GPS data into inertial systems using Kalman filtering. These systems can be supplemented by the barometric or magnetic sensors.

Inertial navigation [4, 5] computes position and attitude of the moving object with respect to its starting position in inertial (non-accelerating) frame of reference. With a small error we might consider surface of the Earth as an inertial base (the error will be discussed later). In the most general case the movement of the rigid object is described in six degrees of freedom – 3 degrees of translation and 3 degrees of rotation. In special case of the wheeled vehicle moving on the horizontal floor only three degrees of freedom are relevant:

- translation in two planar axes (North, East)
- rotation around vertical axis (Yaw)

J. Janech et al. (eds.), *Proceedings of the 2015 Federated Conference on Software Development and Object Technologies*, Advances in Intelligent Systems and Computing 511,
DOI 10.1007/978-3-319-46535-7_21

The other degrees of freedom are required if the surface of the floor is not planar (e.g. contains variable slopes and multiple levels).

2 Attitude Estimation

Attitude (rotation of the object around three axes) defines the transformation between the global coordinate system (bound with the Earth) and local coordinate system (bound with the vehicle). It can be computed in real time from the initial attitude and readings of 3-axial gyroscope (measures angular velocity in local system). Note that without knowing the initial attitude it is impossible to determine actual attitude. Inertial measurement unit x-IMU therefore contains 3-axial accelerometer for roll and pitch estimation and magnetometer for yaw estimation. Accelerometer measures the sum of the gravitational acceleration vector [7] (constant in global coordinate system, defines vertical direction) and the system's own acceleration (can be arbitrary but its mean value in long terms are zero). Magnetometer measures magnetic induction of the Earth's magnetic field (defines magnetic North). This configuration allows fully compensated attitude estimation. Basic sensor fusion [1] algorithms are implemented in default x-IMU firmware.

Attitude can be expressed in the form of Euler angles, rotational matrix or quaternion. Available x-IMU API provides conversion algorithms from quaternions to the other formalisms. Our algorithms use Euler angles in Z-Y-X convention (yaw-pitch-roll) because of their readability by humans and rotational matrix because it allows fast transformation between global and local system and can be easily used in MATLAB® environment. Coordinate systems (both global and local) are Cartesian with North-East-Down (abbr. NED) axis orientation.

3 Influence of the Earth's Rotation

Small portion of the error in yaw estimation is caused by rotation of the Earth around its own axis at angular velocity $\omega_{\text{Earth}} = 7.292115 \times 10^{-5}$ rad.s^{-1} (approximately 15°/h). For wheeled vehicle the worst case occurs around Earth's poles, because vertical complement of the Earth's angular velocity is larger at higher latitude (see Fig. 1) and yaw estimation provided by magnetometer is less precise.

Since the angular velocity of the Earth's rotation is well known, readings from the gyroscope can be compensated by following:

$$\omega_{\text{comp}} = \omega_{\text{raw}} - \boldsymbol{R}.\omega_{\text{Earth}}\begin{bmatrix} -\cos\phi & 0 & \sin\phi \end{bmatrix} \tag{1}$$

where $\boldsymbol{R}$ is rotational matrix expressing attitude with respect to local tangent plane in NED convention on the Earth's surface and ϕ is geographical latitude.

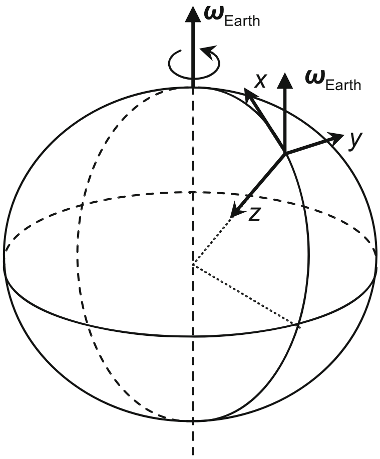

Fig. 1. Rotation of the Earth in local tangent plane

4 Position Estimation

According to the theory of inertia the position of the object can be integrated from acceleration of the object in local coordinate system $\boldsymbol{a}_{\text{local}}$, current rotational matrix $\boldsymbol{R}$, initial velocity $\boldsymbol{v}_0$ and position $\boldsymbol{d}_0$ in global inertial frame of reference. Algorithm of the integration is following:

1. Initialize state variables:

$$\boldsymbol{a} = \boldsymbol{0}, \boldsymbol{v} = \boldsymbol{v}_0, \boldsymbol{d} = \boldsymbol{d}_0 \tag{2}$$

2. Wait for new sample from accelerometer and AHRS (Attitude and Heading Reference System)

$$\boldsymbol{a}_{\text{accel}} = \boldsymbol{a}_{\text{local}} + \boldsymbol{R} \cdot \boldsymbol{g} \tag{3}$$

3. Compute new acceleration in global coordinate system:

$$\boldsymbol{a}_{\text{new}} = \boldsymbol{R}^{-1} \cdot \boldsymbol{a}_{\text{accel}} - \boldsymbol{g}, \tag{4}$$

   where $\boldsymbol{g}$ is constant gravitational acceleration vector $\boldsymbol{g}$ = [0,0, g].
4. Compute new velocity in global coordinate system (trapezoidal integration):

$$\boldsymbol{v}_{new} = \boldsymbol{v} + (\boldsymbol{a}_{new} + \boldsymbol{a})\frac{T}{2}, \tag{5}$$

   where T is sampling period.
5. Compute new position in global coordinate system (trapezoidal integration):

$$\boldsymbol{d}_{new} = \boldsymbol{d} + (\boldsymbol{v}_{new} + \boldsymbol{v})\frac{T}{2} \tag{6}$$

6. Save state, then go back to step 2.

$$\boldsymbol{a} = \boldsymbol{a}_{\text{new}}\ ,\ \boldsymbol{v} = \boldsymbol{v}_{\text{new}},\ \boldsymbol{d} = \boldsymbol{d}_{\text{new}} \tag{7}$$

Experimental results show that given theoretical algorithm is very sensitive to the bias of the accelerometer. Module x- IMU is using 12-bit accelerometer with full scale up to ±8 g. Bias equal to 1 LSB in one axis will produce velocity drift:

$$\boldsymbol{v}_{drift}(t) = \boldsymbol{a}_{bias} \cdot t = \frac{8\,\text{g}}{2^{12-1}} t \approx 0.04\,\text{m.s}^{-2} \cdot t \tag{8}$$

This drift is integrated into position:

$$\boldsymbol{d}_{drift}(t) = \boldsymbol{d}_{drift}(t-T) + \left[\boldsymbol{v}_{drift}(t) + \boldsymbol{v}_{drift}(t-T)\right]\frac{T}{2} \approx \frac{1}{2}\boldsymbol{a}_{bias} t^2 \tag{9}$$

Within t = 100 s of the operation the considered minimal accelerometer bias will produce error of the position estimation $\boldsymbol{d}_{\text{drift}} \approx$ 200 m. This error shows that using only accelerometer for position estimation is not reliable in longer term operation.

Real measured drift characteristics are shown in Fig. 2. Experimental results are slightly better than computed worst case.

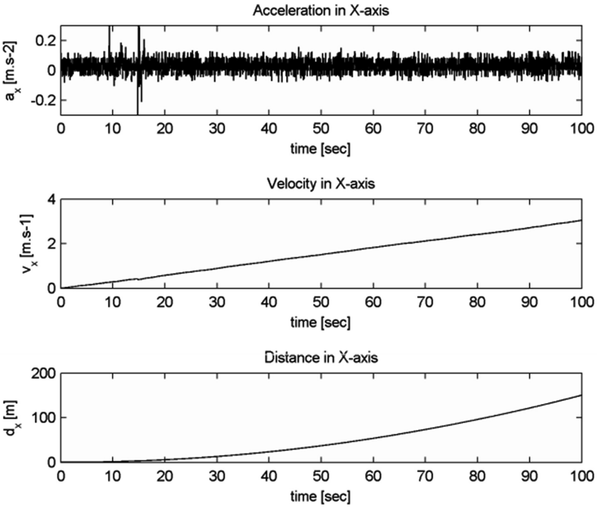

Fig. 2. Measured acceleration, integrated velocity and distance in steady state with respect to global frame of reference. Data are obtained right after sensor- calibration in order to eliminate thermal drift effects.

5 Position Estimation by Odometer

Since the double integration of the accelerometer data does not provide reliable information about distance, it is necessary to use absolute motion sensor [2, 6]. In case of wheeled mobile platform the simplest solution is to use odometer [3] bound with its wheels. We assume that wheels do not slide and difference of the distance run by wheels during turns is negligible.

Following algorithm was used for estimation of the mobile platform's position in three-dimensional space:

1. *Initialize position, reset odometer:*

$$\boldsymbol{d} = \boldsymbol{d}_0 \quad s = 0 \tag{10}$$

2. *Wait for new sample of rotational matrix* ***R*** *from AHRS.*
3. *Get current position of the odometer, compute change of odometer value:*

$$\Delta s = s_{\text{new}} - s \tag{11}$$

4. *Convert dislocation vector into global system:*

$$\boldsymbol{d} \leftarrow \boldsymbol{d} + \boldsymbol{R}^{-1} [ds \quad 0 \quad 0] \tag{12}$$

5. *Go back to step 2.*

Our experimental platform does not contain a real odometer. Since the propulsion DC electromotor moves platform at approximately constant forward speed v_{fwd} we can replace measurement of the distance s by measurement of electromotor on-time on t. Then it is valid:

$$\Delta s \approx v_{\text{fwd}} \cdot T \tag{13}$$

For improved performance of the "virtual odometer" it is possible to take the mass of the vehicle into account. The vehicle is powered by brushed DC electromotor with permanent magnets. Induced voltage in rotor winding is proportional to the rotation frequency f. Supply voltage U_{suppy} (constant when electromotor is turned on) is equal to:

$$U_{suppy} = R_a \cdot I + \frac{f}{K_{\text{V}}} \tag{14}$$

where R_{a} is the rotor winding resistance and K_{V} is the rotation rate per 1 V.

Electromotor torque is proportional to its current I, therefore is valid:

$$\mathrm{M} = K_M \cdot I = K_M \cdot \left(\frac{U_{suppy}}{R_a} - \frac{f}{K_V R_a} \right) = M_0 - C_{\text{M}} f \tag{15}$$

where M_0 is initial torque and C_{M} is motor constant.

The electromotor is coupled with wheels by fixed gear, therefore the previous equation is valid also for propulsion force *F*:

$$F(t) = F_0 - C \cdot v_{\text{fwd}}(t) = \text{m}\frac{dv_{fwd}(t)}{dt} \tag{16}$$

where F_0 is static propulsion force and *C* is a system constant. Since the propulsion force at maximal forward speed max v is zero we can estimate the value of the constant as:

$$C = \frac{F_0}{v_{max}} \tag{17}$$

Because the sampling period of the AHRS system is very short (maximal sampling frequency of the x-IMU module is 512 Hz), we can assume *dt* = *T* and forward speed can be calculated incrementally in each step by following:

$$v_{\text{fwd}}[n+1] = v_{\text{fwd}}[n] + \frac{F_0 T}{m}\left(1 - \frac{v_{\text{fwd}}[n]}{v_{max}}\right) \tag{18}$$

In case when motor is turned off the formula is different:

$$v_{\text{fwd}}[n+1] = v_{\text{fwd}}[n] + \frac{F_0 T}{m n_{max}} \cdot v_{\text{fwd}}[n] \tag{19}$$

Previous equations allow real-time estimation of the robot forward speed without measurement of the actual wheels' speed.

6 Collision Detection

Great advantage of mounted inertial measurement unit is that it allows detection of the collisions of the vehicle with any obstacle. Onboard accelerometer can measure acceleration of the shock in two axes, therefore it is possible to estimate not just occurrence of the shock but also its direction. The shock will cause negative spike of the acceleration in *xy* plane from direction δ (see Fig. 3).

$$\delta = \text{atan2}(-a_{\text{local y}}, -a_{\text{local x}}) \tag{20}$$

where $a_{\text{local x}}$, $a_{\text{local y}}$ are complements of the shock acceleration in local coordinate system.

Since the ability to accelerate of the vehicle is always limited by static friction coefficient μ between its tires and the floor (μ usually does not exceed 1), the maximal acceleration caused by vehicle propulsion or turning is given by:

$$a_{\max} = \mu g \tag{21}$$

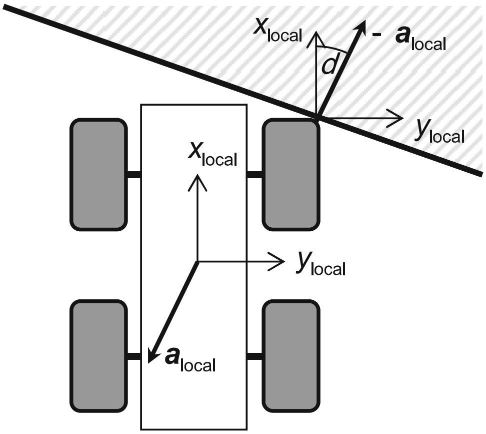

Fig. 3. Acceleration vector during collision with an obstacle

Threshold value of the local acceleration in *xy* plane for identifying collision should be set from 120% to 200% of a_{max}. Experiments show that acceleration during collision of our mobile platform with solid obstacle was approximately 8 g at full forward speed but only 2 g at 25% speed. Collision detection threshold was set to 1.2 g.

7 Control of the Mobile Platform

Our mobile platform was designed for massive production and allows only three-state control of the propulsion (forward – stop – backward) and three-state control of direction (leftstraight- right) (see Fig. 4). Control system is based on on-off regulator with hysteresis. User inputs several waypoints which the robot should visit (approach to it to some distance) in given order.

Every waypoint is defined in 2D space by Cartesian coordinates. We denote current waypoint as W = $[x_{\text{W}}, x_{\text{W}}]$, current position as A = $[x_{\text{A}}, x_{\text{A}}]$ and current heading as ψ_{A}. Distance vector to target is then:

$$\text{D} = \text{W} - \text{A} = [x_{\text{W}} - x_{\text{A}},\, y_{\text{W}},\, y_{\text{A}}] = [x_{\text{D}},\, x_{\text{D}}], \tag{22}$$

Heading to target is:

$$\psi_{\text{A}} = \text{atan2}\ (x_{\text{D}}, x_{\text{D}}) \tag{23}$$

Difference between ψ_{D} and ψ_{A} expresses regulation error:

$$\psi_{\text{E}} = \psi_{\text{D}} - \psi_{\text{A}} \tag{24}$$

Note: In order to obtain angular difference ψ_{E} from interval $(-\pi, \pi)$ it is necessary to remove period 2π out of the result:

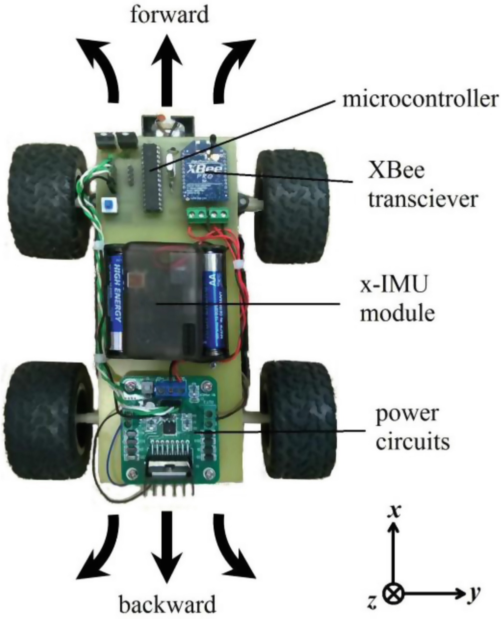

Fig. 4. Experimental mobile robot platform

$$\psi_{\mathrm{E}} \leftarrow \psi_{\mathrm{E}} - 2\pi \cdot \mathrm{round}\left(\frac{\psi_{\mathrm{E}}}{2\pi}\right) \tag{25}$$

Steering control algorithm is three-state on-off regulator with hysteresis. It is described by following state diagram (see Fig. 5).

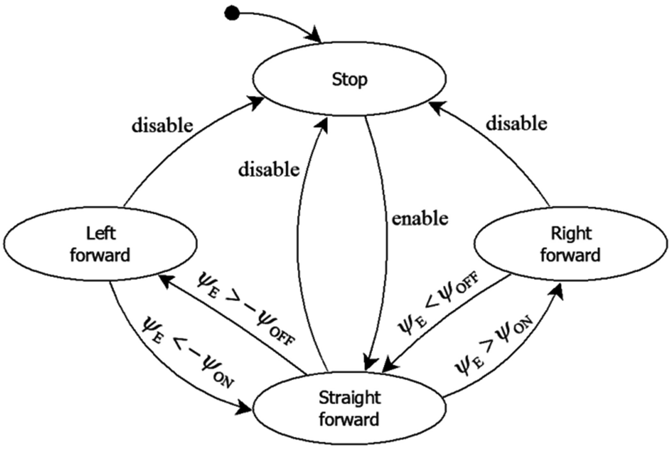

Fig. 5. State diagram of the simple control algorithm

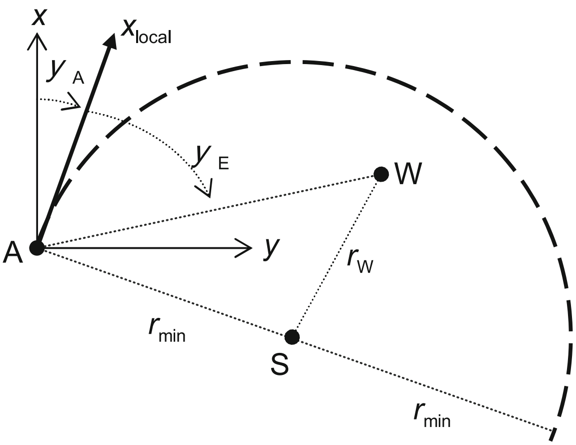

Fig. 6. Unreachable waypoint inside minimal radius trajectory

As can be seen, algorithm does not use backward movement. In case that robot is too close to the waypoint and heading error is too high (waypoint is inside arc with robot's minimal turn radius $r_{\min}$, see Fig. 6), algorithm will not ensure reaching the waypoint (robot will circle around waypoint).

According to Fig. 6 it is valid:

$$S = A \ + r_{\min}[-\sin\psi_A \ \cos\psi_A] \text{ when } \psi_E > 0 \tag{26}$$

$$S = A \ + r_{\min}[\sin\ \psi_A - \cos\ \psi_A] \text{ when } \psi_E < 0 \tag{27}$$

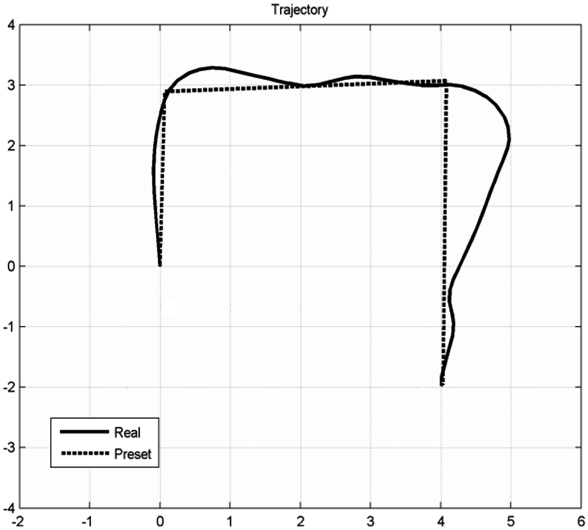

Fig. 7. Real trajectory of experimental mobile robot platform

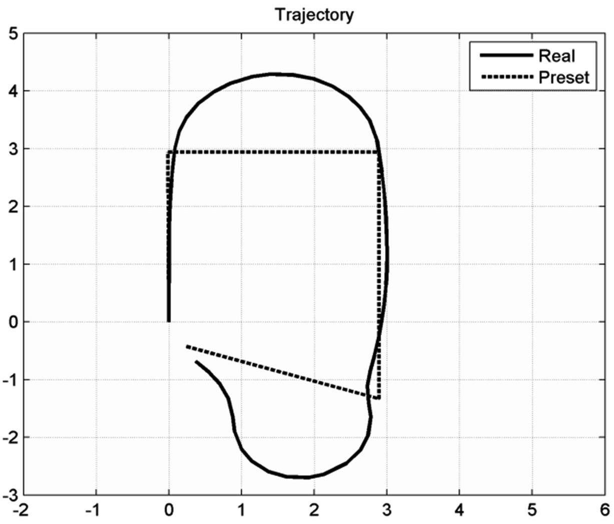

Fig. 8. Real trajectory with bigger deviation

Distance r_W is then computed as length of the segment SW = W − S. If the waypoint is unreachable ($r_W < r_{min}$), vehicle should reverse steering and electromotor direction until waypoint becomes reachable ($r_W > r_{min}$). In order to avoid oscillations it is necessary to use hysteresis in decision whether waypoint is reachable or not.

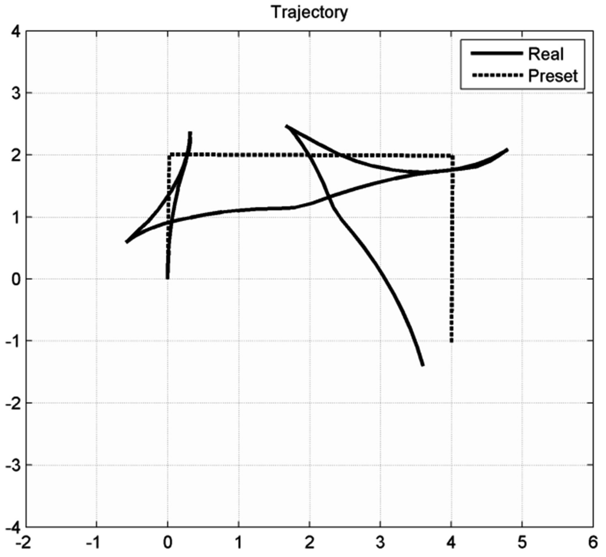

Fig. 9. Backward movement of experimental mobile robot platform

8 Experimental Results

We have applied the proposed algorithm for position control of experimental mobile robotic platform. In the Fig. 7 we can see trajectory of experimental mobile robot platform. Trajectory is defined using 4 points. As can be seen in Figure mobile platform passed all points very accurately even when we set permissible deviation from the real position.

In Fig. 8 is measurement with bigger deviation from the desired position.

In our approach is implemented algorithm which using backward movement for better results. This approach can be seen in Fig. 9.

The robot does not run in perfect circles. This effect is caused by inaccuracy of heading estimation by gyroscope (Fig. 10).

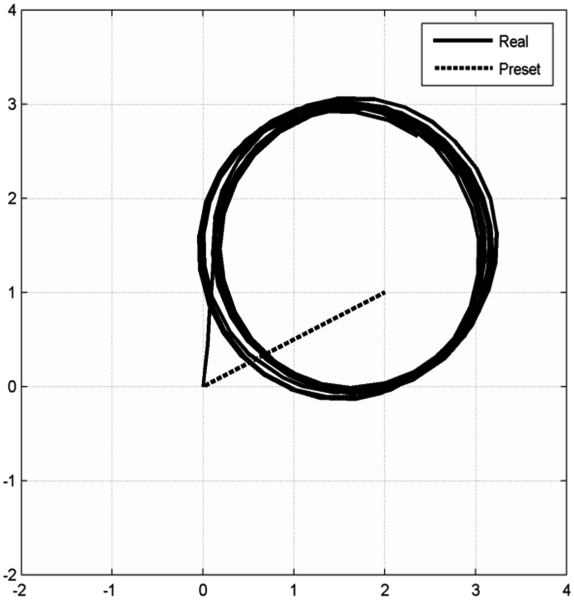

Fig. 10. Inaccuracy of position estimation using gyroscope

9 Conclusion

'Low-cost inertial sensors do not provide sufficient accuracy of the acceleration measurement therefore they cannot be used directly for position estimation. On the other hand, they can be used for very precise measurement of attitude and heading. This work designs, explains and verifies algorithm which allows integration of inertial sensor data (attitude and heading estimation) into control algorithm. This algorithm is designed universally and can be used for automatic control of movement of various devices such as mobile wheeled robots, mobile belt robots, boats or aircraft. Proposed steering control algorithm can be parameterized according to the parameters and limitations of the real controlled vehicle (especially minimal turning radius and acceleration abilities). It also implements detection of the waypoint reachability and provides reversal control for wheeled platforms.

Acknowledgment. The contribution is sponsored by VEGA MŠ SR No 1/0367/15 prepared project “Research and development of a new autonomous system for checking a trajectory of a robot”.

References

1. Feng, S., Murray-Smith, R.: Fusing Kinect sensor and inertial sensors with multi-rate Kalman filter. In: IET Conference on Data Fusion & Target Tracking 2014: Algorithms and Applications (DF&TT 2014), pp. 1–8, 30 April 2014. doi:10.1049/cp.2014.0527
2. Pinney, C., Hawes, M.A., Blackburn, J.: A cost-effective inertial motion sensor for short-duration autonomous navigation. In: Position Location and Navigation Symposium, pp. 591–597. IEEE, 11–15 April 1994. doi:10.1109/PLANS.1994.303402
3. Sampaio, S., Massatoshi Furukawa, C., Maruyama, N.: Sensor fusion with low-grade inertial sensors and odometer to estimate geodetic coordinates in environments without GPS signal. IEEE Lat. Am. Trans. (Revista IEEE America Latina) **11**(4), 1015–1021 (2013). doi:10.1109/TLA.2013.6601744
4. Qasem, H., Gorgis, O., Reindl, L.: Design and calibration of an inertial sensor system for precise vehicle navigation. In: 2008 5th Workshop on Positioning, Navigation and Communication, WPNC 2008, pp. 229–231, 27 March 2008. doi:10.1109/WPNC.2008.4510379
5. Benser, E.T.: Trends in inertial sensors and applications. In: 2015 IEEE International Symposium on Inertial Sensors and Systems (ISISS), pp. 1–4, 23–26 March 2015. doi:10.1109/ISISS.2015.7102358
6. Šimák, V., Končelík, V., Hrbček, J., Folvarčík, J.: Realization and a real testing of the road–fee system. In: Proceedings of the 33rd International Conference on Telecommunications and Signal Processing, TSP 2010, Baden near Vienna, Austria, 17–20 August 2010. ISBN 978-963-88981-0-4
7. Hulsing, R.: MEMS inertial rate and acceleration sensor. IEEE Aerosp. Electron. Syst. Mag. **13**(11), 17–23 (1998). doi:10.1109/62.730613