3D Positioning Algorithm for Low Cost Mobile Robots

Rafael Socas, Sebastián Dormido, Raquel Dormido and Ernesto Fábregas
Department of Computer Science and Automatic Control, UNED, Juan del Rosal 16, 28040, Madrid, Spain

Keywords: 3D Positioning, Mobile Robots, Sensors.

Abstract: A new 3D positioning algorithm for low cost robots is proposed. The algorithm is based on a Finite State Machine to estimate the position and orientation of the robot. The system sets dynamically the parameters of the algorithm and makes it independent of the noise in the sensors. The algorithm has been tested for differential wheel drive robots, however it can be used with different types of robots in a simple way. To improve the accuracy of the system, a new reference system based on the accelerometer of the robot is presented which reduces the accumulative error that the odometry produces.

1 INTRODUCTION

In mobile robots applications a good positioning and orientation estimation are crucial (Borenstein and Feng, 1995). Different techniques can be used to solve this problem. The most common techniques can be divided in seven categories (Borenstein et al., 1997; Lee and Park, 2014): 1. Odometry; 2. Inertial navigation; 3. Magnetic compasses; 4. Active beacon; 5. Global Positioning Systems; 6. Landmark navigation; and 7. Model matching. At the same time, they can be categorized into two groups based on the position measurements: relative, (also called dead-reckoning which includes the categories 1 and 2) and absolute (which includes the rest of the categories). Many applications usually combine two of them, one of each group to compensate the lacks of a single method.

The low cost robots frequently have few sensors. They typically have wheel encoders, accelerometers and obstacle detectors (Siegwart et al., 2011; Everett, 1995; Faisal et al., 2014). For this reason, in this kind of platforms, the techniques that can only be applied are Odometry and Inertial Navigation. Although both techniques produce accumulative errors they provide a good short-term accuracy. The odometry measures the distance that each wheel of the robot has travelled over the time. With this information, the position and the orientation of the robot can be obtained. On the other hand, inertial navigation uses accelerometers and gyroscopes to measure the acceleration and the rate of rotation of the robot. These measurements have to be integrated to obtain the position and the orientation by inertial navigation techniques. However, the cost of the gyroscopes constrains on the environments in which they are practical for use. In low cost robot the techniques that can be used to estimate the position are odometry and inertial navigation with accelerometers. In general way, low cost accelerometers have poor signal to noise ratio when the robot has small accelerations (common situation in many applications, for example in robots with constant speed), for this reason, the positioning and orientation estimation by accelerometers in low cost platforms is a bad solution (Liu and Pang, 2001). On the other hand, the accelerometers have a good performance in tilt angles estimation (Trimpe and D'Andrea, 2010; Luczak et al., 2006).

In this paper a new 3D positioning algorithm for for low cost robots is proposed, it combines odometry and tilt estimation to obtain the position and the orientation in 3D. Also, the tilt estimation is used as a reference system to reduce the orientation error that the classical odometry produces. The paper is organized as follows. Section 2 presents the 3D positioning in mobile robots applications. The proposed algorithm is presented in section 3. The experimental results are analysed in section 4. And finally, the conclusions and future work are presented in section 5.

2 3D POSITIONING AND ORIENTATION

The estimation of the position and orientation of mobile robots is one of the basic preconditions for their

Socas R., Dormido S., Dormido R. and Fabregas E..
3D Positioning Algorithm for Low Cost Mobile Robots.
DOI: 10.5220/0005480900050014
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 5-14
ISBN: 978-989-758-123-6

autonomy. The aim of this work is to estimate the position and orientation of the robot in 3D. Both estimations can be defined as a vector with six components $\mathbf{po} = \{\hat{x}, \hat{y}, \hat{z}, \hat{\varphi}, \hat{\theta}, \hat{\psi}\}$ (Roberson and Schwertassek, 1988). The position $(\hat{x}, \hat{y}, \hat{z})$ and the orientation $(\hat{\varphi}, \hat{\theta}, \hat{\psi})$ are defined with respect to the world frame (O) as it is shown in Figure 1. The elevation $\hat{\theta}$, the bank $\hat{\psi}$ and the heading $\hat{\varphi}$ are the Tait-Bryan angles. Finally, on the robot, the moving frame (O_m) is defined.

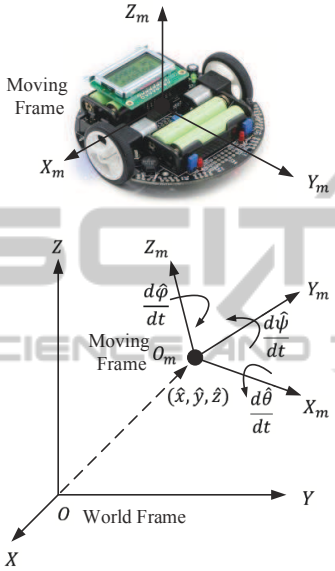

Figure 1: Position $(\hat{x}, \hat{y}, \hat{z})$ and orientation $(\hat{\varphi}, \hat{\theta}, \hat{\psi})$ with respect to the world frame (O).

When the robot is travelling on a flat surface S, a new reference frame can be defined O_s (surface frame). Via odometry the position $(\hat{x}_o, \hat{y}_o)$ and the orientation $\hat{\varphi}_o$ with respect to the surface frame can be calculated. A tilt estimation based on an accelerometer is used to obtain the elevation $\hat{\theta}$ and the bank $\hat{\psi}$ angles of the surface S with respect to the world frame (see Figure 2).

With the parameters $\hat{x}_o$, $\hat{y}_o$, $\hat{\varphi}_o$, $\hat{\theta}$ and $\hat{\psi}$ the components of the vector **po** can be calculated. The methodology to estimate it will be explained in the following sections.

2.1 Position and Orientation Estimation via Odometry

Two differential wheels robots are the most common low cost mobile robots. These systems usually have wheel encoders, an accelerometer and other sensors to avoid obstacles. When the robot is travelling on a flat surface its position and orientation can be estimated using odometry (Abbas et al., 2006; Jha and

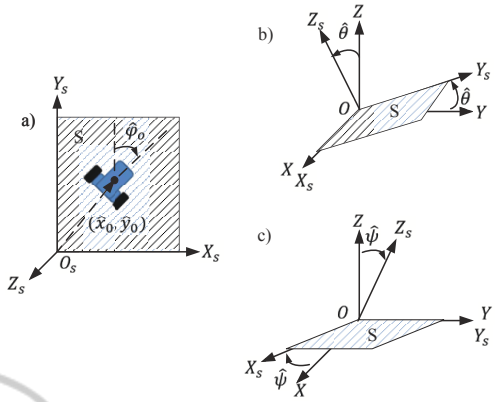

Figure 2: a) Position $(\hat{x}_o, \hat{y}_o)$ and orientation $(\hat{\varphi}_o)$ with respect to the surface frame (O_s). b) Elevation angle $(\hat{\theta})$. c) Bank angle $(\hat{\psi})$.

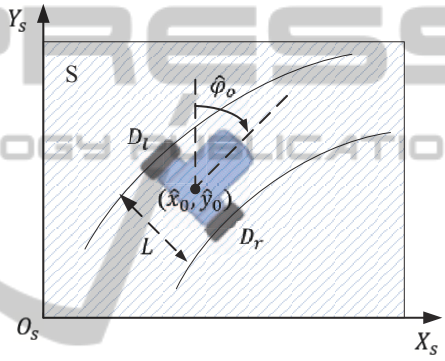

Figure 3: Position $(\hat{x}_o, \hat{y}_o)$ and orientation $(\hat{\varphi}_o)$ with respect to the surface frame (O_s).

Kumar, 2014). Wheel encoders allow to measure the distances that each wheel has travelled (D_l for the left wheel and D_r for the right wheel). With D_l, D_r and the distance between the two wheels L (see Figure 3), the position and the orientation of the robot can be calculated in discrete-time using equations(1)-(3).

$$\hat{\varphi}_o[n] = \hat{\varphi}_o[n-1] + \frac{D_l[n] - D_r[n]}{L} \quad (1)$$

$$\hat{x}_o[n] = \hat{x}_o[n-1] + D_c * sin(\hat{\varphi}_o[n]) \quad (2)$$

$$\hat{y}_o[n] = \hat{y}_o[n-1] + D_c * cos(\hat{\varphi}_o[n]) \quad (3)$$

where
D_l: Distance travelled by left wheel.
D_r: Distance travelled by right wheel.
D_c: Mean distance defined by $D_c = \frac{D_l + D_r}{2}$.
L: Distance between the two wheels.

2.2 Tilt Estimation

The accelerometer is sensitive to the total acceleration of the mobile robot. It is composed of the inertial

acceleration, the gravity field g, the centripetal and the tangential acceleration. This sensor measures the three components (a_x, a_y, a_z) of the total acceleration with respect to the moving frame (O_m). In the two differential wheel drive robots considered in this paper, the gravity field is the most important component of the acceleration, for this reason, the rest of them are not considered.

On the other hand, the robot is made to drive three possible rotations: the elevation $\hat{\theta}$, the bank $\hat{\psi}$ and the heading $\hat{\varphi}$ angles with respect to the world frame (O). With this assumption, the sensor in the robot is measuring the gravity field g (because the movements of the robot do not produce other accelerations or they are irrelevant) with respect to the moving frame on the robot (see Figure 4). In the world frame the acceleration is always the same $(0, 0, -g)$ where, on the other hand, the relationship between the accelerations in the two frames and the rotation angles is defined by (4).

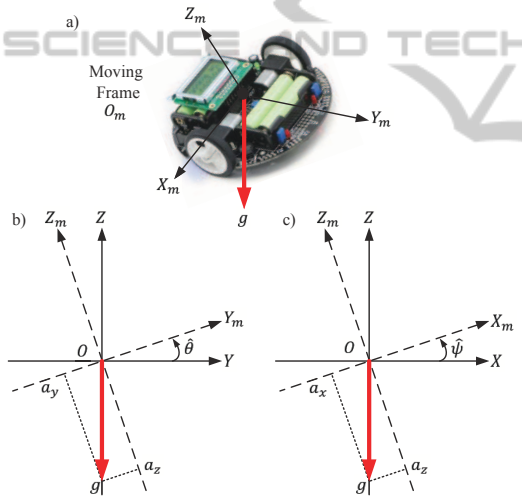

Figure 4: a) Gravity field with respect to the moving frame, the components (a_x, a_y, a_z) are the measurements of the accelerometer. b) Robot with elevation angle $\hat{\theta}$. c) Robot with a bank angle $\hat{\psi}$.

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \mathbf{R}(\hat{\theta}, \hat{\psi}, \hat{\varphi}) * \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \tag{4}$$

The rotation matrix $\mathbf{R}(\hat{\theta}, \hat{\psi}, \hat{\varphi})$ depends on the order of rotations. Therefore to calculate the angles $\hat{\theta}$, $\hat{\psi}$ and $\hat{\varphi}$ it is necessary to know the order of rotation. Considering a robot that rotates first an angle $\hat{\theta}$ and then an angle $\hat{\psi}$, the measurements in the accelerometer are described by (5).

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} -g * cos(\theta) * sin(\psi) \\ -g * sin(\theta) \\ -g * cos(\theta) * cos(\psi) \end{pmatrix} \tag{5}$$

If the robot rotates first an angle $\hat{\psi}$ and then an angle $\hat{\theta}$, the accelerometer measurements are given by (6).

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} -g * sin(\psi) \\ -g * cos(\psi) * sin(\theta) \\ -g * cos(\psi) * cos(\theta) \end{pmatrix} \tag{6}$$

Comparing (5) and (6) the components a_x and a_y are different in each equation, and therefore, we can conclude that the problem is undetermined. If the robot has gyroscopes, the order of rotations can be estimated and consequently the rotation angles can be calculated. As we mentioned before, low cost robots frequently have an accelerometer but they do not have gyroscopes. For this reason, in a general way in this kind of robots the elevation $\hat{\theta}$ and the bank $\hat{\psi}$ angle can not be estimated.

3 ALGORITHM ARCHITECTURE

The limitations of the low cost robots to estimate the elevation and the bank angles has been analysed in Section 2. To avoid these problems in this paper we assume that:

a) The two-wheel differential robot can travel only on a flat surface and,

b) this surface might have an elevation angle or a bank angle, but not both angles at the same time.

With the previous assumptions, to estimate the positioning in 3D and the orientation of the robot, the discrete-time architecture depicted in Figure 5 has been proposed.

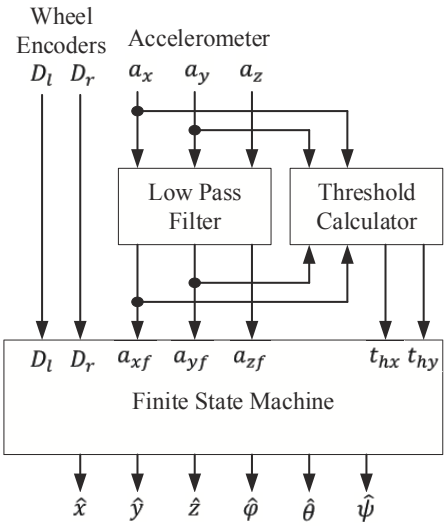

Figure 5: Algorithm architecture.

As it is shown in the Figure 5, the algorithm is composed of the following signals and blocks:

a) Input signals, $D_l[n]$ and $D_r[n]$ directly reading from wheel encoders, and $a_x[n]$, $a_y[n]$ and $a_z[n]$ from the accelerometer.

b) A low pass filter to remove the noise in the sensor's signal. The outputs of this blocks are $a_{xf}[n]$, $a_{yf}[n]$ and $a_{zf}[n]$.

c) A threshold calculator which obtains the threshold values ($t_{hx}[n]$ and $t_{hy}[n]$), these values will be used in the Finite State Machine (FSM) to define the changes between states.

d) The FSM which estimates the position and orientation parameters.

e) And finally, the output signals $(\hat{x}[n],\hat{y}[n],\hat{z}[n],\hat{\varphi}[n],\hat{\theta}[n],\hat{\psi}[n])$.

3.1 Low Pass Filter

Low pass filtering of the signals from the accelerometer is a good way to remove the noise (both mechanical and electrical) (Seifert and Camacho, 2007). In this work, an IIR filter has been proposed (7).

$$a_{if}[n] = (1-\alpha) * a_i[n] + \alpha * a_{if}[n-1] \quad (7)$$

for $i = \{x,y,z\}$

The inputs are the accelerometer's signal $(a_x[n],a_y[n],a_z[n])$ and the outputs are the filtered signals $(a_{xf}[n],a_{yf}[n],a_{zf}[n])$. The parameter α sets the cut off frequency and the group delay of the filter. In the design process, α has to be selected to obtain a good balance between the noise reduction and the delay in the acceleration measurements.

3.2 Threshold Calculator

As we mention before, the positioning algorithm is based on a Finite State Machine. The states, as they will be explained in the next section, depend on the values of the elevation angle $\hat{\theta}[n]$ and the bank angle $\hat{\psi}[n]$. When the robot is on a surface with $\hat{\theta}[n] \approx 0$ or $\hat{\psi}[n] \approx 0$, the noise in the accelerometer sensor could produce changes between states and the system could be unstable. For this reason, the algorithm needs a threshold system to control the changes between the states taking in account the noise in the sensors.

As we will explain later, when the elevation and the bank angle are close to zero ($\hat{\theta} \approx 0, \hat{\psi} \approx 0$), the values of the accelerations $a_{xf}[n]$ and $a_{xf}[n]$ should be zero except by the noise in the sensor. For this reason, the noise in these measurements will be considered in the threshold calculator.

The noise $n_x[n]$ and $n_y[n]$ in the acceleration signals $a_x[n]$ and $a_y[n]$ can be estimated by the difference between the acceleration signals and their filtered signals as shown in (8) and (9).

$$n_x[n] = a_x[n] - a_{xf}[n] \quad (8)$$

$$n_y[n] = a_y[n] - a_{yf}[n] \quad (9)$$

Then a simple moving average has been used to obtain a stable value of the noise, then the function $atan()$ is applied to calculate the threshold angles by (10) and (11).

$$t_{hx}[n] = atan\left(\frac{1}{l}\sum_{k=1}^{l} abs(n_x[k])\right) \quad (10)$$

$$t_{hy}[n] = atan\left(\frac{1}{l}\sum_{k=1}^{l} abs(n_y[k])\right) \quad (11)$$

In this way the proposed system obtains two dynamic threshold values ($t_{hx}[n]$ and $t_{hy}[n]$) one for each acceleration signal. Moreover, it has a degree of freedom, the parameter l, which will be set depending on how the level of noise changes over the time.

The proposed threshold calculator is depicted in Figure 6.

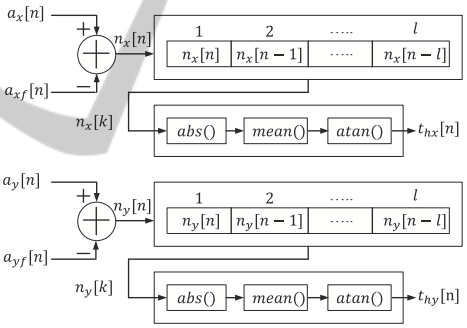

Figure 6: Threshold calculator. The threshold values are obtained as a moving average of the noise in the sensor.

3.3 Finite State Machine

A Finite State Machine (FSM) has been included in the architecture to estimate the position in 3D of the robot (see Fig. 5). The inputs are the wheel encoders, the filtered accelerations and the threshold values. The outputs are the estimation of the position $(\hat{x}[n],\hat{y}[n],\hat{z}[n])$ and the orientation $(\hat{\varphi}[n],\hat{\theta}[n],\hat{\psi}[n])$ of the robot. As we mentioned before, the robot only can travel on a flat surface and this surface could have an elevation angle or a bank angle with respect to the world frame, but it can not have two rotations at the same time. Taking into account these considerations, in this section the behaviour of the FSM have been developed.

As it is shown in the Figure 7, the FSM has five states and five conditions to change between the

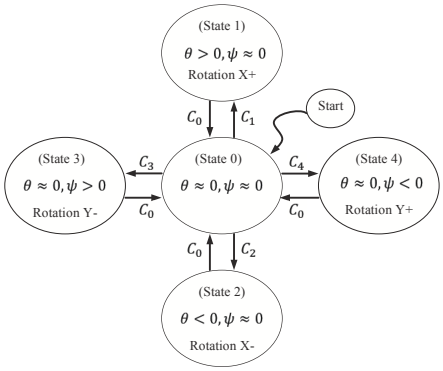

Figure 7: The FSM has five states (State $0,1,2,3,4$ and 5) and five transition conditions (C_0, C_1, C_2, C_3 and C_4).

states. The meaning of each state is presented in the following definitions:

a) State 0: The robot is on a surface without elevation angle ($\hat{\theta} \approx 0$) and without bank angle ($\hat{\psi} \approx 0$) with respect to the world frame.

b) State 1: The robot is on a surface with a positive elevation angle ($\hat{\theta} > 0$) and without bank angle ($\hat{\psi} \approx 0$).

c) State 2: The robot is on a surface with a negative elevation angle ($\hat{\theta} < 0$) and without bank angle ($\hat{\psi} \approx 0$).

d) State 3: The robot is on a surface with a positive bank angle ($\hat{\psi} > 0$) and without elevation angle ($\hat{\theta} \approx 0$).

e) State 4: The robot is on a surface with a negative bank angle ($\hat{\psi} < 0$) and without elevation angle ($\hat{\theta} \approx 0$).

On the other hand, we assume that the system works in discrete-time, in each sample time the measurements of the sensor ($D_l[n]$, $D_r[n]$, $a_{xf}[n]$, $a_{yf}[n]$, $a_{zf}[n]$) are available.

The conditions (C_0, C_1, C_2, C_3, C_4) which control the changes between states are defined as function of the components of the acceleration ($a_{xf}[n]$, $a_{yf}[n]$, $a_{zf}[n]$), the threshold values ($t_{hx}[n]$ and $t_{hy}[n]$) and two positive guard parameters (H and h) as it is presented in the following equations (12)-(18):

$$\theta_{tilt} = atan(a_{yf}[n]/a_{zf}[n]) \tag{12}$$

$$\psi_{tilt} = atan(a_{xf}[n]/a_{zf}[n]) \tag{13}$$

$$C_0 : (abs(\theta_{tilt}) \leq t_{hy}[n] + h) \;\&(abs(\psi_{tilt}) \leq t_{hx}[n] + h) \tag{14}$$

$$C_1 : \theta_{tilt} > t_{hy}[n] + H \tag{15}$$

$$C_2 : \theta_{tilt} < -t_{hy}[n] - H \tag{16}$$

$$C_3 : \psi_{tilt} > t_{hx}[n] + H \tag{17}$$

$$C_4 : \psi_{tilt} < -t_{hx}[n] - H \tag{18}$$

The parameters H and h create two guard bands around the threshold values. These guards bands make a new threshold value ($t_{hy}[n] + H$ or $t_{hx}[n] + H$) when the elevation and bank angles increase and another one ($t_{hy}[n] + h$ or $t_{hx}[n] + h$) when they decrease. This method avoids instabilities when $\theta_{tilt} \approx t_{hy}$ or $\psi_{tilt} \approx t_{hx}$. These parameters have to be set taking in account the level of noise in the sensor.

3.4 Position and Orientation Estimation

In each state of the FSM an estimation of the 3D positioning has to be calculated. In the state 0 the classical odometry has been applied. For the rest of the states a combination of tilt estimation and odometry have been used. The aim of the 3D positioning estimation is to calculate the position vector of the robot ($\mathbf{po}[\mathbf{n}] = \{\hat{x}[n], \hat{y}[n], \hat{z}[n], \hat{\varphi}[n], \hat{\theta}[n], \hat{\psi}[n]\}$). Depending on the state where the robot is, different equations have to be applied. In the following sections, these equations will be defined.

3.4.1 Estimation in State 0

In state 0, due to the tilt is zero, the classical odometry has to applied as it is presented in (19)-(24).

$$\hat{\varphi}[n] = \hat{\varphi}[n-1] + \hat{\varphi}_o[n] \tag{19}$$

$$\hat{x}[n] = \hat{x}[n-1] + D_c[n] * sin(\hat{\varphi}[n]) \tag{20}$$

$$\hat{y}[n] = \hat{y}[n-1] + D_c[n] * cos(\hat{\varphi}[n]) \tag{21}$$

$$\hat{z}[n] = \hat{z}[n-1] \tag{22}$$

$$\hat{\theta}[n] = 0 \tag{23}$$

$$\hat{\psi}[n] = 0 \tag{24}$$

where

$\hat{\varphi}_o[n]$: Heading estimation by odometry $\frac{D_l[n]-D_r[n]}{L}$

$D_c[n]$: Mean distance $D_c[n] = \frac{D_l[n]+D_r[n]}{2}$.

L: Distance between the two wheels.

3.4.2 Estimation in State 1 and 2

The elevation angle $\hat{\theta}[n]$ can be calculated in state 1 by (25) and in state 2 by (26).

$$\hat{\theta}[n] = atan\left(\frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{-a_{zf}[n]}\right) \tag{25}$$

$$\hat{\theta}[n] = atan\left(\frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{a_{zf}[n]}\right) \tag{26}$$

In both states the components $\hat{\varphi}[n]$, $\hat{x}[n]$, $\hat{y}[n]$ and $\hat{\psi}[n]$ have the same expressions (27) - (30)

$$\hat{\varphi}[n] = \hat{\varphi}[n-1] + \hat{\varphi}_o[n] \tag{27}$$

$$\hat{x}[n] = \hat{x}[n-1] + D_c[n] * sin(\hat{\varphi}[n]) \quad (28)$$

$$\hat{y}[n] = \hat{y}[n-1] + D_c[n] * cos(\hat{\varphi}[n]) * cos(\theta_a) \quad (29)$$

$$\hat{\psi}[n] = 0 \quad (30)$$

Where $\theta_a = abs(\hat{\theta}[n])$

The component $\hat{z}[n]$ can be calculated in state 1 by (31) and in state 2 by (32).

$$\hat{z}[n] = \hat{z}[n-1] + D_c[n] * sin(\hat{\varphi}[n]) * sin(\theta_a) \quad (31)$$

$$\hat{z}[n] = \hat{z}[n-1] - D_c[n] * sin(\hat{\varphi}[n]) * sin(\theta_a) \quad (32)$$

3.4.3 Estimation in State 3 and 4

On the other hand, the bank angle $\hat{\psi}[n]$ can be calculated in state 3 by (33) and in state 4 by (34).

$$\hat{\psi}[n] = atan\left(\frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{-a_{zf}[n]}\right) \quad (33)$$

$$\hat{\psi}[n] = atan\left(\frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{a_{zf}[n]}\right) \quad (34)$$

In both states the components $\hat{\varphi}[n]$, $\hat{x}[n]$, $\hat{y}[n]$ and $\hat{\theta}[n]$ have the same expressions (35) - (38)

$$\hat{\varphi}[n] = \hat{\varphi}[n-1] + \hat{\varphi}_o[n] \quad (35)$$

$$\hat{x}[n] = \hat{x}[n-1] + D_c[n] * sin(\hat{\varphi}[n]) * cos(\psi_a) \quad (36)$$

$$\hat{y}[n] = \hat{y}[n-1] + D_c[n] * cos(\hat{\varphi}[n]) \quad (37)$$

$$\hat{\theta}[n] = 0 \quad (38)$$

Where $\psi_a = abs(\hat{\psi}[n])$

Finally, the component $\hat{z}[n]$ can be calculated in state 3 by (39) and in state 4 by (40).

$$\hat{z}[n] = \hat{z}[n-1] + D_c[n] * sin(\hat{\varphi}[n]) * sin(\psi_a) \quad (39)$$

$$\hat{z}[n] = \hat{z}[n-1] - D_c[n] * sin(\hat{\varphi}[n]) * sin(\psi_a) \quad (40)$$

3.5 Heading Reference System based on Tilt Angle

The heading angle $\hat{\varphi}[n]$ is the most important of the navigation parameters in terms of its influence on accumulated errors. In the proposed system, when the robot is in the state 0 the only way to calculate this parameter is by odometry ($\hat{\varphi}[n] = \hat{\varphi}[n-1] + \hat{\varphi}_o[n]$ where $\hat{\varphi}_o[n] = \frac{D_l[n] - D_r[n]}{L}$). But, when the robot is moving on a surface with an elevation angle $\hat{\theta}[n] \neq 0$ or with a bank angle $\hat{\psi}[n] \neq 0$ (states 1, 2, 3 or 4), the values of $a_{xf}[n]$ and $a_{yf}[n]$ can also be used to estimate the heading angle of the robot.

When the FSM is in state 1, 2, 3 or 4, two estimations of the heading can be obtained, one of them via odometry $\hat{\varphi}_o[n]$ and another one from the accelerometer $\hat{\varphi}_a[n]$ by the following equations (41)-(44).

In state 1,

$$\hat{\varphi}_a[n] = \begin{cases} atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}\leq 0) \\ \frac{\pi}{2} + atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}>0) \\ \pi + atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}>0) \\ \frac{3\pi}{2} + atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}\leq 0) \end{cases} \quad (41)$$

in state 2,

$$\hat{\varphi}_a[n] = \begin{cases} \pi + atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}\leq 0) \\ \frac{3\pi}{2} + atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}>0) \\ atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}>0) \\ \frac{\pi}{2} + atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}\leq 0) \end{cases} \quad (42)$$

in state 3,

$$\hat{\varphi}_a[n] = \begin{cases} \frac{\pi}{2} + atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}\leq 0) \\ \pi + atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}>0) \\ \frac{3\pi}{2} + atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}>0) \\ atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}\leq 0) \end{cases} \quad (43)$$

and finally, in state 4

$$\hat{\varphi}_a[n] = \begin{cases} \frac{3\pi}{2} + atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}\leq 0) \\ atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}>0)\&(a_{yf}>0) \\ \frac{\pi}{2} + atan\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}>0) \\ \pi + atan\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf}\leq 0)\&(a_{yf}\leq 0) \end{cases} \quad (44)$$

where $a_{xf} = a_{xf}[n]$ and $a_{yf} = a_{yf}[n]$

Here, it is important to note that the heading estimation from the accelerometer $\hat{\varphi}_a[n]$ is an absolute measure of the heading angle and it does not depend on the previous values. For this reason, this measure is a good way to avoid the accumulated errors that the classical odometry produces.

On the other hand, due to the acceleration signals have to be filtered to eliminate the noise in the sensor, the measurements of the acceleration have a delay which depends on the parameter α of the filter (see equation 7). In this case, if the heading estimation is calculated as $\hat{\varphi}[n] = \hat{\varphi}_a[n]$ it would be a bad approximation, for example, when the robot is making a circular path. Nevertheless, a little time after the robot starts to travel in straight line, the accelerations are stable (because the values of $a_{xf}[n], a_{yf}[n]$ and $a_{zf}[n]$ do not change) and the delay does not affect.

Taking in account the previous considerations, when the robot is travelling in straight line ($D_l = D_r$) after certain time ($time > t_{min}$) and the FSM is in state 1, 2, 3 or 4, the heading estimation from the accelerometer $\hat{\varphi}_a[n]$ can be used as a reference value for the heading angle. In this way, the accumulated error that the classical odometry produces in the heading angle can be avoided. The t_{min} value has to be defined as a function of the parameter α of the filter. The reference system proposed in this paper is depicted in Figure 8.

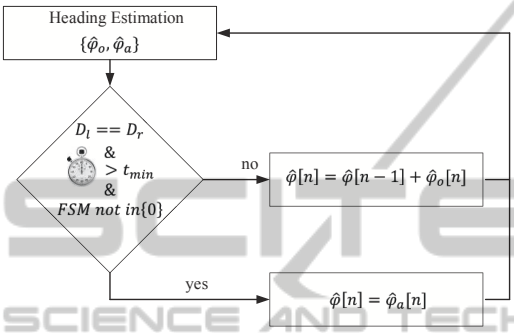

Figure 8: Proposed heading reference system.

4 EXPERIMENTAL RESULTS

To check the proposed algorithm in this paper, the low cost mOway differential wheel drive robot has been used. This robot has wheel encoders and a three-axis MMA7455L accelerometer from Freescale Semiconductor. The accelerometer has a dynamic range from $-2g$ to $2g$ (where g value is $9.81m/s^2$) in the three axis. The positioning algorithm has been implemented in C++ language, the algorithm runs in a laptop with Windows OS. A radio frequency link of 2.4 GHz has been used to communicate the robot and the laptop. A flat surface without tilt angle has been used as the world frame and a moving surface as surface frame (see section 2). Finally, a video camera has been used to capture the real path of the robot. The test laboratory is depicted in Figure 9.

Before checking the algorithm, the sensor and other parameters of the robot were calibrated. The results of these procedures are presented in the Table1. On the other hand, the parameters of the algorithm were set as it is shown in Table 2.

Next sections, show some experiments to analyse the response of the algorithm.

4.1 Behaviour of the FSM

To check the behaviour of the FSM, the robot was put on the moving surface (surface frame) with a constant speed of $10.3cm/s$ in straight line. Then different tilts ($\hat{\theta}$ and $\hat{\psi}$) were applied and the states of the FSM were analysed. The elevation angle was modified from -8° to 8°, in the same way, the bank angle was varied from -7.5° to 7.5°. The result of the experiment is depicted in the Figure 10.

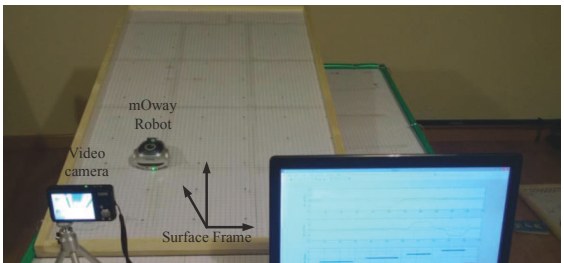

Figure 9: Test laboratory.

Table 1: Robot parameters.

Sampling period	$t_s = 0.2s$
Distance between two wheels	$L = 6.6cm$
Bias X acceleration	$a_{xbias} = -0.1059g$
Bias Y acceleration	$a_{ybias} = -0.3480g$
Bias Z acceleration	$a_{zbias} = 0.0927g$

Table 2: Algorithm parameters.

Filter constant	$\alpha = 0.9$
Length of threshold calculator	$l = 20$
Guard value H	$H = 0.2^\circ$
Guard value h	$h = 0.1^\circ$
Minimum time in reference system	$t_{min} = 1s$

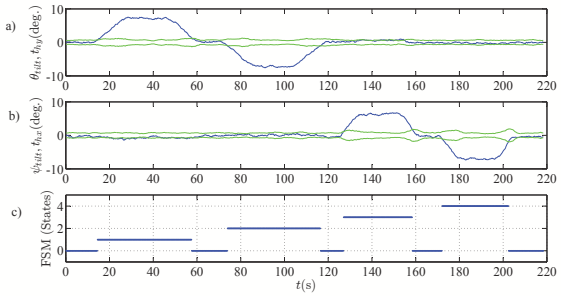

Figure 10: FSM's behaviour. a) Elevation angle in blue and t_{hy} in green. b) Bank angle in blue and t_{hx} in green. c) States of the FSM.

As Figure 10 shows, the behaviour of the system is according to the tilt angle. When tilt angles are close to zero, the system is in the state 0. The FSM is in

the state 1 or 2 depending on the value of θ_{tilt} as it is presented in Figure 10 a) and 10 c). And finally, states 3 and 4 depend on the value of ψ_{tilt} as it is shown in Figure 10 b) and 10 c).

It is important to note that the system can change between states with an error smaller than 1° in the tilt estimation as it is presented in Figure 10.

4.2 3D Positioning Estimation

Once the FSM is analysed, the next step is to check the 3D positioning. To this end two experiments were performed. In the first one, the robot travels in straight line with a speed of $10.3cm/s$ on a surface with a elevation angle of fifteen degrees ($\theta = 15^\circ$). In the second one, the robot drives a circular path (speed of left wheel$= 17.5cm/s$ and speed of right wheel$= 10.3cm/s$) on a surface with bank angle of minus twenty degrees ($\psi = -20^\circ$). The experiments take 10 seconds in both cases. In Figure 11 some snapshots of theses experiments are presented.

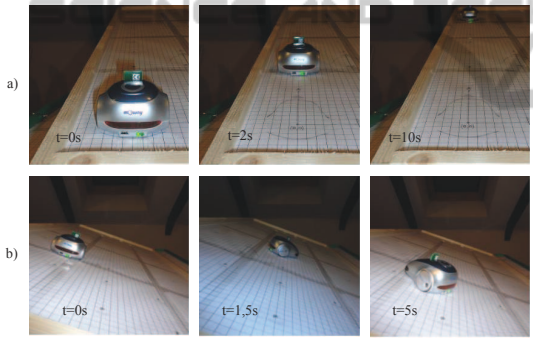

Figure 11: Experiments for checking the 3D positioning estimation. a) Straight line with $\theta = 15^\circ$ and $\psi = 0^\circ$ (state 1). b) Circular path with $\theta = 0^\circ$ and $\psi = -20^\circ$ (state 4).

To check the accuracy of the algorithm, the real position (x,y,z) and orientation (φ,θ,ψ) of the robot is obtained from the images of the video camera. The estimated parameters $(\hat{x},\hat{y},\hat{z},\hat{\varphi},\hat{\theta},\hat{\psi})$ are calculated by the proposed algorithm. The results of the first experiment is depicted in Figure 12 and in Figure 13 for the second one. As it is shown in the Figures 12 and 13 the algorithm is a good 3D positioning estimator with a low estimation error. The elevation and the bank angles do not have accumulative errors because they are obtained directly from the accelerometer. The rest of the parameters, will have accumulative error over the time because they are calculated by odometry.

4.3 The Heading Reference System

To analyse how the heading reference system improves the 3D positioning estimation, a new experiment has been set up. In this case, the robot has been programmed with a trajectory as it is shown in the Figure 14. The real path of the robot has been recorded by the video camera and two estimations have been calculated in the same experiment. One of them without the heading reference system and the another one using this system. The real heading and its estimations by the algorithm are presented in the Figure 15. As it can be noticed from this figure, the error in the heading estimation without the reference system increases over time. However when the reference system is activated, the error in the heading estimation does not increase.

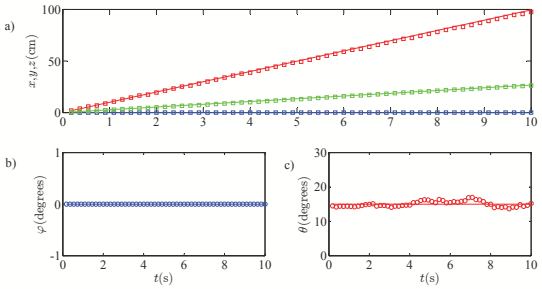

Figure 12: 3D positioning estimation in a straight line. a) Real position (x,y,z) (solid lines) and estimated position $(\hat{x},\hat{y},\hat{z})$ (square line). b) Real heading φ (solid line) and estimated heading $\hat{\varphi}$ (circle line). c) Real elevation θ (solid line) and estimated elevation $\hat{\theta}$ (circle line). The bank angle $\hat{\psi}$ is not depicted because is zero as it was defined in (30).

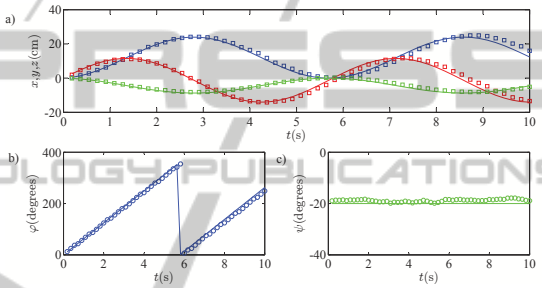

Figure 13: 3D positioning estimation in a circular path. a) Real position (x,y,z) (solid lines) and estimated position $(\hat{x},\hat{y},\hat{z})$ (square line). b) Real heading φ (solid line) and estimated heading $\hat{\varphi}$ (circle line). c) Real bank ψ (solid line) and estimated bank $\hat{\psi}$ (circle line). The elevation angle $\hat{\theta}$ is not shown due to is zero according to (38).

Considering the position parameters, the results of the experiment are depicted in the Figures 16 and 17.

As it is shown in these figures, when the position is obtained with the reference system proposed in this paper, the estimation error is reduced considerably. If the real position vector is defined as $\mathbf{p} = \{x,y,z\}$ and the estimation position vector as $\hat{\mathbf{p}} = \{\hat{x},\hat{y},\hat{z}\}$, the positioning error can be defined as $P_{error} = |\mathbf{p} - \hat{\mathbf{p}}|$. In Figure 18, the positioning error is presented for this

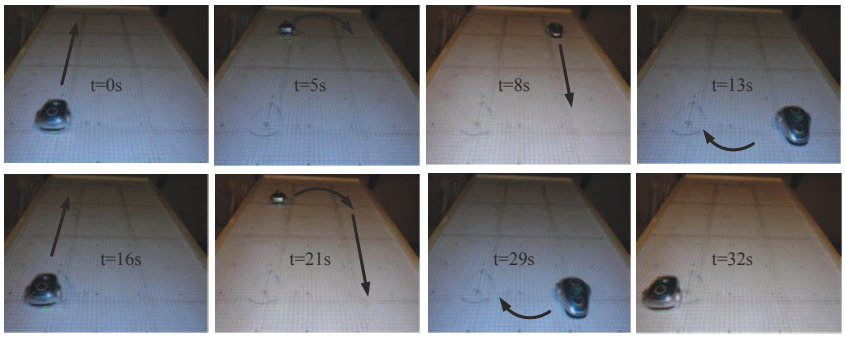

Figure 14: Snapshots of the experiment. In the straight line path the robot has a speed of $10.8cm/s$. In the the circular path the speed of the right wheel is $10.3cm/s$ and the left wheel is $17.5cm/s$.

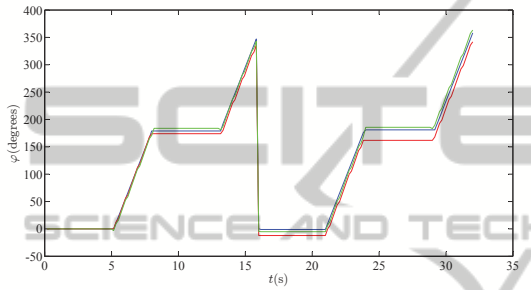

Figure 15: Heading estimation. Solid blue line is the real heading, solid red line is the heading estimation without reference system and solid green line is the heading estimation with the reference system.

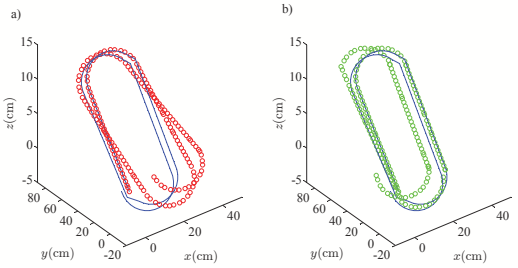

Figure 16: 3D position parameters, solid blue line is the real path. a) Red circle is the position estimation without reference system. b) Green circle is the position estimation with reference system.

experiment.

Comparing the results in the figure, although both methodologies have accumulative error over time, the proposed reference system reduce significantly the positioning error.

5 CONCLUSIONS

In this paper we have proposed a new 3D positioning algorithm for low cost robots. The architecture presented can be applied in robots which have wheel

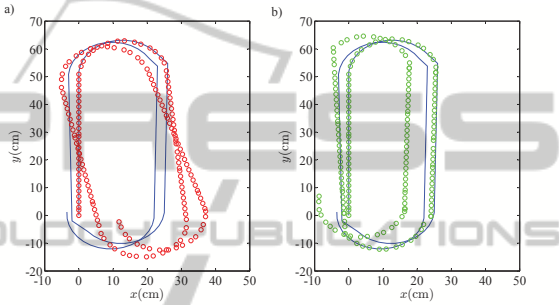

Figure 17: 2D position parameters (to show more clearly the error, z has been set to zero), solid blue line is the real path. a) Red circle is the position estimation without reference system. b) Green circle is the position estimation with reference system.

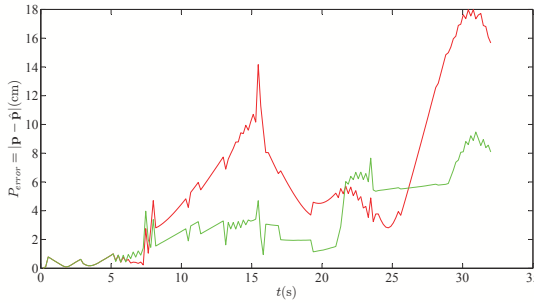

Figure 18: Red solid line is the positioning error when the reference system has not been used. The solid green line is the positioning error when the reference system is used.

encoders and a three axis accelerometer. The problem formulation have been developed for two wheel differential robots, however this algorithm is easy to apply in other kind of robots in a simple way. The methodology presented is based on a Finite State Machine. A threshold calculator allows to set the system dynamically as a function of the noise in the accelerometer. Also a new reference system is proposed, this new system improves considerably the estimation of the algorithm respect to the classical odometry.

The proposed algorithm was checked in the low cost robot mOway. In these experiments, in a simple way, the algorithm has been set and precise estimations of the 3D position were obtained. The threshold calculator worked in a correct way to estimate the states of the FSM and it sets the systems dynamically. Finally with the new reference system the positioning error has been considerably reduced.

As a future work, now we are considering to apply this algorithm in robots which have also gyroscopes to allow estimations on surface with elevation and bank angles at the same time. On the other hand, new methods for reference systems in low cost robots are being explored.

REFERENCES

Abbas, T., Arif, M., and Ahmed, W. (2006). Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots. In *SICE-ICASE, 2006. International Joint Conference*, pages 2073–2078. IEEE.

Borenstein, J., Everett, H. R., Feng, L., and Wehe, D. (1997). Mobile robot positioning-sensors and techniques. Technical report, DTIC Document.

Borenstein, J. and Feng, L. (1995). Correction of systematic odometry errors in mobile robots. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 569–574. IEEE.

Everett, H. (1995). *Sensors for mobile robots: theory and application*. AK Peters, Ltd.

Faisal, M., Hedjar, R., Alsulaiman, M., Al-Mutabe, K., and Mathkour, H. (2014). Robot localization using extended kalman filter with infrared sensor. In *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*, pages 356–360. IEEE.

Jha, A. and Kumar, M. (2014). Two wheels differential type odometry for mobile robots. In *Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2014 3rd International Conference on*, pages 1–5. IEEE.

Lee, Y.-C. and Park, S. (2014). Localization method for mobile robots moving on stairs in multi-floor environments. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 4014–4020. IEEE.

Liu, H. H. and Pang, G. K. (2001). Accelerometer for mobile robot positioning. *Industry Applications, IEEE Transactions on*, 37(3):812–819.

Luczak, S., Oleksiuk, W., and Bodnicki, M. (2006). Sensing tilt with mems accelerometers. *Sensors Journal, IEEE*, 6(6):1669–1675.

Roberson, R. E. and Schwertassek, R. (1988). *Dynamics of multibody systems*, volume 18. Springer-Verlag Berlin.

Seifert, K. and Camacho, O. (2007). Implementing positioning algorithms using accelerometers. *Freescale Semiconductor*.

Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.

Trimpe, S. and D'Andrea, R. (2010). Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2630–2636. IEEE.