DINAME 2017 - Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics
A. T. Fleury, D. A. Rade, P. R. G. Kurka (Editors), ABCM, São Sebastião, SP, Brazil, March 5-10, 2017

ANDROID BASED SENSOR FUSION AND CONTROL NAVIGATION STRATEGY FOR A WHEELED MOBILE ROBOT

Marcelo Rodrigues da Silva[1], **Elvira Rafikova**[1]

[1]Federal University of ABC Av. Do Estado 5001, Santo André, Brazil, marcelo.rodrigues@ufabc.edu.br, elvira.rafikova@ufabc.edu.br

Abstract: In this paper the android bases sensor fusion system architecture for robot navigation is presented. This system contains a position estimation module, which uses the gyroscope of an android mobile device such as the smartphone or a tablet. This device is placed on the differential steering mobile robot platform and connected via wireless network to the robots main controller. The robot performs encoder sensor reading, odometry process, embedded trajectory control and wheel actuation algorithms. The controller algorithms are based on State Dependent Riccati Equation control method and by this means the robot is steered to a reference trajectory. Numerical simulations and experiments are conducted taking as reference path a linear and circular regimes. The main contribution of this work is in the sense of unifying a powerful method of control with state estimation and sensor fusion performed with mobile based device via wireless connection.

Keywords: sensor fusion, android based position sensing, indoor robot navigation, SDRE position control

INTRODUCTION

The autonomous navigation is a task of great challenge for a mobile robot (Mahmood, Baig, and Ahsan 2012). Success in this task depends on four essential skills: perception, localization, cognition and movement control (Siegwart et al. 2004).

The perception involves the environmental sensing and accounting for the internal parameters such as velocity, position, heading, which are essential to estimate its location and targets. Hence, the measurement from different sensor types are desired to measure the same process or variable estimate more accurately the actual robot position. For the robotic navigation is possible to use wide spectrum of sensors (Ladislav, Pavel, and Martin 2014).

The sensor fusion has been applied in a wide plethora of problems, including: mobile robots, autonomous systems, object recognition, navigation, following reference trajectory "target tracking"(Dekhil et al. 1998). At work Indoor Positioning using Sensor Fusion in Android Devices (Rodriguez and Shala 2011) suggests the use of the Kalman filter as a sensor fusion method and a solution to increase this accuracy in measurements.

An integrated navigation system utilizes the complementary characteristics of different navigational devices to improve the overall precision of navigation (Mahmood, Baig, and Ahsan 2012).

In order to offer a navigation option even without the use of GPS (global positioning system), a solution proposed at work (Rodriguez and Shala 2011) suggests the use of IMU available on smartphones, as a form of navigation indoors where the GPS signal is not available, and proposes a solution to increase the accuracy of position estimates. Still on the theme of inertial navigation, (Woodman 2007) deals in detail about the influences of the use of inertial navigation with MEMS technology (microelectromechanical systems) present in smartphones, with the great advantage of the miniaturization of sensors and the disadvantage of a lower accuracy compared to conventional sensors.

In this paper it is proposed and implemented an experimental navigation system that combines *android* device based wireless sensing with robots wheels encoders odometry in a sensor fusion scheme that provides feedback for the trajectory controller. The controller module is based on SDRE (State Dependent Riccati Equation), a suboptimal control method which is simple to implement, fast in operation and provides real-time solution of the control problem. The robot steers to a desired trajectory in linear and circular form. The main contribution of this architecture is in the sense of unifying a powerful control method with a state estimation and wireless mobile sensor fusion. Numerical simulations were performed in 3D simulator available in Labview robotics module. Finally, experimental tests are performed to demonstrate system operation on differential steering mobile robot.

METHODOLOGY

This section presents the set of methods and materials involved in the development of this project. First of all kinematic mobile robot model is described. Then, the experimental platform is presented , as well as sensors and navigation system architecture.

Kinematic Mobile Robot Model

Considering the differential mobile robot model (Al Khatib et al. 2015) which has two identical wheels, positioned in the same axis, moving only around this, independently controlled by motors. Considering the pure rolling wheels, so there are no side slips during movement. Such features have restrictive nature on the system, preventing the movement of this in any direction in an initial moment, characterizing it as non holonomic (Siegwart et al. 2004) .The analysis of the system is made in kinematic level. It is assumed that the center of the mass is located in its geometric center, and hence eliminating possible unbalances accelerations effects. Figure 1 shows the differential robot model, in which position the robot is described by x, y and the angle θ in respect to the x global axis.

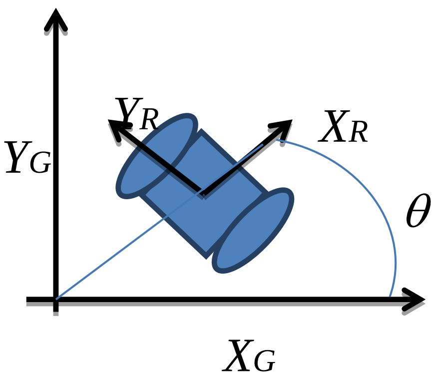

Figure 1 - Mobile Robot Schematics

The global X_G, Y_G coordinates and robots local X_R, Y_R coordinates are related by an orthogonal rotation matrix R.

Experimental Robot Platform

In the practical part of the project a DaNI robot or Robotic Started Kit 2.0 is used, as shown in Figure 2. This robot is a robotics kit NI platform for teaching, research and prototyping using LabVIEW as a development environment. The LabVIEW platform brings some facilities as a dedicated module to DANI robot, as well as libraries for visualization and data analysis.

DaNI robot includes the NI Single Board RIO controller module, mounted on top of a structure of a reconfigurable chassis. The controller board is an industrial application board for embedded systems, which integrates real-time processor 400 MHz with 128 MB of DRAM, 256 MB of non-volatile storage, FPGA (Field Programmable gate Array) with 110 digital inputs and outputs, 32 input channels and 4 output analog channels, 10 / 100 BASE-T Ethernet port, RS232 serial port for peripheral devices(National Instruments 2011) .

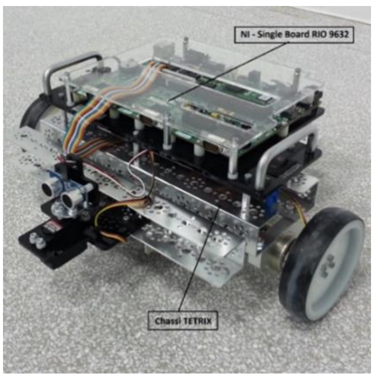

Figure 2 –NI Robotic started kit 2.0, DaNI.

The chassis is mounted on top of the two 4 inches wheels powered by 12V DC motors with 152 rpm and 300 oz-in (Ounce Inch) of torque. There is a free rolling wheel of mecanum type. The kit also includes two optical quadrature encoder, with 400 pulses per revolution, for the rotation of each wheel read as well as an ultrasonic sensor, which measures distances of 2 cm to 3 m, mounted on top of a servo-motor the which allows for sensing an angle of 180 degrees as the rotation of the servo-motor.

MEMS Gyroscope

The use of the MEMS sensor has the advantage of its low cost, small size, low weight and low power consumption (Chen et al. 2012).

The gyroscope is a device that measures angular velocity and position of an object about its rotation axis (Acar and Shkel 2008). The low-cost MEMS gyroscopes used to determine those angles are known to have poor bias stability, which may contribute to worse repeatability of the obtained angles. It can be demonstrated that introduction of some kind of filtering of the gyroscope signal leads to improvement of the measurement quality(Lechowicz and Augustynowicz 2015).

For greater accuracy of the gyroscope device calibration is necessary. The calibration is done by measuring during a few minutes in the idle state. Sensors offset are calculated from acquired data (Ladislav, Pavel, and Martin 2014).

Navigation System Architecture

The experimental navigation system is comprised of an *android* device (smartphone or tablet) , the National Instruments Robotic Starter Kit 2.0, also known as DaNI robot , a wireless communication router device connected to the robot and a computer to prompt and collect trajectory data.

All the processing activity is done embedded in the Single Board- RIO controller, the main DaNI robot board controller. For communication, purpose between all the components a local network is established on a router connected to DANI robot. The raw sensor data of the smartphone's sensors are sent to the robot through an android application that uses the WLAN network. With the data available from the sensors of the smartphone, the robot performs real-time position estimation algorithm calculates the control actuation and sends all the data to the computer terminal prompt. The operation schematic is shown in Figure 3.

Figure 3 - Communication of the system components

A block diagram of the system modules is presented in Figure 4. The reference path is input to the system which is compared with sensor fusion output generating the error vector. This error state vector enters the controller module and a control effort is generated and applied to the plant which is the mobile robot. Then the sensor data is collected and provided to the sensor fusion which generates the state estimation which is fed back to the comparison.

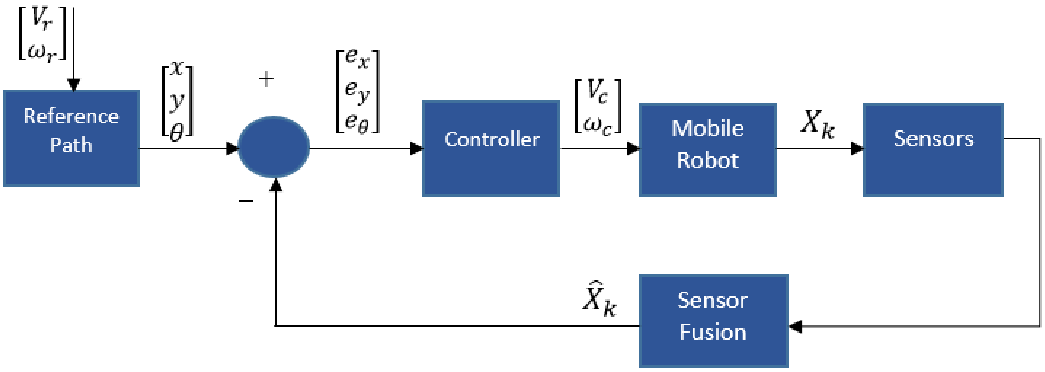

Figure 4 - Navigation system block diagram

State Estimation and Sensor Fusion

Among sensor fusion technique, the most widely used algorithm for state estimation is the Kalman filter and its variations. Kalman filter is an optimal state observer which provides solution for a discrete state estimation linear problem. Considering the system:

$$\begin{aligned} x_k &= \mathbf{A}x_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + w_{k-1} \\ z_k &= \mathbf{H}x_{k-1} + \gamma_{k-1} \end{aligned} \tag{1}$$

Where **A**, **B** and **H** are the state transition, the control and the observation matrices, respectively. The w_{k-1} variable represents the noise due to state transition of the system and γ_{k-1} represents the measurement noise. Both noises are considered independent and of Gaussian distribution and have associated covariance matrices **Q** and **R**, respectively.

The *a priori state* of the filter is given by:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{x}_{k-1} + \mathbf{B}u_k \tag{2}$$

With associated covariance:

$$\mathbf{P}_k^- = \mathbf{A}P_{k-1}\mathbf{A}^T + \mathbf{Q} \tag{3}$$

The Kalman gain is obtained from:

$$\mathbf{K}_k = \mathbf{P}_k{}^{-}\mathbf{H}^T(\mathbf{H}\mathbf{P}_k{}^{-}\mathbf{H}^T + \mathbf{R})^{-1} \tag{4}$$

Then the correction is made from gain ponderation over residual between measured output of the system and the *a priori* state:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k{}^{-} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k{}^{-}) \tag{5}$$

With associated covariance:

$$\mathbf{P}_k = (1 - \mathbf{K}_k\mathbf{H})\mathbf{P}_k{}^{-} \tag{6}$$

In this paper, the Kalman filter was used both like state estimator in the estimation of the coordinates $[\hat{x}_k\ \hat{y}_k\ ]$ of the robot and the orientation angle $[\hat{\theta}_k]$ and like sensor fusion technique, as several filters were implemented in different layers of sensor fusion (Figure 5).

In Figure 5 the first instance of Kalman filter receives as input the velocities read by wheel encoders transformed to the measured state vector. Parallel to the first filter, Filter 2 performs an estimate of the steering angle θ from the raw gyroscope measurements of the android sensor. Finally, to obtain a more reliable value for the steering angle θ of the robot, the fusion is made by a third instance of the Kalman filter between two previous Kalman filter outputs, using as *a priori* state the output from filter 1 and as observed variable the output from de filter 2.

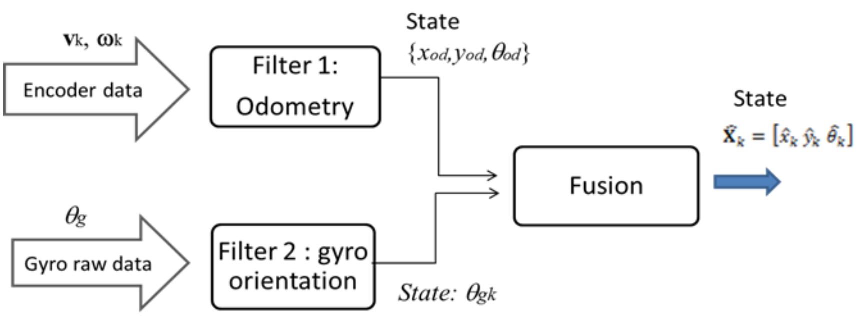

Figure 5 - Schematic of sensor data estimation and fusion

Once obtained the state $\hat{\mathbf{x}}_k = [\hat{x}_k\ \hat{y}_k\ \hat{\theta}_k]$ it is then fed back to the controller module as the presumable robot state. In fact, as one can see later, the estimated state $\hat{\mathbf{x}}_k$ differs from the real robot state by the amount of associated error, described by state covariance of the fusion filter.

The control module

The trajectory control module is based on State Dependent Riccati Equation method which is solves the nonlinear, infinite-horizon, autonomous control problem (Cloutier 1997).

This method considers the nonlinear regulator problem of infinite horizon in the form:

$$\dot{\mathbf{e}} = \mathbf{A}(e)\mathbf{e} + \mathbf{B}(e)\mathbf{u} \quad (7)$$

Where $[e_1 \quad e_2 \quad e_3]^T$ is defined as the difference between the current trajectory and the reference trajectory for each state.

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} x - x_r \\ y - y_r \\ \theta - \theta_r \end{bmatrix} \quad (8)$$

The functional minimized is given by:

$$J = \frac{1}{2}\int_0^{\infty} [\mathbf{e}^T \mathbf{Q}(e)\mathbf{e} + \mathbf{u}^T \mathbf{R}(e)\mathbf{u}]\, dt \quad (9)$$

Where $\mathbf{e} \in \mathbb{R}^{\propto}$ is a state vector and matrices $\mathbf{Q}(e)$ and $\mathbf{R}(e)$ are positive defined for all e. The control $\mathbf{u}$ is given by the equation:

$$\mathbf{u} = -\mathbf{R}^{-1}(e)\mathbf{B}^T(e)\mathbf{P}(e)\mathbf{e} \quad (10)$$

Where the matrix $\mathbf{P}(e)$ is obtained from the equation solution 9.

$$\mathbf{P}(e)\mathbf{A}(e) + \mathbf{A}^{\mathbf{T}}(e)\mathbf{P}(e) - \mathbf{P}(e)\mathbf{B}(e)\mathbf{R}^{-1}(e)\mathbf{B}^T(e)\mathbf{P}(e) + \mathbf{Q}(e) = 0 \quad (11)$$

The equation (11) is a Riccati equation and its the solution is used in (10), on the calculation of control effort $\mathbf{u}$, necessary to minimize the error system (7).

RESULTS AND DISCUSSION

The developments of the navigation system, numerical simulations, as well as experimental tests were performed in Labview environment. The controller module is based on SDRE control method. Initial conditions were chosen for both numerical simulation and experiments as $x_0 = 0{,}75, y_0 = 0\,, \theta_0 = 0$. The reference trajectory is dictated by linear and angular reference $V_r = 0{,}15\, m/s$ e $\omega_r = 0{,}2\, rad/s$. In the following sections simulation and experimental results are presented.

Simulations

Numerical simulations were performed in order to present the efficacy of control scheme and also to test the navigation system. In this case, the gyroscope was not included in simulation. Only odometry was considered. The integration of the controlled system is performed online so the integration method is the step integration. Because of that the integration error is considerable in these simulations. In figure 6 the circular trajectory is depicted. The blue solid line represents a controlled system trajectory. The black trajectory is a simulated odometry with error and green is the Kalman filter output trajectory for odometry process. All trajectories stay close and converge to a red solid line which is a desired trajectory.

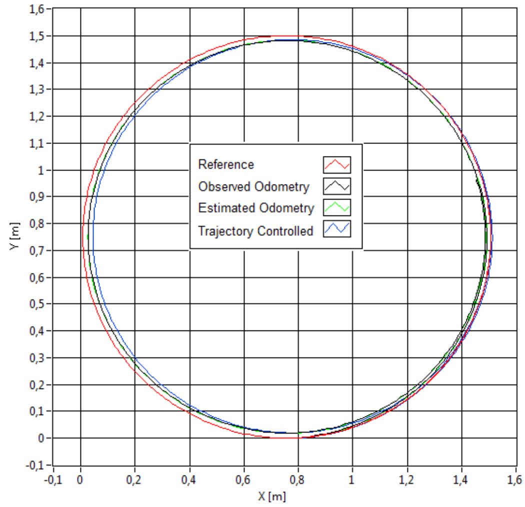

Figure 6 - Robot navigation path in the S(x, y) plane for a circular reference with parameters x_0=0,75, y_0 =0 , θ_0 =0 and V_r=0,15 m/s e ω_r=0,2 rad/s

In Fig. 7 the control vector effort is shown during the simulation time. The u1 é related to the linear velocity and u2 controls the angular velocity. Even in the transient, the control effort is within accepted limits in terms of linear and angular velocities.

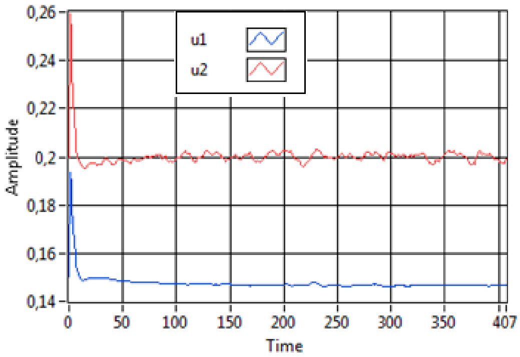

Figure 7 - Control effort evolution over time to the circular reference, knowing that u1 is Vc and u2 is ω_c

Figure 8 present a linear trajectory in which the same trajectories are presented. All trajectories are maintained on the reference.

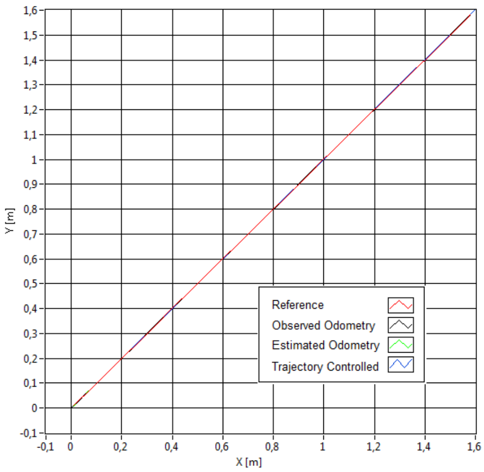

Figure 8 - Robot navigation path in the S(x, y) plane for a linear reference with parameters $x_0 = 0$, $y_0 = 0$, $\theta_0 = {}^{\pi}/_{4}\ rad$ and V_r = 0,15 m/s e $\omega_r = 0\ rad/s$

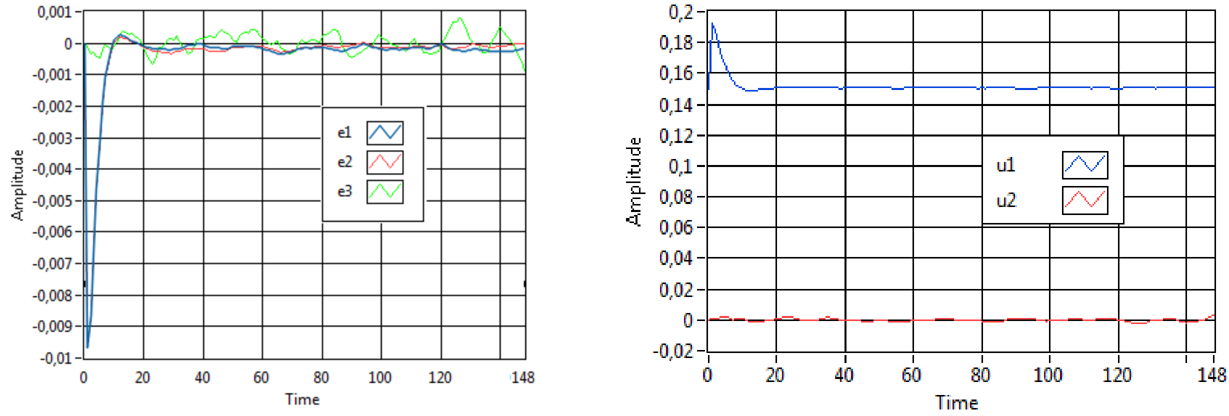

Figure 9 - left - Error between the estimated path of the odometry and the linear reference. Right- Control effort evolution over time to the linear reference.

Figure 9 left shows the errors between estimated trajectory and the reference. This error stays within integration error range.

The numerical simulations in general show the adequacy of the estimation system as well as the control module. The small deviations are attributed to the poor integration technique.

Experimental results

Experiments were conducted on the platform DaNI Robotic Starter Kit 2.0 described previously and implementing the Navigation System Architecture also described previously. All data is retrieved by the interface developped in Labview and the graphics are presented in the next figures. Figure 10 depics a circular reference trajectory in red, also odometry trajecty, the solid blue line, represents the raw data from encoders. The estimated odometry trajectory in green is a fitltered trajectory by the first Kalman filter. The controlled trajectory in magenta is a theoretical trajectory and should not be taken into account as it is not fed back with practical measurement and does not represent the experimental control process. Finally the resultant trajecty from the sensor fusion, which is the output of the third Kalman filter is presented in blue dotted trajectory. The filtered gyroscope output is not depicted in this figures, as it provides only angle output. All trajectories remain close to the reference, in the first turn of the circle. Then the fusion trajectory starts to deviate from the reference as the errors in estimation accumulate.

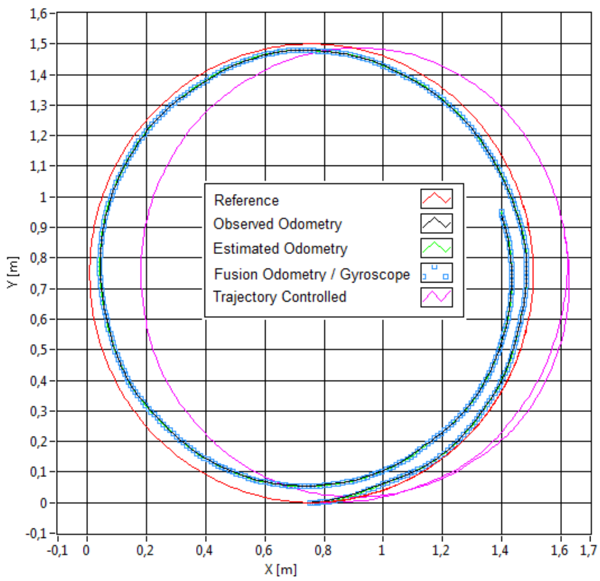

Figure 10 - Robot navigation path in the S(x, y) plane for the reference circle with $x_0 = 0,75, y_0 = 0 , \theta_0 = 0$ and $V_r = 15\ m/s$ e $\omega_r = 0,2\ rad/s$

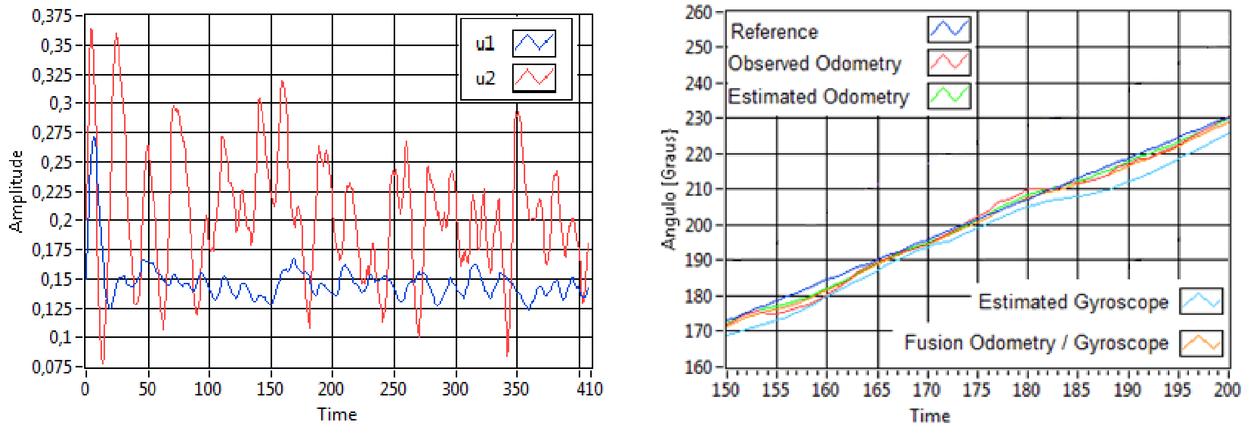

Figure 11 - Control effort evolution over time to the reference circle, knowing that u1 is Vc and u2 is ω_c

Figure 11 left depict the control efforts for the case of circular trajectory performed by robot. The Fig 11 right shows the angle measured and estimated by gyroscope and odometry, as well as the reference. As can be noticed the gyroscope sensor is a source of a great error. In this particular case the odometry process performs better, so the Kalman fusion trusts more the odometry process.

All this previous results only show the estimations of the robots trajectory by different sensors and filtered processes. In order to visualize the real trajectory of the robot, further experiments were conducted placing a camera on the ceiling and filming the robots trajectory as performed. Images were analyzed by a Kinovea free software which allows the object tracking and the some figure overlapping onto the footage. The image on the figure 12 is the result of some frames from the first circle that robot performs combined with the data retrieved (Figure 10). It should be noted that also this process has some error associated with the lens not accurate centralization, distortion, image rotation a scaling. Robot´s initial position is at 0,75 m in *x* axis and 0 at *y* axis. From this picture it is clear that the robot´s localization system, the control module work properly and maintains the system close to the desired trajectory. When deviation occurs and the position estimation can detect it, the regulator takes place and corrects it. Also it is clear that when estimation error occurs in respect to the real position the system it is not possible to correct the robot´s deviation.

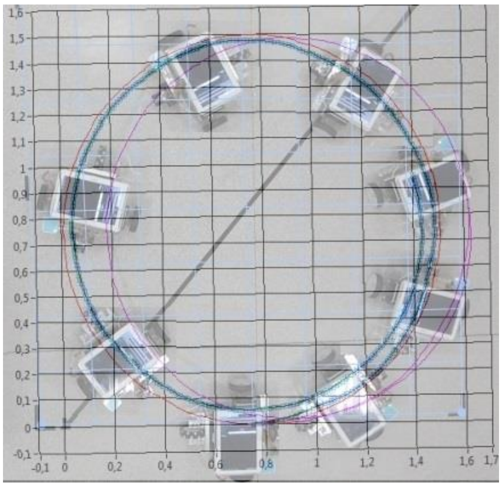

Figure 12 - Illustration with some frames of recorded video to compare the trajectory estimated by the sensor fusion and the actual trajectory of the robot.

Furthermore, experimental test were performed for a linear trajectory regime, in which the robot starts on top of the reference and should maintain the constant 45 degree inclination in respect to the *x* global coordinates axis. The Figure 13 shows the trajectories evolution. All the trajectories remain very close to the reference, slightly deviating in the end of the trajectory, as estimation errors accumulate. Figure 14 shows the effort to perform such a trajectory.

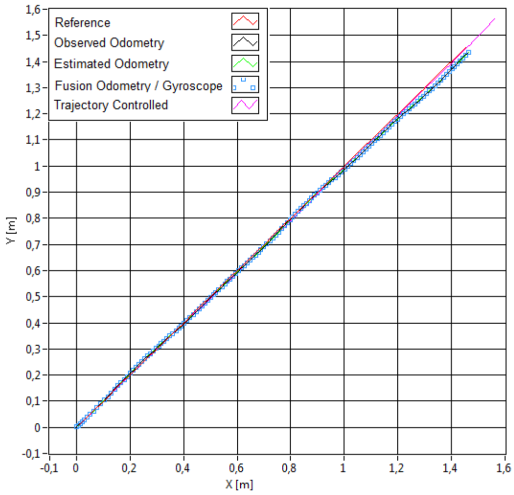

Figure 13 - Robot navigation path in the xy plane for a linear reference with parameters $x_0 = 0$, $y_0 = 0$, $\theta_0 = {}^{\pi}/_{4}\ rad$, and V_r = 0,15 m/s e $\omega_r = 0\ rad/s$

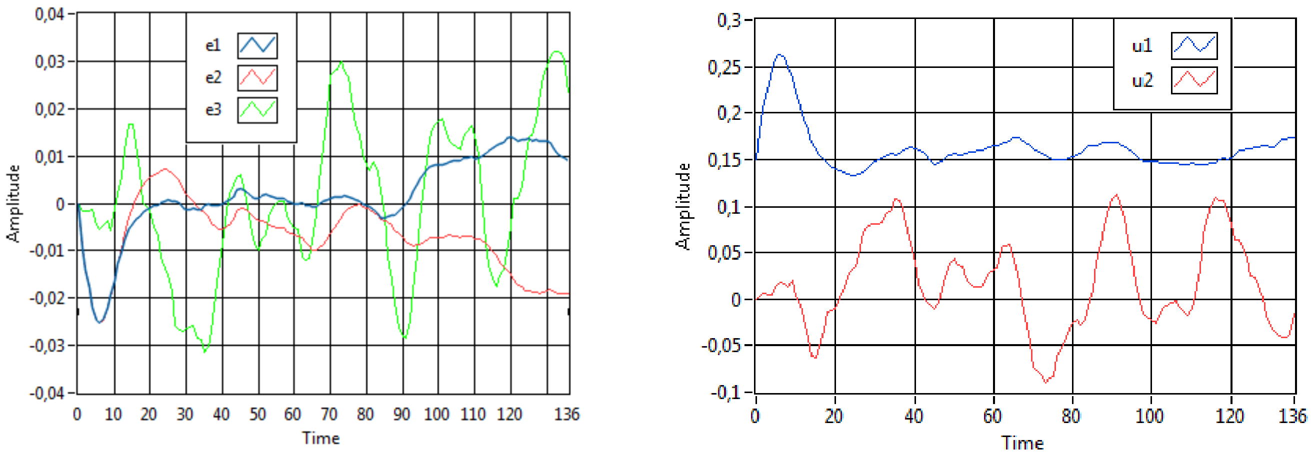

Figure 14 - Control effort evolution over time for linear reference

In the Fig. 14 left it is shown the errors between estimated position of the robot by a sensor fusion and a desired trajectory. In the right picture: the control u2 fluctuates and is not equal to zero which depict the real effort to maintain the robot following the line. This is mainly because of non equal, unbalanced and unsynchronized actuation in robots wheels motors.

Analyzing the experimental tests, it is clear that after a certain amount of time both the gyroscope and the odometry, start to drift due to the incremental nature of its measurement. This effect is more pronounced in gyroscope. Futher error quantization analysis can precisely show the instant in which this sensor reading can become useless. Another sensor readings can be introduced in order to enhance the performance. Also to deal with this issue in future implementation it is possible to use a different robotic model, the one that considers its dynamics and not only kinematics. This way it will be possible to insert directly the linear acceleration and angular velocity measurements.

CONCLUSION

In this paper the *android* bases sensor fusion system architecture for robot navigation is presented. This system contains a position estimation module, which uses the gyroscope of an android mobile device such as the smartphone or a tablet. This device is placed on the mobile robotic platform and connected via wireless network to the robots main controller. The robot performs encoder sensor reading, odometry process, embedded trajectory control and wheel actuation algorithms and sends all the data to a computer terminal for telemetry purpose.

Numerical simulations and experimental work has been conducted to assess the overall performance of the navigation. Two main regimes, the linear and circular trajectories were performed and presented as the main results. It has been shown that the system performs the state estimation within a certain small error range of accuracy and maintains the robot close to the reference. Future test can address regimes were a robot is placed outside the reference to show the convergence of the overall system to the desired trajectory.

Also it is noticeable that the sensor scheme for this application deals with the low-cost, compact, non dedicated sensors as well as the wireless communication between system components. The first one certainly degrades the performance of the coordinate estimation over time. The second one causes important delays which can be the fact between the poor and proper convergence of the trajectory. These entire situations were observed during the experimental test and can be the inspiration for the improvement for the incremental position estimation methods. Altough the poor performance over time of some sensors this work main contribution is to put together a navigation system which unites powerful controller with a sensor fusion and state estimation architecture. Other sensor components, as well as the plant models can be used to enhance the performance of the navigation system.

REFERENCES

Acar, Cenk, and Andrei Shkel. 2008. springer *MEMS Vibratory Gyroscopes: Structural Approaches to Improve Robustness*. Springer Science & Business Media. BOOK.

Chen, Cheng, Wennan Chai, Ahmad Kamal Nasir, and Hubert Roth. 2012. “Low Cost IMU Based Indoor Mobile Robot Navigation with the Assist ofOdometry and Wi-Fi U Sing Dynamic Constraints.” *Proc. IEEE/ION Position Location and Navigation Symposium (PLANS’12)*: 1274–79.

Cloutier, JR. 1997. “State-Dependent Riccati Equation Techniques: An Overview.” *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)* (June 1997): 932–36 vol.2. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=609663\nhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=609663.

Dekhil, Mohamed et al. 1998. “Sensor Fusion.” *Control Problems in Robotics and Automation* 230: 193–207. http://dx.doi.org/10.1007/BFb0015084.

Al Khatib, Ehab I., Mohammad A Jaradat, Mamoun Abdel-Hafez, and Milad Roigari. 2015. “Multiple Sensor Fusion for Mobile Robot Localization and Navigation Using the Extended Kalman Filter.” In *2015 10th International Symposium on Mechatronics and Its Applications (ISMA)*, IEEE, 1–5. http://ieeexplore.ieee.org/document/7373480/.

Ladislav, Beran, Chmelar Pavel, and Dobrovolny Martin. 2014. “Navigation of Robotic Platform with Using Inertial Measurement Unit and Direct Cosine Matrix.” In *Proceedings ELMAR-2014*, IEEE, 1–4. http://ieeexplore.ieee.org/document/6923322/.

Lechowicz, Andrzej, and Andrzej Augustynowicz. 2015. “Mechatronics - Ideas for Industrial Application.” *Advances in Intelligent Systems and Computing* 317: 473–82. http://www.scopus.com/inward/record.url?eid=2-s2.0-84921367766&partnerID=tZOtx3y1.

Mahmood, Azhar, Asim Baig, and Qaisar Ahsan. 2012. “Real Time Localization of Mobile Robotic Platform via Fusion of Inertial and Visual Navigation System.” *2012 International Conference of Robotics and Artificial Intelligence*: 40–44. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6413407.

National Instruments. 2011. “NI LabVIEW Robotics Started Kit.” http://www.ni.com/datasheet/pdf/en/ds-217.

Rodriguez, Angel, and Ubejd Shala. 2011. “Indoor Positioning Using Sensor-Fusion in Android Devices.” (September): 58.

Siegwart, Roland et al. 2004. *Autonomous Mobile Robots*. Bradford Book. BOOK.

Woodman, Oliver J. 2007. “An Introduction to Inertial Navigation.” *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696* 14: 15.

RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the material included in this paper.