Proceedings in Manufacturing Systems, Volume 15, Issue 1, 2020, 27-34 ISSN 2067-9238

PID CONTROLLER BASED ON A GYROSCOPE SENSOR FOR AN OMNIDIRECTIONAL MOBILE PLATFORM

Adrian MAROȘAN[1,*], George CONSTANTIN[2]

[1)] Assist. Prof., PhD Student, Robots and Manufacturing Systems Department, University "Politehnica" of Bucharest, Romania
[2)] PhD, Prof., Robots and Manufacturing Systems Department, University "Politehnica" of Bucharest, Romania

Abstract: *This paper refers to the design of the control of an omnidirectional mobile platform, by implementing a PID controller having as input signal a gyroscope sensor, which will indicate in real time the orientation angle of the platform. The use of this system follows the position and angle of orientation to control the angular and linear speed of the omnidirectional robot. A comparison is made between the system with direct control and the one with the PID control algorithm based on the gyroscope sensor. The entire system is controlled by an Arduino Mega 2560 development board, based on an Atmega microcontroller. Following the tests performed, the controller presented in this paper offers an optimal solution to minimize the differences between the reference and the output trajectory. The obtained results demonstrate the effectiveness of the proposed algorithms in the control of omnidirectional mobile platforms.*

Key words: *PID controller, omnidirectional robot, gyroscope sensor, Arduino Mega.*

1. INTRODUCTION

Omnidirectional mobile platforms are widely used in the production line card, but most have a fixed trajectory. The tasks performed by robots, mobile on wheels can be: loads, transport, monitoring, space, environment, etc. Compared to platforms, with conventional wheels, platforms, omnidirectional have advantages, large in terms of mobility in environments, crowded. Omnidirectional platforms can change, without changing direction, position or orientation, the wheels, being able, to perform, tasks easily, in crowded environments, with, static obstacles, and dynamic and in color, narrow [1, 2, 5]. Amazon is one of the most important examples of the use of omnidirectional platforms, they have significantly improved productivity. One of the most important problems encountered on these platforms is that related to their control, the most accurate positioning and speed. Proportional-integral-derivative (PID) controllers have a large scope in industrial control systems due to the small number of parameters that need to be adjusted. High-performance PLCs have the PID TUNING option, based on high-performance mathematical calculations. This PID tuning is done quickly and with very high accuracy [3, 6, 7].

PID parameters can be determined experimentally or by mathematical equations. It also uses specialized software such as Matlab with which you can determine a transfer function, and display a graph with the index of the system. Depending on this answer, the parameters can be adjusted manually until the most accurate stability of the whole system is reached.

* Corresponding author: Emil Cioran nr.4, Sibiu,Romania
Tel.:0742945750;
E-mail addresses: *adrian.marosan@ulbsibiu.ro* (A. Maroșan), *george.constantin@icmas.eu* (G. Constantin)

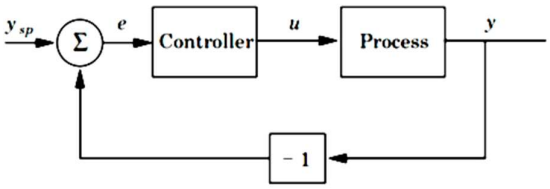

Fig. 1. Close loop control [3].

Fig 1 shows the basic control of a feedback loop system [11, 12, 13].

The linearity of the system is given by the transfer function:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1+C(s)G(s)}. \quad (1)$$

In many cases $G(s)$ is an unknown, which leads to a difficult design of the controller. In other cases, $G(s)$ is determined experimentally (where appropriate) or is a constant. The PID always offers a signal that is proportional to the error. This error is of two types, namely integral error and derivative error [2, 8, 9].

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}. \quad (2)$$

where: $u(t)$ and e (t) are signals and K_p, K_i, K_d are the setting parameters of the PID controller. The transfer function that characterizes these PID controllers is given by relation (3).

$$u(s) = K_p + K_i \frac{1}{s} + K_d s. \quad (3)$$

Next in this paper will be presented the control system of the omnidirectional mobile platform and the results obtained by integrating the gyroscope sensor in the PID controller [14, 15].

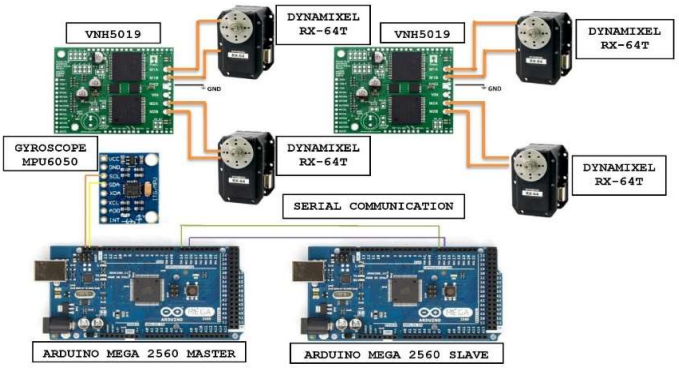

Fig. 2. Control system.

2. CONTROL SYSTEM FOR THE OMNIDIRECTIONAL MOBILE PLATFORM

Figure 2 shows the control system of the omnidirectional mobile platform based on the 2 Atemega2560 microcontrollers integrated on the Arduino Mega 2560 development boards. The robot is equipped with 4 Dynamixel Rx-64T servomotors for each omnidirectional wheel. These are based on Maxon brushless motors. According to Fig. 3, this type of motors was used due to their reliability, high torque and low price.

In the graph shown in Fig. 3 it can be seen that the actuators used need a supply voltage of 12 V and a maximum current of 4.1 A. For these reasons we chose to use two VNH5019 drivers, one for each of the two motors. These drivers are shown in Fig. 4.

These drivers operate at a voltage between 5.5–24 V, and the maximum current they supported on each motor is 12 A continuous (30 A peak) or 24 A continuous (60 A peak) on a single motor connected to both channels. These drivers provide feedback on the direction of current and support a PWM command with a frequency of up to 20 kHz. The microcontroller can monitor the

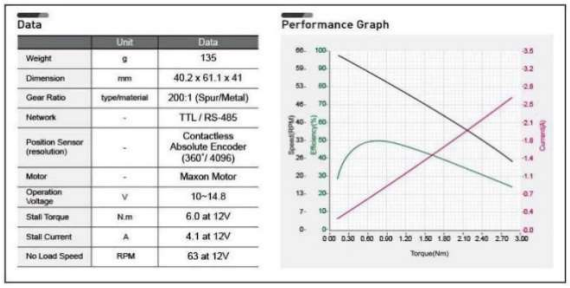

Fig. 3. Features Dynamixel Rx-64T [22].

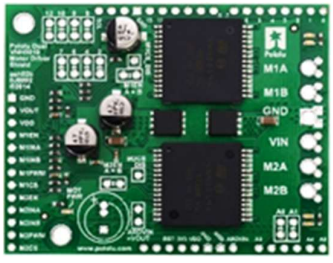

Fig.4. Pololu dual VNH5019[21].

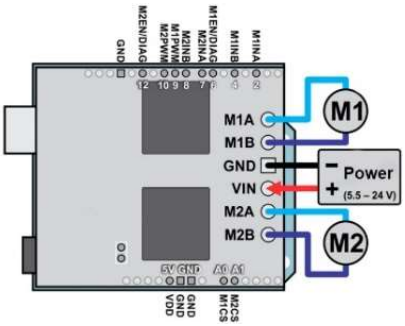

Fig. 5. VNH5019 driver pins for the Arduino Mega 2560 [21].

real-time power consumption of the motors, and the LEDs with which they provide the necessary feedback to help visually diagnose any problems or breakdowns. The drivers can be used as a shield for Arduino or as an independent driver with a PWM signal control from a microcontroller or a PLC.

The connection between the Arduino development board and the VNH5019 driver are presented in Fig. 5, and in Table 1 are explained the correspondence between the two electronic components.

Table 1

VNH5019 driver pins for the Arduino Mega 2560

Arduino PIN	VNH5019 PIN	COMMAND
Digital 2	M1INA	Engine 1 direction-input A
Digital 4	M1INB	Engine 1 direction - input B
Digital 6	M1EN/DIAG	Engine 1 activating input / output errors
Digital 7	M2INA	Engine 2 direction-input A
Digital 8	M2INB	Engine 2 direction - input B
Digital 9	M1PWM	Motor 1 speed-input
Digital 10	M2PWM	Motor 2 speed-input
Digital 12	M2EN/DIAG	Engine 2 enable input / output errors
Analog 0	M1CS	Current direction Motor 1 output
Analog1	M2CS	Current direction Motor 2 output

3. INTEGRATION OF THE GYROSCOPE SENSOR IN THE PID CONTROL SYSTEM

3.1. MEMS, Gyroscope sensor

The gyroscope is based on mechanical coupling and energy transfer from one axis to another. The principle of operation is based on the combination of the vibration of a sample mass and an orthogonal angular velocity input that together generates the Coriolis sinusoidal force. The general dynamic system is usually a spring damping system with two degrees of freedom, where the Coriolis force induced by rotation determines the energy transfer proportional to the angular input speed. If simultaneously the mass of the sample is driven in oscillation on the x-axis and subjected to rotation about the Z axis, then the displacement of the mass can be felt on a perpendicular y-axis [4, 10, 18].

The term used to define MEMS varies in different parts of the world. In the United States, they are predominantly called MEMS, while in some other countries they are called "Microsystems Technology" or "micromachined devices". More interesting are microsensors and micro-actuators. Microsensors and micro-actuators are appropriately classified as "translators", which are defined as devices that convert energy from one form to another. In the case of microsensors, the device usually converts a measured mechanical signal into an electrical signal. The latest research shows a major development of MEMS for a number of micro-actuators, including: microvalves for controlling gas and liquid flows; optical switches and mirrors for redirecting or modulating light beams; independently controlled micromirror panels for displays, micro-resonators for a number of different applications, micropumps to develop positive fluid pressures, micro-flaps to modulate airflows on air planes, and more. MEMS inertial sensors, especially accelerometers and gyroscopes, are increasingly common in all markets around the world. For example, MEMS accelerometers have replaced conventional accelerometers for in-car airbag deployment systems. MEMS technology has made it possible to integrate the accelerometer and electronics on a single silicon chip, at a cost of only a few dollars. These MEMS accelerometers are much smaller, more functional, lighter, more reliable and are produced for a fraction of the cost of conventional accelerometer elements at the macro level. More recently, MEMS gyroscopes (i.e., speed sensors) have been developed for both automotive and consumer electronics applications. MEMS inertia sensors are now used in every car sold, in the field of robotics (drones), and the most common are smartphones that have more and more microsystems [16, 17, 19].

3.2. MPU-6050 MEMS

In order to obtain a continuous feedback regarding the rotation angle, an MPU 6050 gyroscope sensor was used in the control system of the omnidirectional mobile platform. The block diagram is represented in Fig. 6 [5].

The abbreviation MPU comes from Motion Processing Unit and this module contains an integrated circuit with accelerometer, gyroscope and temperature sensor. This device combines a 3-axis gyroscope and a 3-axis accelerometer on the same silicon tablet, together with an on-board digital motion processor (DMP™), which processes complex 6-axis Motion-Fusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I2C bus, which allows the devices to gather a complete set of sensor data without the intervention of the system processor.

Fig. 6. MPU-6000 Family Block Diagram [20].

Gyroscopes and accelerometers are available in a wide variety. The choice of these devices depends on several features, as well as gyroscopes and accelerometers separately, they depend on the number of axes required, the measuring range of the sensor and the interface. The main parameters for MPU6050 are presented in Table 2.

For this application, only the position angle on the *Z* axis is needed. The information provided by the sensor has the following notations:

Axis, *Ay*, *Az* – raw acceleration measured along each axis (m/s^2); *Gx*, *Gy*, *Gz* – unprocessed gyroscope measurements measured around each axis (°/s); φ – roll rotation around the *X* axis; θ – pitch rotation around the *Y* axis; ψ – yaw rotation around the *Z* axis.

The orientation of the axes as well as the connection pins of the sensor are represented.

In this application we need to know only the angle of rotation around the *Z* axis (ψ – yaw). The gyroscope is connected to the Arduino development board and an I2C communication is used. Table 3 shows the connection pins between the two devices.

Table 2

The main parameters for MPU6050 [5]

Parameter	Rating
Supply Voltage, VDD	0.5V to +6V
VLOGIC Input Voltage Level (MPU-6050)	−0.5V to VDD + 0.5V
REGOUT	−0.5V to 2V
Input Voltage Level (CLKIN, AUX_DA, AD0, FSYNC, INT, SCL, SDA)	−0.5V to VDD + 0.5V
CPOUT (2.5 V ≤ VDD ≤ 3.6 V)	−0.5V to 30V
Acceleration (Any Axis, unpowered)	10,000g for 0.2ms
Operating Temperature Range	−40°C to +105°C
Storage Temperature Range	−40°C to +125°C
Electrostatic Discharge (ESD) Protection	2kV (HBM); 250V (MM)
Latch-up	JEDEC Class II (2),125°C ±100mA

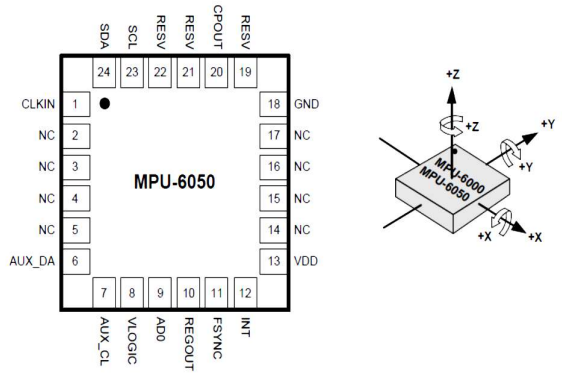

Fig. 7. Orientation of Axes of Sensitivity, Polarity of Rotation and output pins [20].

Table3

Pin connections between Arduino Mega2560 and MPU6050

Arduino Mega2560	MPU6050
PIN 43, SCL	SCL
PIN 44. SDA	SDA
GND	GND
5V	VCC

3.3. Control design

The block diagram in Fig.8 represents an amount of automatic control for the omnidirectional mobile platform.

"Reference position" represent the value of the angle imposed. This value is the "input" of the controller.

"Position error" is the difference between the imposed position and the value that is read from the gyroscope sensor.

The PID controller consists of the three constants K_p (proportional), K_i (integral), K_d (derivative). At the output of this controller we will have the response signal which in this case is a PWM signal. This will represent the "output" of the controller. The signal will go further to the "VNH5019" drivers which will control two DC motors. The actual position angle is given by the MPU-6050 gyroscope sensor, which is actually the element that will closes the feedback loop of this system. Between the two development boards from Arduino there is a serial communication, illustrated in Fig. 2 where depending on the type of movement that the robot must make from the master board a signal will be sent to something slave. Master control board will in turn receive the PWM command seed from the PID controller.

The analytical expressions that define the controller in Fig. 8. can been expressed in the following form of the following calculation relations:

$$P_r = R_p - \theta, \tag{4}$$

where R_p is position error and R_p over reference position, and θ is the value of the angle read from the gyroscope.

$$D = P_r - P_{r1}, \tag{5}$$

where D is derivative and P_r is position error, and P_{r1} is the previous position error.

From relation (4) and (5) it results that in order to determine the PWM necessary for the control of the motors we need the following calculations:

$$C_s = (K_P \cdot P_r) - (K_d \cdot D), \tag{6}$$

where: C_s is the correction speed, and K_P and K_d are the derivative and proportional constants in the PID controller.

From all the above relations it results that the control PWM required for the $M1$ motor and the $M2$ motor is described by the following relations:

$$PWM\ M_1 = I_s - C_s, \tag{7}$$

$$PWM\ M_2 = I_s + C_s, \tag{8}$$

where I_s is imposed speed. Following these calculations and experimental tests we found that in the design of the controller we no longer need the integration factor, namely the constant K_i, for these reasons we will have only a PD controller, which after the experimental tests performed is functional for this omnidirectional robot.

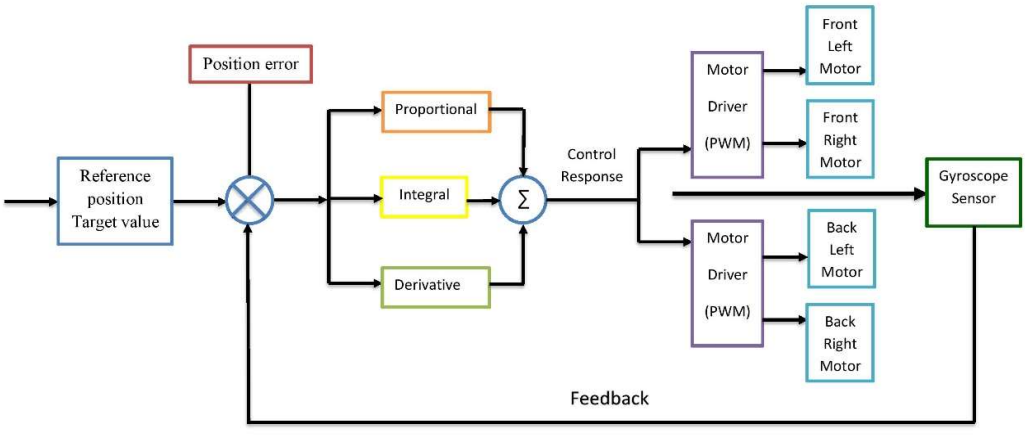

Fig. 8. Block diagram for the control of the omnidirectional mobile platform.

4. EXPERIMENTAL RESULTS

This section presents the real-time experimental results for motion control and positioning of the omnidirectional mobile platform. The aim is to monitor the position error at different rotations that the robot has to execute. PID, a system that is based only on a direct command, so there is no control or verification whether the robot will reach or exceed the desired angle.

Following the experimental procedures, a set of experimental results was obtained, which will be presented in several cases.

The test results show the variations of the angle θ in an uncontrolled system without feedback loop. After the integration of the PID system, the evolution of the positioning error P_r in relation to the angle θ read on the gyroscope will be illustrated.

The programming logic diagram is presented in Fig. 9 for the master control board and in Fig. 10 for the slave control board.

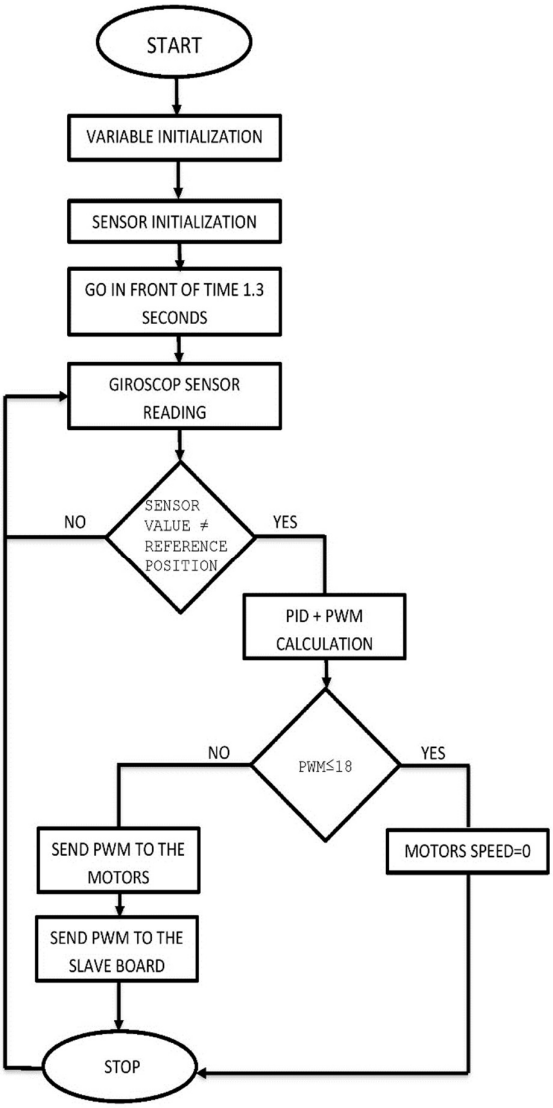

Fig. 9. Programming logic diagram for the master control board.

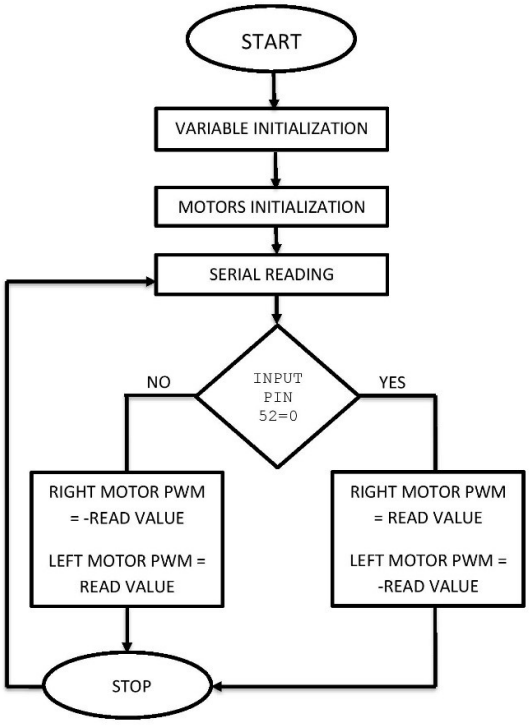

Fig. 10. Programming logic diagram for the slave control board.

4.1 Case no.1

In the first case there is a situation in which the omnidirectional mobile platform is forced to perform a 90 ° rotation. To perform this command is sent directly to the motors PWM signal. In Fig. 11 the values read with the gyroscope sensor for 6 repeated rotations are presented. The first three attempts are made at low speeds, then the others at maximum speeds. From these tests it results that at low speeds good values can be obtained, the angle being quite close to the real value. The differences are quite small ±(2–3) °. Instead for high speeds and high interference forces, this open loop control system is no longer efficient and the difference between the required angle and the real value is between (5–10) °.

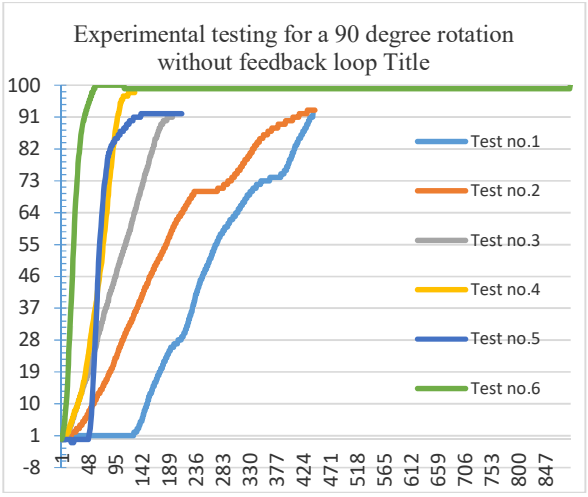

Fig. 11. Variation of rotation angle for an open loop system.

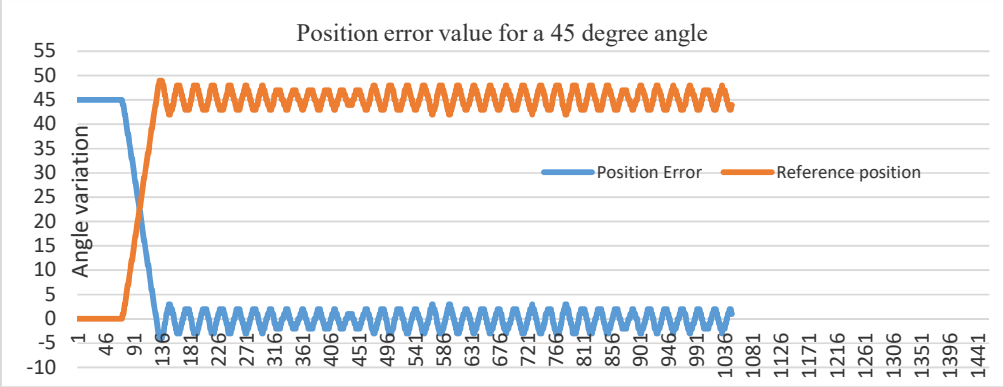

Fig. 12. Position error variation for an imposed angle of 45 °.

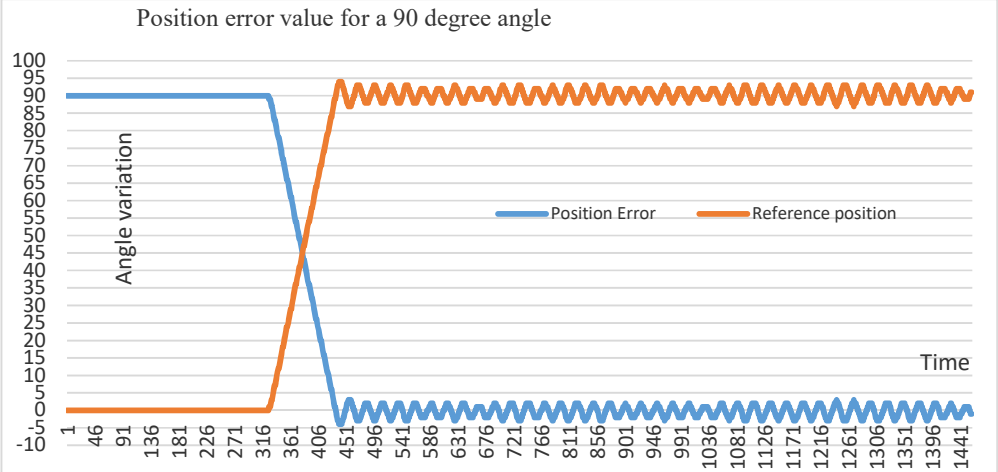

Fig. 13. Position error variation for an imposed angle of 90 °.

4.2 Case no. 2

Starting from the experimental results obtained in the first case, a system with feedback loop was introduced in the control program of the robot, whose design is illustrated in Fig. 8. This system is a PID controller without the integration factor, having as signal gyroscope sensor input and as output signal PWM command for the motor driver. The graph from Figs. 12 and 13 present results obtained having as scenario the following case:

- the robot travels a short linear distance of a certain period of time after which it performs a rotation of 45 °. For this test the constants are $K_p = 30$ and $K_d = 5$ and the required angle is 45 °. The PID controller is adjusted manually and the constants change after the experimental tests until the position error becomes 0.

- the robot travels a short linear distance of a certain period of time after which it performs a rotation of 90 °. For this test the constants are $K_p = 30$ and $K_d = 5$ and the required angle is 90 °. Following these tests represented graphically in Figs. 12 and 13, one can notice that the value of the angle read by the gyroscope sensor is quite close to the required value, but we have large variations between the minimum and maximum value, in other words there are oscillations, which means that we have an override of the PID and an unstable system. To fix this problem the constants K_p and K_d need to be modified.

4.3 Case no. 3

In the third case, the constants K_p and K_d are changed, trying to achieve the best possible adjustment of the PID. The values for the constants with which the experimental tests were performed are the following: $K_p = 10$ and $K_d = 20$. Two sets of tests were performed, one for a 45 ° angle, illustrated in Fig. 14 and one for a 90 ° angle illustrated in Fig. 15. From both graphs it can be seen that after changing the constants the stability of the system increased significantly, and by decreasing the constant K_p the override was eliminated and the difference between the positioning error and the reference one has a very small value.

5. CONCLUSIONS

Following these experimental tests, it can be seen that the integration of the control system with feedback loop based on a gyroscope sensor on the omnidirectional mobile platform, significantly improves the orientation and positioning accuracy. it is not a very precise one, and obtaining the parameters can only be done experimentally. However, these parameters are greatly influenced by any change in the system, in this case the

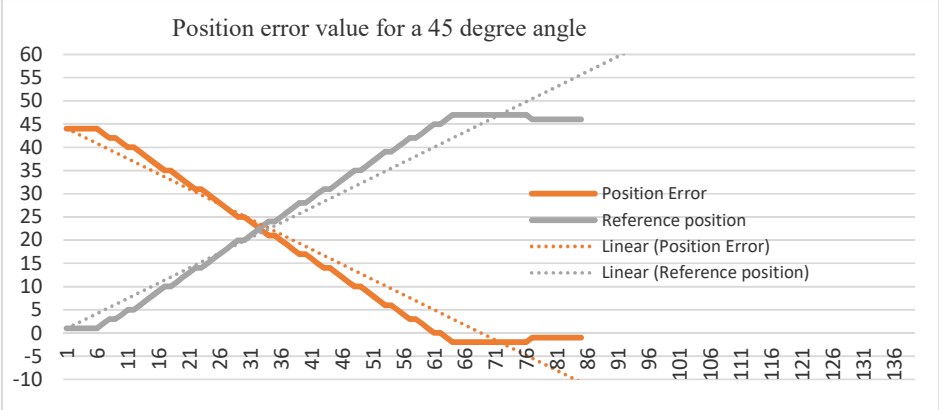

Fig. 14. Position error variation for an imposed angle of 45 °.

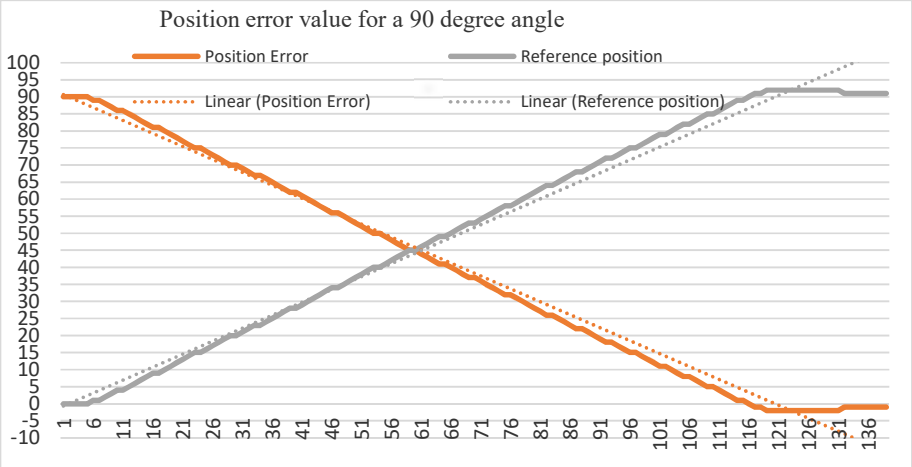

Fig. 15. Position error variation for an imposed angle of 90 °.

next stage will be focused on developing an algorithm that offers the possibility of automatic adjustment of the PID controller. This will lead to finding optimal parameters for the system. they will be able to upgrade along with other system changes such as speeding or sudden braking. Another proposed objective is the integration in the control system of 2 rotary encoders mounted on free omidirectional wheels that provide information for the linear movements for the *X* and *Y* axes. In this way we will have a completely controlled and automatic system. In this way they can be integrated. other algorithms that increase the accuracy of orientation and positioning in real time.

6. REFERENCES

[1] S.G. Racz, M. Crenganiș, A. Bârsan, I. A. Maroșan, *Omnidirectional Autonomous Mobile Robot With Mecanum Wheel*, the Scientific Bulletin Addendum, No. 4, 2019, pp. 112–124.

[2] M.O. Tătar, C. Popovici, D. Mândru, I. Ardelean, A. Pleșa, (2014, May, *Design and development of an autonomous omni-directional mobile robot with Mecanum wheels*, 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, pp. 1–6, IEEE.

[3] Xh. Mehmeti, *Adaptive PID controller design for joints of Humanoid Robot*, IFAC-PapersOnLine, Vol. 52, No. 25, 2019, pp. 110–112.

[4] L. Karl, M. Simandl, Parameter Estimation of MEMS Gyroscope Using Local State Estimation Methods, IFAC-PapersOnLine, Vol. 48, No. 28, 2015, pp. 279–284.

[5] C. Arantes, J. S. Esteves, *Development de a four mecanum wheels omnidirectional mobile platform enabling remote motion control through a net graphical application or an inertial measurment unit, University of Minho Guimarães, Portugal*, International Journal, Vol. 4, 2018, p. 243.

[6] L. Cedro, K. Wieczorkowski, *Optimizing PID controller gains to model the performance of a quadcopter*, Transportation Research Procedia, Vol. 40, 2019, pp.156–169.

[7] R. Marino, S. Scalzi, M. Netto, *Nested PID steering control for lane keeping in autonomous vehicles*, Control Engineering Practice, Vol. 19, 2011, pp. 1459–1467.

[8] T.J. Ren, T.C. Chen, C.J. Chen, *Motion control for a two-wheeled vehicle using a self-tuning PID controller*, Control Engineering Practice, Vol. 16, 2008, pp. 365–375.

[9] D. Vrancic, M. Huba, P.M. Oliveria, *PID controller tuning for integrating processes*, IFAC Papers On-Line, Vol. 51, No. 4, 2018, pp. 586–591.

[10] S. J. Huang, Y.W. Shiao, *2D path control of four omni wheels mobile platform with compassand gyroscope sensors,* Sensors and Actuators A: Physical, Vol. 234, 2015, pp. 302–310.

[11] J. M. Chairez, V. Santibanez, J. M. Valenzuela, *Adaptive control schemes applied to a control moment gyroscope of 2 degrees of freedom*, Mechatronics 57 (2019) 73–85.

[12] S. J. Hammoodi, K.S. Flayyih, A.R. Hamad, *Design and implementation speed control system of DC Motor based*

on PID control and Matlab Simulink, International Journal of Power Electronics and Drive System (IJPEDS), Vol. 11, No. 1, March 2020, pp. 127–134.

[13] A. Bârsan, *Position Control of a Mobile Robot through PID Controller*, ACTA Universitatis Cibiniensis, Vol. 71, No. 1, pp. 14–20, doi: https://doi.org/10.2478/aucts-2019-0004.

[14] J. Li, X. Tang, F. Xu, R. Huang, *Motion Analysis and Stability Control of Four-Wheeled Omnidirectional Mobile Robot under Limited Hardware Resource*, IOP Conf. Series: Journal of Physics: Conf. Series, Vol. 1487, 2020, p. 012040, doi:10.1088/1742-6596/1487/1/012040, CCEAI 2020.

[15] T. Peng, J. Qian, B. Zi, J. Liu, X. Wang, *Mechanical Design and Control System of an Omni-directional Mobile Robot for Material Conveying*, Procedia Cirp, Vol. 56, 2016, pp. 412–415.

[16] M. Rahmani, *MEMS gyroscope control using a novel compound robust control*, ISA Transactions, Vol. 72, 2018, pp. 37–43.

[17] M. Mirzaei, I. Hosseini, V. Ghaffari, *MEMS gyroscope fault detection and elimination for an underwater robot using the combination of smooth switching and dynamic redundancy method*, Microelectronics Reliability, Vol. 109, 2020, p. 113677.

[18] R. Votrubec, *Stabilization of Platform using Gyroscope*, Procedia Engineering, Vol. 69, 2014, pp. 410–414.

[19] https://www.mems-exchange.org, accessed: 2020-05-15.

[20] Sensor System on Chip, https://invensense.tdk.com, accessed: 2020-05-20.

[21] Pololu Catalog, https://www.pololu.com, accessed: 2020-05-14.

[22] Robotis, http://www.robotis.us, accessed: 2020-05-15.