COBEM 2021 26th International Congress of Mechanical Engineering
November 22-26, 2021, Virtual Congress, Brazil

# COB-2021-2068
# DEVEOPMENT OF ROBOTIC SYSTEM IN OPEN-SOURCE PLATFORM FOR MOBILE ROBOT

**Everton Silva Libânio and Claudio Garcia**
Polytechnic School of São Paulo University
Av. Prof. Luciano Gualberto, trav. 3 nº 158, Cidade Universitária,
São Paulo/SP, Brazil – CEP 05508-900
(e-mail: everton.libanio@gmail.com, clgarcia@lac.usp.br )

***Abstract.*** *In mobile robots, it is very important to have a functional robotic system, which makes it possible to carry out predetermined paths and makes it possible to execute further tests with non-linear dynamic controllers. This system can also be used, in addition to running the routes, the monitoring of these paths, involving both positions and executed speeds, in order to be able to analyze the performance of future controllers to be used. The objective of this work is to show the stages of the development of a robotic system on an open-source platform, for mobile robots, from the description of the robot, through the stages of odometry system validation, modeling and validation of the motor parameters by simulation and the tuning of a proportional integral controller by the method of Direct Synthesis. Once this is done, a trajectory controller based on reference speeds is inserted. Sensors were also used to measure the electric current of the motors, in order to measure the torque exerted by the robot and the static Kalman filter was applied to improve the acquisition of signals, due to the noise caused by the PWM of the motor drivers. Finally, displacement monitoring tests of the robot are performed, using the following monitoring modes: manual measuring of distance and time, measurement made by the own robotic system with the control loop with PI controller, with the trajectory controller included and monitoring the displacement of the robot with the use of the LIDAR sensor (Light Detection and Ranging), so that, by comparisons, carry out the validation of the monitoring of the route.*

***Keywords:*** *Mobile Robots, Direct Synthesis method, Static Kalman Filter, Monitor Robot Displacement, LIDAR Sensor,*

## 1. INTRODUCTION

Today, with technological advances and the high number of embedded devices, the use of mobile robots in many activities, such as logistical transportation of production (ALEJANDRO, 2019), external applications such as delivery of goods and use in agricultural activities for planting and crop inspection (DONOVAN, 2018). In urban centers, robots could perform various tasks, such as monitoring people (MARTIN, 2011), for surveillance applications (CORREIA, 2019) and autonomous wheelchairs (ANDALUZ et al. 2014).

When working on the development of robots, it is very important to have a functional robotic system, which is able to perform predetermined paths and to be able to monitor the system variables, position and speed of the robot and the paths traveled, thus making it possible to carry out further development with the implementation of controllers.

### 1.1 USED ROBOT

The robot used (see Figure 1) has a structure with four wheels, two motor rear wheels and the front wheels are driven by belts. The design of the H bridges, responsible for controlling the motors, was developed by (ALBRECHT, 2013) in previous work. The Arduino Atmega 2560 microcontroller is used, where the integral proportional controllers are executed, the trajectory controller based on reference speeds and where the desired paths are created, in a real time execution, such as following straight lines and circumferences.

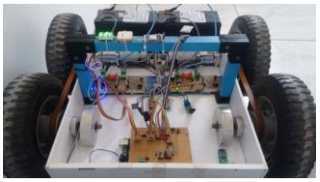

Figure 1: Mobile robot used.

The HC05 bluetooth serial communication module was used, responsible for receiving the commands of the routes to be carried out with the desired speed, executed in real time, and also responsible for sending data from the robot during the route for further processing of the information with the MATLAB software. The latter was also used to perform the simulations.

BaneBots RS-775 18V motors are used, with operating voltage from 6 to 20 V, 78% efficiency, stall current of 130A, unloaded speed of 19500 rpm without reduction and reduction factor 1:64 (ALBRECHT, 2013). The robot has a mass of approximately 18.5 kg, and based on the tests to determine the motor coefficients, the theoretical maximum speeds of the robot were determined, with the linear speed of 1.95 m / s and the angular speed of 6.195 rad / s.

To measure the wheel speed, a Hall effect sensor, model SS41, from Honeywell, was used, whose output is in an open collector, that is, an external pull-up resistor is used. The signal changes in value when the magnetic field changes orientation (ALBRECTH, 2013). In this project, a 6-pole stepper motor circular magnet was added, as shown in Figure 2, thus increasing the accuracy of measuring the motor rotation by 6 pulses per revolution. The sensor was fixed directly on the motor shaft, without being influenced by the motor reduction box.

Figure 2: Detail of the odometry system.

The sensor, ACS712-30A, was also added to measure the electric current of the motors, to measure the torque exerted by the robot. The sensor has a measuring range of -30 A to 30 A, measuring direct and alternating currents and detects inversion in the current flow. Despite being invasive in the measuring circuit, it has the great advantage of having a minimum insulation of 2.1 kVrms between the current acquisition terminals and the sensor supply terminals.

### 1.2 LIDAR SENSOR

For the robot tracking tests, the SICK LMS291-S05 LIDAR (Light Detection and Ranging) sensor was used, not integrated with the robot, having the principle of operation based on the measurement of time of flight laser pulses, with optical mirror that scans the laser and scan with openings of 180º and 100º and resolutions of 1º and 0.5º. To acquire the signals, the sicktoolbox.1.0.1 library was used, a virtual machine with an image of Ubuntu 12.04, in the latest version of Ubuntu 20.04 there was a compatibility problem with the current version of the gcc library and the LIDAR library and because of this was not possible to use this library with Ubuntu version 20.04. The development environment was CodeBlocks.

## 2. THEORETICAL FUNDAMENTALS

### 2.1 MODEL AND DETERMINATION OF MOTOR COEFFICIENTS

To determine the viscous coefficient of the motors, voltage values were applied and in steady state, where the acceleration is zero and in experiments the respective speeds were measured and then the average values of the coefficients were obtained. The motor manufacturer usually provides characteristics such as speed and torque constants, while others such as viscous coefficient and coefficient of inertia need to be determined through practical tests. Due to the dimensions of the motor, the resistance / inductance ratio (electrical constant) is much smaller than the inertia / (viscous coefficient) ratio (mechanical constant). Inductance was disregarded, thus obtaining the Equation (1) for the electrical model.

$$V(t) = R_a . i_a(t) + E(t) \quad (1)$$

where $R_a$ is the armature resistance, $i_a$ the armature current, $E(t)$ the induced electromotive force and $V(t)$ the armature voltage.

According to Newton's laws we have the following rotational equation for the electromagnetic torque of the motor:

$$T_g(t) = T_w(t) + T_j(t) + T_L(t) \quad (2)$$

where $T_j(t) = J.\dot{w}(t)$ refers to the conjugate due to the axis inertia, with tests in steady state this portion was considered null due to no have acceleration, $T_L(t)$ refers to the load conjugate, as the tests were performed with the motors without load, this portion was considered null. $T_w(t) = B.w(t) + T_j(t)$ is called the loss conjugate where *B.w (t)* is the viscous damping of the motor and wheels, being linear and proportional to the rotation and $T_f(t)$ refers to the friction contributions in the load and in the motor, being non-linear and of a value much lower than the first portion, being thus neglected in this analysis. Based on these considerations, torque results in Equation (3):

$$T_g(t) = B.w(t) \quad (3)$$

Considering the constant magnetic flux, the relationship between the electrical and mechanical parts of the motor is given below, where $w(t)$ is the rotation speed of the motor:

$$T_g(t) = K_t.i_a(t) \quad (4)$$
$$E(t) = K_e.w(t) \quad (5)$$

with $K_t$ being the torque constant, $K_e$ and the counter electromotive force constant and were calculated according to the tests provided by the manufacturer. $K_t$ was obtained by the ratio of the blocking torque divided by the blocking electrical current. In order to determine $K_e$, values provided for unloaded speed and unloaded current were used in the following relationship: $i_{free} = \frac{V(t)-K_e.w_{free}(t)}{Ra}$ from Equations (1) and (5). Thus, by equating Equations (3) and (4) and isolating the electric current in the equation, the Equation (6) is obtained to found the viscous motor coefficient:

$$B = \frac{K_t}{w(t)} . \left[\frac{V-K_e.w(t)}{Ra}\right] \quad (6)$$

The J coefficients of inertia of the motors were determined approximately with a deceleration test or "run down test" (BEZERRA, 2016). In which using the same previous test after a certain period of time of the motor in steady state, in that was removed the motor voltage and based on the speed versus time graph, a time $t_m$ was sought until the motor reduces the exp (-1) or 0.368 of the speed of the steady state. Knowing that the resulting torque of the motor is equal to zero, due to the removal of tension we will have the Equation (7):

$$J\dot{w}(t) + B.w(t) = 0 \quad (7)$$

And solving it we have:

$$w(t) = w_i(t).\mathrm{e}^{-t\frac{B}{J}} \quad (8)$$

being $w_i(t)$ the speed of regime and simplifying Equation (8) using the logarithm function, we arrive at:

$$J = t_m.B \quad (9)$$

In this estimate, Coulomb friction or dry friction and other types of friction that have nonlinear characteristics (ZEILMANN et al., 2010) were disregarded.

## 2.2 INDEPENDENT REPRESENTATION BY ACTUATOR

In actuator independent representation, each robot actuator is controlled as a SISO system. After replacing Equation (5) in Equation (1) and isolating the electric current in Equation (1), the expression of the electric current obtained is replaced in Equation (4) with $T_g(t) = K_t \frac{V(t)-K_e.w(t)}{Ra}$, this being the equation that represents the torque generated by the motor and equaling this with Equation (2) (equation for the electromagnetic torque of the motor), in which the load conjugate $T_L(t)$ is multiplied by the reduction ratio $r$ and finally the Laplace transform is applied in relation to the motor speed, obtaining the following model expression for each actuator in Equation (10).

$$\left(J_m s + B_m + \frac{(K_t K_e)}{R}\right) \mathrm{w_m}(s) = K_t \frac{V(s)}{R} - r.\tau_l(s) \quad (10)$$

where $\tau_l(s)$ is the torque of the robot acting on the motor, which is considered as a disturbance in the system, as shown in Figure 3.

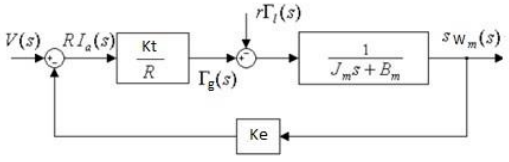


Figure 3: Diagram of the 1st order motor plant.
Adapted from: (CRUZ, 2018).

## 2.3 DISCRETE PID CONTROLLER

The controller will deliver a control voltage after the system feedback where the speed error will serve as an input to the controller.

To implement the controller in the microcontroller, it was decided to use a positional PI structure, in which the proportional, integral and derivative portions are included independently, as in Equation (11), and with a discretization by the back rectangular method $T_s$ is the sampling period.

$$u(t) = K_P.\left(e(t) + \frac{1}{T_I}.\int_0^t e(\tau)d\tau\right) \quad (11)$$

Equation (12) results from the proportional portion in discrete time.

$$u_p[n] = K_P.e[n] \quad (12)$$

Equation (13) results from integral portion in discrete time.

$$u_I(t) = \frac{1}{T_I}.\int_0^t e(\tau)d\tau \rightarrow u_I[n] = \frac{K_P.T_S}{T_I}.\sum_{k=0}^{n} e[k] \rightarrow u_I[n] = \frac{K_P.T_S}{T_I}.(e[n] - e[n-1]) \quad (13)$$

## 2.4 CONTROLLER TUNING

After obtaining the motor coefficients, it was decided to use the Direct Synthesis Method to tune the PI controllers, as this method is particularly efficient for fast control loops and helps to suppress oscillations (GARCIA, 2017) and due to the motors during the simulation do not show even dead time and much less fluctuations with noticeable periods, when increasing the plant's gain because it was not possible to use the methods of Ziegler-Nichols (OGATA, 2010) to effect the tuning of the gains.

The adjustment of the PI parameters was made for a first order plant using the Direct Synthesis method, in which the following relationships were adopted for the gains $k_p$ and $k_i$:

$$k_p = \frac{T}{k*T_c} \quad (14)$$

$$k_i = \frac{1}{T}. \quad (15)$$

$$k = \frac{Outut}{Input} \quad (16)$$

where $T_c$ is a time constant that defines how fast the plant will reach the equilibrium point, being chosen empirically according to the system requirements and $T$ is the time constant, which is made equal to the integration time constant $T_I$ and $k$ is the reason of the output for the input of the plant.

In practice, if there is a high overshoot in the control voltage, you can try to correct this with the addition of an acceleration ramp or even reducing the proportional gain obtained.

## 2.5 DIFFERENTIAL ROBOT KINEMATIC MODEL

The state of a vehicle can be determined by:

$$X = [\, x\ y\ \theta\ v\ \omega\,]^{\mathrm{T}} \quad (17)$$

where $x$ and $y$ are the position in the Cartesian coordinates, $\theta$ is the vehicle's angle in relation to the origin, $v$ is the referential linear speed of the vehicle and $\omega$ is the angular speed of the vehicle. In Figure 4 there is a schematic representation of the vehicle.

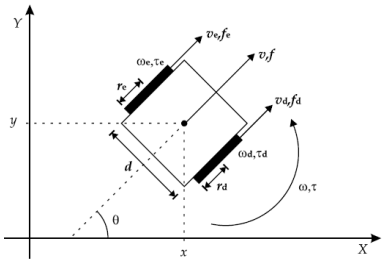


Figure 4: Schematic representation of the vehicle.
Author: (Vieira, 2005).

The reference speeds for the movements of the controller, linear speed and angular speed of the vehicle can be obtained by the speed ratios of each wheel, exposed by the following matrix relation:

$$u = T_w^u.w \quad (18)$$

where $u = \begin{bmatrix} v \\ \omega \end{bmatrix}$ $w = \begin{bmatrix} w_d \\ w_e \end{bmatrix}$ $T_w^u = \begin{bmatrix} \frac{r_d}{2} & \frac{r_e}{2} \\ \frac{r_d}{d} & -\frac{r_e}{d} \end{bmatrix}$ $w_d$ , $w_e$ are wheels speeds, $r_d$, $r_e$ are wheels radius and $d$ is the distance of wheels.

## 2.6 TRAJECTORY CONTROLLER

Due to the robot's constructive structure, orientation control is not necessary, non-holonomic restrictions prevent the robot from following speeds perpendicular to the orientation of the wheels, reaching the following equation.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}\cos\theta & -a.\operatorname{sen}\theta\\ \operatorname{sen}\theta & a.\cos\theta\end{bmatrix}.\begin{bmatrix}v\\ \omega\end{bmatrix} \tag{19}$$

being at *a* distance value between the robot's center of rotation and the tracking point that the robot wants to make its path, therefore *a* is different from zero, this displacement is used so that the matrix equation has its real inverse relation and thus the trajectory controller can be designed.

Because this configuration the trajectory controller of Equation (20) is equal to the inverse rotation matrix of the transformation of coordinates from the global reference plane to the local reference plane of the robot, because of it this controller accepts to work with global coordinates in the local reference of the robot without needing its conversion. The trajectory controller or inverse kinematic controller is expressed below.

$$\begin{bmatrix}v_{ref}\\ \omega_{ref}\end{bmatrix} = \begin{bmatrix} cos\theta & sen\theta\\ -\frac{1}{a}sen\theta & \frac{1}{a}cos\theta\end{bmatrix}.\begin{bmatrix}\dot{x}d + lx.\tanh\left(\frac{kx}{lx}.\tilde{x}\right)\\ \dot{y}d + ly.\tanh\left(\frac{ky}{ly}.\tilde{y}\right)\end{bmatrix} \tag{20}$$

where as input of this controller we have the current orientation angle $\theta$ obtained by the robot or by a dedicated sensor or by the integration of the angular speed obtained by the robot, $\dot{x}d$, $\dot{y}d$ are the speeds on the *x* and *y* axes, being *lx* and *ly* Є R saturation constants, $ky$ and $ky$ the controller gains e $\tilde{x}$ and e $\tilde{y}$ position errors. The terms with the hyperbolic tangent are included to limit the value of the desired speeds and avoid saturation of the robot's actuators in case of large position errors. Speed errors are considered to be appropriately limited (MARTINS, 2009). Also according to (MARTINS, 2009), an increase in *kx* and *ky* gains influences an improvement in trajectory tracking errors, position errors, but there is no change in trajectory errors, robot speed errors. In (TOMMASI, 2015) there is a description of an empirical method for tuning the parameters for this trajectory controller, called the fixed gains controller.

## 2.7 SCALAR KALMAN FILTER

The Kalman Filter (OLIVEIRA et al., 2017) is a widely used method when it is desired to mitigate the effect of measurement noise via software. This method aims to "use measurements of quantities carried out over time (contaminated with noise and other uncertainties) and generate results that tend to approximate the real values of the measured quantities" (WELCH; BISHOP, 2006). The Kalman Filter is a predictive recursive filter for noisy measurements.

In addition to the Kalman filter used in linear dynamic systems, the Extended Kalman filter used for non-linear systems, the Kalman filter in its scalar form can be applied, a convenient way to improve the quality of data acquisition from a sensor that presents readings contaminated by noise. Next is the scalar form of the Kalman filter algorithm.

A priori stage:

$$\hat{x}_k^- = \hat{x}_{k-1}^- \tag{21}$$

$$\sigma_{Pk+1}^2 = \sigma_{Pk}^2 + \sigma_Q^2 \tag{22}$$

A posteriori stage:

$$K_k = \frac{\sigma_{Pk}^2}{\sigma_{Pk}^2 + \sigma_R^2} \tag{23}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(\mathrm{z_k} - \hat{x}_k^-) \tag{24}$$

$$\sigma_{Pk+1}^2 = (1 - K_k)\sigma_{Pk}^2 \tag{25}$$

The signal is considered to have no dynamic characteristics. Therefore, we will only have the measurement signal from Equation (21). In Equation (22) we have the *a priori* update of the Kalman filter, where $\sigma_{Pk}^2$ is the initial variance of the output (which is related to the learning of the filter), defining how fast the filter will converge to the nearest value of the true value, $\sigma_Q^2$ is the variance of the model (related to the quality of the model). In Equation (23) the step *a posteriori* begins, calculating the Kalman gain for the scalar case, where $\sigma_R^2$ is the variance of the sensor (related to its quality). In Equation (24), *a posteriori* estimation of the measured value of the sensor occurs based on the previous estimate, on the Kalman gain and on the difference between the value read $\mathrm{z_k}$ and the previously estimated value $\hat{x}_k^-$. Finally, in Equation (25) there is an update of the initial variance of the output a posteriori.

## 2.8 ROBOTIC SYSTEM PLANT

The following is a block diagram of the plant in Figure 5.

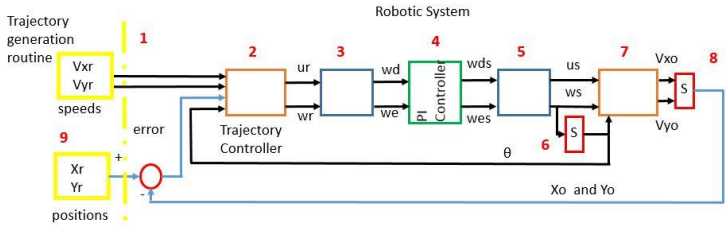


Figure 5: Block diagram of the representation of the embedded robotic plant.

In Figure 5, in item **1** there are the reference speed inputs generated by the trajectory generation routine, in **2** there is the trajectory controller described in Equation (20), in **3** the linear and angular speeds are transformed for the wheel speeds of the robot seen in Equation (18), but multiplied on both sides by the inverse of the transformation matrix $T_w^u$, already in **4** are the robot motors with PI controllers, in **5** we have the Equation again (18) transforming the measured speeds of the wheels into linear and angular speeds of the robot, in **6** there is the integration of the angular speed, obtaining the current angular position used in trajectory controller and in item **7** of the figure, where the linear and angular speeds are converted to speeds of the X, Y plane. In **8** there is the integration of the velocities obtained in the Cartesian plane and thus the positions of the robot in the Cartesian plane are obtained during the path. Finally, in **9** there are the entries of the positions generated by the reference trajectory and the calculation of the error of the positions occurred during the courses, position errors that are used in the feedback of the trajectory controller of Equation (20).

## 3. SYSTEM CONSTRUCTION AND TESTS

### 3.1 ODOMETRY

According to the data sheet of the SS41 sensor, was possible to analyze if the chosen sensor was suitable for the robotic system under development, having the following operating characteristics: rise time and fall time from 10% to 90% with duration of 1, 5 microseconds, resulting in a total period of 3 microseconds for the passage (input and output) of a magnetic pole of the sensor.

Knowing that the maximum speed of the motor without load is 19500 RPM or 325 turns / s (Hz), which results in a minimum period of oscillation of the motor and multiplying 6 magnetic poles of the magnet attached to the motor, the operating frequency the motor changes to 1950 revolutions / s (Hz) resulting in a minimum oscillation period of the motor of 523 us. Therefore, according to the Nyquist Theorem, it is known that the sampling period of a signal must be at least half the value of the signal to be collected to avoid the aliasing effect. As the minimum sensor acquisition period is 3 us and the minimum motor operation period is 513 us, the sensor will not present a sampling problem.

To verify and validate the robot's speed measurement data obtained by the odometry system, tests were performed and the average speed obtained by the SS41 Hall effect sensor, in steady state, was compared with the speed measured with a tachometer from the manufacturer IT Instruments model ITTAC 7200, by measurements performed. These results are included in Table 1, where the motor coefficients are exposed, except for the application of PWM equal to 25, where the percentage error was around 23%, the other percentage errors were around 1% to 3.5%.

### 3.2 OBTAINING MOTOR COEFFICIENTS

Data acquisitions were performed as follows: with the motors in open mesh and suspended, with the wheels coupled to the axle, the motor speeds were recorded by applying spaced variations of PWM pulses of 25 (10%) up to the limit of 225 PWM (90%) with an approximate 2% spacing between the variations. The data collected in the steady state part were used for the average value of the viscous coefficient of the motor and the deceleration stretches were used to determine the average value of the coefficient of inertia of the motors, according to the run down test. The values in parentheses in Table 1 refer to the percentage error of the average speed of the sensor in relation to the speed measured by the tachometer.

Table 1: Sample of data collected from the left motor and obtaining the coefficients of viscous friction and moment of inertia and validation of the sensor speed acquisition.

| Left Motor | | | | | | |
|---|---|---|---|---|---|---|
| PWM | %PWM | Voltage Multimeter | Average sensor speed (rad/s) | Speed (rad/s) Tachometer | *B* m(N.m/rad/s) | *J* m(N.m/rad/s).s |
| 25 | 10 | 0,810 | 1,0965 (-23,24%) | 0,8897 | 3,4070 | 0,2380 |
| 75 | 30 | 4,845 | 8,8625 (1,08%) | 8,9599 | 2,5172 | 1,1050 |
| 150 | 60 | 8,430 | 15,9928 (3,40%) | 16,5583 | 2,4264 | 1,6990 |
| 225 | 90 | 10,300 | 19,2081 (3,51%) | 19,9076 | 2,4687 | 1,9000 |
| | | | | Average of values | 2,46275 | 1,54223 |

### 3.3 VALIDATION OF MOTOR COEFFICIENT OBTAINED

To validate the coefficients of inertia *J* and viscous friction *B* of both motors, we tried to reproduce the data acquisition tests in simulation of the motors with the Simulink / MATLAB software, applying the input voltages in the open loop motors, voltages values obtained during the tests to obtain the coefficients. And during the simulations with the obtained coefficients, they resulted in simulated speeds close to those obtained in tests with the robot.

In Table 2, with the exception of the percentage error for 25 PWM, which was around 39% due to the low pwm value at the input and the coeficients obtained were differents at this range of pwm, the remaining errors were in the range of 1% to 3%. The values in parentheses refer to the percentage errors related to the average speed and the simulated speed.

Table 2: Validation of the viscous friction coefficient and the moment of inertia of the left motor.

| **Left Motor** | | | | |
|---|---|---|---|---|
| PWM | **%PWM** | **Measured voltage (V)** | **Avegare speed (rad/s)** | **Simulated speed (rad/s)** |
| 25 | 10 | 0,81 | 1,0960 | 1,525 (39,14%) |
| 75 | 30 | 4,84 | 8,8625 | 9,116 (-2,86%) |
| 100 | 40 | 6,45 | 12,1040 | 12,140 (-2,97%) |
| 200 | 80 | 9,80 | 18,2140 | 18,450 (-1,29%) |

### 3.4 PARAMETER SETTINGS FOR THE KALMAN SCALAR FILTER

During the collection of electric current from the motors, it was noted that the collected signals were very noisy, due to the electromagnetic interference generated by the PWM of the H bridges. Both for the electric current signals and the wheel speed signals, we opted for use the static Kalman filter, considering the system without dynamic characteristics. After the design of the filters, they were tested with the collected data and after determining the parameters, the filters were implemented in the robot.

In the tests, the parameters $\sigma_R^2$ were defined, which is related to the precision of the sensor, which is obtained by the variance of the measurements after the system enters a steady state, adopting the highest value of variance measured in the tests. The parameter $\sigma_Q^2$ is related to the precision of the model and influences the quality of the model. Through empirical adjustments, it was realized that 0.05 would be a good value for electrical current signals. The parameter $\sigma_{Pk}^2$ is related to the initial covariance of the filter. As $\sigma_{Pk}^2$ influences the learning of the filter by simulations with the data of measured electric currents, it was noticed that for value of P = 100, the filter showed a fast learning. The same simulation procedures previously described were used to filter the signals obtained by the speed measurement sensors.

Next there is the Figure 6, where the value measured by the current sensor is in blue and in red there is the value of the signal with the application of the Kalman Escalar filter. Still in this same figure, we tried to compare it with a 1st order filter with 0.0674s time constant in yellow, but due to the oscillatory behavior of the electric current, the 1st order filter did not show a good result. The value measured in orange is by the clamp meter coincided with the average value of the current in steady state. 225 PWM were applied to the open loop motor. The Minipa ET-3320A clamp meter, which measures AC and DC currents, was used in tests to measure the electrical current.

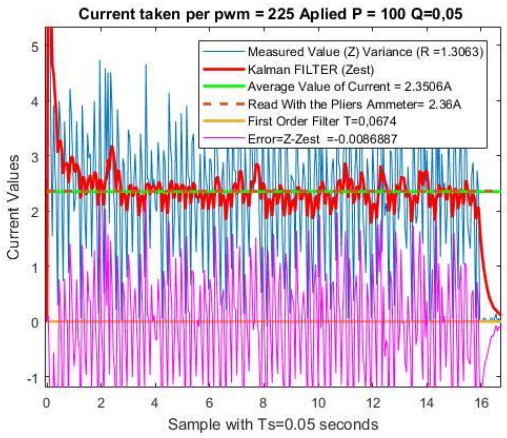


Figure 6: Application of Static Kalman Filter in signal of electric current generated with 225 PWM

Figure 7 shows the same comparisons made with the electric current signal, but for the speed signal measured by the robot's motor. Unlike what happened with the electric current, the 1st order filter, presented a good result but with a small amplitude offset. The average speed value showed a small error in relation to the value obtained by the tachometer.

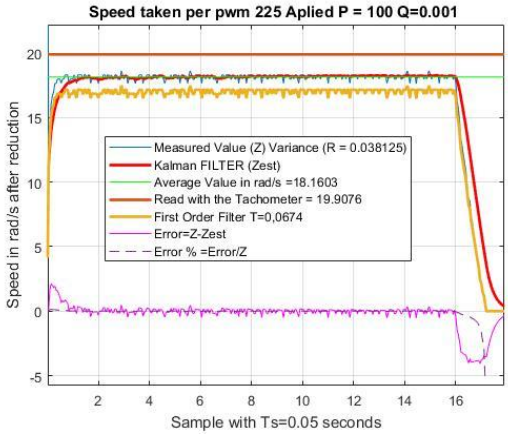


Figure 7: Application of Static Kalman Filter in a speed signal measured with the insertion of 225 PWM.

### 3.5 ROBOT MONITORING TESTS

With a reflective object fixed in the robot at the level of the scanning of the fixed laser sensor, it was possible to carry out the monitoring tests of the robot in an offline process. To acquire data from the LIDAR sensor, the polar coordinates data, raw scan acquisition mode, were converted to Cartesian coordinates and afterwards a filter was used to eliminate the edges of the environment, leaving only the reflective object in question and finally the average points were obtained in each scan and with them it was possible to monitor the displacement of positions and with the average time of the acquisition process in a set of scans it was possible to calculate the speeds performed by the robot.

For the monitoring, a manual measurement method was also used, with measuring tape and time measurement and with the robot data, both with the use of a trajectory controller and without, for the monitoring of the robot positions. In the monitoring without the trajectory controller, an offline calculation was realized, considering the linear and angular velocities measured by the robot, transforming them into velocities in the Cartesian plane and after performing an integration to obtain the positions. Making the collection with the trajectory controller, it was possible in the same test to obtain data as if this controller was not being used.

Figure 8 shows the comparison of the position monitoring related to Test 1 contained in Table 3 in red line was the measure without trajectory controller in blue with this controller and in yellow was the made by LIDAR sensor.

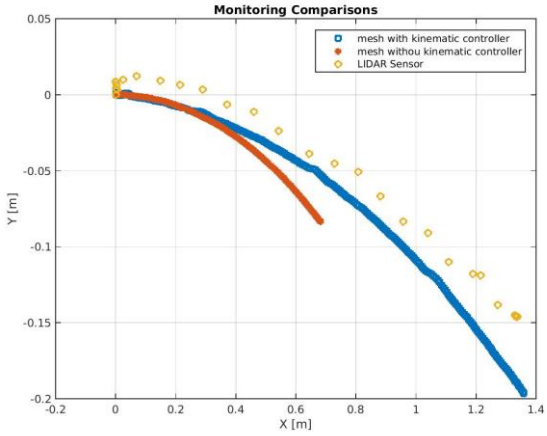


Figure 8: Compares the monitoring of the position traveled by the robot.

In Figure 9 there is a comparison between the speed monitored related to Test 1 contained in Table 3 in red line was the robot's measurement and in blue was of the LIDAR.

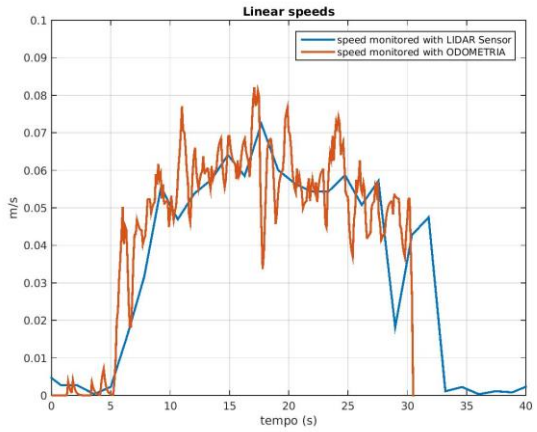


Figure 9: Comparison of monitored speeds.

Table 3 shows the results of the tests carried out in linear displacement, comparing the displacements monitored by the robot with the manual verification, with LIDAR sensor, with and without the use of the trajectory controller and the comparison of the velocities calculated by the manual verification by LIDAR and those obtained directly by the robot's control loops (mean values of speeds in stead state). In parentheses are the values of the percentage errors in relation to the values obtained by manual verification.

Table 3: Robot displacement tests and comparisons of positions and speeds

| **TEST 1** | **Manual verification** | **Without trajectory controller** | **With trajectory controller** | **LIDAR Sensor** |
|---|---|---|---|---|
| Displacement in X | 1,3500 m | 0,6817 m (49,50%) | 1,3580 m (-0,59%) | 1,3380 m (0,90%) |
| Displacement in Y | -0,0500 m | -0,0833m (-66,60%) | -0,1963 m (-292%) | 0,1500 m (400%) |
| Linear Speed | 0,0675 m/s | ---------- | 0,0569 m/s (16%) | 0,0554 m/s (18%) |
| **TEST 2** | **Manual verification** | **Without trajectory controller** | **With trajectory controller** | **LIDAR Sensor** |
| Displacement in X | 2,2600 m | 1,1270 m (50,13 %) | 2,2430 m (0,75%) | 2,23 m (1,32%) |
| Displacement in Y | 0 | 0,0685 m | 0,1376 m | 0,04m |
| Linear Speed | 0,1883 m/s | ---------- | 0,1379 m/s (26,76%) | 0,19 m/s (-0,91%) |
| **TEST 3** | **Manual verification** | **Without trajectory controller** | **With trajectory controller** | **LIDAR Sensor** |
| Displacement in X | 2,9600 m | 1,4710 m (50,30%) | 2,9140 m (1,55 %) | 2,9100 m (1,70%) |
| Displacement in Y | 0 m | 0,0039 m | 0,2456 m | 0,1110 m |
| Linear Speed | 0,2960 m/s | -------- | 0,2934 m (0,89 %) | 0,2861 m (3,34%) |
| **TEST 4** | **Manual verification** | **Without trajectory controller** | **With trajectory controller** | **LIDAR Sensor** |
| Displacement in X | 1.4 m | 0,5561 m (60,27 %) | 1,035 m (26%) | 1,3835 m (1,18%) |
| Displacement in Y | 0,4 m | 0,09082 m (77,29%) | 0,3827 m (4,3%) | 0,4300 m (-7,50%) |
| Linear Speed | 0.3333 m/s | ---- | 0,3914 m/s (-17,43%) | 0,2400 m/s (28%) |

In all tests, can be observed that the monitoring of robot positions without the kinematic controller did not generate good results with high percentage errors, as shown in Table 3, because when trying to monitor the path in this way, there was no feedback on position errors, these errors are not compensated, unlike when using the external loop with the trajectory controller. In all position monitoring, when there were lateral deviations in Y, errors occurred.

Comparing the displacement monitoring in X, both using the LIDAR sensor and using the data with the use of the trajectory controller, the percentage errors were below 1%, as shown in Table 3, except in Test 4, in that odometry showed a 28% error. In test 4 the speed error was due to the application of a high speed in a short space, making the robot not reach the steady state, always working in a transient regime and showing a high lateral deviation at the moment of departure, but even in this experiment, monitoring the displacement in X with the LIDAR maintained an error of around 1%, with satisfactory performance.

Finally, the seed monitoring was exposed with higher percentage errors and with smaller errors alternating between the monitoring with the speed data obtained by the robot and the monitoring data obtained with the LIDAR sensor, with a 0,89% error variation to 28%, but even so, based on the short distances in which the tests were carried out, the errors can be considered acceptable.

## 4. CONCLUSIONS

In this article, the stages of development of a system for autonomous mobile robot were presented, with a greater emphasis related to the robot instrumentation, due to the needs of the project, with the description of the components used in the robot, with the implementation of the controllers being carried out, with the determination of motors coefficients and their validation.

Afterwards, the models of the motors were built, opting to use the Direct Synthesis method to tune the PI controllers, due to the robot being a quick response plant. In parallel, in addition to the speeds, the values of the electric currents were also collected and the empirical method of determining parameters for the Static Kalman filter was shown, in addition to performing the validation of the odometry system by comparison with measuring instruments. The general model of the plant and the reference velocity trajectory controller, also called the fixed gains controller, were shown.

Finally were shown experiments with linear displacement, where comparisons were made, taking manual measurements as a reference and the executed speeds were compared, monitored by the robot and calculated by the LIDAR sensor, showing deviations below 30%. Position monitoring was compared by maintaining manual measurements as reference and comparing with LIDAR and using the trajectory controller on the X axis with percentage errors around 1%. The lateral deviations were not monitored correctly, the monitoring indirectly without considering the trajectory controller showed significant errors, because the form of monitoring does not have the correction of the position errors that occurred during the path.

## 5. REFERENCES

ALBRECHT, B. L. IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE E APRIMORAMENTO DO SISTEMA DE MOVIMENTAÇÃO DE UM VEICULO AUTÔNOMO TERRESTRE; Monografia apresentada ao Programa de Educação Continuada da Escola Politécnica da USP para obtenção do título de MBA em Automação Industrial, 2013

ANDALUZ, VICTOR H.; CANESCO, PAUL; VARELA, JOSÉ, ORTIZ, JESSICA S.; PÉREZ, MARIA G.; ROBERTI, FLAVIO; CARELLI, RICARDO.: Chapter, Robust Control with Dynamic Compensation for Human-wheelchair System. in Intelligent Robotics and Applications, 2014, Volume 8917, editora Springer. ISBN: 978-3-319-13965-4

ALEJANDRO, Omar. Automatização da logística interna da indústria, site itforum, Setembro/2019, Disponível em <https://itforum.com.br/colunas/robo-autonomo-movel-e-a-automatizacao-da-logistica-interna-da-industiria/>Acesso:em 06/04/2021.

BEZERRA, Matheus C. D. S, Métodos de estimação dos parâmetros mecânicos de geradores síncronos, Monografia apresentada ao Curso de Engenharia Elétrica com ênfase em Sistemas de Energia e Automação, da Escola de Engenharia de São Carlos da Universidade de São Paulo, Projeto de Formatura em Engenharia Elétrica, 2016.

CORREIA, António M. A. A. de A. ; Robô Móvel para Vigilância, Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2019.acesso em: 29/04/2021 <https://repositorio-aberto.up.pt/bitstream/10216/121385/2/343921.pdf>

CRUZ, J. J. PTC 3441 Modelagem e Controle de Manipuladores Robóticos; Escola Politécnica da USP, Departamento de Engenharia de Telecomunicações e Controle; Laboratório de Automação e Controle. 2018.

DONOVAN, Alexander. 9 Robots That Are Invading The Agiculture Industry, site interestingengineering, Novembro/2018 Disponível em <https://interestingengineering.com/9-robots-that-are-invading-the-agriculture-industry> Acesso em :06/04/2021.

GARCIA, C. Controle de Processos Industriais – Volume 1 – Estratégias Convencionais. Editora Edgard Blücher, 2017.

MARTIN, Antonio. Robôs móveis para vigilância e monitoramento de pessoas que necessitam de assitência. site dicyt, Fevereiro 2011, Disponível em <https://www.dicyt.com/noticia/robos-moveis-para-vigilancia-e-monitoramento-de-pessoas-que-necessitam-de-assistencia> Acesso em :06/04/2021.

MARTINS, F. N. Modelagem e compensação da dinâmica de robôs móveis e sua aplicação em controle de formação. Tese de Doutorado, Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo – 2009. Repositório UFES 1975-M386m.

MARTINS, F. N.; BRANDÃO, A. S. Mobile Robots – volume1 - Motion Control and Velocity-Based Dynamic Compensation for Mobile Robots, InTechOpen, 2018. Disponível em: https://www.intechopen.com/books/applications-of-mobile-robots/motion-control-and-velocity-based-dynamic-compensation-for-mobile-robots

OGATA, Katsuhiko; Engenharia de Controle Moderno. 5ª ed. São Paulo: Pearson Prentice Hall, 2010.

OLIVEIRA, W. S.; GONÇALVES, E. N. Implementação em C: filtro de Kalman, fusão de sensores para determinação de ângulos. ForScience: revista científica do IFMG, Formiga, v. 5, n. 3, e00287, jul./dez. 2017.

The Sick LIDAR MATLAB/C++ Toolbox. Disponível em < http://sicktoolbox.sourceforge.net/> acesso em: 29/04/2021.

TOMMASI, Eduardo E. V.; DE FARIA, Heitor G.; CUADROS, Marco Antônio de S. L.; ALMEIDA, Gustavo M.; RESENDE, Cassius Z.; GAMARRA, Daniel F. T.: ESTUDO COMPARATIVO DE CONTROLADORES DE SEGUIMENTO DE TRAJETÓRIA PARA ROBÔS DE TRAÇÃO DIFERENCIAL: GANHOS FIXOS E BACKSTEPPING. In: XII Simpósio Brasileiro de Automação Inteligente (SBAI), Natal – RN, 25 a 28 de Outubro de 2015.

VIEIRA, F. C. Controle Dinâmico de Robôs Móveis com Acionamento Diferencial, Universidade Federal do Rio Grande do Norte, Centro de Tecnologia; Programa de Pós Graduação em Engenharia Elétrica. Fevereiro 2005.

WELCH, G.; BISHOP, G. An Introduction to the Kalman Filter. UNC-Chapel Hill, TR 95-041, July 24, p. 1-16, 2006. Disponível em: <https://www.cs.unc.edu/~welch/media/.

ZEILMANN, A., GOMES, S., TERRES, M., SOARES, L., GOMES, L. (TEORIA E EXPERIMENTAÇÃO NO ESTUDO SOBRE ATRITOS EM ATUADORES ROBÓTICOS Adriano P. Zeilmanna, b, Samuel S. Gomes, Marco A.S. Terres, Leonardo B. Soares e Sebastião C.P. Gomes) Mecânica Computacional Vol. XXIX, págs. 2689-2708 (artigo completo) Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.) Buenos Aires, Argentina, 15-18 Noviembre 2010.