TRAJECTORY TRACKING CONTROL OF A TRACKED MOBILE ROBOT

KAIO D. T. ROCHA[1], THIAGO A. LIMA[1], MARCUS D. N. FORTE[1], MICHAEL COMBERIATE[2], FABRÍCIO G. NOGUEIRA[1], BISMARK C. TORRICO[1], WILKLEY B. CORREIA[1]

1. *Department of Electrical Engineering, Federal University of Ceará, Fortaleza, CE*
E-mails: kaiodtr@gmail.com, thiago.lima@alu.ufc.br, davi2812@dee.ufc.br, fnogueira@dee.ufc.br, wilkley@dee.ufc.br

2. *NASA Goddard Space Flight Center / Capitol Technology University, USA*
E-mail: nasamike@nasamike.com

Abstract— This paper presents a strategy for trajectory tracking control of a tracked differential drive mobile robot. The utilized controller is non-linear and has feedforward and feedback actions, with the latter using data from a SLAM (simultaneous localization and mapping) algorithm. This strategy allows for the rejection of disturbances such as skidding and slipping during tracking of a reference trajectory. The dynamic model that describes the tracks is identified by means of a system identification method and a speed controller is designed in order to satisfy a desired dynamic behavior of the tracks. The performance of the proposed controllers is tested and validated through experiments with a real mobile robot.

Keywords— Mobile Robot Control, Robot Platform, Trajectory Tracking, System Identification.

Resumo— Este artigo apresenta uma estratégia para controle de seguimento de trajetórias de um robô móvel com acionamento diferencial com esteiras. O controlador utilizado é não-linear e apresenta ações *feedforward* e *feedback*, sendo na segunda utilizada informação de realimentação proveniente de um algoritmo de SLAM (do inglês, *simultaneous localization and mapping*). Essa estratégia possibilita a rejeição de perturbações externas, tal como deslizamento e derrapagem, durante o seguimento de uma trajetória de referência. O modelo dinâmico que descreve as esteiras é identificado por meio de uma técnica de identificação de sistemas e um controlador de velocidade é projetado para satisfazer o comportamento dinâmico desejado para as esteiras. O desempenho dos controladores propostos é avaliado através de testes experimentais em um robô móvel real.

Palavras-chave— Controle de Robôs Móveis, Plataformas Robóticas, Seguimento de Trajetória, Identificação de Sistemas.

1 Introduction

Control of mobile robots on a reference trajectory has been the object of study in some previous works (Klančar, Matko and Blažič, 2005). In order to overcome the challenge of localizing the robot in real time, some different sensorial techniques have been used, such as encoder odometry, computer vision, and sensor fusion.

In Chung, Hou and Chen (2015), magnetometer and gyroscope data are fused using a Kalman filter with fuzzy compensation to estimate a mobile robot localization. Wang, Sun and Zhao (2015) used camera images to identify an outdoor road in order for the robot to execute a reference tracking algorithm. More specifically, on the tracked mobile robot case, Low (2014) used GPS information to locate the robot in an outdoor environment and fed the robot pose back to the trajectory tracking control loop. However, note that the GPS-based strategy is not suitable for indoor environments and has limited precision. An alternative is to use modern LIDAR (light detection and ranging) sensors, which allows for the operation in both outdoor and indoor environments with precision in the centimeters range and high scan rates. This type of sensor is commonly used in SLAM (simultaneous localization and mapping) algorithms, mostly to generate maps of an environment.

Taking advantage of the high resolution and accuracy of this kind of sensor, in this work, it is evaluated the application of a LIDAR as part of a robot trajectory tracking control system. More specifically, the robot pose is estimated by means of a LIDAR-based SLAM algorithm and used as feedback to the controller. This strategy allows for the rejection of skidding and slipping disturbances, which are naturally present in real mobile robots. Lima et al. (2016) applied this strategy in a wheeled mobile robot and, in this work, it is assessed in a tracked mobile robot.

The implemented nonlinear controller is based on Klančar, Matko and Blažič (2005), which is composed of feedforward and feedback actions. Often, this control system uses odometry (estimated from encoders and kinematic model) information as feedback to the nonlinear controller.

The trajectory tracking controller generates the speed reference for the tracks speed controller that runs at a higher rate in an inner loop. Two digital RST controllers were used to control the speed of the two motor-wheel-track systems. The controller tuning was made through a pole placement method. For this, the LS (least squares) method was used to identify a parametric dynamical model from data sets acquired experimentally with the robot.

Tests were performed in the real robot named *Nanook*. The robot was developed at NASA–GSFC in 2007 with the purposes of acquiring point cloud images and testing communication protocols that would be later used in out of earth exploration rovers. The robot was modernized with new control hardware, de-

veloped at DEE-UFC. Also, it runs the Robot Operating System (ROS), which allows for easier implementation of common robotics tasks, such as autonomous navigation.

This work is organized as follows. In Section 2, the kinematics mathematical model and hardware components of the mobile robot are described. In Section 3, the speed controller is designed after the system identification. In Section 4, the trajectory tracking controller is designed and the experimental results are shown. Finally, Section 5 brings concluding remarks about this paper.

2 Mobile robot description

2.1 Hardware description

Nanook, shown in Figure 1, is the tracked differential drive mobile robot used to carry out the experiments. Its dimensions are 50 cm in width, 70 cm in length, and 40 cm in height. Each track is driven by three wheels, two of which are actuated and the middle one is for better support. The wheels have 17 cm in diameter. A diagram of the components and how they are connected is shown in Figure 2.

Figure 1. *Nanook* tracked mobile robot.

The power supply consists of two 12 V batteries connected in series, yielding 24 V. Since the components require different input voltages, two DC-DC converters are used, also providing short circuit, overload and surge protection to the system.

Four Pittman GM9236S027 geared DC motors drive the tracks. A Sabertooth 2x25 motor driver is the interface between the microcontroller unit and the motors, generating the PWM signals to set the desired speeds. The driver provides two power outputs, one for each pair of motors attached to the tracks, giving the robot the differential drive characteristic.

Two Pittman E30B quadrature incremental optical encoders attached to one of the motors on each side of the robot provide feedback to a speed control loop.

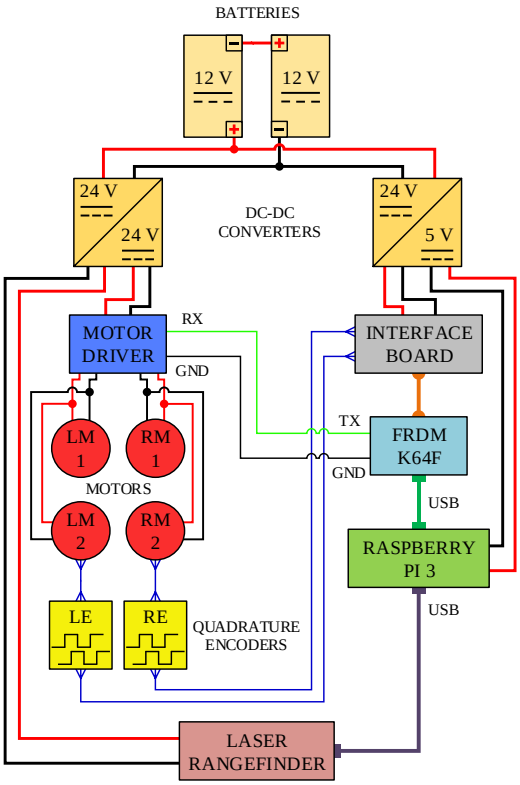

Figure 2. *Nanook* hardware diagram.

The microcontroller unit is an ARM Cortex-M4 based on the FRDM-K64F development board by NXP. It is responsible for the speed control of the tracks, reading the encoders signals and sending commands to the motor driver via serial communication.

The robot also features a Sick LMS221 LIDAR sensor, used for pose estimation, necessary for the trajectory tracking control.

Finally, a Raspberry Pi 3 single-board computer is responsible for executing the trajectory tracking control algorithm, receiving data from the LIDAR sensor and sending tangential and angular velocity references to the microcontroller unit. The Raspberry Pi 3 has the ROS framework installed, taking advantage of the large number of existing packages that perform common robotics tasks, such as SLAM, developed in Kohlbrecher et al. (2011), which in this particular application is used to obtain the pose estimates from the LIDAR sensor scans.

2.2 Mathematical modelling

The kinematic model of the tracked robot is obtained in this section, with Figure 3 showing a representation of the robot and its symbols. The tracks are independently driven, therefore the robot's kinematics can be described similarly with that of a two wheeled nonholonomic mobile robot.

It is assumed that the robot's geometric and gravity centers are coincident, and L is the distance between the two tracks. The position of the robot (x, y) is the midpoint C between the two tracks, and θ is its orientation angle in the inertial frame (X_I, Y_I).

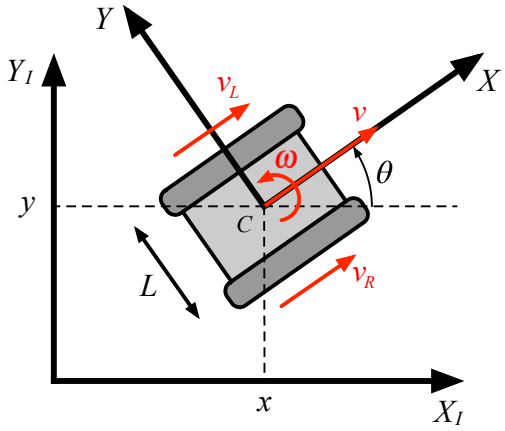

Figure 3. Robot architecture and symbols.

The robot's tangential and angular velocity in terms of the velocities of the left and right tracks are given by equations (1) and (2), respectively.

$$v = \frac{v_R + v_L}{2} \tag{1}$$

$$\omega = \frac{v_R - v_L}{L} \tag{2}$$

The kinematic model for the robot is given by equation (3).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ sen\ \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

Given a reference trajectory $\boldsymbol{p_r}(t) = [x_r(t), y_r(t), \theta_r(t)]^T$, defined for time $t \in [0, T]$, and provided with the robot inverse kinematics, it is possible to determine a feedforward control law. However, the calculated robot inputs can only drive the robot through a desired trajectory when there are no disturbances or initial errors in the system.

The necessary feedforward tangential (v_r) and angular (ω_r) velocities to drive the robot are obtained from the reference trajectory and are given by equations (4) and (5), respectively.

$$v_r(t) = \pm\sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{4}$$

$$\omega_r(t) = \frac{\dot{x}_r(t)\,\ddot{y}_r(t) - \dot{y}_r(t)\,\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{5}$$

3 Tracks speed control

3.1 System identification

Each of *Nanook*'s tracks couples two motors that are responsible for the robot's movement. This coupling yields a system of high order due to electromechanical oscillations between the pair of motors. In order to capture the dynamical behavior and obtain the model of each track, the LS identification method was used (Landau and Zito, 2006).

For this purpose, two datasets $D_N = \{u(k), y(k)\}_{k=1}^{N}$ of N measurements were collected for each track, where $u(t)$ is a pseudorandom binary sequence (PRBS) designed to persistently excite the plant to all of the dynamical behavior and $y(t)$ is the angular speed of the track. The first dataset is used to identify the system using the LS method, while the second one is used to provide a non-biased data fitting analysis. The sampling time for the experiments is $T_s = 10$ ms.

Figure 2 shows the command and data acquisition scheme used to obtain the datasets D_N. The Sabertooth 2x25 driver output is a PWM signal with mean voltage varying from -24 V to 24 V.

The command signal is sent via serial communication between the FRDM-K64F microcontroller unit and the motor driver. The signal was conditioned to be in the continuous range between -1 and 1, with -1 meaning the minimum output voltage of -24 V and 1 meaning maximum output voltage, which is 24 V. The motor encoder's quadrature signals are read by the microcontroller, converted to speed readings in RPM and stored to be later applied in the LS method.

The PRBS signals were conditioned to be -0.5 and 0.5 during the tests, meaning that the motors were driven by voltages of -12 V and 12 V. All four experiments lasted 120 seconds, so each dataset D_N has $N = 12000$ measurements. The autoregressive model with exogenous input (ARX model) shown in (6) is chosen to represent the system.

$$\begin{aligned} A(q^{-1})y(t) &= q^{-d}B(q^{-1})u(t) + e(t) \\ A(q^{-1}) &= 1 + a_1(q^{-1}) + \cdots + a_{n_A}(q^{-n_A}) \\ B(q^{-1}) &= 1 + b_1(q^{-1}) + \cdots + b_{n_B}(q^{-n_B}) \end{aligned} \tag{6}$$

Where $u(t)$ is the command, or input signal, $y(t)$ is the output and $e(t)$ is the modelling error. The orders of the polynomials are chosen as $n_A = n_B = 2$.

After applying the LS algorithm using the identification datasets D_N, the polynomials $A(q^{-1})$ and $B(q^{-1})$ are obtained as shown in Table 1.

Table 1. Identified models for the tracks.

Left	$\boldsymbol{A(q^{-1})}$	$1 - 0.3666\,q^{-1} + 0.0064\,q^{-2}$
	$\boldsymbol{B(q^{-1})}$	$-8.3695\,q^{-1} - 8.6422\,q^{-2}$
Right	$\boldsymbol{A(q^{-1})}$	$1 - 0.3327\,q^{-1} + 0.01\,q^{-2}$
	$\boldsymbol{B(q^{-1})}$	$9.8494\,q^{-1} - 9.3951\,q^{-2}$

The achieved fitting between the simulated model and the validation data is 97.08 % for the left track, and 94.48 % for the right track.

3.2 Pole placement method

After the models for the tracks were identified, RST digital speed controllers were designed for each of the tracks using pole placement method. This method yields some advantages when designing controllers, such as the ability to work with both stable and unstable systems and without restriction upon the degrees of $A(q^{-1})$ and $B(q^{-1})$. The control scheme shown in Figure 4 is used to obtain the equations for the controller design.

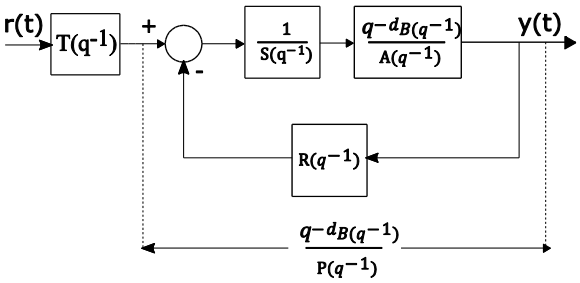

Figure 4. RST speed controller block diagram.

The closed loop transfer function in the backward shift operator q^{-1} is given by equation (7), where dependency on q^{-1} is omitted for simplicity. The desired dynamical behavior of the system is defined in time domain specifications, specifically the rise time and the maximum overshoot of a second order system. Then, the continuous time transfer function that presents the desired behavior for the plant is obtained and discretized in order to gather the polynomial $P(q^{-1})$ in equation (8), which defines the desired closed loop poles. $P(q^{-1})$ may include additional non-dominant poles in order to find a unique solution for polynomials $S(q^{-1})$ and $R(q^{-1})$, where $S(q^{-1})$ includes an integrator in order to achieve zero steady-state error for reference tracking and $T(q^{-1})$ is equal to the sum of the coefficients of $R(q^{-1})$ so that no additional zeros are introduced in the closed loop transfer function.

$$H_C(q^{-1}) = \frac{q^{-d}TB}{AS + q^{-d}BR} \tag{7}$$

$$P = AS + q^{-d}B = 1 + \sum_{i=1}^{n} p_i(q^{-i}) \tag{8}$$

After specifying a rising time of 100 ms and 0 % overshoot for both tracks, the controller parameters were found, as shown in Table 2.

Table 2. Controller parameters for each track.

Left	$\boldsymbol{R(q^{-1})}$	$0.0013 - 0.0056q^{-1} + 0.0001q^{-2}$
	$\boldsymbol{S(q^{-1})}$	$1.0000 - 1.1088q^{-1} + 0.1088q^{-2}$
	$\boldsymbol{T(q^{-1})}$	-0.0042
Right	$\boldsymbol{R(q^{-1})}$	$-0.0022 - 0.0061q^{-1} + 0.0001q^{-2}$
	$\boldsymbol{S(q^{-1})}$	$1.0000 - 1.1328q^{-1} + 0.1328q^{-2}$
	$\boldsymbol{T(q^{-1})}$	0.0038

Real tests were executed in order to assess the controller performance. The test for each track lasted 5 seconds. A constant speed reference of 5 RPM was applied initially to the controller, with addition of a constant perturbation via software in the control signal between 250 and 300 ms. Also, the reference switches from 5 to 10 RPM between the time of 1.2 and 1.7 s, and then goes back to 5 RPM.

Figure 5 shows the controller experimental results for the left and right tracks. Although some overshoot caused by the disturbance happened during the experiment, both the controllers were able to react and bring the speed back to the reference.

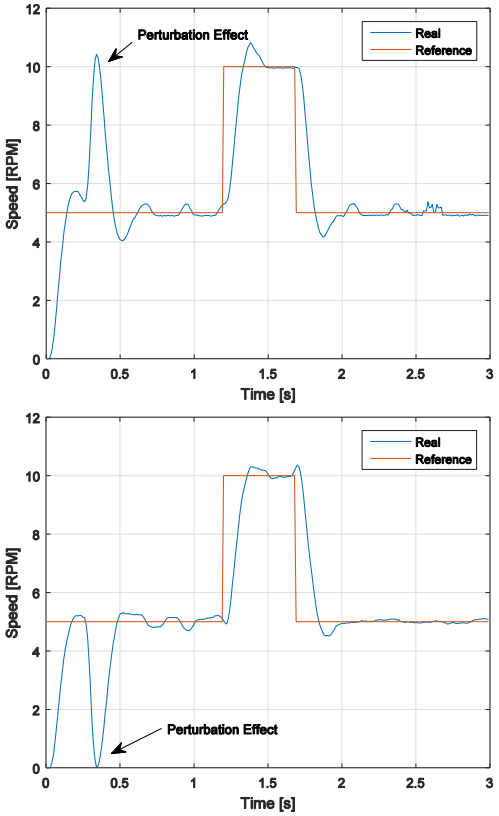

Figure 5. Experimental results. Left track on top, and right track on bottom.

4 Trajectory tracking control

4.1 Controller Design

Figure 6 shows a block diagram of the trajectory following controller. The controller is composed of a feedforward and a feedback terms. A SLAM system using the Sick LMS221 LIDAR sensor is responsible for providing the robot pose feedback, which is then compared with the reference pose. The output of the feedforward and feedback terms are combined in order to drive the robot. The output is in the form of reference tangential (u_v) and angular (u_ω) velocities, which are then converted to left and right tracks reference speeds and fed to the controller designed in Section 3.

According to Klančar, Matko and Blažič (2005), the reference trajectory is defined by a vector $\boldsymbol{p_r}(t) = [x_r(t), y_r(t), \theta_r(t)]^T$ and $\boldsymbol{p}(t) = [x(t), y(t), \theta(t)]^T$

is the current robot pose. Omitting the time t dependency, one can write the pose error in the robot reference frame as in (9).

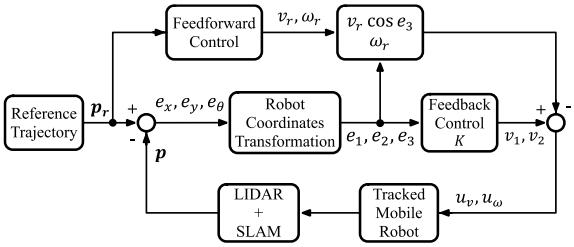

Figure 6. Trajectory tracking control scheme.

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (9)$$

Differentiating equation (9) with respect to time and using equations (3), (4) and (5), a dynamic model for the tracking error is obtained, as shown in (10).

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_v \\ u_\omega \end{bmatrix} \quad (10)$$

Where v_r and ω_r are the feedforward reference tangential and angular velocities given by equations (4) and (5). Then, the controller outputs, which are applied to the robot, are given by (11).

$$\begin{aligned} u_v &= v_r \cos e_3 - v_1 \\ u_\omega &= \omega_r - v_2 \end{aligned} \quad (11)$$

Where v_1 and v_2 are the closed loop outputs. The error model in equation (10) is then linearized and the following control law is obtained:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -sign(v_r)\, k_2 & -k_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (12)$$

The controller tuning is based on the desired damping ratio ζ and natural frequency ω_n. The closed loop response is related to these specifications. The controller gains are calculated in real time using the equations in (13). Proof and stability analysis are presented in Klančar, Matko and Blažič (2005).

$$\begin{aligned} k_1 &= k_3 = 2\,\zeta\,\omega_n(t) \\ k_2 &= g\,|v_r(t)| \\ \omega_n(t) &= \sqrt{\omega_r^2(t) + g\,v_r^2(t)} \end{aligned} \quad (13)$$

4.2 Experimental Tests and Results

Successful application of this strategy in wheeled differential drive mobile robots has been reported in Zermas (2011) and Lima et al. (2016). In this section, the performance of the controller is analyzed when applied to the tracked differential drive robot described in Section 2.

The first experimented trajectory is an eight-shaped trajectory, with reference tangential velocity v_r of 0.1 m/s and angular velocity ω_r of 0.125 rad/s. In this case, all three reference pose components $[x_r, y_r, \theta_r]^T$ change at each time instant, and the angular velocity reference ω_r changes signal when half of the trajectory is reached. The controller parameters are $\zeta = 0.8$ and $g = 30$, chosen so that the robot follows the trajectory as smoothly as possible.

In order to further test the capabilities of the controller, an initial error was included, with the robot starting the trajectory with pose $\boldsymbol{p_0} = [-0.5, -0.5, \pi/2]^T$.

Figure 7 presents the result of this first experiment. The red dashed curve represents the eight-shaped reference trajectory. The blue continuous curve shows the estimated position of the real robot, and the green triangles indicate its estimated orientation at some points. The controller is able to overcome the initial pose error and reach a point from which the reference trajectory is smoothly followed by the robot. Figure 8 shows the pose error evolution along the trajectory.

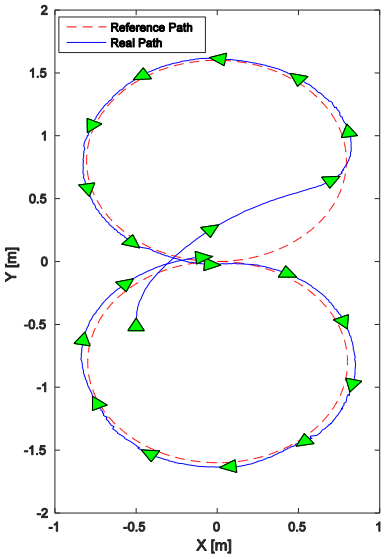

Figure 7. Result of the first experiment.

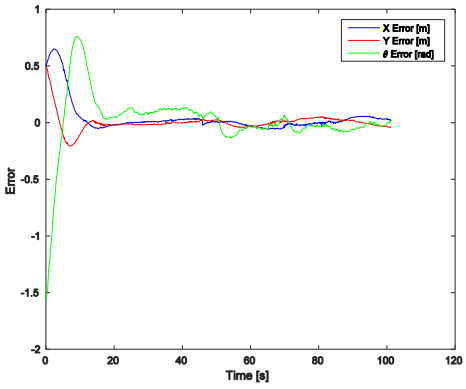

Figure 8. Pose error evolution in the first experiment.

An L-shaped trajectory was also tested, with the purpose of analyzing the behavior of the robot while

following straight trajectories and how the controller reacts to a sudden change in the reference orientation. The reference tangential velocity v_r is 0.1 m/s. The controller parameters for this trajectory are $\zeta = 0.9$ and $g = 40$. ω_n was calculated using (13).

The result of the second experiment is presented in Figure 9. In the first part of trajectory, the robot should follow a straight line in the positive X direction, maintaining the orientation equal to zero, however, the robot also moves slightly towards the positive Y direction, and the controller takes longer to correct this error, since it is prioritizing the error in X, which varies more rapidly. In the second part, there is a sudden change in the reference orientation, now $\pi/2$ rad, and the robot must move straight along the Y direction. The controller also manages to mitigate this severe change and, after approximately 2 meters, the robot is on the reference trajectory again. The pose error evolution in this trajectory is shown in Figure 10.

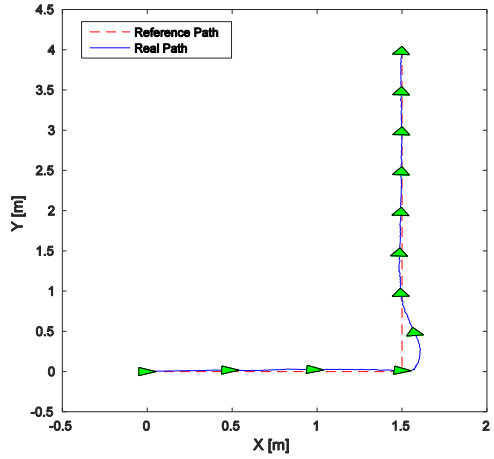

Figure 9. Result of the second experiment.

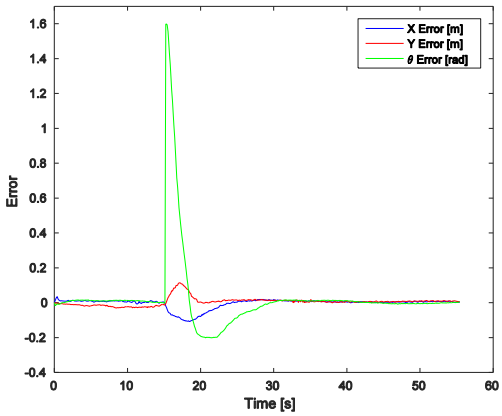

Figure 10. Pose error evolution in the second experiment.

5 Conclusion

This work presented the model identification and speed control of the tracks of a real tracked mobile robot, as well as application and tuning of a trajectory tracking controller. Using the minimum square error identification method and a digital RST controller, it was possible to successfully design a digital embedded speed controller for the mobile robot tracks. The adopted trajectory tracking control strategy was also effective, since the robot pose error was always being minimized in a relative short amount of time, despite the natural skidding and slipping disturbances.

Future work includes developing algorithms for other mobile robot applications such as area mapping, trajectory planning and obstacle avoidance.

Acknowledgements

The authors acknowledge the support from CNPQ, CAPES, and FUNCAP.

References

Chung, H. Y., Hou, C. C., and Chen, Y. S. "Indoor Intelligent Mobile Robot Localization Using Fuzzy Compensation and Kalman Filter to Fuse the Data of Gyroscope and Magnetometer," in IEEE Transactions on Industrial Electronics, vol. 62, no. 10, pp. 6436-6447, Oct. 2015.

Klančar, G., Matko, D. and Blažič, S., 2005, June. "Mobile robot control on a reference trajectory". In Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation (pp. 1343-1348). IEEE.

Kohlbrecher, S., Von Stryk, O., Meyer, J. and Klingauf, U., 2011, November. A flexible and scalable slam system with full 3d motion estimation. In Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on (pp. 155-160). IEEE.

Landau, I. D., and Zito, G., "Digital Control Systems: Design, Identification and Implementation", in Springer. London, 2006.

Lima, T. A., Forte, M. Davi do Nascimento, Nogueira, F. G., Torrico, B. C., and de Paula, A. R. "Trajectory tracking control of a mobile robot using lidar sensor for position and orientation estimation", 2016 12th IEEE International Conference on Industry Applications (INDUSCON), Curitiba, 2016, pp. 1-6.

Low, C. B., "Experimental implementation of a novel trajectory tracking control design on a full-sized nonholonomic tracked mobile robot with low-level velocities control systems," 2014 IEEE Conference on Control Applications (CCA), Juan Les Antibes, 2014, pp. 1318-1323.

Wang, J., Sun, S. and Zhao, X., "Unstructured road detection and trajectory tracking for tracked mobile robot," 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, 2015, pp. 535-539.

Zermas, D., "Control of a leader-follower mobile robotic swarm based on the NXT educational LEGO platform", 2011 IEEE International Symposium on Industrial Electronics, Gdansk, 2011, pp. 1381-1386.