OMNIDIRECTIONAL MOBILE ROBOT'S MULTIVARIABLE TRAJECTORY TRACKING CONTROL: A ROBUSTNESS ANALYSIS

T. P. Nascimento[1], A. G. S. Conceição[2], A. P. Moreira[1],

[1]*FEUP-INESC, Porto, Portugal*
tiagopn@ieee.org, amoreira@fe.up.pt

[2]*Electrical Engineering Department, Polytechnic College, UFBA, Brazil*
andre.gustavo@ufba.br

Abstract: A trajectory control for mobile robots goes through its robustness analysis as a final stage to control application in different systems. This work presents the robustness analysis for a multivariable trajectory tracking controller of the omni-directional mobile robot 5DPO for the Middle size League of Robot Soccer competition. The results were compared to the results from the robots in the Small size League that have less dynamic issues. This controller is validated with the variation of the dynamic system and the timing constant to follow different trajectories, as well as the sum of square error and the maximum absolute error.

Keywords: Mobile robotics, multivariable control, robustness analysis.

1. INTRODUCTION

An omni-directional robot is a robot able to move itself in any direction in the horizontal plane, without the pose re-orientation. This mobility is not usual in the most common mobile robots, with two wheels or four wheels, called differential mobile robots. Several studies have become frequent about omni-directional robot trajectory tracking control research due to the need of agility of these vehicles providing a comprehensive study of case in the area of mobile robotics (Watanabe, 1998) , (Kodagoda, *et al*., 2002), (Liu and Zhu, 2007), (Liu, *et al*., 2008).

A previous work in this subject presents a multivariable trajectory tracking controller applied to a Small Size league robot called AxeBot (Nascimento, *et al*., 2009). This control system considers the influence of the coupling nonlinearities when controlling a state-entrance linearized system. The main objective of this paper is to do a robustness analysis of this multivariable trajectory tracking controller by applying it to a robot with much different dynamics than the Axebot. This omni-directional mobile robot would be the 5DPO mobile robot.

The 5DPO is an omni-directional mobile robot, developed to play autonomously the soccer robots match, under, the RoboCup Federation Middle Size league rules. This paper applies the same controller directly to a nonlinear model of the 5DPO robot, which has more dynamic issues, analyzing its robustness by the response to the variations of trajectory size and velocity of tracking. This paper also presents some results for trajectory tracking from Matlab Simulink environment. The paper organization attempt to the following sequence: at section 2 the omni-directional mobile robot is presented with the respective Dynamic Modeling according a comprehensive approach using a space state format. Section 3 presents the cascade control system analyzed. At section 4 the results for trajectory tracking, under Matlab Simulink environment are shown. Finally at section 5 final comments and future works are presented.

Fig. 1. The 5DPO robot.

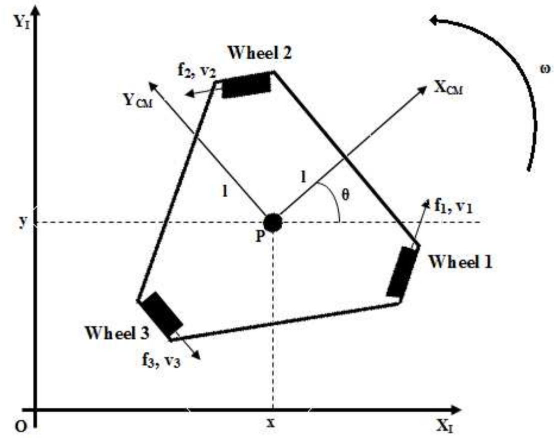

Fig. 2. The coordination system.

2. THE 5DPO ROBOT

The mobile robot 5DPO used as a study case of this paper is presented here. Fig. 1 shows a photo of the real robot equipped with three omni-directional wheels arranged at 120 degrees from each other. Three DC Maxon motors with digital speed encoder and a planetary 12:1 gearbox are used as actuators for the robot locomotion system. The electronic system of the 5DPO robot is composed by three motor drives to control the motor speed. These drives have ATMega8 microcontrollers embedded with a Proportional-Integral speed controller set by the Ziegler–Nichols method (Conceição, *et al.*, 2008).

The complete study of the kinematic and dynamic modelling of an omni-directional mobile robot can be seen in Nascimento, *et al.* (2009). Nevertheless, the final kinematic model can be seen in equation (1) through the analysis of Fig. 2. There it can be seen that the velocities of the robot with a reference to a inertial point is given by a rotation matrix times the motors angular velocities, its transformation to linear velocity (by multiplying to the wheel's radius en divided by the gear coupling) and a B matrix that translates the velocities from the wheels to the center of mass.

$$V_I = R(\theta)\left(B^T\right)^{-1}\frac{r_\omega}{\eta N}\vec{\omega}_m \tag{1}$$

with $\vec{\omega}_m = \begin{bmatrix}\omega_{m1}\\ \omega_{m2}\\ \omega_{m3}\end{bmatrix}$, $B^T = \begin{bmatrix}0 & -1 & l\\ \frac{\sqrt{3}}{2} & \frac{1}{2} & l\\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & l\end{bmatrix}$

$$R(\theta)=\begin{bmatrix}Cos(\theta) & -Sen(\theta) & 0\\ Sen(\theta) & Cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix}, V_I = \begin{bmatrix}v_{Ix}\\ v_{Iy}\\ \omega_I\end{bmatrix}$$

The dynamic model is also presented below. This model is nonlinear model in space state form where the f(x(t)) is the first two terms in the sum, g(x(t)) is the last term in the sum, $x(t)=\vec{\omega}_m(t)$, $\dot{x}(t)=\dot{\vec{\omega}}_m(t)$ and $u(t)=u$, with is both entrance of control and tension of the motors (Slotine and Li, 1991).

$$\dot{\vec{\omega}}_m = -\left(P^{-1}G^{-1}HB\left(\frac{k_t k_{em}}{R}+b_0\right)\times\frac{\eta N}{r_\omega}\vec{\omega}_m\right) - \\ -P^{-1}G^{-1}HBF_{at} + P^{-1}G^{-1}HB\frac{k_t \eta N}{r_\omega R}u \tag{2}$$

with, $G=\left(I+HBB^T\left(\frac{J_m(\eta N)^2}{r_\omega^2}\right)\right)$, $P^{-1}=\left(\frac{\eta N}{r_\omega}\right)B^T$

and $H^{-1}=\begin{bmatrix}m & 0 & 0\\ 0 & m & 0\\ 0 & 0 & J\end{bmatrix}$.

From the above equations it can be said that r_ω is the radius of the wheel, *l* is the robot's diameter, N is the coupling factor and η is the efficiency of the motor-wheel engagement, *m* is the robot's mass and *J* is the robot's moment of inertia. Also It has *u* as the motor's tension; R the resistance of the armature; k_t the constant torque of the motor, k_{em} the constant of counter-electromotor force; J_m the moment of inertia of each rotor (i.e. the sum of the moments of inertia of the motor shaft and reducing the system, respectively), b_0 the coupling damping constant.

3. THE CASCADE CONTROL SYSTEM

The control loop to be analyzed for the 5DPO robot can be seen in Fig. 3. The 5DPO nonlinear model can be seen by the first block (from right to left). One can see that the speed controller used is a PID, although as said before, it is actually a PI controller for each motor. Together with the nonlinear model, it makes the internal control loop. It is remarkable that the system has three SISO PI controllers being a nonlinear multivariable strongly coupled system, but the control law for the inner loop is not. Nevertheless, it generates an error at steady state that can be corrected by another external control loop characterizing this as a cascade controller system.

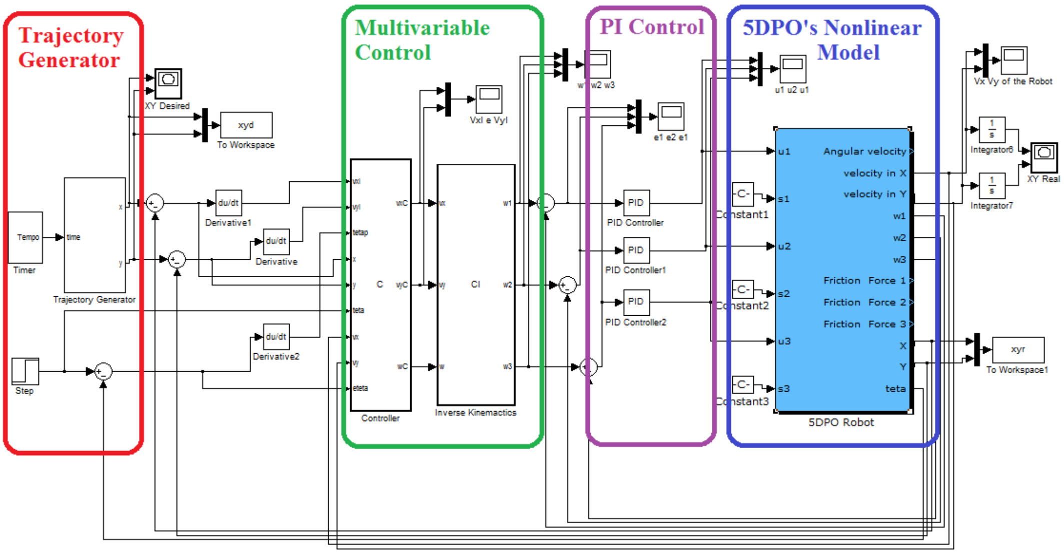

Fig. 3. The cascade control used.

This external and multivariable trajectory tracking control block can be seen after the PI controllers (from right to left), receiving data from the error between the robot's position and the Trajectory Generator's reference position and from the robot's velocities.

The cascade control allows the stationary errors generated by speed controllers to be corrected by the trajectory controller. This architecture is based on the singular perturbation theory, commonly known as the time-scale separation principle, which assumes that the internal loop is exponentially stable and the bandwidth of internal loop is much greater than the external dynamics of the mesh, so that the loop of the external controller can be designed ignoring the dynamics of the internal loop (Liu, *et al.*, 2008). The inner control loop was set by the Ziegler–Nichols method and its values of P, I and D were respectively $P = 0.245$, $I = 0.0112$ and $D = 0$.

The external loop is the multivariable control to be analyzed. This controller was implemented in the same way as in Nascimento, *et al.* (2009), through the theory based in O'Reylly (1987). The positioning system is known as the homogeneous transformation of coordinates. This equation can also be seen as:

$$V_I = \dot{P}_I = f(P_I, V_{CM}) \tag{3}$$

with $P_I = \begin{bmatrix} X_I \\ Y_I \\ \theta_I \end{bmatrix}$.

Therefore, it can be said that, linearizing V_I (robot's velocity vector with respect to inertial position) with respect to P_I (robot's position vector with respect to inertial position) and V_{CM} (robot's velocity vector with respect to the robot's center of mass) around the equilibrium point, it becomes:

$$\begin{cases} \dot{P}_I = A_1 P_I + B_1 V_{CM} \\ y = C_1 P_I \end{cases} \tag{4}$$

with $A_1 = \begin{bmatrix} 0 & 0 & -v_{CMx}.Sen(\theta) - v_{CMy}.Cos(\theta) \\ 0 & 0 & v_{CMx}.Cos(\theta) - v_{CMy}.Sen(\theta) \\ 0 & 0 & 0 \end{bmatrix}$,

$B_1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix}$

and $C_1 = I_3$.

Therefore, the multivariable trajectory tracking control law is such as the following.

$$\dot{V}_{CM}(t) = K_P.\dot{e}_P(t) + K_I.e_P(t). \tag{5}$$

with $e_P(t) = \begin{bmatrix} X_I \\ Y_I \\ \theta \end{bmatrix}$, $\dot{e}_P(t) = \begin{bmatrix} V_{Ix} \\ V_{Iy} \\ \dot{\theta} \end{bmatrix}$,

$K_P = B_1^{-1}\left(A_1 + diag\left[k_{P1} \quad k_{P2} \quad k_{P3}\right]\right)$,

$K_I = B_1^{-1} diag\left[k_{I1} \quad k_{I2} \quad k_{I3}\right]$,

To say a controller is robust, is to say it can be applied in different dynamic variations in a kind of system. To analyze the multivariable controller it was necessary to put the same controller in a similar environment (an omni-directional mobile robot) with a much hard dynamics and submit this system to a variety of external parameters configuration such as timing on trajectory tracking. Considering this analysis, it was used in the 5DPO robot controller the same parameters set in the Axebot robot. Therefore, the external controller obtained its best performance with the values of $k_P = 0.8$ and $k_I = 16$.

4. RESULTS

To analyse the multivariable controller's robustness, several trajectory tracking tests were made. The principal issue is to put a controller design to small robots, with small dynamic issues, in middle size robots with bigger problems concerning the dynamic of the system to be controlled. To resume, a bigger robot calls for a better and more robust controller to overpower the difficulty in minimizing the trajectory tracking error.

It is necessary also to have in mind that this controller is to be compared with the results obtained by Nascimento, *et al*. (2009). A comparison between the dynamic parameters of the Axebot robot and the 5DPO robot can show the growth of the dynamic problem and it can be seen in the Table 1.

The tests were made in Matlab Simulink environment. The robot was put to follow two different trajectories: an eight and a complex trajectory, both with a considered level of complexity. Both trajectories were made with three different tracking time: 5, 10 and 100 seconds. Also, two sizes of trajectories were: a small size (1x1.5 meters for the 8 trajectory and 2x1.5 meters for the complex) and a big size (3x5.5 metres for the 8 trajectory and 6x6 meters for the complex). These testes generated twelve graphics that can be seen below.

4.1 *Small size trajectories*

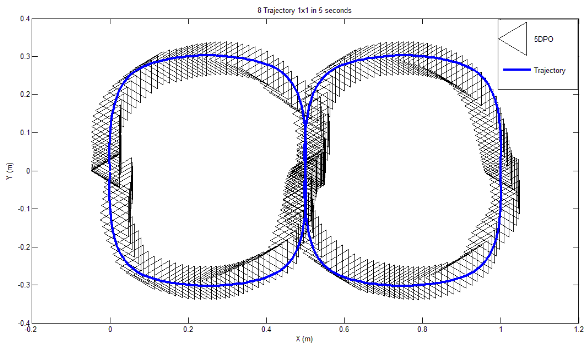

Fig. 3. The 8 trajectory in 5 seconds.

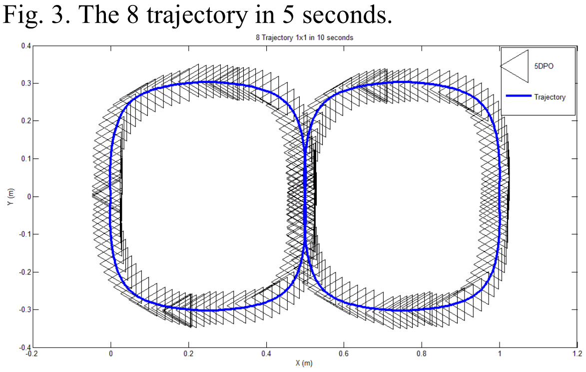

Fig. 4. The 8 trajectory in 10 seconds.

Table 1 Dynamic parameters values

Parameter	Axebot	5DPO
M (kg)	1,83	26
d (m)	0,09	0,195
r_ω (m)	0,072	0,51
k_t (Nm/A)	0,0059	0,0302
k_{em} (Nm/A)	0,0059	0,0302
N	19	12
R (Ω)	1,79	0,316
J (kgm^2)	2,125	0,705
J_ω (kgm^2)	$8{,}1x10^{-4}$	0,338
J_m (kgm^2)	$3{,}88x10^{-7}$	$1{,}39\ x10^{-5}$

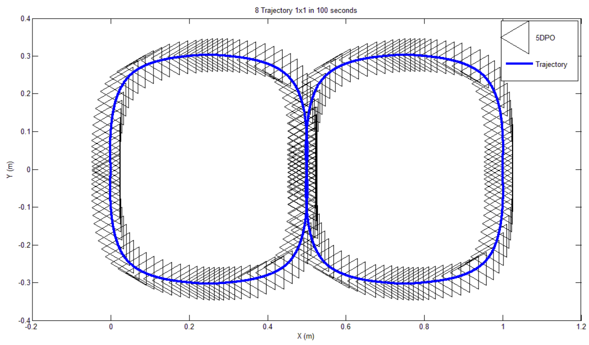

Fig. 5. The 8 trajectory in 100 seconds.

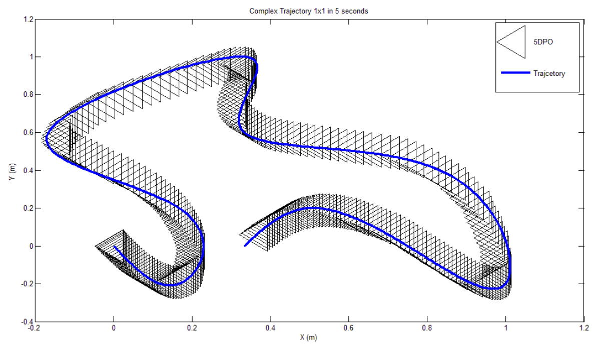

Fig. 6. Complex trajectory in 5 seconds.

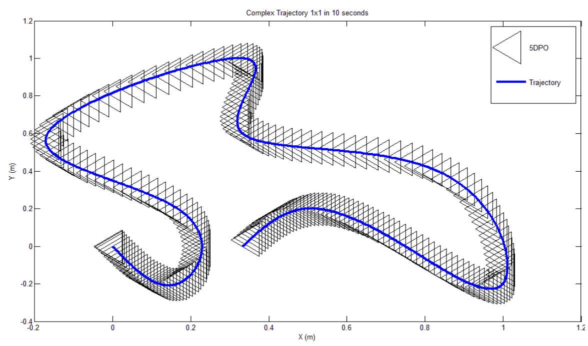

Fig. 7. Complex trajectory in 10 seconds.

It is obvious that for smaller trajectories the controller has a better performance. Also for trajectories that has less rush variations in the curvature the controller works better. Even though, the controller works with a significant good performance.

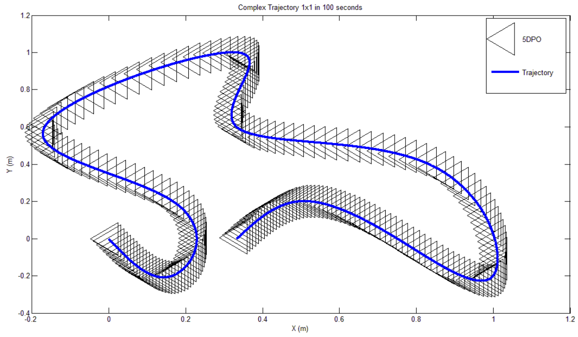

Fig. 8. Complex trajectory in 100 seconds.

4.2 Big size trajectories

Here it can be seen the performances for a bigger trajectory, both in eight format and a more complex trajectory.

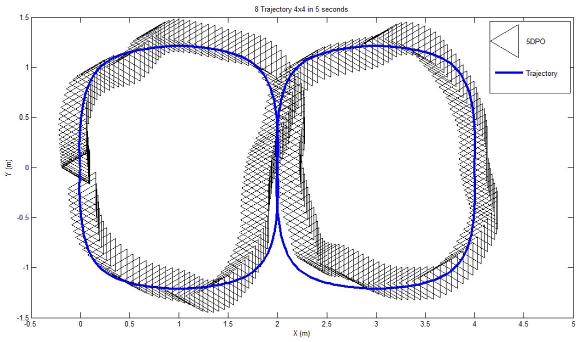

Fig. 9. The 8 trajectory in 5 seconds.

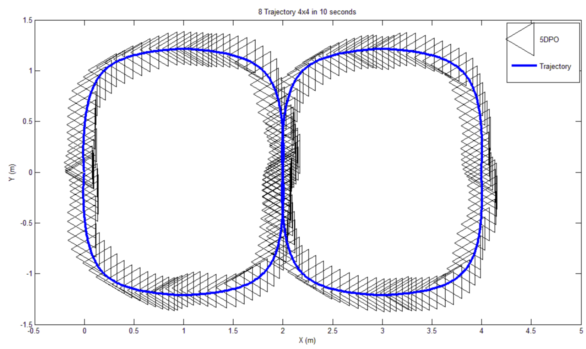

Fig. 10. The 8 trajectory in 10 seconds.

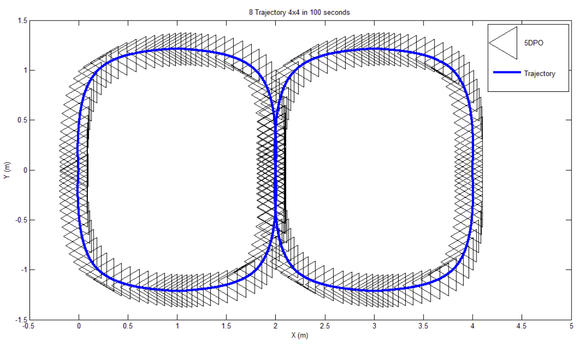

Fig. 11. The 8 trajectory in 100 seconds.

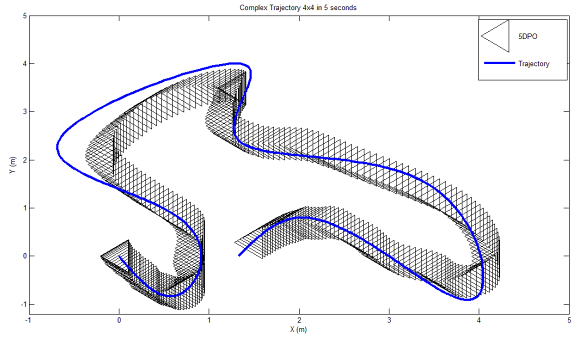

Fig. 12. Complex trajectory in 5 seconds.

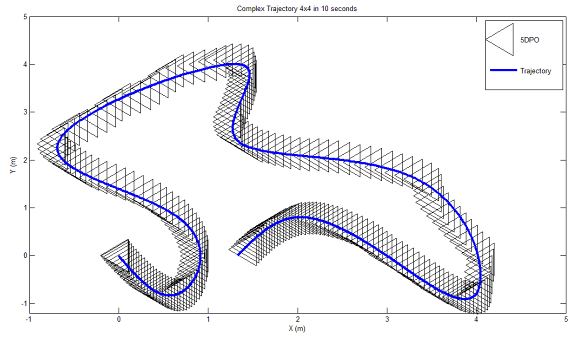

Fig. 13. Complex trajectory in 10 seconds.

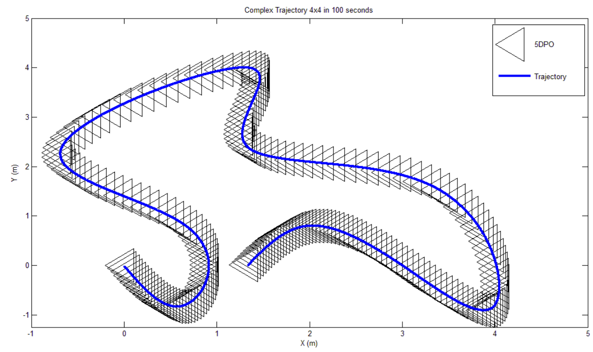

Fig. 14. Complex trajectory in 100 seconds.

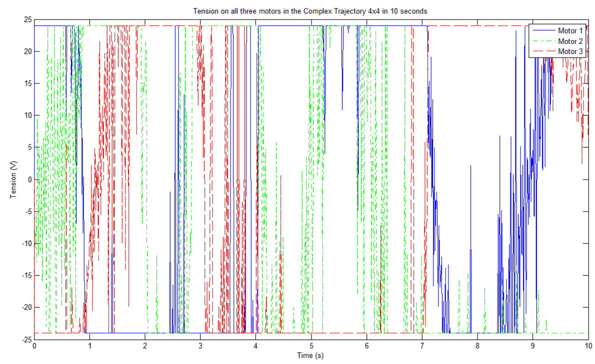

Fig. 15. Tension of all three motors in a Complex trajectory big size in 10 seconds.

Those graphics above show that the controller can handle a realistic situation in both types of trajectories. Although the tracking of a big trajectory in 5 seconds is the only unrealistic situation, the controller was successful in tracking it. It can be observed that the maximum velocity of the robot was achieved in fast situations (i.e. 5 and 10 seconds in the big trajectory size).

Another observation is about the motor's saturation in the 24 V. This tension also was achieved as it can be seen in the Fig. 15 when performing a complex trajectory in a little time as 10 seconds. Finally, for each trajectory reference, the sum of square error in X axis and Y axis in square meters (m^2) was measured. The results are shown at Table 2. The Table 3 shows the maximum absolute error in X and Y for all trajectories. Remembering that the robot has 40 cm of diameter, it can be said that, once again, except for the 5 seconds tracking (which is unrealistic) the controller had a good performance in tracking both trajectories.

Table 2 Sum of Square Error

X (m^2)	Y (m^2)	Time (s)	Trajectory	Size
1,04	1,22	5	8	Small
0,53	0,66	10	8	Small
0,03	0,06	100	8	Small
2,59	3,64	5	Complex	Small
1,24	1,85	10	Complex	Small
0,12	0,18	100	Complex	Small
51,7	67,3	5	8	Big
12,6	15,5	10	8	Big
0,74	1,04	100	8	Big
106,9	179,8	5	Complex	Big
32,9	39,4	10	Complex	Big
1,94	2,79	100	Complex	Big

Table 3 Maximum Absolute Error

E_X (m)	E_Y (m)	Time (s)	Trajectory	Size
0,076	0,101	5	8	Small
0,053	0,039	10	8	Small
0,004	0,006	100	8	Small
0,142	0,168	5	Complex	Small
0,072	0,086	10	Complex	Small
0,007	0,008	100	Complex	Small
0,673	0,822	5	8	Big
0,216	0,231	10	8	Big
0,024	0,020	100	8	Big
0,996	1,507	5	Complex	Big
0,477	0,401	10	Complex	Big
0,030	0,033	100	Complex	Big

5. FINAL COMMENTS

A multivariable approach for a trajectory tracking control system for omni-directional mobile robots and the respective results for two different trajectory references tracking were presented in this paper to analyze this multivariable controller's robustness. The efficiency in using a multivariable approach for trajectory tracking control can be seen from the presented trajectories: in eight format and the complex. Also a quantitative evaluation about the sum of square error in X axis and Y axis in square meters (m^2) is shown at Table 2, as well as the maximum absolute error in X axis and Y axis in meters (m) is shown at Table 3. It certifies the efficiency of the proposed approach for the trajectory tracking control system for omni-directional mobile robots such as the AxeBot (Nascimento, *et al.*, 2009) and the 5DPO robot. Note that the results considered the diameter of the robot (40 cm) that leaves the error measurement almost irrelevant. The results point out to the importance to consider both systems to certify that this controller could work in both of them even with the difference in the dynamic parameters that increase the controlling problem.

REFERENCES

Conceição, A. G. S., A. P. Moreira and P. Costa (2008). A nonlinear model predictive control strategy for trajectory tracking of a four-wheeled omni-directional mobile robot. *Optimal Control, Applications and Methods, John Wiley & Sons, Ltd. OPTIMAL CONTROL APPLICATIONS AND METHODS*, DOI: 10.1002/oca.827, **Volume 29 Issue 5**, Pages 335 - 411.

Kodagoda, K. R. S., W. S. Wijesoma and E. K. Teoh (2002). Fuzzy Speed and Steering Control of an AGV, *IEEE Transactions on control systems technology*, **Volume 10**, No. 1

Liu, Y. and J. J. Zhu (2007). Singular perturbation analysis for trajectory linearization control, in: *American Control Conference*, pp. 3047–3052.

Liu, Y., J. J. Zhu, R. L. Williams II and J. Wu (2008). Omni-directional mobile robot controller based on trajectory linearization. *Robotics and Autonomous Systems*, 56, p461–479.

Nascimento, T. P., C. C. Paim and A. L. da Costa (2009). Controle Multivariável de Trajeória para robôs móveis omni-direcionais. In: *IX Simpósio Brasileiro de Automação Inteligente (SBAI),* Brasilia.

O'Reilly, J. (1987). Multivariable Control for Industrial Aplications. *Peter Peregrinus*, United Kingdom.

Slotine, J.-J. and Li, W. (1991). Applied Nonlinear Control, *Prentice Hall*.

Watanabe, K. (1998). Control of an Omnidirectional Moblie Robot. *IEEE Second International Conference on Knowledge Based Intelligent Electronics Systems*: 51-60.