CONTROLLER OPTIMIZATION AND MODELLING OF AN OMNI-DIRECTIONAL MOBILE ROBOT

André Scolari Conceição *,1 **A. Paulo Moreira** * **Paulo J. Costa** *

* *Department of Electrical and Computer Engineering University of Porto - Porto - Portugal.*
[scolari,amoreira,paco]@fe.up.pt

Abstract: This paper presents a dynamic and kinematic model and a trajectory controller for an omni-directional mobile robot. The parameters of the controller are optimizated based on trajectory following simulations, with the mobile robot model, take into account aspects like time and errors of position and orientation of the robot. Simulation and real results of trajectory following are presented.

Keywords: Trajectory control, modelling and simulation, omni-directional mobile robot.

1. INTRODUCTION

Omni-directional mobile robots have the ability to move simultaneously and independently in translation and rotation (Pin and Killough, 1994). However, non-linearities, like motor dynamic constraints, and others characteristics like friction, inertia moment and mass of the robot, should be modelled, because can greatly affect the robot behaviour. Hence, dynamic modelling of mobile robots is very important to design of controllers, as in (Liu *et al.*, 2003)(Watanabe, 1998)(Fraga *et al.*, 2005)(Kalmár-Nagy *et al.*, 2002), mainly when the robots need to follow trajectories at higher velocity, with sudden change in its direction and orientation.

The suggested controller presents interesting features to follows the path correctly, how the possibility to define different linear velocities and angular positions to the robot during the trajectory following. A trajectory can be approximated with line segments. A line segment has two distinct endpoints. We use in this paper the name of a line segment with endpoints A and B as "line segment $\overline{AB}$". Hence, we can define linear velocities and angular position to the robot in each endpoint of the line segment, moreover we can adjust its velocities and angular position to the long of the line segment. Another feature of the controller is low computational time, which is essential in real time applications.

The optimization of the parameters of the controller is based on robot model. Due to values of time and errors of position and orientation of the robot, in trajectory following simulations with the robot model, we can calculate the best parameters to the controller.

We focus attention on a omni-directional mobile robot with four motors, as shown in Fig.1, built for the 5dpo Robotic Soccer team from the Department of Electrical and Computer Engineering at the University of Porto at Porto, Portugal(Moreira *et al.*, 1999). The organization

[1] Supported by the Program Alβan, the European Union Program of High Level Scholarships for Latin America, scholarship n.E04D028256BR.

of the paper is as follows. In section 2, the omni-directional mobile robot model is developed. The controller for trajectory following is presented in section 3. In section 4, the optimization of the controller parameters, simulation results and real results are presented. Finally, the conclusion is drawn in section 5.

2. THE OMNI-DIRECTIONAL MOBILE ROBOT MODEL

The omni-directional mobile robot model is developed based on the dynamics, kinematics and DC motors of the robot.

Fig. 1. Mobile robot.

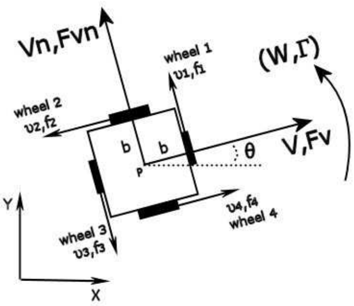

Fig. 2. Geometric parameters and coordinate frames.

The World frame (X, Y, θ), the robot's body frame and the geometric parameters is shown in Fig. 2. The following symbols, in SI unit system, are used to modelling:

- b $[m]$ $\rightarrow$ distance between the point P(center of chassis) and robot's wheels
- M $[kg]$ $\rightarrow$ robot mass
- r $[m]$ $\rightarrow$ wheel radius
- l $\rightarrow$ motor reduction
- J $[kg.m^2]$ $\rightarrow$ robot inertia moment
- B_v, B_{vn} $[N/(m/s)]$ $\rightarrow$ viscous friction related to V and V_n velocities
- B_w $[N/(rad/s)]$ $\rightarrow$ viscous friction related to W velocity
- C_v, C_{vn} $[N]$ $\rightarrow$ coulomb friction related to V and V_n velocities
- C_w $[N.m]$ $\rightarrow$ coulomb friction related to W velocity
- V, V_n $[m/s]$ $\rightarrow$ linear velocities of the robot
- W $[rad/s]$ $\rightarrow$ angular velocity of the robot
- θ $[rad]$ $\rightarrow$ orientation angle of the robot
- F_v, F_{vn} $[N]$ $\rightarrow$ traction forces of the robot
- Γ $[N.m]$ $\rightarrow$ rotation torque of the robot
- v_1, v_2, v_3, v_4 $[m/s]$ $\rightarrow$ wheels linear velocities
- f_1, f_2, f_3, f_4 $[N]$ $\rightarrow$ wheels traction forces
- T_1, T_2, T_3, T_4 $[N.m]$ $\rightarrow$ wheels rotation torque

2.1 Robot Dynamics

By Newton's law of motion and the robot's body frame, in Fig. 2, we have

$$F_v(t) = M\frac{dV(t)}{dt} + B_vV(t) + C_v sgn(V(t)) \quad (1)$$

$$F_{vn}(t) = M\frac{dVn(t)}{dt} + B_{vn}V_n(t) + C_{vn} sgn(V_n(t)) \quad (2)$$

$$\Gamma(t) = J\frac{dW(t)}{dt} + B_wW(t) + C_w sgn(W(t)) \quad (3)$$

where,

$$sgn(\alpha) = \begin{cases} 1, & \alpha > 0, \\ 0, & \alpha = 0, \\ -1, & \alpha < 0. \end{cases}$$

The relationships between the robot's traction forces and the wheel's traction forces are,

$$F_v(t) = f_4(t) - f_2(t) \quad (4)$$

$$F_{vn}(t) = f_1(t) - f_3(t) \quad (5)$$

$$\Gamma(t) = (f_1(t) + f_2(t) + f_3(t) + f_4(t))b \quad (6)$$

The wheel's traction force(f) and the wheel's torque(T), for of each DC motor, is as follow:

$$f(t) = \frac{T(t)}{r} \quad (7)$$

$$T(t) = l.K_t.i_a(t) \quad (8)$$

where $i_a(t)$ is the armature current and K_t is motor torque constant. The dynamics of each DC motor can be described using the following equations,

$$u(t) = L_a\frac{di_a(t)}{dt} + R_ai_a(t) + K_vw_m(t) \quad (9)$$

$$T(t) = K_ti_a(t) \quad (10)$$

where L_a is the armature inductance, R_a is the armature resistance, $u(t)$ is the applied armature voltage, $w_m(t)$ is the rotor angular velocity in rad/sec, k_v is the emf constant.

2.2 Robot Kinematics

By geometric parameters of the robot and the robot's body frame, in Fig. 2, is possible to derive the motion equations,

$$\begin{aligned}\frac{dx(t)}{dt} &= V(t)cos(\theta(t)) - Vn(t)sen(\theta(t))\\ \frac{dy(t)}{dt} &= V(t)sen(\theta(t)) + Vn(t)cos(\theta(t))\\ \frac{d\theta(t)}{dt} &= W(t)\end{aligned} \tag{11}$$

The relationships between wheel's linear velocities (v_1, v_2, v_3 and v_4) and robot velocities (V,Vn and W) are,

$$\begin{aligned} v_1(t) &= V_n(t) + bW(t)\\ v_2(t) &= -V(t) + bW(t)\\ v_3(t) &= -V_n(t) + bW(t)\\ v_4(t) &= V(t) + bW(t)\end{aligned} \tag{12}$$

Where $x(t)$ and $y(t)$ is the localization of the point P, and $\theta(t)$ the orientation angle of the robot.

3. LINE SEGMENT CONTROLLER

The proposed controller adjusts the position and orientation of the robot to follow a line segment, defined in the plane XY, as shown in Fig.3. From a line segment and the position of the robot, we can define the velocity vectors to the robot. The velocity vectors, robot position(P) and a line segment($\overline{AB}$) are shown in Fig.3.

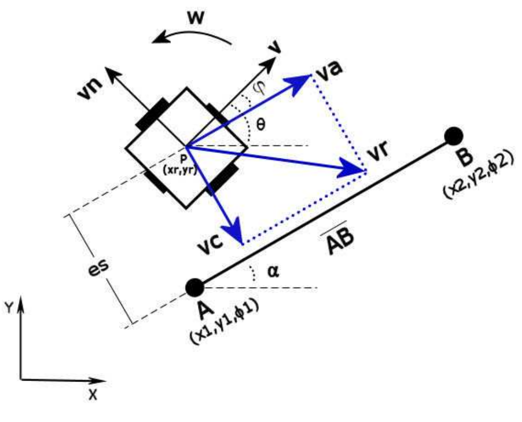

Fig. 3. Schematic of the controller.

The robot position is $P(x_r, y_r)$ and θ is the orientation angle of the robot in the plane XY. The velocity vectors V and Vn are perpendicular, and represent the linear velocities of the robot. The angular velocity of the robot is W. The angle φ is the difference between the line segment angle (α) and the robot angle (θ):

$$\varphi = \alpha - \theta. \tag{13}$$

The velocity vector v_r is the desired linear velocity to robot, called too reference velocity. The velocity v_r can receive different values in both points(A and B) of the line segment, hence the robot can follow trajectories with different reference velocities. The reference velocity to the long of the line segment is:

$$v_r = v_{r1}(1 - d) + v_{r2}d. \tag{14}$$

Where v_{r1} is the reference velocity of the point A and v_{r2} is the reference velocity of the point B, of the line segment $\overline{AB}$. The variable d is the projection from robot position($P(xr, yr)$) to line segment $\overline{AB}$, it is normalized to length of the line segment. The vector velocities of the controller, are as follow:

$$v_c = e_s k_1, \tag{15}$$

$$v_a = \begin{cases} 0 & , \; v_r^2 - v_c^2 < 0,\\ \sqrt{v_r^2 - v_c^2}, & v_r^2 - v_c^2 > 0. \end{cases} \tag{16}$$

With the vector v_a parallel to $\overline{AB}$($v_a \parallel \overline{AB}$) and the vector v_c perpendicular to $\overline{AB}$($v_c \perp \overline{AB}$). The distance between the robot and the line segment is e_s, and k_1 is a gain. We can calculate the vector velocities V and V_n, using a rotation matrix:

$$\begin{bmatrix} V\\ Vn \end{bmatrix} = \begin{bmatrix} cos(\varphi) & -sen(\varphi)\\ sen(\varphi) & cos(\varphi) \end{bmatrix} \begin{bmatrix} v_a\\ v_c \end{bmatrix}.$$

The robot angular velocity(W) is calculated based on robot angular position(θ) and desired angular positions(ϕ) in both points(A and B) of the line segment. For all line segment, the angular velocity W is calculated with a similar variation used in equation 14, in function of d. Therefore, when is defined the line segment $\overline{AB}$, we define the desired angular positions(ϕ_1 and ϕ_2) in each point of the line segment. The controller to robot angular position is defined as follow:

$$W = e_\theta k_2. \tag{17}$$

with:

$$e_\theta = \theta_{ref} - \theta, \tag{18}$$

$$\theta_{ref} = \phi_1(1 - d) + \phi_2 d. \tag{19}$$

where k_2 is a gain, e_θ is the error between desired angular position(θ_{ref}) and robot angular position(θ).

4. OPTIMIZATION OF THE CONTROLLER

After define the controller structure, we need to choose the appropriate values to gain k_1 and gain k_2. A cost function (C) was created to measure the performance of trajectory following. The cost function is described as follow:

$$C(k_1, k_2) = E_d P_d + E_a P_a + (T_r - T_i) P_t.$$

Where:

$E_d \hookrightarrow$ Mean square error(MSE) related to e_s, for all trajectory following;
$E_a \hookrightarrow$ Mean square error related to e_θ, for all trajectory following;
$T_i \hookrightarrow$ ideal time to follow the trajectory;
$T_r \hookrightarrow$ robot time to follow the trajectory;
P_d, P_a, $P_t \hookrightarrow$ gain related to errors.

We used the robot model, described in section 2, to define k_1 and k_2 gains. Trajectory following simulations, see Fig. 4, with different values of the k_1 and k_2, make possible obtain the values of the cost function. The objective is found the values of the k_1 and k_2 that result the minimum value of the cost function.

Through real experiments with the robot, we know that gains (k_1 and k_2) above of 15 can cause oscillation in trajectory following. Hence, we used values between 0 and 20, with resolution of 0.5 in trajectory following simulations. The trajectory following simulations were made for 3 values of the linear velocity v_r: 1, 0.7 e 0.4 $[m/s]$. So, it computed 1600 simulations for each velocity, resulting in 4800 total simulations. Without the robot model, it will be impossible. Furthermore, the total simulation time for each velocity(1600 simulations) is about 10 seconds, therefore we did not need to use minimization algorithms.

The trajectory used in the simulations has special features, as sudden change of direction and orientation to the robot, in order to test the controller in hard condition. The Fig. 4 shows the trajectory used in the simulations, with the points(x,y). The Fig. 5 shows values of the desired angular positions(ϕ) for each point(x_i,y_i), i=1, ...13 of the trajectory.

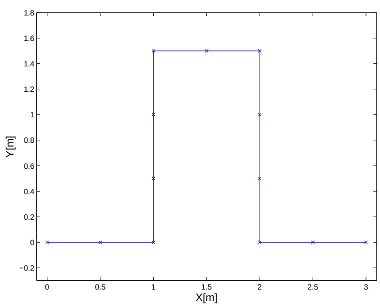

Fig. 4. Trajectory to calculate the cost function.

The cost functions for 3 linear velocities v_r and the gains (k_1 and k_2) are shown in Figs. 6, 7 and 8. The minimum values of the cost function(C) and correspondents gains are shown in table 1.

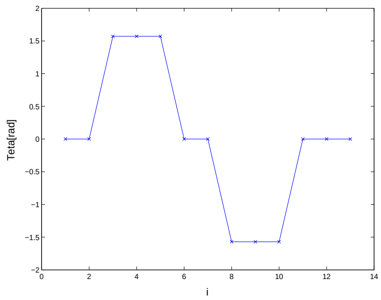

Fig. 5. Angular positions(ϕ) of the trajectory.

$v_r[m/s]$	C	k_1	k_2	P_d	P_a	P_t
1	5.57	8	12.5	1000	10	1
0.7	3.87	7	10.5	1000	10	1
0.4	1.21	7.5	9	1000	10	0.5
mean≈	-	7.5	10.5	-	-	-

Table 1. Minimum values of the cost function.

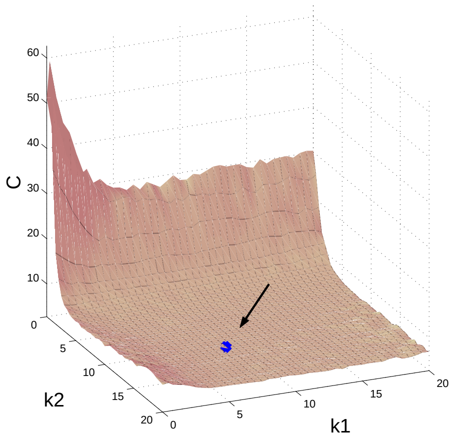

Fig. 6. Cost function, $v_r = 1\ m/s$.

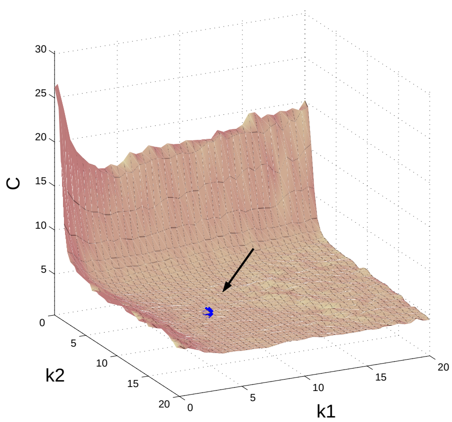

Fig. 7. Cost function, $v_r = 0.7\ m/s$.

The values of the gains P_d, P_a and P_t were defined in order to be even the values of the errors E_d, E_a and $(T_r - T_i)$. Currently we use the mean of the gains values, shown in table 1. We tested this values in robot trajectory following for the linear velocities $v_r = 0.4, 0.7, 1[m/s]$. It had a good performance so, the use of different gains for different linear velocities is not necessary. Real and simulated results with gain $k_1 = 7.5$, gain

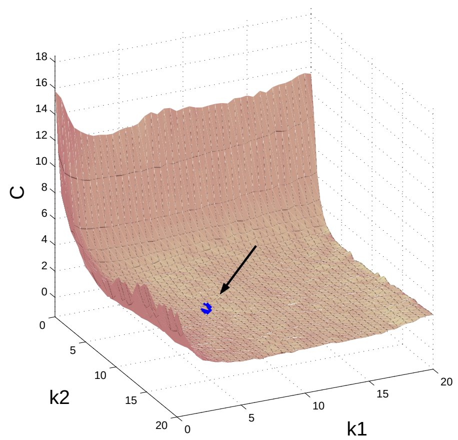

Fig. 8. Cost function, $v_r = 0.4\ m/s$.

$k_2 = 10.5$ and linear velocity $v_r = 1[m/s]$ are shown if Figs. 9, 10 and 11.

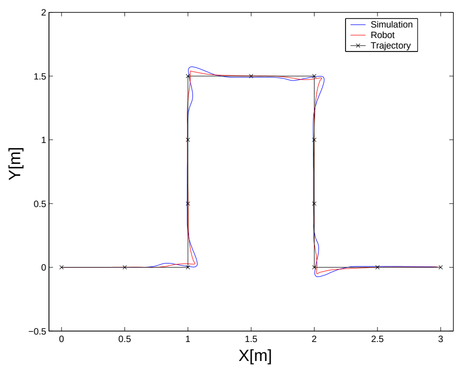

Fig. 9. Simulated and real trajectory.

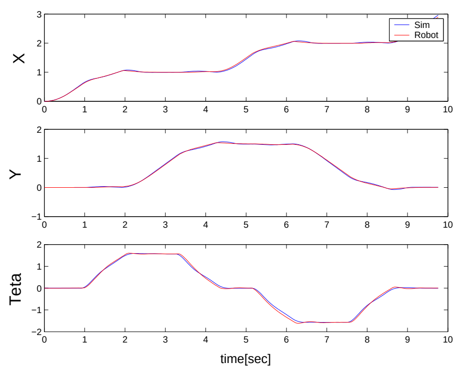

Fig. 10. $X(m), Y(m), \theta(rad)$.

The procedure to define the gains k_1 and k_2 was repeated with a trajectory like "8", see Fig. 12, this trajectory has a different feature than first one, in Fig. 4. The robot does not need to do sudden change of direction and orientation. The table 2 shows the minimum values of the cost function(C) and correspondents gains. In this simulations we kept the same gains to P_d, P_a and P_t, to get a comparison. The results for the gain k_2 were equal in both procedures. The results for the gain k_1 were not equal, the gain diminished with the reduction of the linear velocity v_r. It happened due to characteristics of the trajectory "8", this

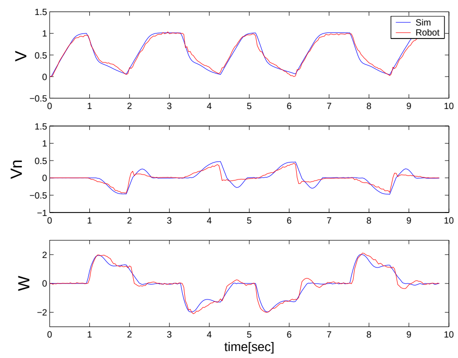

Fig. 11. $V(m/s), Vn(m/s), W(rad/s)$.

trajectory is more soft than the first trajectory, consequently the error E_d diminished too.

Finally, we tested the gains k_1 and k_2 obtained with the first trajectory, in the robot to follow the trajectory "8", the robot had a satisfactory performance. We decided to use the bigger gains ($k_1 = 7.5$ and $k_2 = 10.5$), because in robotic soccer application the mobile robot needs to execute trajectories quickly and with a perfect position to the objective, for example, positioning to the ball, or to the goal, or to avoid dynamic obstacles.

$v_r[m/s]$	C	k_1	k_2	P_d	P_a	P_t
1	2.84	9	12.5	1000	10	1
0.7	0.72	5	10.5	1000	10	1
0.4	0.22	3	9	1000	10	0.5
mean≈	-	5.5	10.5	-	-	-

Table 2. Minimum values of the cost function, trajectory "8".

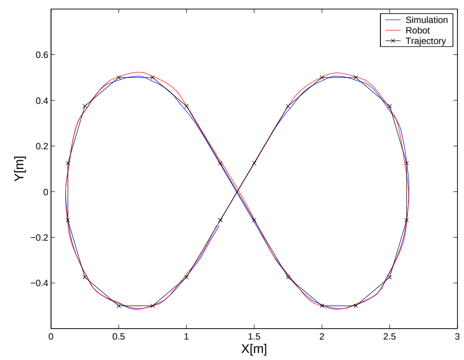

Fig. 12. Simulated and real trajectory.

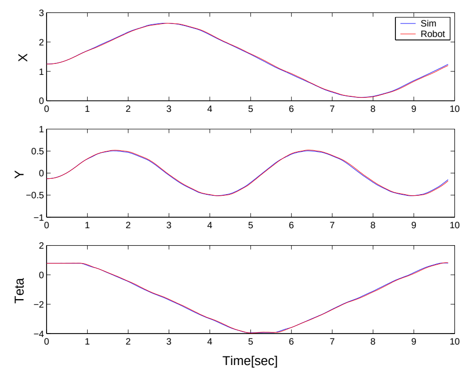

Fig. 13. $X(m), Y(m), \theta(rad)$.

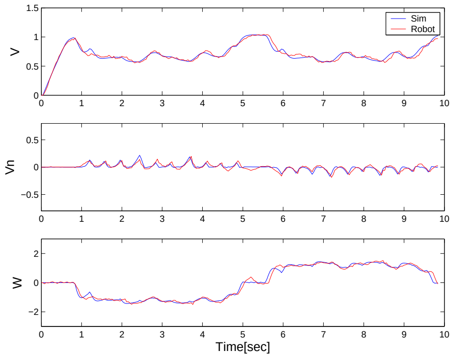

Fig. 14. $V(m/s), Vn(m/s), W(rad/s)$.

5. CONCLUSION

The main purpose of this paper was the optimization of the parameters of the controller for trajectory following. Dynamic and kinematic models of a omni-directional mobile robot were presented, as well as their importance on the controller design, due to the possibility of getting more information through the simulations than using only the robot. The proposed controller presents important features, as the possibility of defining different translation velocities and angular positions to the robot during the trajectory following. Besides, it does not demand a high computational time, which is essential in real time applications and applications of high velocity.

REFERENCES

Fraga, Sérgio Loureiro, João borges de Sousa and Fernando Lobo Pereira (2005). Optimização dinâmica para planeamento de movimento e controlo de veículos autónomos. *Robótica 2005 - Festival Nacional de Robótica*.

Kalmár-Nagy, Tamás, Pritam Ganguly and Raffaello D.Andrea (2002). Real-time trajectory generation for omnidirectional vehicles. *Proceedings of the American Control Conference* pp. 286–291.

Liu, Yong, Xiaofei Wu, J Jim Zhu and Jae Lew (2003). Omni-directional mobile robot controller design by trajectory linearization. *Proceedings of the American Control Conference* **4**, 3423 – 3428.

Moreira, A.Paulo, P. Costa, A.Sousa, P. Marques, P. Costa and A. Matos (1999). 5dpo team description robocup. *Robot World Cup Soccer Games and Conference*.

Pin, Fracois G. and Stephen M. Killough (1994). A new family of omnidirectional and holonomic wheeled platforms formobile robots. *IEEE Transactions on Robotics and Automation* **10**, 480–489.

Watanabe, Keigo (1998). Control of ominidirectional mobile robot. *2nd Int. Conf. on Knowledge-Based Intelligent Electronic Systems*.