TRAJECTORY TRACKING FOR OMNI-DIRECTIONAL MOBILE ROBOTS BASED ON RESTRICTIONS OF THE MOTOR'S VELOCITIES.

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: In this paper, we propose an algorithm that combine the restriction on motor's velocities and the kinematic model of Omni-Directional mobile robots to improve the trajectory's following. The algorithm verifies the reference velocities of the robot and redefine them if necessary, in order to prevent possible saturation on motor's velocities. Simulation results of the algorithm applied to an omni-directional mobile robot are presented. *Copyright © 2006 IFAC*

Keywords: Mobile robots, omni-direction, trajectory design.

1. INTRODUCTION

The motivation of this work comes from the necessity of mobile robots to follow trajectories in a correct and fast form. For this to happen, it is necessary to take into account aspects like the limits of motor's velocities of mobile robot and holonomics movements (see (Pin and Killough, 1994)(Yi and Kim, 2002)). Omni-directional mobile robots have the ability to move simultaneously and independently in translation and rotation. However, non-linearities, like motor dynamic constraints can greatly affect the robot behaviour, especially when the robot is accelerated and decelerated.

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 (Costa *et al.*, 1999). For this application (Robotic Soccer) 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. So, the dynamic characteristics of the motion are essential to path tracking objective.

Fig. 1. Omni-directional mobile robot.

Methods to establish a real-time control strategy (voltages to the motors) that will move the robot to a given location, with zero final velocity,

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

as quickly as possible are presented by (Kalmár-Nagy *et al.*, 2002). Some constraints on the vehicle's design are proposed in an effort to simplify its motion equation by (He *et al.*, 2004). In this work we analyse the limitations of motor's velocities and its influence in real-time trajectory design. Due to robot characteristic (omni-directional), reference velocities may cause saturation in motor's velocities, mainly when trajectories have linear velocities (V,Vn) and angular velocity (W) different from zero (see Fig. 4). The algorithm, called IRV (ideal reference velocities), is used to verify the reference velocities of the robot and redefine them, if necessary, in order to prevent possible saturation on robot's motors and to obtain the best motor's performance. This is done keeping the right direction of the velocity's vectors(V,Vn and W). On the other hand, in motor's saturation cases, errors can happen in the robot trajectory, because the robot cannot follow the reference velocities.

The robot's control and communication structure can be seen in Fig. 2. The computer (PC) controls all the actions of the robot. The communication with the microcontroller is done through the serial port (RS232). It controls the motors using signals of PWM (Pulse Width Modulator) and a drive of power. The software of control in PC works in a frequency of $25Hz$, from the camera. So, in each 40(ms) the trajectory and the reference velocities of the mobile robot are recalculated, based at the current mobile robot position, and the new desired position.

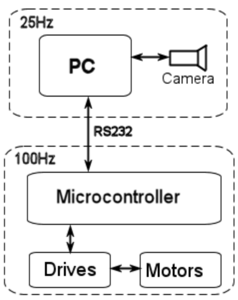

Fig. 2. Robot's control and communication structure.

The schematic in Fig. 3, shows the trajectory control loop of the mobile robot.

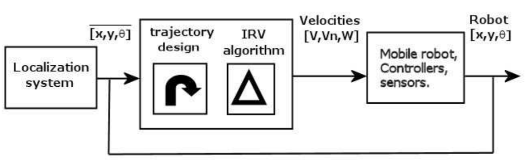

Fig. 3. Schematic - trajectory control loop.

This paper is structured in the following way. In section 2, we present the kinematic model of the mobile robot. In section 3, the proposed IRV(ideal reference velocities) algorithm is presented. The simulation results are presented in Section 4. Finally, the conclusion is drawn in section 5.

2. KINEMATIC MODEL

The reference velocities of the mobile robot are the linear velocities ($V(t)$,$Vn(t)$) and angular velocity ($W(t)$). They are converted in the linear wheel velocities $v1(t)$, $v2(t)$, $v3(t)$ and $v4(t)$:

$$v1(t) = Vn(t) + fW(t) \quad (1)$$

$$v2(t) = -V(t) + gW(t) \quad (2)$$

$$v3(t) = -Vn(t) + fW(t) \quad (3)$$

$$v4(t) = V(t) + gW(t) \quad (4)$$

where (see Fig. 4):

- f: the distance between the point C and wheels of the motors M1 e M3;
- g: the distance between the point C and wheels of the motors M2 e M4.

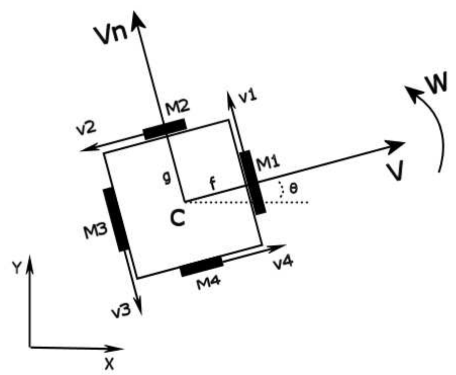

Fig. 4. Definitions of posture and velocities.

Using the Fig. 4, it is possible to write the motion equations of the mobile robot,

$$V(t) = \frac{1}{2}(-v2(t) + v4(t)) \quad (5)$$

$$Vn(t) = \frac{1}{2}(v1(t) - v3(t)) \quad (6)$$

$$W(t) = \frac{1}{2f}(v1(t) + v3(t)) \quad (7)$$

$$W(t) = \frac{1}{2g}(v2(t) + v4(t)) \quad (8)$$

where:

- $V(t)$ and $Vn(t)$: linear velocities of the point C;
- $W(t)$: angular velocity of the mobile robot;
- $v1(t)$,$v2(t)$,$v3(t)$ e $v4(t)$: linear velocity of the wheels of the mobile robot;

3. IRV ALGORITHM

The IRV algorithm calculates the robot velocities $V(t)$,$Vn(t)$ and $W(t)$, taking into account the limitations of motor's velocities and the kinematic model of the robot. So, its equation can be described with three control variables ($V(t)$,$Vn(t)$ and $W(t)$), related with linear wheel velocities of the mobile robot($v1(t)$,$v2(t)$,$v3(t)$ and $v4(t)$).

Therefore, for one determined velocity W, from the equation 7:

$$v3 = 2fW - v1 \quad (9)$$
$$v1 \in [-v1_{max}, v1_{max}]$$

Using 9 in 6,

$$Vn = \frac{1}{2}(v1 - 2fW + v1),$$
$$Vn = v1 - fW. \quad (10)$$

where, for $v1 = v1_{max}$ and $v1 = -v1_{max}$, we obtain for this value of $W(t) = 0$ the maximum positive and negative linear velocity $Vn(t)$.

With the same procedure for others motors:

$$V = -v2 + gW \quad (11)$$
$$v2 \in [-v2_{max}, v2_{max}]$$

$$Vn = -v3 + fW \quad (12)$$
$$v3 \in [-v3_{max}, v3_{max}]$$

$$V = v4 - gW \quad (13)$$
$$v4 \in [-v4_{max}, v4_{max}]$$

The equations (10...13) can be analysed how equation of a plane,(see (Anton and Rorres, 2000)),

$$ax + by + cz + d = 0 \quad (14)$$

Changing the coordinates system x,y and z for V,Vn and W respectively,

$$aV + bVn + cW + v_i = 0 \quad (15)$$

where v_i is linear velocities of robot wheels, for $i = 1, 2, 3, 4$. Rearranging the equations, we obtain eight equations of the a plane, considering W positive and negative,

$$Vn + fW - v1 = 0 \quad (16)$$
$$V - gW + v2 = 0 \quad (17)$$
$$Vn - fW + v3 = 0 \quad (18)$$
$$V + gW - v4 = 0 \quad (19)$$
$$Vn - fW - v1 = 0 \quad (20)$$
$$V + gW + v2 = 0 \quad (21)$$
$$Vn + fW + v3 = 0 \quad (22)$$
$$V - gW - v4 = 0 \quad (23)$$

The Fig. 5 shows eight planes with the follow limits of velocities for the wheels robot: $v1 = v3 = 1(m/s)$ and $v2 = v4 = 1.5(m/s)$. The Fig. 5 was generated with W positives and negatives, until the instant that velocity Vn reached zero ($Vn = 0$).

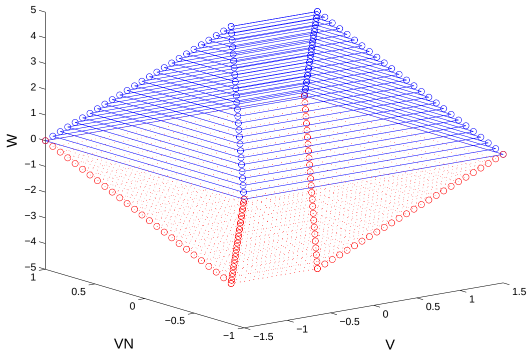

Fig. 5. Solid that represents the possible robot velocities(V, Vn, W).

Analysing Fig. 5, the reference velocities (V_{ref}, Vn_{ref}, W_{ref}), should be inside the solid. To redefine the reference velocities, that are outside the solid (in Fig. 5), we can find the intersection point($\overline{P}_{ref}$) of the line l and the planes of the equations(16...23).

Referring to Fig.6, let the line l from the origin to the point $P_{ref} = (V_{ref}, Vn_{ref}, W_{ref})$. Then, intersection point($\overline{P}_{ref}$) give us the maximum robot velocities, respecting the limits of robot's motors, as the follow formulation,

- $G = aV + bVn + cW + v_n = 0$: equation of plane;
- $P_{ref} = (V_{ref}, Vn_{ref}, W_{ref})$: point of reference velocities;
- $\overline{P}_{ref} = \alpha P_{ref}$: point of redefined reference velocities;
- α : factor of scale.

so,

$$\alpha P_{ref} = (\alpha V_{ref}, \alpha Vn_{ref}, \alpha W_{ref}) \quad (24)$$
$$\alpha P_{ref} \in G$$

substituting αP_{ref} in the equation of plane, to obtain α,

$$a\alpha V_{ref} + b\alpha Vn_{ref} + c\alpha W_{ref} + v_i = 0,$$
$$\alpha(aV_{ref} + bVn_{ref} + cW_{ref}) + v_i = 0,$$

$$\alpha = \frac{-v_i}{aV_{ref} + bVn_{ref} + cW_{ref}} \quad (25)$$

For example, the equation of plane 16, the values are,

- $v_i = -v1$,
- $a = 0$,
- $b = 1$,
- $c = f$.

Now, we present an example with the point $P_{ref} = (1, -1.2, 2)$, representing the robot velocities, with following parameters,

- $v1 = v3 = 1(m/s)$,
- $v2 = v4 = 1.5(m/s)$,
- $f = g = 0.2(m)$.

The point $P_{ref} = (1, -1.2, 2)$ is outside of solid (see Fig.6 and Fig.7), after redefine the robot velocities, with the formulation above, we obtain the point $\overline{P}_{ref} = (0.625, -0.75, 1.25)$ (point of the intersection with the plane) and the $\alpha = 0.625$.

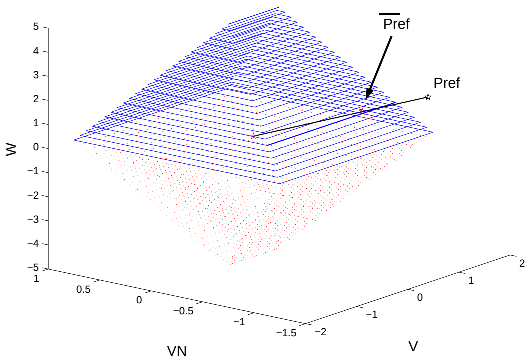

Fig. 6. Axis V,Vn,W and points P_{ref} and $\overline{P}_{ref}$.

The Fig. 7 shows the axis (Vn,W), where we can see the point of intersection $\overline{P}_{ref}$ with the plane.

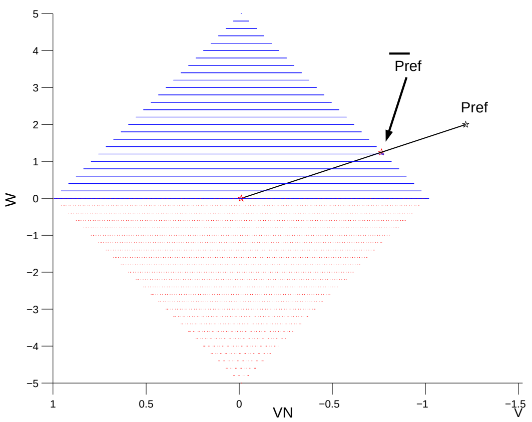

Fig. 7. Axis Vn,W and points P_{ref} and $\overline{P}_{ref}$.

The reference velocities (P_{ref}) were redefined keeping the right direction of the velocity's vectors (V, Vn and W), so the robot trajectory will be done slowly, but not have errors.

4. SIMULATION RESULTS

We used to test the IRV algorithm the trajectory shown in Fig. 8. The initial position of mobile robot was $x_i = 0(m)$, $y_i = 0(m)$ and $\theta_i = 0(rad)$. The final position was $x_f = -2(m)$, $y_f = 0(m)$ and $\theta_f = \pi(rad)$.

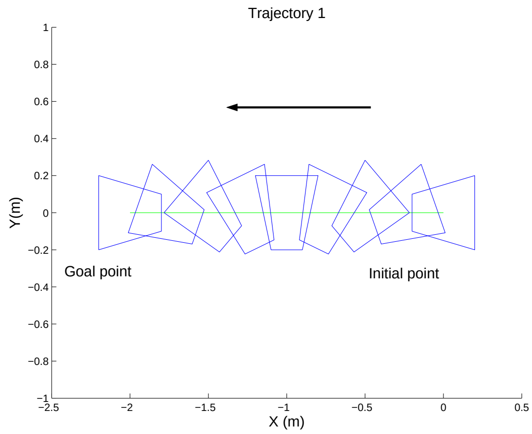

Fig. 8. Axis coordinates XY - trajectory for test.

Three simulation cases had been tested:

(1) with an ideal model of the mobile robot;
(2) with the model of the mobile robot, without the IRV algorithm;
(3) with the model of the mobile robot, with the IRV algorithm.

With the following parameters,

- velocity limits in motors 1 and 3: $v1 = v3 = 1(m/s)$,
- velocity limits in motors 2 and 4: $v2 = v4 = 1.5(m/s)$,
- $f = g = 0.2(m)$.

The Fig. 9 shows the robot velocities executed by ideal model(case 1), this model does not have velocity restrictions, following high reference velocities($V_{ref}, Vn_{ref}, W_{ref}$). In Fig. 12 we show that the trajectory was tracked correctly.

The Fig. 10 shows the robot velocities executed by model of the mobile robot (case 2), that take into account the limitations of motor's velocities, but without use the IRV algorithm. It results that the motor's velocities has reached saturation point, generating errors in robot's velocities and consequently, executing the trajectory in a wrong form, as in Fig. 12.

The Fig. 11 shows the robot velocities executed by model of the mobile robot, that take into account the limitations of motor's velocities, but with the IRV algorithm(case 3). The velocities of the mobile robot respects the motors limits, following the trajectory correctly, as in Fig. 12.

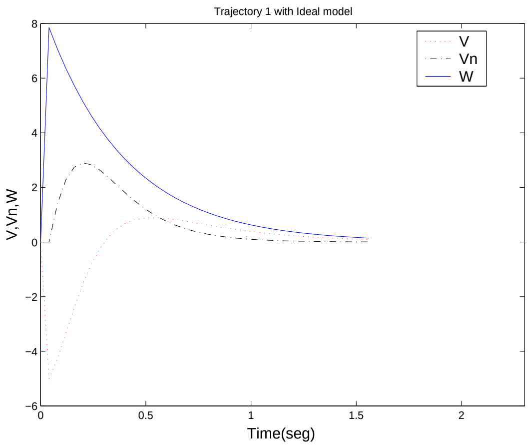

Fig. 9. Ideal mobile robot model.

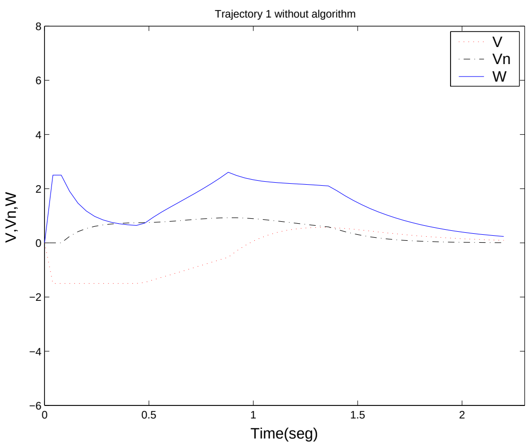

Fig. 10. Mobile robot model, without the IRV algorithm.

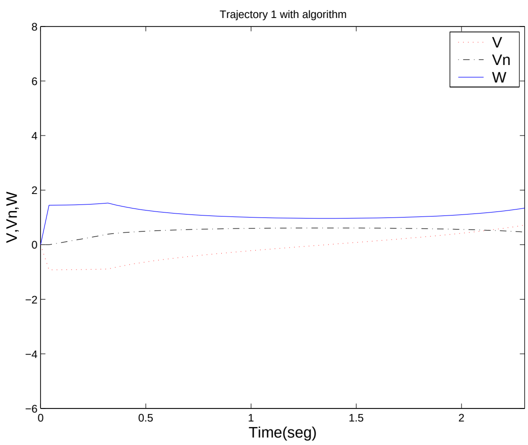

Fig. 11. Mobile robot model, with the IRV algorithm.

5. CONCLUSION

This paper formulates an algorithm to define the robot's reference velocities in order to enable the trajectories to be followed at maximum velocity without significant errors. This algorithm is helpful for several applications where the exact trajectory's following is priority. In robot soccer case, where the trajectories change all the time because the ball is always in movement, accurate trajectories and positioning are needed. The procedure presented in this paper is implemented in 5dpo Robotic Soccer team from the Department of Electrical and Computer Engineering at the University of Porto. The same procedure can be used for omni-directional robots, with different configuration (three wheels) and in all applications where critical trajectories must be perfectly executed.

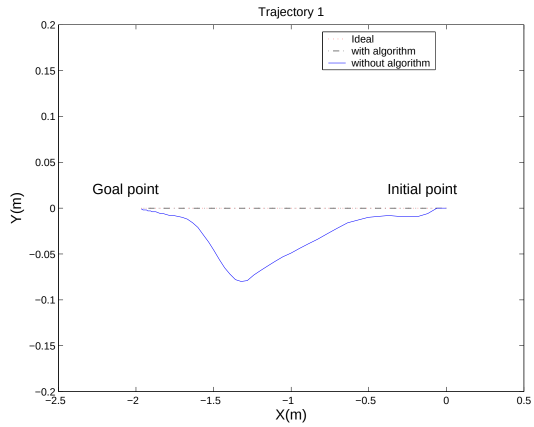

Fig. 12. Robot's trajectory in the three cases.

REFERENCES

Anton, H. and C. Rorres (2000). *Elementary Linear Algebra - Applications version*. John Wiley Sons Inc.

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

He, Zhenfeng, Lanfen Lin and Xueying Ma (2004). Design, modeling and trajectory generation of a kind of four wheeled omni-directional vehicle. *IEEE International Conference on Systems, Man and Cybernetics* pp. 6125–6130.

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.

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.

Yi, Byung-Ju and Whee Kuk Kim (2002). The kinematics for redundantly actuated omnidirectional mobile robots. *Journal of Robotic Systems* **19**, 255–267.