

A KINEMATIC AND DYNAMIC MODEL-BASED MOTION CONTROLLER FOR MOBILE ROBOTS

J. R. Asensio L. Montano

*Department of Computer Science and Systems Engineering
University of Zaragoza
María de Luna, 3, E-50015 Zaragoza, SPAIN
{jrasensio, montano}@posta.unizar.es*

Abstract: The paper presents a model for motion generation of differential-drive mobile robots. The parameters of the dynamic model allow adjusting the robot translational and rotational behaviours separately. The model takes into account the robot kinematic and dynamic constraints, making the velocities and accelerations bounded and compatible with those the robot can perform. The main contribution of the paper is to use the model itself as a motion controller: under soft hypothesis on the velocities and accelerations, this approach allows an easy tuning of the controller parameters. A system stability and parameters sensitivity analysis is developed, in order to get guidelines for controller tuning. The clear physical sense of the parameters make this tuning easy and intuitive. Experimental results involving a real mobile robot show the performance of this approach.

Keywords: Model-Based Control, Model Reference Control, Robot Dynamics, Robot Kinematics, Robot Navigation.

1. INTRODUCTION

In the field of mobile robotics is usual to work with kinematic models to obtain stable motion control laws for trajectory following or goal reaching (Khatib *et al.*, 1997; Ramírez and Zeghloul, 2000). Other authors have proposed dynamic models relating the setpoints to the servos and the robot linear and angular velocities (Espinosa *et al.*, 1998; Fierro and Lewis, 1997). (Topalov *et al.*, 1998) use a model in which the torques of the servos appear as the input vector. Similarly, (Yun and Yamamoto, 1997) link the robot coordinates and the turned angle of each wheel with the servo's torques.

In this paper a new model for differential drive mobile robots is presented (§2). It is a dynamic model relating a virtual force applied on a point on the robot with

the linear and angular velocities. The model takes into account the robot kinematic and dynamic constraints, leading to bounded velocities and accelerations that are compatible with those a real mobile robot can perform. The main contribution of this paper is to use the model itself as a controller in a closed-loop control scheme. This approach has several advantages. Having the parameters of the model/controller a clear physical sense, they can be easily tuned to comply with the desired robot dynamic behaviour. Moreover, there is no need to identify the robot parameters, whenever some soft hypothesis on the robot velocities and accelerations are respected. Velocities to follow a path or reach a goal are automatically computed by the model/controller, needing no additional tuning.

The controller is used as a motion generator (§3) jointly with different navigation methods to generate the robot velocities needed to perform a navigation task (both for trajectory generation or for reactive navigation). Moreover, a stability analysis is in-

¹ This work has been partially supported by spanish project DPI2000-1272

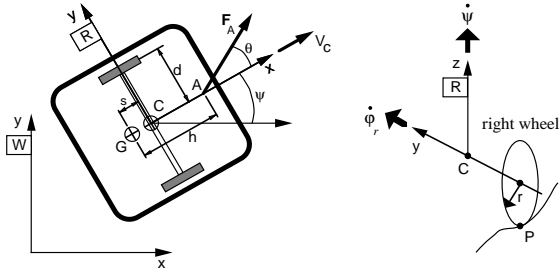


Fig. 1. Robot geometric parameters, and some used references

cluded (§4), obtaining a sufficient stability condition. A method to tune the parameters to comply with particular requirements on linear and angular velocities is also explained (§5). Experiments involving a real robot give validity to this approach (§6). Some conclusions are drawn in §7.

2. OBTAINING THE MODEL

Fig. 1 shows the geometric parameters of the model (r , d , h and s) and the references associated to the workspace W and to the robot R . Let $\mathbf{x} = (v, \omega)^T$ be the state vector, where v and ω are the robot linear and angular velocities, respectively. Let \mathbf{F}_A be a virtual force applied on the application point A in order to move the robot. Applying the equations of the Dynamics, $\Sigma \mathbf{F} = m\mathbf{a}$ and $\Sigma \mathbf{M} = \mathbf{I}_G \Omega$ (see (Montano and Asensio, 1997)) the following equations are obtained:

$$\begin{aligned} (I_{zz} + ms^2)\dot{\omega} - 2msv\omega + \frac{2bd^2}{r}\omega &= F(h-s)\sin\theta \\ m\dot{v} + ms\omega^2 + \mu mg + \frac{2b}{r}v &= F\cos\theta \end{aligned}$$

where the following dynamic parameters are involved: the module of \mathbf{F}_A (denoted as F), the mass of the robot (m), the inertia moment around the vertical axis (I_{zz}), the viscous friction coefficient (b), and the Coulomb friction coefficient (μ). After some simplifications related to the negligible terms (s is considered to be null or small enough), the previous equations transform in linear differential ones, which can be expressed as a state space equation $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$:

$$\dot{\mathbf{x}} = \begin{bmatrix} -2b/mr & 0 \\ 0 & -2bd^2/rI_{zz} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1/m & 0 \\ 0 & h/I_{zz} \end{bmatrix} \begin{bmatrix} F\cos\theta \\ F\sin\theta \end{bmatrix} \quad (1)$$

This equation models the motion of a differential drive mobile robot subjected to a force \mathbf{F}_A .

3. USING THE MODEL AS A CONTROLLER

The main contribution of this paper is to use the model of the robot as a motion controller. The advantage of this proposal is that not only the kinematic constraints

but also the dynamic of the robot are directly taken into account on the controller, its parameters having a clear physical sense.

3.1 Control Scheme

The basic idea is depicted in Fig. 2: a reference model could be used to compute the location $L_M = (x_M, y_M, \psi_M)^T$, which is compared to the robot location $L = (x, y, \psi)^T$. The error between both locations (e_M) would be used to tune the controller parameters. From this idea, the scheme is now simplified in order to develop a simpler controller.

Hypothesis. Let us assume the robot has an embedded velocity control system that tries to make the robot velocities equal to the desired ones $\mathbf{x}_d = (v_d, \omega_d)^T$. This approximation is valid if 1) the dynamic of the velocity controller is much faster than the one of the whole control scheme, and 2) the needed accelerations are in the robot acceleration range. In other words, the transfer functions for the velocities are ≈ 1 .

Assuming these hypothesis, it can be seen (Fig. 2) that the reference model itself can be used as a controller in the direct chain. In this case the error $e_M(L, L_M)$ between the robot location L and the location computed by the model L_M is $e_M = 0$, and then the final control scheme is the one represented into the double-lined block. The controller equations obtained from the model (1) are (see Fig. 3 for the meaning of θ):

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{u} &= NL(e(L_{goal}, L)) = (F\cos\theta, F\sin\theta)^T \\ F &= f(\text{distance}(L_{goal}, L)) \\ \theta &= \psi_{ref} - \psi = \arctan \frac{y_{goal} - y_A}{x_{goal} - x_A} - \psi \end{aligned}$$

If $e_M(L, L_M) \neq 0$ is because the velocities cannot follow the setpoint \mathbf{x}_d : the robot is working out of the range of admissible velocities or/and accelerations. As a consequence the controller parameters would be retuned as a function of this error e_M (the dotted line in Fig. 2). In the current implementation the parameter tuning is carried out off-line, forcing the generated accelerations to be in the range of the admissible ones for the particular robot used.

With this control scheme there is no need to identify the robot dynamic parameters (m, I_{zz}, b, μ): the parameters are only chosen to adjust the robot dynamic behaviour, generating motions compatible with the kinematic and dynamic constraints.

3.2 Parameter normalization

In order to simplify the further analysis of both stability and robot dynamic behaviour, some parameters of the Eq. 1 can be considered as unitary. It is possible to

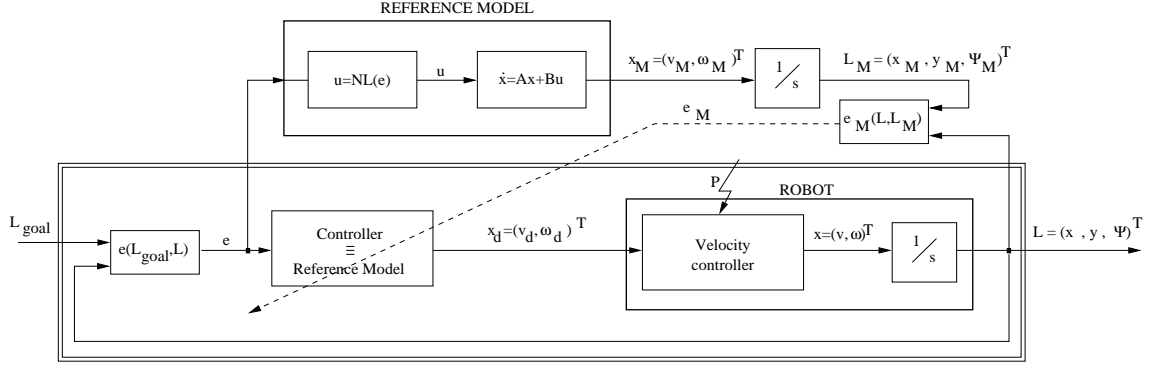


Fig. 2. How the reference model can be used as the controller in the direct chain of a closed-loop control.

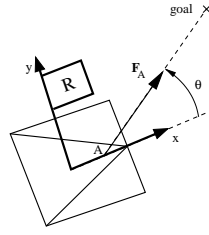


Fig. 3. The direction of \mathbf{F}_A is computed from the goal and the application point A.

work with a *normalized* robot making $m = 1$, $r = 1$, and $d = 1$. Thus, the parameters are b , F , h , and $k_i \triangleq m/I_{zz}$ (i. e., the inverse of a mass-normalized inertia moment), and the *normalized* equation becomes

$$\dot{\mathbf{x}} = -2b \begin{bmatrix} 1 & 0 \\ 0 & k_i \end{bmatrix} \mathbf{x} + F \begin{bmatrix} 1 & 0 \\ 0 & k_i h \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \quad (2)$$

This equation keeps the generality of the model, since there are yet four parameters to impose the desired behaviour for both velocities.

4. STABILITY ANALYSIS

In (Dudek and Jenkin, 2000; Fox *et al.*, 1997) the difficulty to integrate motion equations for differential drive robots to obtain the trajectories is discussed. Using the Zames method –a variation of Popov criterion, see (Atherton, 1975)– the analysis can be made without explicitly consider the non-linear part of the system (it is only needed that the non-linearity is limited to an area in the plane $e - NL(e)$). The system stability is studied in two phases: first for the rotational part of the equation (for ω) and then for its translational part (for v). This can be made in this way since the linear parts for v and ω are not coupled.

From the state space equation (2), and being $\omega \equiv \dot{\psi}$:

$$\ddot{\psi} + 2bk_i\dot{\psi} = Fk_i h \sin \theta = Fk_i h u \quad (3)$$

where

$$\begin{aligned} e &= \theta = \psi_{ref} - \psi \\ NL(e) &= \sin \theta \end{aligned}$$

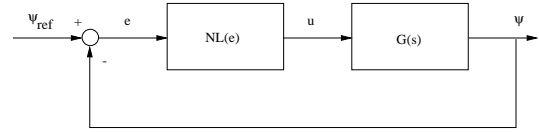


Fig. 4. Control scheme for the rotational part.

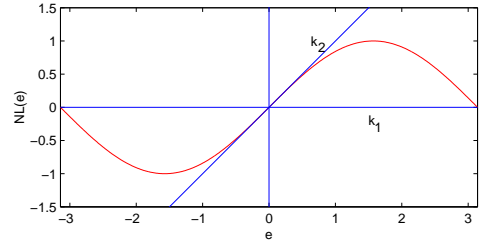


Fig. 5. $NL(e)$ has odd symmetry and is confined between two lines with slopes k_1 and k_2 .

The rotational subsystem control scheme is shown in Fig. 4. The transfer function for the linear part is:

$$G(s) = \frac{\psi(s)}{u(s)} = \frac{Fk_i h}{s(s + 2bk_i)} \quad (4)$$

The non-linear part of the system verifies:

$$k_1 e(t) \leq NL(e(t)) \leq k_2 e(t) \quad (5)$$

being $k_1 = 0$ y $k_2 = 1$ the slopes of two lines defining the region in which $NL(e)$ exists (see Fig. 5). Moreover, NL has odd symmetry, which is other condition to apply the Popov criterion on system stability. However, since k_1 is not strictly positive, the system having a pole at $s = 0$, we must apply the Zames criterion rather than the Popov criterion. On the other hand, the Zames criterion leads to conservative values for the parameters, corresponding to the sufficient stability conditions.

Zames criterion. The frequential transfer function of the rotational part is:

$$G(j\omega) = -\frac{Fk_i h}{\omega^2 + 4b^2k_i^2} - j\frac{2Fk_i^2 h b}{\omega(\omega^2 + 4b^2k_i^2)} \quad (6)$$

Using the Zames criterion, the sufficient condition for stability is

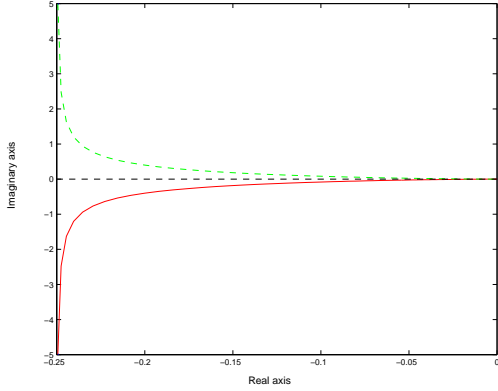


Fig. 6. Nyquist diagram for $G(j\omega)$, with $b = k_i = h = F = 1$.

$$\frac{Fh}{4b^2k_i} < 1 \quad (7)$$

The Nyquist diagram for this system with $F = h = b = k_i = 1$ is shown in Fig. 6. It has an asymptota at -0.25 , which implies that F or h could be multiplied by 4, or that b or k_i could be divided by 4, and the system will remain stable.

Another condition for the Zames criterion to be applied is that the setpoint ψ_{ref} must be impulsional or must tend towards zero as time passes. This implies that $\theta \rightarrow 0$, i. e., the robot must orientate towards the goal before reaching it.

We can see in Fig. 7 the behaviour of the robot trajectory in a case in which the stablishment time is high. In this example the parameter k_i has been set to 10^{-5} (far from the stability condition values) and then the rotational response is very slow. If the initial and final points defining the path are near (with respect to the system stablishment time) the generated trajectory is strange, but even in this case it approaches to the goal.

Once stated the condition for the stability of the rotational part of the system, the stability of the translational part is easier to be demonstrated. Assuming that the robot is orientated to the goal ($\theta = 0$), the equation for the translational part is ($v = \dot{x}$):

$$\ddot{x} + 2b\dot{x} = f(x) \quad (8)$$

Choosing a force whose module $f(x)$ tends to zero when the robot approaches the goal:

$$f(x) = \begin{cases} K(x_{goal} - x) = K\tilde{x} & \text{if } d < d_0 \\ F & \text{if } d \geq d_0 \end{cases}$$

equation 8 corresponds to a second order stable linear system (if $K > 0$, $b > 0$). Choosing $K = b^2$ (which implies $d_0 = F/b^2$ to keep the continuity of $F(x)$) the system has not overshooting, which assures a smooth approach to the goal. Then we can conclude that the system is asymptotically stable.

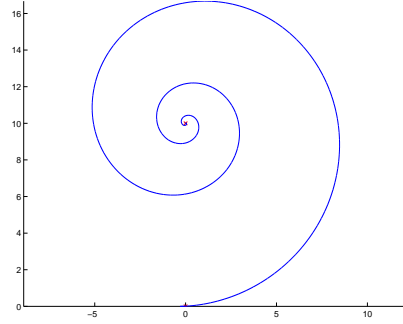


Fig. 7. Trajectory performed with $k_i = 10^{-5}$ from $(0,0)$ to $(0,10)$.

5. PARAMETER SENSITIVITY

In this section the problem of how to set the parameter values for a desired robot behaviour is solved. We analyze the sensitivity of the robot behaviour to each one of the four relevant parameters (b , k_i , F and h). Due to space limitations, let us analyze only one of them: b , the viscous friction coefficient. Since the equations for v and ω are not integrable, the following analysis is based on simulations. The rest of the parameters are fixed to 1. The initial robot location is $(0,0,0)^T$ and the desired final robot location is $(0,10,90^\circ)^T$ (a rotation of 90° is very common in navigation tasks performed in human made indoor environments: door passing, entering into a corridor, etc).

The robot orientation ψ is considered the relevant output. Fig. 8 shows the temporal response of ψ against different values of b . For $b < 0.3$ the robot trajectories are strange (see Fig. 9) but even in these cases the robot is driven to the goal, reinforcing the idea that the system is asymptotically stable for any set of not null parameters. We can state that the value for which the system has the faster response without overshooting is $b \approx 0.975$. This analysis is countersigned by the intuition: growing b implies growing the amount of energy dissipated by friction, and this makes the system *more stable* (in terms of relative stability) but slower. Is this kind of reasoning what makes the model so powerful, and since the model is used also as controller, this allows tuning the parameters even using only the intuition.

Using similar reasoning for the other three parameters, a set of *quasi-optimal* values can be obtained for them: $b = 0.975$, $k_i = 0.95$, $h = 1.05$, and $F = 1.05$. With them there exists an overshoot of $\approx 0.081\%$ on ψ . The trajectory performed using this set of parameters when the robot must turn 90° without stopping is shown in Fig. 10. The only information provided to the controller to compute the motion commands is the initial and final locations and the waypoint. In Fig. 11 the profiles for v and ω are shown. It is noteworthy that both profiles are smooth, having finite accelerations (unlike other approximations in the literature). The maxima accelerations are: $a_{\max} = F = 1.05 \text{ m s}^{-2}$ and

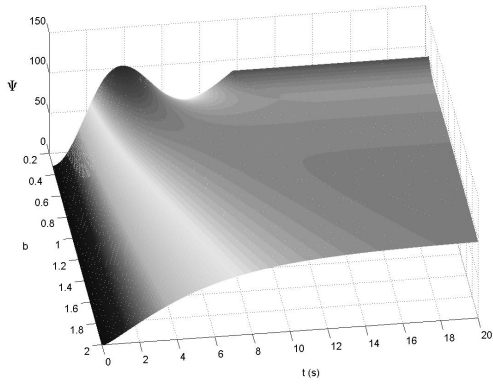


Fig. 8. Temporal response of the robot orientation ψ while b varying in the range $b \in [0.3, 2.0]$.

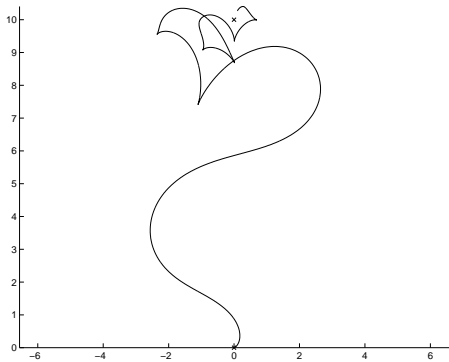


Fig. 9. Trajectory performed with $b = 0.1$.

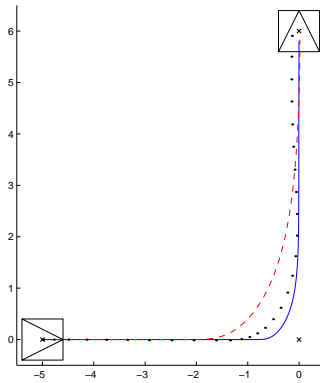


Fig. 10. Trajectories obtained using the *quasi-optimal* set of parameters (solid line), with the modified ones to have $v_{\max} = 1 \text{ ms}^{-1}$ (dashed), and that performed by the real robot (obtained from odometry) for this last set of parameters (dotted).

$\alpha_{\max} = Fk_i h = 1.047 \text{ rad s}^{-2}$. For every sample period, the controller generates the path and velocities needed to have the desired dynamic behaviour (adequate velocities, turning without overshooting), while keeping the accelerations in the admissible range.

Tuning the parameters for some special needs. The maximum linear velocity obtained with the set of quasi-optimal parameters is $v_{\max} = F/2b \approx 0.54 \text{ ms}^{-1}$. If a different linear velocity v_1 is desired, but preserving the rotational behaviour, some parameters of the

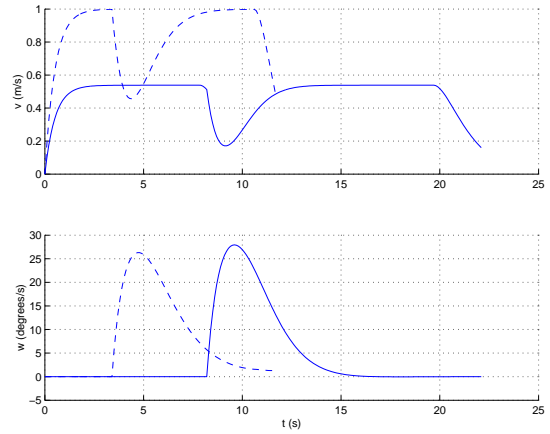


Fig. 11. Linear (v) and angular (ω) velocity profiles with the *quasi-optimal* set of parameters (solid line), and with the modified ones for $v_{\max} = 1 \text{ ms}^{-1}$ (dashed line).

controller must be modified. The easiest way is to keep b unchanged (because changing it implies that k_i and h must be modified, too), and to make $F_1 = Fv_1/v$. Looking at the rotational part of the state space equation (eq. 2), and taking into account that b and k_i are the same as before, it is enough to keep constant the product $Fk_i h$ to maintain the same rotational behaviour. Moreover, since changing F implies changing the maximum linear acceleration of the robot, both the new acceleration and maximum linear velocity must be reachable by the robot. Fig. 10 shows the performed trajectory with the parameters for $v_{\max} = 1 \text{ ms}^{-1}$: $b = 0.975, k_i = 0.95, F_1 = 1.95, h_1 = 0.565$. The maximum acceleration for this case is $a_{\max} = F = 1.95 \text{ ms}^{-2}$. Fig. 11 shows the velocity profiles for this case (note that the rotational behaviour has been kept). Other parameters could be tuned in a similar way to comply with some velocity/acceleration restrictions, being these changes very intuitive.

6. EXPERIMENTAL RESULTS

To test the validity of the proposed approach, the experiment with $v_{\max} = 1 \text{ ms}^{-1}$ has been repeated with a Labmate robot. In Fig. 12 the simulated (solid line) and the real (dotted line) velocity profiles are shown. The real profiles are near to the simulated ones, but initially there is a delay of about 0.45 s in the real case, that is due to the communication through a serial link between the host computer and the platform controller. Furthermore, the linear acceleration requirement for this case ($a_{\max} = 1.95 \text{ ms}^{-2}$) is near the limit the real robot can perform. This implies that the initial response does not exactly match the simulated one. Together with the communication delay, this leads to the profiles appearing delayed. However, even in this case the trajectory –that, as measured from the odometry, corresponds to the dotted line of Fig. 10– is near enough the simulated one, having a error that we consider small for a navigation at the high speed of 1 ms^{-1} .

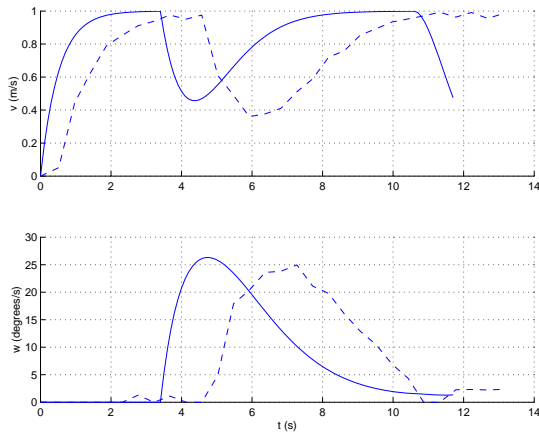


Fig. 12. Velocity profiles for v and ω obtained through simulation (solid) and those of the real robot (dashed), for $v_{\max} = 1 \text{ ms}^{-1}$.

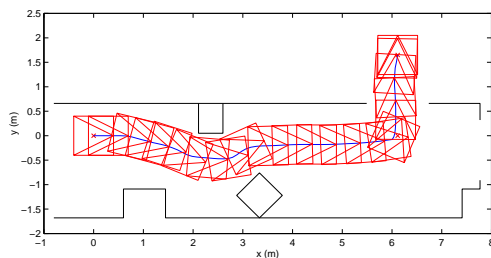


Fig. 13. Trajectory performed by the real robot using the set of quasi-optimal parameters.

Using the controller for reactive navigation. The previous method can work jointly with any navigation method generating a *virtual* force F_A to be applied on the robot. For the experiment shown in Fig. 13 a reactive navigation method based on potential fields has been used (whose details can be seen in (Montano and Asensio, 1997)). In order to perform a safe navigation between the unknown walls and obstacles, the chosen parameters are the quasi-optimal ones, with the exception of F , that has been reduced to 0.525, which gives a $v_{\max} = 0.27 \text{ ms}^{-1}$. Keeping $h = 1.05$ the maximum angular acceleration is also reduced in order to avoid sudden turns in presence of obstacles, being yet enough to correctly avoid them.

The presented controller has been extensively and successfully used, jointly with reactive navigation techniques, in several robots with different dynamic constraints: Labmate (Zaragoza), Hilare 2, Hilare 2bis, and Lama (LAAS, Toulouse) and Nomadic Scout (ISR-IST, Lisboa)(Mínguez and Montano, 2002). This shows the adaptability and good performance of the proposed model.

7. CONCLUSIONS

In this paper a model to control the motion of differential drive mobile robots has been presented. Since the model takes into account the robot kinematic and dynamic constraints, the accelerations and velocities

are compatible with that of the real robot (in particular, the accelerations are always bounded, unlike other models). The most important contribution of the paper is to use the model as a controller, acting as a reference model. This presents several advantages: 1) the controller computes, for every sample period, motion commands which are compatible with the kinematic and dynamic robot constraints; 2) the model has a clear physical sense, being its parameters easily tuned (even intuitively tuned); 3) as reference model, the real parameters of the robot do not need to be identified; 4) the velocity profiles are directly generated by the controller, and no additional computation of velocity setpoints to follow trajectories are needed. Stability and parameter sensitivity analysis have been presented in order to show a method to choose the parameters for the desired behaviour.

The same controller can be used as motion generator (trajectories and velocities) or jointly to a reactive navigation technique for obstacle avoidance. In this case, the only condition is that the reactive technique must compute a virtual force to drive the robot. The controller filters the sudden direction changes of F_A , which is computed every sample period by the navigation method, providing feasible motions compatible with the real robot dynamics.

REFERENCES

- Atherton, D. P. (1975). *Nonlinear Control Engineering*. van Nostrand Reinhold Company.
- Dudek, G. and M. Jenkin (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press.
- Espinosa, F., R. García, M. Mazo, E. López and R. Mateos (1998). Modelling and Simulation of the Kinematic and Dynamic Behaviour of a Fork-Lift-Truck. In: *3rd IFAC Symposium on Intelligent Autonomous Vehicles, IAV'98*. pp. 75–79.
- Fierro, R. and F. L. Lewis (1997). Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics. *Journal of Robotics Systems* **14**(3), 149–163.
- Fox, D., W. Burgard and S. Thrun (1997). The Dynamic Window Approach to Collision Avoidance. *IEEE Transactions on Robotics and Automation*.
- Khatib, M., H. Jaouni, R. Chatila and J. P. Laumond (1997). Dynamic Path Modification for Car-Like Nonholonomic Mobile Robots. In: *IEEE Intl. Conf. on Robotics and Automation*. pp. 2920–2925.
- Mínguez, J. and L. Montano (2002). Robot Navigation in Very Complex Dense and Cluttered Indoor/Outdoor Environments. In: *15th IFAC World Congress*. Barcelona, Spain.
- Montano, L. and J. Asensio (1997). Real-Time Robot Navigation in Unstructured Environments Using a 3D Laser Rangefinder. In: *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*. pp. 526–532.
- Ramírez, G. and S. Zeghloul (2000). A New Local Path Planner for Nonholonomic Mobile Robot Navigation in Cluttered Environments. In: *IEEE Int. Conf. on Robotics and Automation*. pp. 2058–2063.
- Topalov, A. V., D. D. Tsankova, M. G. Petrov and T. Ph. Proychev (1998). Intelligent Sensor-Based Navigation and Control of Mobile Robot in a Partially Known Environment. In: *3rd IFAC Symposium on Intelligent Autonomous Vehicles, IAV'98*. pp. 439–444.
- Yun, X. and Y. Yamamoto (1997). Stability Analysis of the Internal Dynamics of a Wheeled Mobile Robot. *Journal of Robotics Systems* **14**(10), 698–709.