Robotica (2009) volume 27, pp. 981–997. © Cambridge University Press 2009
doi:10.1017/S0263574709005402

Trajectory tracking of mobile robots in dynamic environments—a linear algebra approach

Andrés Rosales*, Gustavo Scaglia, Vicente Mut and Fernando di Sciascio

Instituto de Automática (INAUT), Universidad Nacional de San Juan, Av. Libertador San Martín 1109 (oeste) – J5400ARL, San Juan, Argentina.

(Received in Final Form: January 26, 2008. First published online: February 26, 2009)

SUMMARY

A new approach for navigation of mobile robots in dynamic environments by using Linear Algebra Theory, Numerical Methods, and a modification of the Force Field Method is presented in this paper. The controller design is based on the dynamic model of a unicycle-like nonholonomic mobile robot. Previous studies very often ignore the dynamics of mobile robots and suffer from algorithmic singularities. Simulation and experimentation results confirm the feasibility and the effectiveness of the proposed controller and the advantages of the dynamic model use. By using this new strategy, the robot is able to adapt its behavior at the available knowing level and it can navigate in a safe way, minimizing the tracking error.

KEYWORDS: Collision avoidance; Dynamic model; Force field method; Linear algebra; Mobile robot; Trajectory tracking.

1. Introduction

Mobile robots navigation problem in real environments has been an interesting area for researches in the last years. There are numerous papers focused on solving this matter in different ways. The challenge is to obtain a controller to navigate autonomously in environments with static and dynamic obstacles, especially when these objects have an unknown movement. But, to obtain satisfactory results during its assigned mission, a mobile robot needs to follow, in an exact way, a previously established path or trajectory. If the aforementioned characteristic is achieved with a minimum of error, the mobile robot will be able to avoid collisions and come back to its path or trajectory with a better performance. Traditional approaches as path planning methods are slow to be applied in real time[44] and due to this drawback, new techniques are necessary.

When a mobile robot navigates in a real environment, unexpected static and dynamic obstacles are a big problem for a suitable autonomous navigation, hence a good reactive behavior is a reasonable way to avoid collisions, in this case, the important thing is not the optimality but the safety of the robot's movement. On the other hand, if obstacles have full known movements (or positions), an optimal motion can be generated by using space–time planning. In several applications, it can perform a localization of the environment by means of the construction of a work-area map from robot sensors. These maps are useful to find the free collisions path with static objects as walls or closed doors, etc. However, for mobile objects (e.g. humans, other robots, etc.), it is not an easy task, because the robot must draw dynamic maps with alternative paths or trajectories for the avoidance and hold a desired distance as regards the obstacle. The robot requires a quick dynamic reaction to avoid collisions with all objects instead of depending on static (or dynamic) maps, which are updated very slowly.

Global methods (e.g. path planning algorithms) calculate a whole path to the goal from the actual robot position.[24] If there are mobile obstacles, a common technique entails to add another dimension—the time—at the space state, so that the dynamic problem is reduced to a static one.[14] However, even when global methods supply optimal solutions, their greater disadvantage is assuming a full knowledge of the environment. In practical applications, these techniques are usually combined with local methods to avoid unexpected obstacles.[2,19,20,39,42] Local methods, also called reactive methods, only generate the next control signal; they just use a closer part of the environment and update the world model in accordance with the actual observation of the sensors. Most development techniques, such as Dynamic Window Approach,[12] Curvature Velocity Approach,[30] Curvature Path Approach,[41] and Inevitable Collision States Concept,[13,34] depend on a full knowledge of the environment to get a good performance.

Some researches have developed a great variety of methods based on the potential field concept. Laplace equation was used in ref. [5] to avoid local minimums. To overcome deficiencies in the potential field method, Vector Field Histogram $(VFH)^2$ was developed, it looks for spaces in polar histograms locally built. To protect the robot against collisions, a repulsive field localized in its neighborhood was presented in ref. [28]. Later, the VFH+[47] and the VFH*[46] methods were developed. A fractional potential method was developed in ref. [16], which define attractive and repulsive potentials, taking into account the relative position and velocity of a robot as regards the obstacles and objectives. Wang *et al.*[49] introduced a variable speed force field method for the multirobot collaboration.

* Corresponding author. E-mail: arosales@inaut.unsj.edu.ar

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

The use of path tracking in a navigation system is justified in structured workspaces as well as in partially structured workspaces in which unexpected obstacles can be found during navigation. In the first case, the reference trajectory can be set from a global trajectory planner. In the second case, algorithms used to avoid obstacles re-plan the trajectory in order to avoid a collision; then, a new reference trajectory, which must be followed by the robot, is generated. Furthermore, there are algorithms that express the reference trajectory of the mobile robot as a function of a descriptor called r[7] or s, called "virtual time,"[25] whose derivative is a function of the tracking error and the time t. For example, if the tracking error is large, the reference trajectory should wait for the mobile robot; on the other hand, if the tracking error is small, then the reference trajectory must tend to the original trajectory calculated by the global planner. Accordingly, the module of trajectory tracking will use the original path or the online re-calculated path as reference to obtain the smallest error when the mobile robot follows the path.[32] Consequently, the path tracking is always important independently from whether the reference trajectory has been generated by a global trajectory planner or a local one.

Several studies have been published regarding the design of controllers to guide mobile robots during trajectory tracking. Some of the controllers designed so far are based only on the kinematics of the mobile robot, like the controllers presented in refs. [37, 38]. Other types of tracking controllers based on robot dynamics are presented by Liu *et al.*[26] and Dong & Guo,[9] but the results shown are just based on simulations. In Hwang & Chang[17] a decentralized control by using a mixed robust control is presented. Shuli[40] introduces a controller based on the error model of Kanayama *et al.*[21] As a result, instead of just one controller, two are handed, which are used depending on whether the angular velocity is null or not. In ref. [50] adaptive control via Lyapunov techniques is developed. Kim[22] proposes a receding horizon tracking control for time-varying linear systems with constraints both on the control signal and on the tracking error, based on the minimization of a functional for finite-time costs. Besides, Linear Matrix Inequalities (LMI) are used in order to synthesize the controller. Fukao *et al.*[15] presents the design of an adaptive tracking controller for the dynamic model based on torque with unknown parameters, but only simulation results are presented. Yang & Kim[51] proposed a robust tracking controller for nonholonomic wheeled mobile robots using sliding mode. In ref. [10] fuzzy logic and neural network control approaches were used to deal with the disturbances and dynamic uncertainties. Yang *et al.*[52] introduced an intelligent predictive control approach for the path tracking problem. In ref. [11], the backstepping technique was used to design the adaptive and robust controller for the nonholonomic system. Kanayama *et al.*[21] developed smooth static time invariant state feedback for a velocity-controlled mobile robot with nonholonomic constraint. Most papers about the trajectory tracking have interesting solutions for this theme, but they do not present experimental results or do not work in real environments with the presence of different kinds of obstacles.

The trajectory tracking for mobile robots is characteristically a nonlinear problem. Diverse model-based classic techniques, which propose controllers with a zero-error tracking, have been applied to solve this problem. However, these classic approaches involve an online matrix inversion,[23,48] which represents a drawback in the implementation of the aforementioned methods. In this paper, the proposed algorithm leads the trajectory tracking errors to zero, and it does not involve online matrix inversion problems. Most of the surveys do not present a final expression for the control signals of their controllers,[18,27,45,50] because the computation of these control variables must be made by using demanding computer operations. On the other hand, some current straightforward methods present just simulations).[9,26,53] In this paper, the design of the proposed control law by using Linear Algebra tools is intuitive, and furthermore the final expression for the control signals, which will be directly implemented on the mobile robot, is presented.

The aim of this paper is to use the linear algebra theory and numerical methods to compute control actions, so that, the mobile robot achieves a position (x, y), with a pre-established orientation ψ at each sample time (kTo) without collisions during its navigation. To achieve this objective, in a nonholonomic vehicle, two control variables are available: the linear velocity u and the rotational velocity ω. The proposed controller, based on the dynamic model of a nonholonomic mobile robot, computes the optimal control action (according to least squares[43]), which allows the mobile robot to go from the actual state to the desired one. Furthermore, this paper proposes working with the dynamic model of the mobile robot, because this characteristic allows the controller to face up to sudden velocity changes and to improve the performance of the system.[4] This linear algebra based approach has been tested on other works such as work by Rosales *et al.*[35] and Scaglia *et al.*[36–38]

In this work, simulations and experimental results have been applied to a PIONEER 3DX mobile robot. The efficacy and feasibility are then demonstrated in a practical sense through a set of experiments where the speed-range is similar to the one reported in other papers about trajectory tracking using laboratory equipment.[31]

The paper is organized as follows: Section 2 describes the dynamic model of the mobile robot. Section 3 presents the methodology to compute the linear algebra based controller. Modified force field method is described in Section 4. Section 5 presents simulations and experimental results using the proposed controller on a PIONEER 3DX mobile robot. Conclusions are detailed at Section 6. Finally, the Appendix introduces the stability analysis of the proposed controllers.

2. Dynamic Model of A Mobile Robot

To perform tasks with requirements of high speed and/or transport of heavy loads, it is very important to consider the dynamics of the mobile robot, because such tasks exert very large external forces on the robot and they will inevitably influence its path and direction, hence, a kinematic model is not sufficient. Dynamic characteristics of the robot such as mass and inertia center change if the robot is loaded. Previous studies very often ignore the dynamics of mobile robots and

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

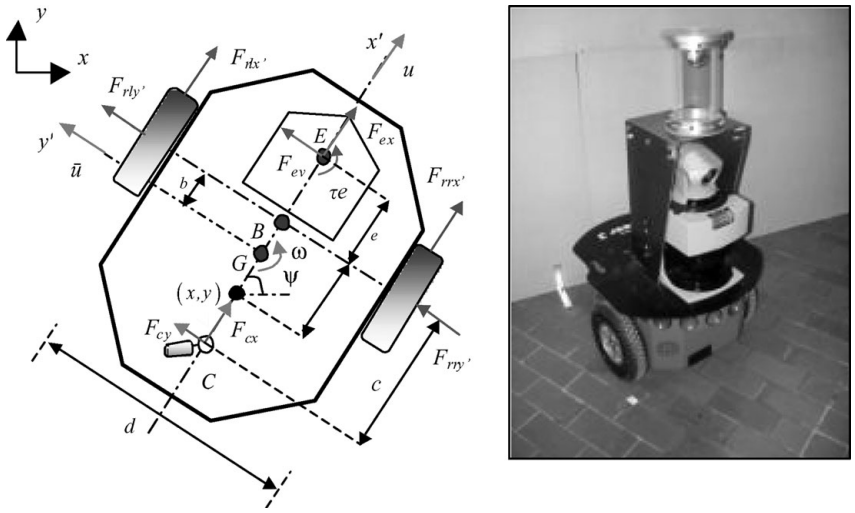


Fig. 1. PIONEER 3DX mobile robot, (a) model; (b) laboratory equipment.

also suffer from algorithmic singularity. A nonholonomic dynamic model of a unicycle-like mobile robot[6] is shown in Fig. 1a and it is presented in Eq. (1).

The robot position is defined by (x, y); this point is located at a distance a from rear axis center of the robot, u and $\bar{u}$ are the longitudinal and side speeds of mass center, ω is the angular speed and ψ is the orientation angle, G is the gravity center and B is the base line center of the wheels, $F_{rrx'}$ and $F_{rry'}$ are the longitudinal and lateral tyre forces of the right wheel, $F_{rlx'}$ and $F_{rly'}$ are the longitudinal and lateral tyre forces of the left wheel, $F_{cx'}$ and $F_{cy'}$ are the longitudinal and lateral forces exerted on C by the castor, $F_{ex'}$ and $F_{ey'}$ are the longitudinal and lateral forces exerted on E by the tool (e.g. a robotic arm), b, c, d, and e are distances, and τ_e is the moment exerted by the tool.

From the diagram in Fig. 1a, the dynamic model of the mobile robot is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} u\cos\psi - a\omega\sin\psi \\ u\sin\psi + a\omega\cos\psi \\ \omega \\ \dfrac{\theta_3}{\theta_1}\omega^2 - \dfrac{\theta_4}{\theta_1}u \\ -\dfrac{\theta_5}{\theta_2}u\omega - \dfrac{\theta_6}{\theta_2}\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{1}{\theta_1} & 0 \\ 0 & \dfrac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} uc \\ \omega c \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix}. \quad (1)$$

The identified parameters $\boldsymbol{\theta}$ of the dynamic model[6] are

$$\theta_1 = \left(\frac{R_a}{k_a}(mR_t r + 2I_e) + 2rk_{DT}\right) \Big/ (2rk_{PT}) = 0.24089;$$

$$\theta_2 = \left(\frac{R_a}{k_a}(I_e d^2 + 2R_t r(I_z + mb^2)) + 2rdk_{DR}\right) \Big/ (2rdk_{PR}) = 0.2424;$$

$$\theta_3 = \frac{R_a}{k_a} mbR_t / (2k_{PT}) = -0.00093603;$$

$$\theta_4^0 = \frac{R_a}{k_a}\left(\frac{k_a k_b}{R_a} + B_e\right) \Big/ (rk_{PT}) + 1 = 0.99629;$$

$$\theta_5 = \frac{R_a}{k_a} mbR_t \Big/ (dk_{PR}) = -0.0037256;$$

$$\theta_6 = \frac{R_a}{k_a}\left(\frac{k_a k_b}{R_a} + B_e\right) d \Big/ (2rk_{PR}) + 1 = 1.0915,$$

where m is the robot mass; r is the right and left wheel radius; k_b is equal to the voltage constant multiplied by the gear ratio; R_a is the electric resistance constant; k_a is the torque constant multiplied by the gear ratio; k_{PR}, k_{PT}, and k_{DT} are positive constants; I_e and B_e are the moment of inertia and the viscous friction coefficient of the combined motor rotor, gearbox, and wheel; and R_t is the nominal radius of the tire.[6]

The elements of the uncertainty vector $\boldsymbol{\delta}$, related to the mobile robot are $\boldsymbol{\delta} = [\,\delta_x \;\; \delta_y \;\; 0 \;\; \delta_u \;\; \delta_\omega\,]^T$, where δ_x and δ_y depend on velocities due to wheels slide and robot orientation; δ_u and δ_ω depend on mechanic parameters of the robot, such as mass, inertial moment, wheel diameter, engine and servo-controllers parameters, and forces on the wheels. All these parameters are considered as disturbances.

Remark 1. *If the slip speed of the wheels, the forces and torques exerted by the tool, and the forces exerted by the castor wheel are of no significant value, the uncertainties vector* $\boldsymbol{\delta}$ *will not be considered.*

In general, most market-available robots have low-level PID velocity controllers to track input reference velocities and do not allow the motor voltage to be driven directly.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

Therefore, it is useful to express the mobile robot model in a suitable way by considering rotational and translational reference velocities as control signals.

3. Linear Algebra Based Controller

The controller proposed in this work is based on the linear algebra theory and numerical methods.[37,38] By knowledge of the desired state at the next sample time, it is possible to compute the necessary control actions so that the mobile robot tracks the reference trajectory with a good performance. We assume that the mobile robot is moving on a horizontal plane without slip.

3.1. Euler approximation

First, through Euler approximations* of the dynamic model of mobile robot (1), the following set of equations is obtained,

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \psi_{k+1} \\ u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \psi_k \\ u_k \\ \omega_k \end{bmatrix} + To \left\{ \begin{bmatrix} u_k \cos\psi_k - a\omega_k \sin\psi_k \\ u_k \sin\psi_k + a\omega_k \cos\psi_k \\ \omega_k \\ \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} uc_k \\ \omega c_k \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} \right\}, \tag{2}$$

where values of x at the discrete time $t = kTo$, will be denoted as x_k; To is the sample time; and $k = 0, 1, 2, \ldots$. Afterwards, the state vector $\mathbf{x}_{k+1}$ is replaced by the desired state vector

$$\mathbf{x}d_{k+1} = [xd_{k+1} \quad yd_{k+1} \quad \psi d_{k+1} \quad ud_{k+1} \quad \omega d_{k+1}]^T.$$

Remark 2. *The use of numerical methods to compute the systems evolution is based mainly on the possibility to determine the system state at instant $k+1$, if the state is known at instant k.† So, a variable at instant $k+1$ can be substituted for the desired one and subsequently computing the necessary control action to make the states of the system move from its current value to a desired one.*

From (2), the following system of linear equations is created:

$$\mathbf{A}\boldsymbol{\mu}_k = \mathbf{b} \tag{3}$$

where

$$\boldsymbol{\mu}_k = [uc_k \quad \omega c_k]^T \tag{4}$$

$$\mathbf{b} = \begin{bmatrix} xd_{k+1} - x_k \\ yd_{k+1} - y_k \\ \psi d_{k+1} - \psi_k \\ ud_{k+1} - u_k \\ \omega d_{k+1} - \omega_k \end{bmatrix} - To \begin{bmatrix} u_k \cos\psi_k - a\omega_k \sin\psi_k \\ u_k \sin\psi_k + a\omega_k \cos\psi_k \\ \omega_k \\ \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} - To \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} \tag{5}$$

$$\mathbf{A} = To \begin{bmatrix} 0 & 0 & 0 & \frac{1}{\theta_1} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\theta_2} \end{bmatrix}^T. \tag{6}$$

From (3), which is a set of five equations with two unknown variables, and by using normal equations‡ ($\mathbf{A}^T\mathbf{A}\boldsymbol{\mu}_k = \mathbf{A}^T\mathbf{b}$), the optimal solution (according to least squares§)[43] for $\boldsymbol{\mu}_k$ is found,

$$\begin{bmatrix} uc_k \\ \omega c_k \end{bmatrix} = \begin{bmatrix} \frac{\theta_1 (ud_{k+1} - u_k) - To\left(\theta_3\omega_k^2 - \theta_4 u_k + \theta_1\delta_u\right)}{To} \\ \frac{\theta_2 (\omega d_{k+1} - \omega_k) - To\left(-\theta_5 u_k\omega_k - \theta_6\omega_k + \theta_2\delta_\omega\right)}{To} \end{bmatrix} \tag{7}$$

where ud_{k+1} and ωd_{k+1} are linear and rotational desired velocities, respectively.

3.2. Column space criterion

Now, the objective is to find ud_{k+1} and ωd_{k+1}, so that the tracking error will be minimal. To fulfill this condition the system in (3) should have an exact solution. Thus, vector $\mathbf{b}$ must belong to Column space¶ of matrix $\mathbf{A}$, that is, vector $\mathbf{b}$ must be a linear combination of column vectors of matrix

* Euler approximations are a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value.

† Having the Markov property means that, given the present state, future states are independent of the past states. In other words, the description of the present state fully captures all the information that could influence the future evolution of the process.

‡ Given an overdetermined matrix equation $\mathbf{Ax} = \mathbf{b}$, the normal equation is that which minimizes the sum of the square differences between left and right sides $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. Here, $\mathbf{A}^T\mathbf{A}$ is a normal matrix, that is, $\mathbf{A}^T\mathbf{A} - \mathbf{A}\mathbf{A}^T = \mathbf{0}$.

§ The least squares solution to an inconsistent system $\mathbf{Ax} = \mathbf{b}$ satisfies $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. If the columns of $\mathbf{A}$ are linearly independent $\mathbf{A}^T\mathbf{A}$ is invertible and $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$.

¶ In linear algebra, the column space of a matrix is the set of all possible linear combinations of its column vectors. The column space of an $m \times n$ matrix is a subspace of m-dimensional Euclidean space.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

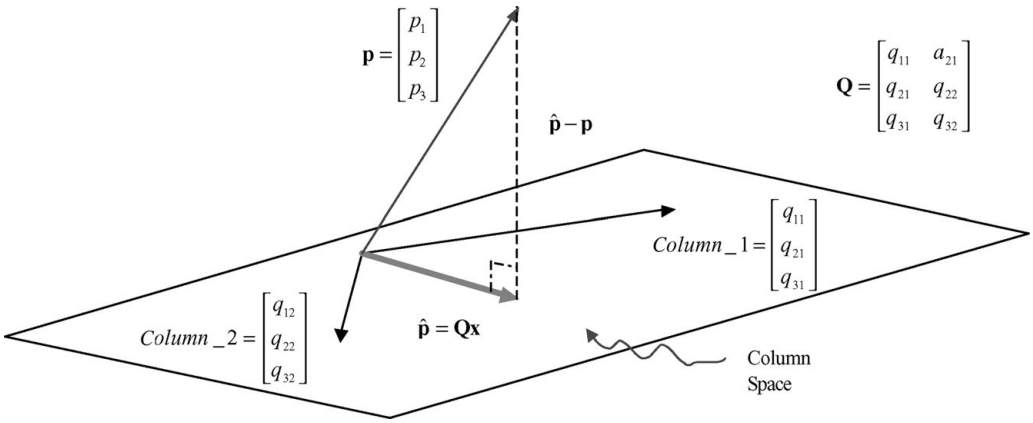


Fig. 2. Example of a projection of a matrix 3×2 on the column space.

$\mathbf{A}$[43]; a column space base of the matrix $\mathbf{A}$ is

$$C(\mathbf{A}) = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \overline{\theta_1} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \overline{\theta_2} \end{bmatrix} \right\} = \{\mathbf{v}_1, \mathbf{v}_2\}. \tag{8}$$

System (3) must have an exact solution, hence, it must satisfy $\mathbf{b} = \varsigma_1 \mathbf{v}_1 + \varsigma_2 \mathbf{v}_2$ (linear combination of $\{\mathbf{v}_1, \mathbf{v}_2\}$), where, ς_1, $\varsigma_2 \in \mathbb{R}$, that is,

$$\mathbf{b} = \varsigma_1 \mathbf{v}_1 + \varsigma_2 \mathbf{v}_2 = \begin{bmatrix} 0 & 0 & 0 & \dfrac{\varsigma_1}{\theta_1} & \dfrac{\varsigma_2}{\theta_2} \end{bmatrix}^T. \tag{9}$$

To visualize in a better way the previous concept about column space, Fig. 2 presents an example of a projection of a vector 3×1 on the column space, where, $\mathbf{Q}\mathbf{x} = \mathbf{p}$ is an inconsistent system, that is, vector $\mathbf{p}$ is not a combination of columns of the matrix $\mathbf{Q}$; $\hat{\mathbf{p}}$ is the vector $\mathbf{p}$ projection on the column space of the matrix $\mathbf{Q}$ and it is the nearest point to $\mathbf{p}$ in this space; and $\|\hat{\mathbf{p}} - \mathbf{p}\|$ is just the distance from $\mathbf{p}$ to the point $\mathbf{Q}\mathbf{x}$ in the column space of $\mathbf{Q}$, that is, the error. The error vector $\|\hat{\mathbf{p}} - \mathbf{p}\|$ is perpendicular to the column space (Fig. 2). The norm is defined by $\|\mathbf{z}\|_{\mathbf{P}}^2 = \mathbf{z}^T\mathbf{P}\mathbf{z}$ with $\mathbf{P} > 0$.

The optimal solution (according to least squares) of this system satisfies $\mathbf{Q}^T\mathbf{Q}\mathbf{x} = \mathbf{Q}^T\mathbf{p}$ (normal equations), therefore, $\hat{\mathbf{p}} = \mathbf{Q}\mathbf{x} = \mathbf{Q}^\dagger\mathbf{x}$, where $\mathbf{Q}^\dagger$ is the pseudo-inverse matrix of $\mathbf{Q}$. Then, if the columns of $\mathbf{Q}$ are linearly independent: $\mathbf{Q}^\dagger = (\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T$.

On the way back to our controller, and from (5) and (9) the relations in (10) can be established,

$$\begin{cases} xd_{k+1} - x_k = To u_k \cos\psi_k - To a\omega_k \sin\psi_k + To\delta_x \\ yd_{k+1} - y_k = To u_k \sin\psi_k + To a\omega_k \cos\psi_k + To\delta_y \\ \psi d_{k+1} - \psi_k = To\omega_k \end{cases}. \tag{10}$$

From (10) and taking the Remark 1 into account, the following system of linear equations is created:

$$\mathbf{B}\boldsymbol{\upsilon}_k = \mathbf{c} \tag{11}$$

where

$$\boldsymbol{\upsilon}_k = [u_k \quad \omega_k]^T \tag{12}$$

$$\mathbf{c} = [xd_{k+1} - x_k \quad yd_{k+1} - y_k \quad \psi d_{k+1} - \psi_k]^T \tag{13}$$

$$\mathbf{B} = To \begin{bmatrix} \cos\psi_k & -a\sin\psi_k \\ \sin\psi_k & a\cos\psi_k \\ 0 & 1 \end{bmatrix}. \tag{14}$$

From (11), which is a set of three equations with two unknown variables and again, by using normal equations ($\mathbf{B}^T\mathbf{B}\boldsymbol{\upsilon}_k = \mathbf{B}^T\mathbf{c}$), the optimal solution (according to least squares) for $\boldsymbol{\upsilon}_k$ is found.

The optimal solution for u_k and ω_k represents the linear and rotational velocities that the mobile robot must have at instant k, so that system (3) has an exact solution, therefore, the use of these values as the desired velocities ud_{k+1} and ωd_{k+1} is justified, that is,

$$\begin{bmatrix} ud_{k+1} \\ \omega d_{k+1} \end{bmatrix} = \frac{1}{To} \begin{bmatrix} k_u(\Delta x \cos\psi_k + \Delta y \sin\psi_k) \\ k_\omega \left(\dfrac{-a\Delta x \sin\psi_k + a\Delta y \cos\psi_k + \Delta\psi}{a^2 + 1} \right) \end{bmatrix},$$

$$\text{where} \begin{cases} \Delta x = xd_{k+1} - x_k \\ \Delta y = yd_{k+1} - y_k \\ \Delta\psi = \psi d_{k+1} - \psi_k \end{cases} \tag{15}$$

where k_u and k_ω are positive constants that allow us adjusting the performance of the proposed control system; they satisfy $0 < k_u < 1$, and $0 < k_\omega < 1$, allowing to reduce the variations in state variables.[37]

The proposed controller design is completely finished by using the relations given in (15), which will be used to generate the control signals in (7).

3.3. Trajectory tracking performance

To show the outstanding performance of this proposed approach, two experimental results are presented next. The first reference trajectory was a circle defined by $x_r =$

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

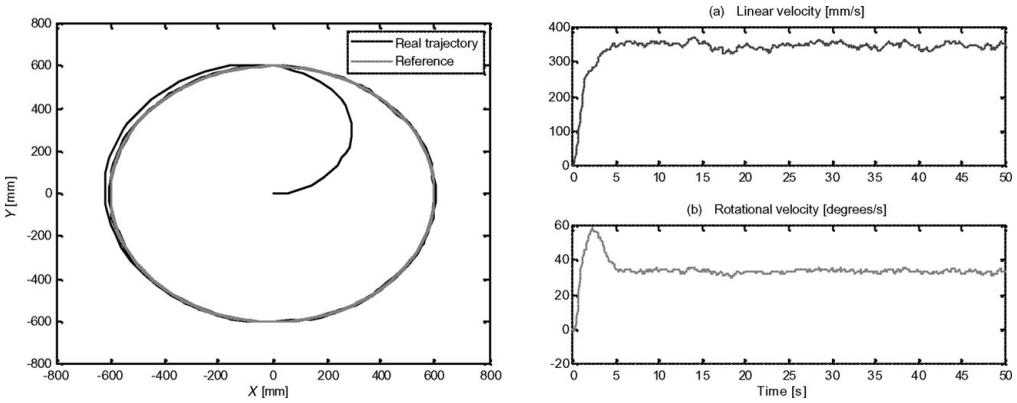


Fig. 3. (a) Circular reference trajectory tracking; (b) Velocity profiles (1) linear (2) rotational.

$r\cos(\omega_r t)$ and $y_r = r\sin(\omega_r t)$, where $r \in \mathbb{R}$ is the circle radius and ω_r is the rotational velocity in reference trajectory. For this test $r = 600$ [mm], $u_{ref} = 400$ [mm/s], and $\omega_r = 38.2$ [degrees/s]. The sample time was $To = 0.1$ [s], and initial conditions used in this example were $\mathbf{x}_0 = [0\ 0\ 0\ 0\ 0]^T$ [mm].

The second reference trajectory was an eight-shaped curve defined by $x_r = r\sin(\omega_r t)$ and $y_r = r\cos(0.5\omega_r t)$, where $r \in \mathbb{R}$. For this test $r = 800$ [mm], $u_{ref} = 300$ [mm/s] and $\omega_r = 21.49$ [degrees/s]. Also the sample time was $To = 0.1$ [s] and the initial conditions were $\mathbf{x}_0 = [500\ 0\ 0\ 0\ 0]^T$ [mm].

Figures 3 and 4 show velocity profiles of the mobile robot for a circle and an eight-shaped trajectory, respectively. The mobile robot follows the desired trajectory with a maximum error (absolute value of the difference between the desired and the real trajectories, once the mobile robot has reached the geometric predefined path) of 20 [mm] in the circular trajectory and 60 [mm] in the eight-shaped one (in the sharp turning parts). These errors are very small compared to the distance between wheel axes (400 [mm]). If a comparison between our experimental results and results recently published is made,[8] which presents an algorithm based on the dynamic model of the mobile robot showing simulation results, we can conclude that the control system proposed in this paper presents a similar performance, working at the same range of speeds.

Results corroborate the goodness of the proposed controller; they show that the mobile robot follows the reference trajectory precisely. The next section will present how this reference trajectory is modified to avoid collisions.

4. Modified Force Field Method

In the last year, the force field method, based on the generation of a virtual potential field, has been much used for path planning of autonomous mobile robots because it has efficiency and mathematical simplicity. The basic concept in this approach is to cover the work environment with an artificial force, which causes the robot to be attracted by the goal and rejected by obstacles.[3]

4.1. Fictitious force

The modified force field method, proposed in this work, generates a virtual force field that continually changes depending on the distance to the obstacle, and the real and maximal velocities of the mobile robot. This modified fictitious force is defined as follows:

$$\left|\vec{F}_k\right| = \begin{cases} 0 & \text{if} \quad \text{dist}_k > D_k \\ \dfrac{c_f \cdot \eta}{|d_k - \text{dist}_k|} & \text{if} \quad \text{dist}_k \leq D_k \end{cases} \tag{16}$$

where dist_k is the distance between the robot position (x_k, y_k) and the obstacle position (xb_k, yb_k); D_k defines a repulsive zone, where the collision avoidance strategy is active; d_k represents the minimum distance without contact of the

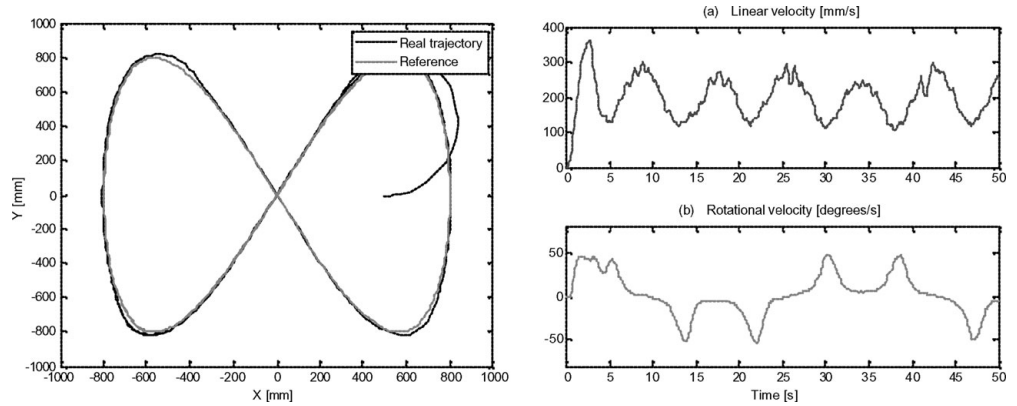


Fig. 4. (a) Eight-shaped reference trajectory tracking; (b) Velocity profiles (1) linear (2) rotational.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

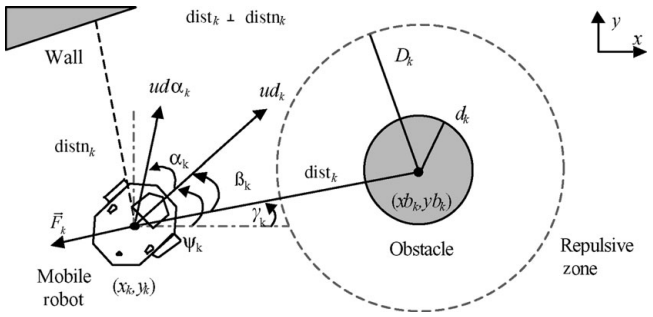


Fig. 5. Modified force field method diagram.

mobile robot with the obstacle, also, this distance takes the robot and obstacle radius into account; c_f is a positive constant that determines the force field coverage; and η is defined as

$$\eta = \rho \frac{u_k}{u_{\max}}, \quad 0 < \rho \leq 1 \tag{17}$$

where u_k and $u_{\max}$ are real and maximal velocities of the mobile robot, respectively. Figure 5 shows a diagram of the force field method, where ψ_k is the orientation angle of the mobile robot; γ_k is the angle between the obstacle and the robot; β_k is an angle that shows the direction of the obstacle with respect to the robot (consequently, this angle also provides the direction to deflect the trajectory); α_k is the rotation angle due to virtual force; dist_k is the distance to the obstacle, and distn_k is a distance in the normal direction to dist_k, which indicates the presence of other obstacles in a perpendicular direction to the principal obstacle to be avoided, ud_k and $ud\alpha_k$ are the desired velocity vector and the modified one, respectively; and (xb_k, yb_k) is the position of the obstacle.

Remark 3. *The following assumptions are given on the robot and the obstacles:*

A1) The mobile robot measures the distance dist_k *between the robot and the obstacle and the relative angle* β_k *with a sampling time To.*

A2) The controlled variables of the robot are u_k *and* ω_k*, and they are constrained as* $|u_k| \leq |u_{\max}|$ *and* $|\omega_k| \leq |\omega_{\max}|$*, where* $u_{\max}$ *and* $\omega_{\max}$ *are the maximum translational and rotational velocities, respectively.*

A3) The mobile obstacles are moving with a smooth trajectory and their maximal velocity is limited by the 80% of the maximal velocity of the mobile robot.

Assumptions (A1) and (A2) are held for distance sensors and actuators such as ultrasonic sonars, infrared sensors, and electric motors. Assumption (A3) is a constraint given in this paper.

4.2. Velocity reduction criterion

The components of the desired velocity vector are modified as follows:

$$\begin{bmatrix} \frac{xd\alpha_{k+1} - xd\alpha_k}{To} \\ \frac{yd\alpha_{k+1} - yd\alpha_k}{To} \end{bmatrix} = \sigma_k \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix} \begin{bmatrix} \frac{xd_{k+1} - xd_k}{To} \\ \frac{yd_{k+1} - yd_k}{To} \end{bmatrix}. \tag{18}$$

The parameter $0 < \sigma_k < 1$ is used to reduce the magnitude of the modified desired velocity vector, when the mobile robot avoids an obstacle. Moreover, when $|\vec{F}_k| = 0$, $\sigma_k = 1$ (normal trajectory tracking).

The rotation angle α_k is computed as follows:

$$\alpha_k = |\vec{F}_k| \text{sign}(\sin(\beta_k)). \tag{19}$$

4.3. Obstacle detection

To know if an obstacle is static or dynamic, the modified force field method computes the velocity of this object, by using a simple comparison between the actual obstacle position and its position at the previous sample time. Appling the same criterion, we compute the steering of the dynamic obstacle.

Once the obstacle has been detected and its parameters have been computed, we are able to determine if this obstacle is static or dynamic. Normally, complex algorithms can be used to compute the velocity and direction of a dynamic obstacle, however in this work; we are interested in the use of the simplest approach by using (20).

$$\left.\begin{aligned} ubx_k &= \frac{xb_{k+1} - xb_k}{To} \\ uby_k &= \frac{yb_{k+1} - yb_k}{To} \end{aligned}\right\} \Rightarrow \begin{cases} ub_k = \sqrt{ubx_k^2 + uby_k^2} \\ \psi b_k = \arctan\left(\frac{uby_k}{ubx_k}\right) \end{cases}, \tag{20}$$

where ubx_k and uby_k are the Cartesian components of the obstacle. Furthermore, if the obstacle is dynamic, the distance between the mobile robot and the obstacle is modified as follows:

$$\text{dpred}_k = \sqrt{((xb_k + ubx_k Tp) - (x_k + ux_k Tp))^2 + ((yb_k + uby_k Tp) - (y_k + uy_k Tp))^2} \tag{21}$$

where dpred_k is the predicted distance, and Tp is the predicted time. Note that for static obstacles, ubx_k and uby_k are null, so that, $\text{dpred}_k = \text{dist}_k$ (see Fig. 5).

Figure 6 shows the case, where the mobile robot tracks a straight line with a static obstacle and a dynamic one. When the mobile robot detects the first obstacle (static), it turns to the left until avoiding the object, and then it returns to its trajectory, after this event, the mobile robot detects another obstacle, but now this obstacle is dynamic, therefore, the mobile robot goes round the back of this object. In both cases, the desired velocity is reduced by using the σ_k parameter. Reference velocity was $u_{ref} = 400$ [mm/s].

So that, when the obstacle is dynamic, the modified force field method causes the mobile robot to avoid this nonstatic obstacle moving toward the direction opposite to the object movement, that is, the mobile robot goes round the back of the

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

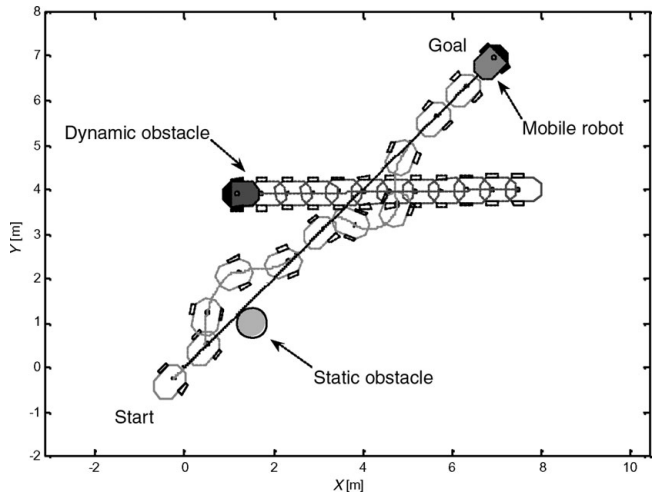


Fig. 6. Collision avoidance (static and dynamic objects) by using the modified force field method (experimental data).

dynamic obstacle (see Fig. 6), so that, the collision avoidance is safe for the mobile robot. To achieve this movement change, the sign of α_k (see Fig. 5) is modified.

4.4. End of the obstacle avoidance

To finish the influence of the modified force field method on the mobile robot, that is, to know when the mobile robot has avoided an obstacle, the cosine theorem* has been chosen. Figure 7 shows a diagram for this criterion.

Consider the triangle $P_r\ P_d\ P_o$, where P_r is the initial position of the mobile robot; P_d is the desired position at instant $k+1$; and, P_o is the obstacle position; φ is the opposite angle to the $P_r P_d$ line, that is, it will be the angle which indicates whether the obstacle was avoided or not. When $\pi/2 < \varphi < 3\pi/2$, $\cos(\varphi)$ is negative, that means the mobile robot still does not avoid the obstacle (Fig. 7a), and when $-\pi/2 < \varphi < \pi/2$, $\cos(\varphi)$ is positive, so the mobile robot has already avoided the obstacle (Fig. 7b).

A general diagram of obstacle motion detection algorithm is shown in Fig. 8.

To show how the proposed parameters affect the force field method, a simple case is presented in Fig. 9: the mobile robot must follow a straight trajectory from the position (0.0, 0.0) until the position (5.5, 5.5), where an obstacle has been placed at position (2.5, 2.5). The used parameters are presented in Table I, for this experiment a maximal speed $u_{\max} = 600$ [mm/s] and $\sigma_k = 0.2$ have been used.

The force field method has been modified in this work to improve its performance. Firstly, this modified technique detects obstacles in the work environment; then, the method momentarily modifies the velocity of the mobile robot and its desired trajectory, so that, obstacles can be avoided effectively and safely.

To prevent local minimum problems (such as a large U-shaped obstacle),[29] the modified force field method simply uses a normal distance distn_k to dist_k (see Fig. 5), so that, when $\text{distn}_k \leq D_k$, the algorithm adds $\pm\pi/2$ [rad] to the α_k angle, depending on the orientation of this normal distance, that is, when an obstacle is in front of the mobile robot and another object is on its right, the algorithm adds $+\pi/2$ [rad] to the α_k angle, otherwise, if the another object is on its left, the algorithm adds $-\pi/2$ [rad] to the α_k angle.

Table I. Comparison of different parameters for the modified force field method..

	c_f	ρ	**Cover distance [m]**	**Collision**
1	0.99	0.467	8.630	No
2	0.99	0.167	8.484	No
3	**0.85**	**0.167**	**8.078**	**No**
4	0.85	0.937	8.243	No
5	0.23	0.167	7.716	Yes

An example of this situation is presented in Fig. 10: the mobile robot must follow a straight trajectory from the position (−2.0, −2.0) [m], and it must avoid a V-shaped obstacle placed in the environment.

Figure 10a shows a tracked trajectory for the mobile robot; when the obstacle has been detected ($\text{dist}_k \leq D_k$) the speed is reduced and the desired trajectory is modified (by using (18)), then due to the V-shape of the obstacle the normal distance distn_k is also activated ($\text{distn}_k \leq D_k$), so that, the mobile robot turns $-\pi/2$ [rad] in addition to the α_k angle. Figure 10b presents the velocity profiles for the current experiment. When the V-shaped obstacle causes the activation of the force field method, the mobile robot reduces its speed, then the desired trajectory overtakes the mobile robot, hence when the collision has been already avoided, the mobile robot must accelerate (until its maximum velocity) to reach again the desired trajectory.

Remark 4. *In the force field method, a robot is assumed to travel in a 2D environment and its location can be precisely known. A force field is defined as a virtual field of repulsive force in the vicinity of a robot when it travels in a working space. The magnitude and orientation of a force field are determined by and vary with the robot status. This virtual repulsive force increases with the decrease of the distance to the robot (by using (16)).*

5. Simulations and Experimental Results

5.1. Simulation results

The computer simulations were generated by MobileSim* software for debugging and experimentation with ARIA* or other software that supports mobile robots platforms. The simulated robots are based on the physical mobile robots; therefore the same algorithms were used both in real experiments and simulations. All the graphical results presented in this work were generated by using Matlab† software.

The simulated reference trajectory was a circle of 7000 [mm] radius with a linear reference speed $u_{\text{ref}} = 390$ [mm/s] and a rotational reference speed $\omega_{\text{ref}} = 3.19$ [degrees/s]. The

* Cosine theorem is a statement about a general triangle which relates the lengths of its sides to the cosine of one of its angles. Let a triangle ABC, with sides a, b, c, and angles α, β, γ; c is the side opposite of angle γ, and a and b are the two sides enclosing γ. Using this notation, the cosine theorems states that: $c^2 = a^2 + b^2 - 2ab\cos(\gamma)$.

* http://robots.mobilerobots.com/
† http://www.mathworks.com

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

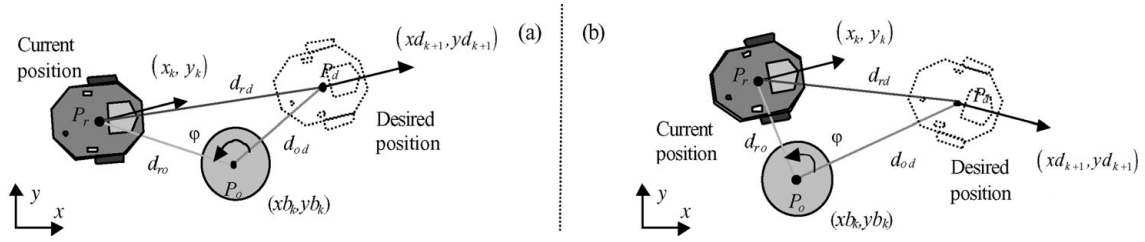


Fig. 7. (a) Modified force field method with $\cos(\varphi) < 0$ ($\pi/2 < \varphi < 3\pi/2$) (b) Modified force field method with $\cos(\varphi) > 0$ ($-\pi/2 < \varphi < \pi/2$).

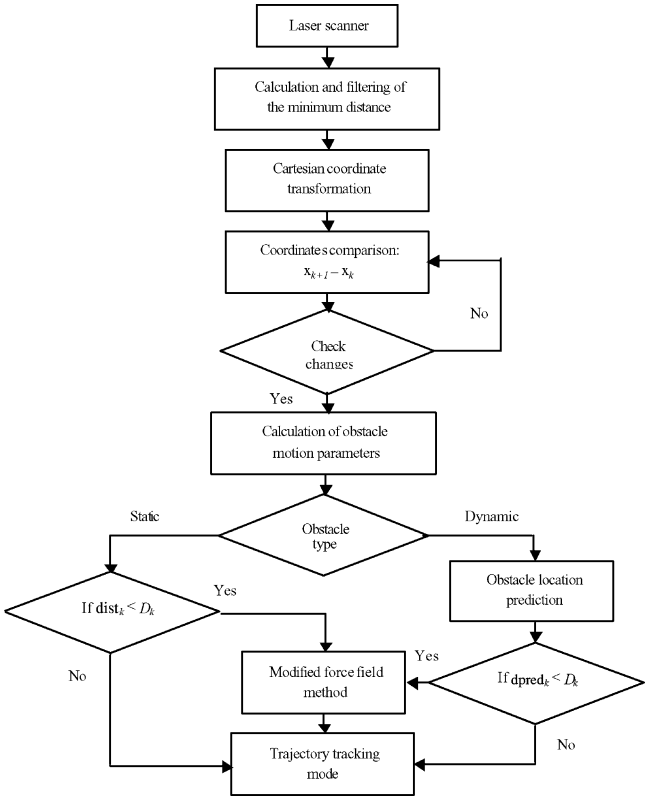


Fig. 8. Diagram of obstacle motion detection.

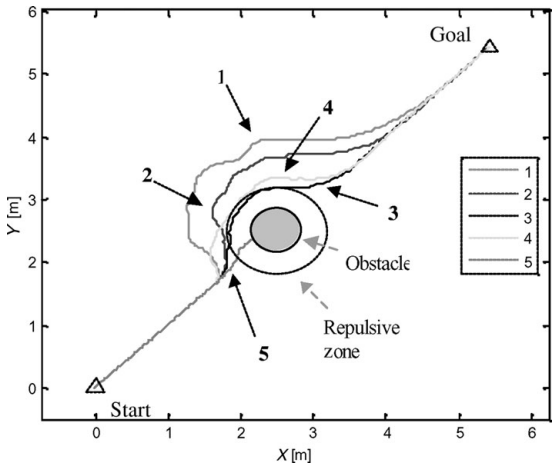


Fig. 9. Parameters influence on the modified force field method (experimental data).

initial conditions were $\mathbf{x}_0 = [-250\ [\text{mm}]\ -250\ [\text{mm}]\ 0\ [\text{degree}]\ 0\ [\text{mm/s}]\ 0\ [\text{deg/sec}]]^T$.

Figure 11 shows the mobile robot follows in a precise way the desired circular trajectory. At the beginning, the robot detects a static obstacle on the right and avoids it, and then it returns to the reference trajectory. Next, three dynamic obstacles, which are moving with different smooth trajectories, are avoided during the tracking. At the end two different obstacles (one static and one dynamic) are detected and avoided. The whole test was developed with a minimum tracking error.

5.2. Experimental results

The proposed methodology has been tested with PIONEER 3DX mobile robots (with approx. 400 [mm] of radius) equipped with a SICK LMS200 laser sensor (see Fig. 12).

The PIONEER mobile robot includes an estimation system based on odometry, which adds accumulative errors to the system.[33] From this, data updating through external sensors are necessary. This problem is separated from the strategy of trajectory tracking and collision avoidance, and it is not considered in this paper.[31] The real test environment for PIONEER robots includes doors, halls, and unknown obstacles (e.g. another mobile robot). In this environment the mobile robot follows a specific pre-established trajectory autonomously. For a reactive avoidance of obstacles, we have taken $d_k = 300$ [mm] and $D_k = 700$ [mm]. Data acquisition and all necessary algorithms are computed in real time with the on-board robot computer. A sample time $To = 0.1$ [s] was used and $a = 200$ [mm] (see Fig. 1).

The first experimental test was performed through halls and corridors. Figure 13 shows the experimental environment map with real (continuous line) and desired (dotted line) trajectories of the mobile robot; thick lines correspond to different obstacles detected online (walls, boxes, other robots, etc.), which are incrementally added into the map, as the mobile robot moves forward.

The mobile robot was able to achieve the desired trajectory with a very good performance minimizing the tracking error.

A comparison between the desired and real trajectories of the mobile robot is presented in Fig. 14. In this figure the places where the mobile robot avoids the obstacles have also been represented. There are three static obstacles (boxes) and two dynamic ones (mobile robots).

Remark 5. *The real trajectory is represented by a continuous line (black) and the desired trajectory by a dotted one (dark green). It is important to remark that the absolute value of the difference between the desired and the real*

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

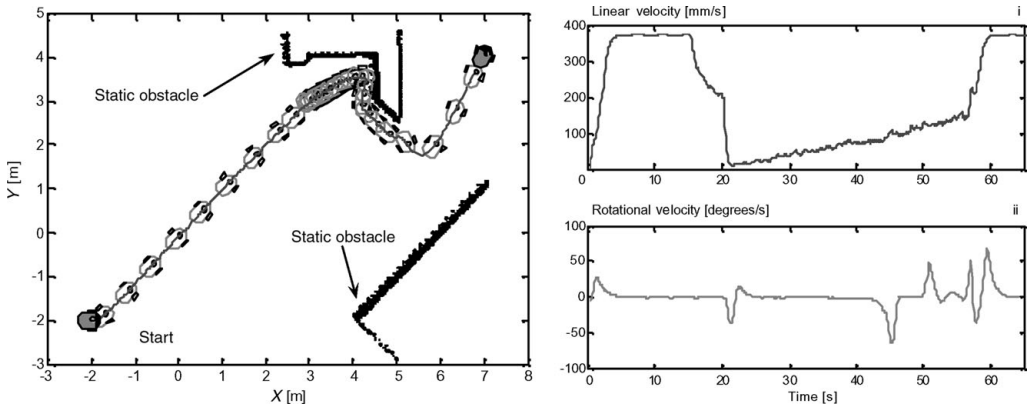


Fig. 10. (a) Collision avoidance (V-shaped obstacle) by using the modified force field method (experimental data); (b) Translational and rotational velocity profiles.

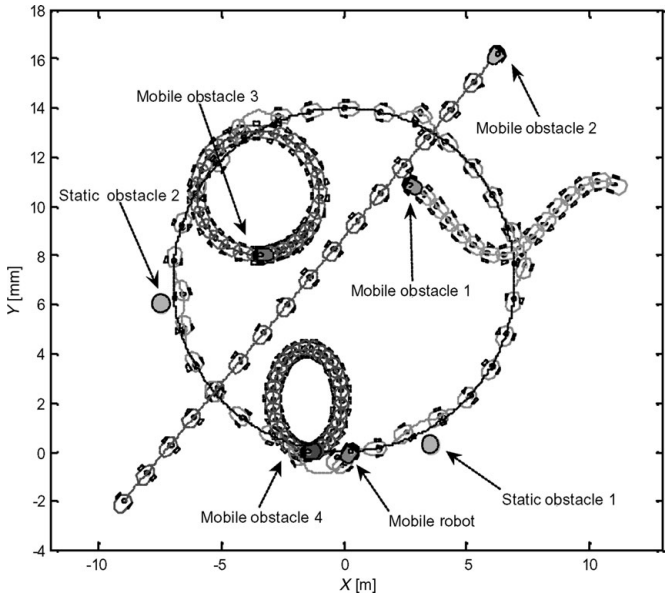


Fig. 11. Simulation of a circular trajectory tracking with static and mobile obstacles.

trajectory, once the mobile robot has reached the geometric pre-defined path, will be called error.

In Fig. 14, it can also be noticed that, when the trajectory direction suddenly changes (e.g. unexpected obstacles presence), the tracking error increases, nevertheless it decreases afterwards, with a maximum value of about 87 [mm] when the modified force field method does not act, that is, when the mobile robot tracks the desired trajectory without the obstacles presence. Besides, the error is not too large when compared with the size of the PIONEER robot, considering the demanding desired trajectory chosen. This trajectory type is used to test the performance of the system, because it is a *worst-case* situation, where the error is acceptable since it is smaller than half the distance between the axes of the mobile robot (400 [mm]).

Linear and rotational velocity profiles are presented in Figs. 15a and 15b, respectively. Linear velocity is limited to $|u_{\max}| = 600$ [mm/s] and rotational velocity to $|\omega_{\max}| = 60$ [degrees/s].

The mobile robot gets a good trajectory tracking. When it detects an obstacle, the modified force field method comes into action: the mobile robot reduces its velocity until avoiding the obstacle, and after that increases its speed to come back to the original desired trajectory. The reference trajectory was generated with a constant linear velocity of $u_{\text{ref}} = 300$ [mm/s].

A set of experiences in this environment was carried out at different reference velocities and a summary of these tests is presented in Table II, the most representative results of the experimental tests were shown in the previous figures.

Remark 6. *The maximum trajectory tracking error value is computed when the modified force field method does not act, that is, when the mobile robot tracks the desired trajectory without the obstacles presence. When an obstacle is detected by the sensors, the reference trajectory is modified, therefore during the avoidance time the trajectory tracking error is not evaluated.*



Fig. 12. PIONEER mobile robots and their environment.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

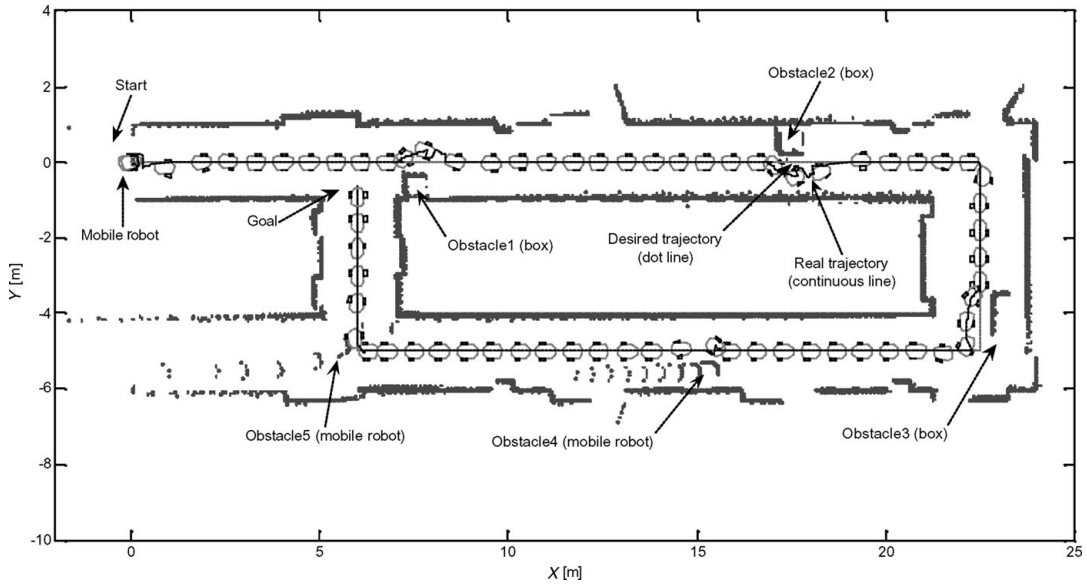


Fig. 13. Experimental environment with real (continuous line) and desired (dotted line) trajectories of the mobile robot.

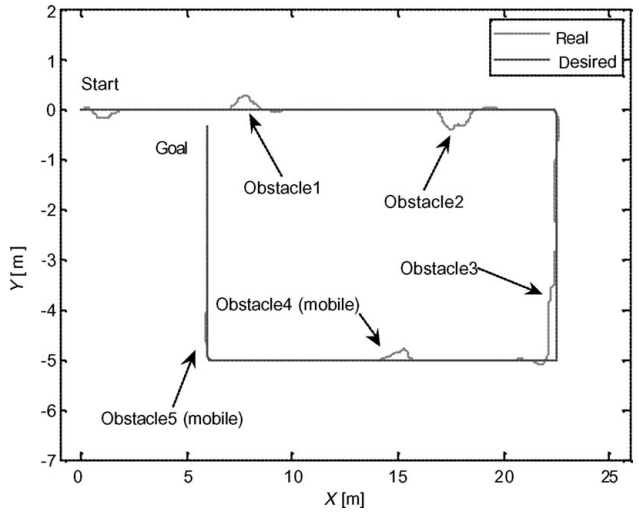


Fig. 14. Real (red continuous line) and desired (blue dotted line) trajectories.

The second experiment was performed in an office environment. Figure 16 shows the experimental environment map with real (continuous line) and desired (dotted line) trajectories of the mobile robot; thick lines correspond to different obstacles detected online (walls, boxes, people, etc.), which are incrementally added into the map, as the mobile robot moves forward.

Table II. Summary of the tracking errors in the trajectory for experimental tests..

u_{ref} (mm/s)	Maximum error (trajectory tracking) (mm)	Maximum linear velocity (mm/s)	Maximum rotational velocity [degrees/s]
100	36	257.6	44.1
200	52	374	42.34
300	87	597.2	−55.59

When the mobile robot finds an obstacle, its speed is reduced until the collision has been avoided. For this test the σ_k parameter (see (18)) was smaller than that in the previous experiment. Linear and rotational velocity profiles are presented in Figs. 17a and 17b, respectively. Linear velocity is limited to $|u_{\max}| = 400$ [mm/s] and rotational velocity to $|\omega_{\max}| = 60$ [degrees/s]. The reference trajectory was generated with a constant linear velocity of $u_{\text{ref}} = 320$ [mm/s].

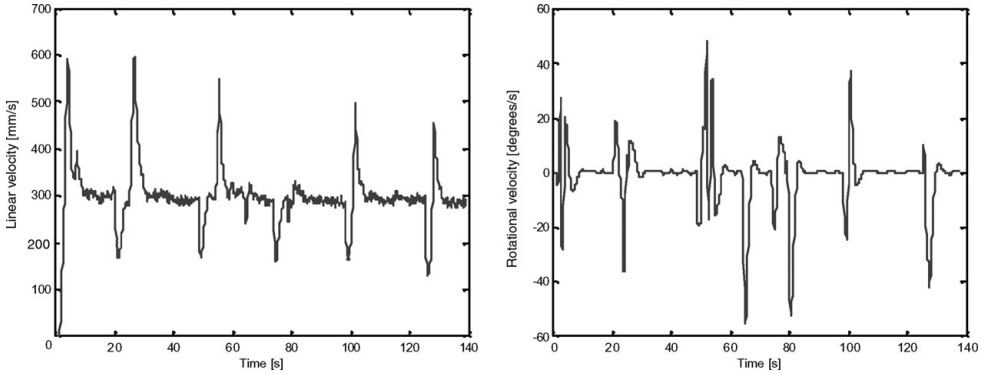


Fig. 15. First experiment (a) linear velocity profile; (b) rotational velocity profile.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

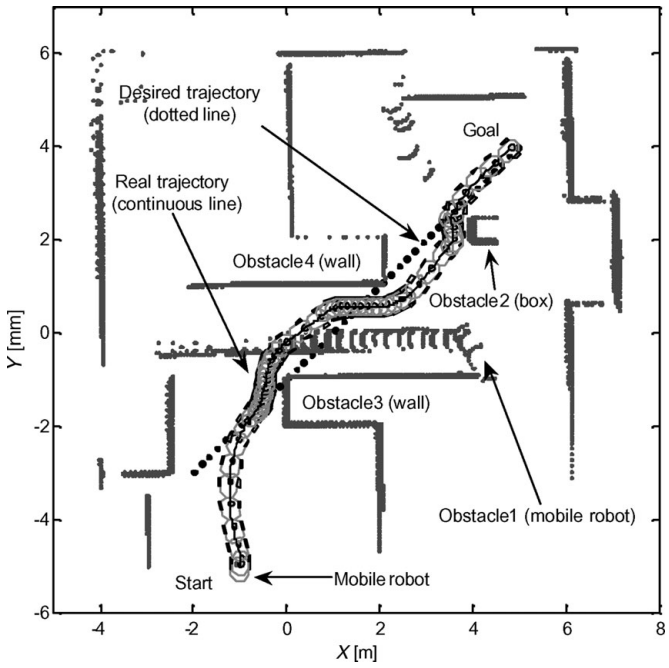


Fig. 16. Experimental environment with real (continuous line) and desired (dotted line) trajectories of the mobile robot.

It can be observed that the proposed control system is dependent on the precision and accuracy of the sensor system; however, it is independent from the sensor method used. This relies on the fact that not only intern sensors (odometry), but also extern sensors (laser) can be used, depending on the application, complexity, or the problem to be solved.

Remark 7. *Note that in the focused trajectory tracking problem of this paper, the reference never waits for the mobile robot, on the contrary the reference advances as the time passes. There are other methods to stop the reference while the mobile robot is approaching),*[7,25] *but in this paper, the most demanding situation has been analyzed, because the mobile robot has to reach the reference as soon as possible without colliding with the obstacles.*

Remark 8. *One important aspect to clarify the behavior of the mobile robot during the experiments is that in a trajectory tracking, unlike in a path tracking, the desired point does not wait for the mobile robot, as a trajectory is a time-parameterized path. The mobile robot tries to reach the pre-established trajectory while the desired point is moving. The mobile robot's velocity and the trajectory are on a par, when the desired trajectory is reached by the mobile robot. If an obstacle causes the activation of the force field method, the mobile robot reduces its speed, then the trajectory overtakes the mobile robot, hence when the collision has been already avoided, the mobile robot must accelerate (until its maximum velocity) to reach again the desired trajectory (see Figs. 10, 15, and 17). In spite of this challenging assignment, the performance of the mobile robot during the experiments is successful.*

6. Conclusions

In this work, a linear algebra based approach to the trajectory tracking of mobile robots in dynamic environments by using a modified force field method has been presented. This controller allows free collision navigation with a good tracking trajectory/path performance. The proposed approach combines the linear algebra based controller with a reactive strategy for the collision avoidance. Experimental results show the combination of these techniques produces a straightforward and effective controller for mobile robots navigation in unknown environments with dynamic and static obstacles. An appealing characteristic of this controller is its simple implementation in any programming language.

In this paper, the proposed algorithm leads the trajectory-tracking errors to zero, and it does not involve online matrix inversion problems. Moreover, the design of the proposed control law by using linear algebra tools is intuitive, and the final expression for the control signals, which will be directly implemented on the mobile robot, is presented.

Simulation and experimental results of the developed controllers on a PIONEER 3DX mobile robot have also been addressed; these tests show the feasibility of the developed algorithms. Through the analysis of the experiments, it can be concluded that the trajectory error between the desired and the real trajectories of the mobile robot is very small. In addition, a stability analysis of the proposed controllers has been presented.

In order to classify and develop our work properly, we have supposed the model of the mobile robot is a good approximation of the real system and we have considered that the uncertainties are small enough to be insignificant. Simulation and experimental results have guaranteed both hypotheses.

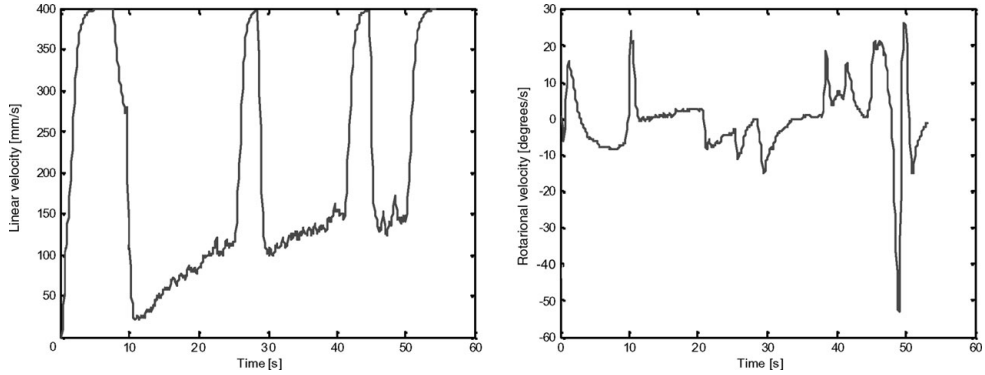


Fig. 17. Second experiment (a) linear velocity profile; (b) rotational velocity profile.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

The proposed methodology for the controller design can be applied to other types of systems. The required precision of the proposed numerical method for the system approximation is smaller than the one needed to simulate the behavior of the system. This is because, when the states for the feedback are available, in each sampling time, any difference from accumulative errors is corrected (e.g. rounding errors). Thus, the approach is used to find the best way to go from one state to the next one, according to the availability of the system model.

Appendix: Stability Analysis

This appendix provides the stability analysis of the proposed controller. The used criterions to demonstrate the convergence of the proposed algorithms are mainly based on ref. [1].

Condition for zero-error convergence: In order to analyze the stability of the proposed controller, the system (11)–(14) will be used again. By analyzing this system the required conditions for a zero-error convergence will be provided.

$$\underbrace{\begin{bmatrix}\cos(\psi_k) & -a\sin(\psi_k)\\ \sin(\psi_k) & a\cos(\psi_k)\\ 0 & 1\end{bmatrix}}_{\mathbf{B}}\underbrace{\begin{bmatrix}u_k\\ \omega_k\end{bmatrix}}_{\upsilon}=\underbrace{\frac{1}{To}\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta\psi\end{bmatrix}}_{\mathbf{c}}, \tag{A1}$$

where $\Delta x = xd_{k+1} - x_k$, $\Delta y = yd_{k+1} - y_k$, and $\Delta\psi = \psi d_{k+1} - \psi_k$.

The desired orientation for the mobile robot must guarantee that nonholonomic system reaches the pre-established trajectory with a zero-error convergence. To compute this required orientation, the column space criterion will be again used. Subsequently, system (A1) must have an exact solution. To fulfil this condition, vector $\mathbf{c}$ is obliged to belong to column space of matrix $\mathbf{B}$, that is, vector $\mathbf{c}$ has to be a linear combination of column vectors of matrix $\mathbf{B}$.[43]

A column space base of the matrix $\mathbf{B}$, $C(\mathbf{B})$ is given by

$$C(\mathbf{B}) = \left\{\begin{bmatrix}\cos(\psi_k)\\ \sin(\psi_k)\\ 0\end{bmatrix}, \begin{bmatrix}-a\sin(\psi_k)\\ a\cos(\psi_k)\\ 1\end{bmatrix}\right\}. \tag{A2}$$

The orthonormal column space base of the matrix $\mathbf{B}$, $C^{\perp}(\mathbf{B})$ is given by

$$C^{\perp}(\mathbf{B}) = \left\{\begin{bmatrix}\cos(\psi_k)\\ \sin(\psi_k)\\ 0\end{bmatrix}, \frac{1}{\sqrt{a^2+1}}\begin{bmatrix}-a\sin(\psi_k)\\ a\cos(\psi_k)\\ 1\end{bmatrix}\right\} = \{\mathbf{w}_1, \mathbf{w}_2\}. \tag{A3}$$

Vector $\mathbf{c}$ of (A1) is composed of two parts: the component in the column space $\mathbf{c}_{C(\mathbf{B})}$ and the component in the left null space $\mathbf{c}_{N(\mathbf{B}^T)}$ of the matrix $\mathbf{B}$, that is

$$\mathbf{c} = \mathbf{c}_{C(\mathbf{B})} + \mathbf{c}_{N(\mathbf{B}^T)} \tag{A4}$$

$$\mathbf{c}_{C(\mathbf{B})} = (\mathbf{c}^T\mathbf{w}_1)\mathbf{w}_1 + (\mathbf{c}^T\mathbf{w}_2)\mathbf{w}_2 \tag{A5}$$

$$\mathbf{c}_{C(\mathbf{B})} = \frac{1}{To}\begin{bmatrix}\Delta x\cos^2(\psi_k) + \Delta y\sin(\psi_k)\cos(\psi_k)\\ \Delta x\sin(\psi_k)\cos(\psi_k) + \Delta y\sin^2(\psi_k)\\ 0\end{bmatrix} + \frac{1}{To(at^2+1)} \times \begin{bmatrix}a^2\Delta x\sin^2(\psi_k) - a^2\Delta y\sin(\psi_{(k)})\cos(\psi_k) - a\Delta\psi\sin(\psi_k)\\ -a^2\Delta x\sin(\psi_k)\cos(\psi_k) + a^2\Delta y\cos^2(\psi_k) + a\Delta\psi\cos(\psi_k)\\ -a\Delta x\sin(\psi_k) + a\Delta y\cos(\psi_k) + \Delta\psi\end{bmatrix}. \tag{A6}$$

For an exact solution, $\mathbf{c}$ must be equal to $\mathbf{c}_{C(\mathbf{B})}$, that is

$$\mathbf{c}_{C(\mathbf{B})} = \frac{1}{To}[\Delta x \quad \Delta y \quad \Delta\psi]^T. \tag{A7}$$

Operating and simplifying some terms in the above equation, the following condition is obtained:

$$\begin{bmatrix}\sin(\psi_k)(\Delta x\sin(\psi_k) - \Delta y\cos(\psi_k) + a\Delta\psi)\\ \cos(\psi_k)(-\Delta x\sin(\psi_k) + \Delta y\cos(\psi_k) - a\Delta\psi)\\ a\Delta\psi + \Delta x\sin(\psi_k) - \Delta y\cos(\psi_k)\end{bmatrix} = \begin{bmatrix}0\\ 0\\ 0\end{bmatrix}. \tag{A8}$$

That is,

$$\Delta x\sin(\psi_k) - \Delta y\cos(\psi_k) + a\Delta\psi = 0. \tag{A9}$$

From (A9), the required angle to the error converges to zero can be obtained,

$$\psi ez_{k+1} = \psi_k - \frac{\Delta x\sin(\psi_k) - \Delta y\cos(\psi_k)}{a}, \tag{A10}$$

where ψez_k is the angle for a zero-error convergence.

From (A10), the following relation is obtained; this expression will be used for future computations:

$$a(\psi ez_{k+1} - \psi_k) = -\Delta x\sin(\psi_k) + \Delta y\cos(\psi_k). \tag{A11}$$

Assumption 1. Because the reference velocities u_{ref} and ω_{ref} satisfy (2), the mobile robot can follow a pre-established trajectory. Moreover, the reference trajectory also fulfils (A1), that is

$$\begin{aligned} x_{\text{ref}k+1} &= x_{\text{ref}k} + u_{\text{ref}k}To\cos(\psi_{\text{ref}_k}) - \omega_{\text{ref}k}aTo\sin(\psi_{\text{ref}k})\\ y_{\text{ref}k+1} &= y_{\text{ref}k} + u_{\text{ref}k}To\sin(\psi_{\text{ref}k}) + \omega_{\text{ref}k}aTo\cos(\psi_{\text{ref}k})\\ \psi_{\text{ref}k+1} &= \psi_{\text{ref}k} + To\omega_{\text{ref}k} \end{aligned} \tag{A12}$$

where variables x_{ref}, y_{ref}, ψ_{ref}, u_{ref}, and ω_{ref} are the reference state variables that composed the pre-established trajectory.

Assumption 2. So that the error tends to zero, the following relations for the desired variables are proposed:

$$\begin{aligned} xd_{k+1} &= x_{\text{ref}\,k+1} - k_x\underbrace{(x_{\text{ref}\,k} - x_k)}_{\tilde{x}_k}\ yd_{k+1}\\ &= y_{\text{ref}\,k+1} - k_y\underbrace{(y_{\text{ref}\,k} - y_k)}_{\tilde{y}_k}\ \psi d_{k+1}\\ &= \psi_{\text{ref}k+1} - k_\psi\underbrace{(\psi_{\text{ref}\,k} - \psi_k)}_{\tilde{\psi}_k} \end{aligned} \tag{A13}$$

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

where $\tilde{x}$, $\tilde{y}$, and $\tilde{\psi}$ are the errors in x, y, and ψ, respectively; k_x, k_y, and k_ψ are constants of weight for $\tilde{x}$, $\tilde{y}$, and $\tilde{\psi}$, respectively. Also, $0 < (k_x, k_y, k_\psi) < 1$.

Additionally, to achieve the zero-error convergence condition the reference orientation angle $\psi_{\text{ref}k}$ in (A13) will be replaced with the zero-error convergence angle ψe_k, that is,

$$\psi d_{k+1} = \psi e_{k+1} - k_\psi \underbrace{(\psi e_k - \psi_k)}_{\tilde{\psi}_k}. \tag{A14}$$

Assumption 3. Translational and rotational real velocities will be equal to the linear and rotational desired velocities, respectively, plus the particular error for each case, that is,

$$\begin{aligned} u_k &= ud_k + \tilde{u}_k \\ \omega_k &= \omega d_k + \tilde{\omega}_k \end{aligned} \tag{A15}$$

where $\tilde{u}$ and $\tilde{\omega}$ are the errors in u and ω, respectively.

Analysis for $\tilde{\psi}$: From (A14), and by using $\Delta\psi = \psi d_{k+1} - \psi_k$

$$\Delta\psi = (\psi e_{k+1} - \psi_k) - k_\psi \tilde{\psi}_k \tag{A16}$$

From (10), (15), and (A15),

$$\begin{aligned} \psi_{k+1} &= \psi_k + To\omega_k = \psi_k \\ &+ \frac{-a\Delta x \sin(\psi_k) + a\Delta y \cos(\psi_k) + \Delta\psi}{a^2+1} + To\tilde{\omega}_k \end{aligned} \tag{A17}$$

From (A16) and (A17), and by using (A11),

$$\begin{aligned} \psi_{k+1} &= \psi_k \\ &+ \frac{a^2(\psi e_{k+1} - \psi_k) + (\psi e_{k+1} - \psi_k) - k_\psi \tilde{\psi}_k}{a^2+1} + To\tilde{\omega}_k \\ \psi_{k+1} &= \psi e_{k+1} - \frac{1}{a^2+1} k_\psi \tilde{\psi}_k + To\tilde{\omega}_k \\ &\underbrace{(\psi e_{k+1} - \psi_{k+1})}_{\tilde{\psi}_{k+1}} - \frac{1}{a^2+1} k_\psi \tilde{\psi}_k + To\tilde{\omega}_k = 0 \end{aligned}$$

$$\tilde{\psi}_{k+1} - \frac{1}{a^2+1} k_\psi \tilde{\psi}_k + To\tilde{\omega}_k = 0 \tag{A18}$$

By the analysis of (A18), it can be concluded that if $\tilde{\omega}_k \to 0$ with $k \to \infty$, then $\tilde{\psi}_k \to 0$ with $k \to \infty$. Therefore, to demonstrate the stability of ψ, the stability of ω also must be verified.

Before verifying the stability of ω, an analysis for x and y will be carried out.

Analysis for x and y: First the state x will be analyzed.

From (10), (15), and (A15),

$$\begin{aligned} x_{k+1} &= x_k + [\Delta x \cos^2(\psi_k) + \Delta y \sin(\psi_k)\cos(\psi_k)] \\ &+ To\cos(\psi_k)\tilde{u}_k - \frac{\sin(\psi_k)}{a^2+1} \\ &\times [-a^2\Delta x \sin(\psi_k) + a^2\Delta y \cos(\psi_k) + a\Delta\psi] \\ &- aTo\sin(\psi_k)\tilde{\omega}_k \end{aligned} \tag{A19}$$

Replacing (A11) in (A19)

$$\begin{aligned} x_{k+1} &= x_k + [\Delta x \cos^2(\psi_k) + \Delta y \sin(\psi_k)\cos(\psi_k)] \\ &+ To\cos(\psi_k)\tilde{u}_k - aTo\sin(\psi_k)\tilde{\omega}_k \\ &- \frac{\sin(\psi_k)}{a^2+1}[-a^2\Delta x \sin(\psi_k) + a^2\Delta y\cos(\psi_k) \\ &+ a(\psi ez_{k+1} - \psi_k) - ak_\psi\tilde{\psi}_k] \end{aligned}$$

Again, by using (A11) in the above equation

$$\begin{aligned} x_{k+1} &= x_k + \Delta x + \frac{a}{a^2+1}\sin(\psi_k)k_\psi\tilde{\psi}_k \\ &+ To\cos(\psi_k)\tilde{u}_k - aTo\sin(\psi_k)\tilde{\omega}_k \end{aligned}$$

Replacing the expression $xd_{k+1} = x_k + \Delta x$ in the previous equation

$$\begin{aligned} x_{k+1} &= xd_{k+1} + \frac{a}{a^2+1}\sin(\psi_k)k_\psi\tilde{\psi}_k + To\cos(\psi_k)\tilde{u}_k \\ &- aTo\sin(\psi_k)\tilde{\omega}_k. \end{aligned} \tag{A20}$$

Finally, replacing (A13) in (A20) the expression for the x-error is obtained,

$$\begin{aligned} x_{k+1} &= x_{\text{ref}\,k+1} - k_x\tilde{x}_k + \frac{a}{a^2+1}\sin(\psi_k)k_\psi\tilde{\psi}_k \\ &+ To\cos(\psi_k)\tilde{u}_k - aTo\sin(\psi_k)\tilde{\omega}_k \\ &(x_{\text{ref}\,k+1} - x_{k+1}) - k_x\tilde{x}_k + \frac{a}{a^2+1}\sin(\psi_k)k_\psi\tilde{\psi}_k \\ &+ To\cos(\psi_k)\tilde{u}_k - aTo\sin(\psi_k)\tilde{\omega}_k = 0 \end{aligned}$$

So,

$$\begin{aligned} &\tilde{x}_{k+1} - k_x\tilde{x}_k \\ &\underbrace{+ \frac{a}{a^2+1}\sin(\psi_k)k_\psi\tilde{\psi}_k + To\cos(\psi_k)\tilde{u}_k - aTo\sin(\psi_k)\tilde{\omega}_k}_{\text{nonlinearity}} = 0 \end{aligned} \tag{A21}$$

If a similar analysis is carried out for the state y, an analogous expression to (A21) will be obtain, that is

$$\begin{aligned} &\tilde{y}_{k+1} - k_y\tilde{y}_k \\ &\underbrace{- \frac{a}{a^2+1}\cos(\psi_k)k_\psi\tilde{\psi}_k + To\sin(\psi_k)\tilde{u}_k + aTo\cos(\psi_k)\tilde{\omega}_k}_{\text{nonlinearity}} = 0. \end{aligned} \tag{A22}$$

So that, from (A21) and (A22),

$$\begin{aligned} \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{y}_{k+1} \end{bmatrix} &= \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \end{bmatrix} \\ &\underbrace{+ \frac{a}{a^2+1}\begin{bmatrix} -\sin(\psi_k) \\ \cos(\psi_k) \end{bmatrix} k_\psi\tilde{\psi}_k + To\begin{bmatrix} \cos(\psi_k) & -a\sin(\psi_k) \\ \sin(\psi_k) & a\cos(\psi_k) \end{bmatrix}\begin{bmatrix} \tilde{u}_k \\ \tilde{\omega}_k \end{bmatrix}}_{\text{nonlinearity}}. \end{aligned} \tag{A23}$$

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

By analyzing (A23), it can be concluded that if the nonlinearity behavior is bounded and $\tilde{u}_k \to 0$ and $\tilde{\omega}_k \to 0$ with $k \to \infty$, then $\tilde{x}_k \to 0$ and $\tilde{y}_k \to 0$ with $k \to \infty$. Therefore, like in the case of ψ, the stability of x and y also depends on the stability of u and ω.

Analysis for u and ω: From (2), the states u and ω at instant $k+1$, will be given by

$$\begin{bmatrix} u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ \omega_k \end{bmatrix} + To \begin{bmatrix} \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} + To \begin{bmatrix} \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} uc_k \\ \omega c_k \end{bmatrix} \tag{A24}$$

Replacing the controller expression (7) in (A24)

$$\begin{bmatrix} u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ \omega_k \end{bmatrix} + To \begin{bmatrix} \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} + \begin{bmatrix} \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \times \begin{bmatrix} \theta_1 u d_{k+1} - (\theta_1 - To\theta_4)u_k - To\theta_3\omega_k^2 \\ \theta_2\omega d_{k+1} - (\theta_2 - T_0\theta_6)\omega_k + T_0\theta_5 u_k\omega_k \end{bmatrix}$$

$$\begin{bmatrix} u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ \omega_k \end{bmatrix} + To \begin{bmatrix} \frac{\theta_3}{\theta_1}\omega_k^2 - \frac{\theta_4}{\theta_1}u_k \\ -\frac{\theta_5}{\theta_2}u_k\omega_k - \frac{\theta_6}{\theta_2}\omega_k \end{bmatrix} + \begin{bmatrix} ud_{k+1} - \left(1 - To\frac{\theta_4}{\theta_1}\right)u_k - To\frac{\theta_3}{\theta_1}\omega_k^2 \\ \omega d_{k+1} - \left(1 - To\frac{\theta_6}{\theta_2}\right)\omega_k + To\frac{\theta_5}{\theta_2}u_k\omega_k \end{bmatrix}$$

So,

$$\begin{bmatrix} u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} ud_{k+1} \\ \omega d_{k+1} \end{bmatrix} \tag{A25}$$

Assumption 4. So that the error tends to zero, the following relations for the desired variables are proposed:

$$\begin{aligned} ud_{k+1} &= u_{\text{ref}\,k+1} - k_u \underbrace{(u_{\text{ref}\,k} - u_k)}_{\tilde{u}_k} \\ \omega d_{k+1} &= \omega_{\text{ref}\,k+1} - k_\omega \underbrace{(\omega_{\text{ref}\,k} - \omega_k)}_{\tilde{\omega}_k}, \end{aligned} \tag{A26}$$

where k_u and k_ω are constants of weight for $\tilde{u}$ and $\tilde{\omega}$, respectively. Also, $0 < (k_u,\, k_\omega) < 1$. Replacing (A26) in (A25)

$$\begin{bmatrix} u_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} u_{\text{ref}\,k+1} - k_u\tilde{u}_k \\ \omega_{\text{ref}\,k+1} - k_\omega\tilde{\omega}_k \end{bmatrix}.$$

So,

$$\begin{bmatrix} \tilde{u}_{k+1} - k_u\tilde{u}_k \\ \tilde{\omega}_{k+1} - k_\omega\tilde{\omega}_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{A27}$$

Therefore, from (A27), it is concluded $\tilde{u}_k \to 0$ ands $\tilde{\omega}_k \to 0$ with $k \to \infty$. Now, the behavior of the nonlinearity in (A23) for x and y will be analyzed.

Analysis of the nonlinearity in x and y: Equation (A23) can be written as

$$\mathbf{v}_{k+1} = \mathbf{A}\mathbf{v}_k + \mathbf{P}_k \tag{A28}$$

where

$$\mathbf{v}_k = \begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \end{bmatrix},\ \mathbf{A} = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}$$

$$\mathbf{P}_k = \frac{a}{a^2+1}\begin{bmatrix} -\sin(\psi_k) \\ \cos(\psi_k) \end{bmatrix} k_\psi\tilde{\psi}_k = \begin{bmatrix} P1_k \\ P2_k \end{bmatrix}\tilde{\psi}_k \text{ and}$$

$$\mathbf{Q}_k = To\begin{bmatrix} \cos(\psi_k) & -a\sin(\psi_k) \\ \sin(\psi_k) & a\cos(\psi_k) \end{bmatrix}\begin{bmatrix} \tilde{u}_k \\ \tilde{\omega}_k \end{bmatrix}$$

Matrices $\mathbf{P}_{(k)}$ and $\mathbf{Q}_{(k)}$ are limited and from (A18) and (A27),

$$\lim_{k\to\infty}\tilde{\psi}_k = 0, \quad \lim_{k\to\infty}\tilde{u}_k = 0, \text{ and } \lim_{k\to\infty}\tilde{\omega}_k = 0 \tag{A29}$$

So that, it is true that $\mathbf{P}_k \to 0$ and $\mathbf{Q}_k \to 0$ with $k \to \infty$.

Proof. By using ref. [1, Theorem 5.3.1, p. 248 and Theorem 5.2.3, p. 240], each solution $\mathbf{v}_k$ of (A28) satisfies

$$\mathbf{v}_k = \mathbf{A}^k\mathbf{v}_0 + \sum_{l=1}^{k}\mathbf{A}^{k-l}\left[\mathbf{P}_{l-1} + \mathbf{Q}_{l-1}\right] \tag{A30}$$

$$\mathbf{v}_k = \mathbf{A}^k\mathbf{v}_0 + \sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{P}_{l-1} + \sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{Q}_{l-1} \tag{A31}$$

$$\|\mathbf{v}_k\| \le \left\|\mathbf{A}^k\mathbf{v}_0\right\| + \left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{P}_{l-1}\right\| + \left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{Q}_{l-1}\right\| \tag{A32}$$

Equation (A31) can be written as

$$\|\mathbf{v}_k\| \le c_0\left[\delta\left(1 + c_2\right)\right]^{k-k_1} + \left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{P}_{l-1}\right\| + \left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{Q}_{l-1}\right\| \tag{A33}$$

where

$$\left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{P}_{l-1}\right\| \le \max\sqrt{\left(P1_k^2 + P2_k^2\right)}\left|\tilde{\psi}_0\right|\left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}k_\psi^{l-1}\right\| \tag{A34}$$

$$\left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}\mathbf{Q}_{l-1}\right\| \le |\tilde{u}_0|\left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}k_u^{l-1}\right\| + |\tilde{\omega}_0|\left\|\sum_{l=1}^{k}\mathbf{A}^{k-l}k_\omega^{l-1}\right\| \tag{A35}$$

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

where $\tilde{\psi}_0$, $\tilde{u}_0$ and $\tilde{\omega}_0$ are the initial angle and velocities errors, respectively, and $0 < (k_\psi, k_u, k_\omega) < 1$. We can always choose $\delta(1 + c_2) < 1$ according to ref. [1, Theorem 5.2.3, pp. 240–241], and applying the Toeplitz Lemma ref. [1, p. 682], it is true that,

$$\lim_{k\to\infty} \sum_{l=1}^{k} \mathbf{A}^{k-l} k_\psi^{l-1} = 0, \quad \lim_{k\to\infty} \sum_{l=1}^{k} \mathbf{A}^{k-l} k_u^{l-1} = 0,$$
$$\lim_{k\to\infty} \sum_{l=1}^{k} \mathbf{A}^{k-l} k_\omega^{l-1} = 0 \tag{A36}$$

Then, from (A32–A36),

$$\lim_{k\to\infty} \|\mathbf{v}_k\| = 0 \tag{A37}$$

Subsequently, from (A23), (A28), and (A37) $\tilde{x}_k \to 0$, $\tilde{y}_k \to 0$, and $\tilde{\psi}_k \to 0$ with $k \to \infty$.

Remark A1. *If the trajectory tracking problem with $u_{ref} \neq 0$ is considered:*

$$\|\tilde{\mathbf{x}}_k\| = \|[\tilde{x}_k \quad \tilde{y}_k \quad \tilde{\psi}_k \quad \tilde{u}_k \quad \tilde{\omega}_k]^T\| \to \mathbf{0} \quad \text{with} \quad k \to \infty.$$

Remark A2. *If the positioning problem with $u_{ref} = 0$ is considered, by using ref. [1, Theorem 5.3.1, Chapter 5, p. 248] it is also fulfilled that*

$$\|\tilde{\mathbf{x}}_k\| = \|[\tilde{x}_k \quad \tilde{y}_k \quad \tilde{\psi}_k \quad \tilde{u}_k \quad \tilde{\omega}_k]^T\| \to \mathbf{0} \text{ with } k \to \infty.$$

Remark A3. *The necessary and sufficient condition for the system (A1) has a zero-error convergence is (A9), because this condition causes an exact solution in the aforementioned system.*

Acknowledgments

This work was partially funded by the German Academic Exchange Service (DAAD – Deutscher Akademischer Austausch Dients), by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET – National Council for Scientific Research), and by the Universidad Nacional de San Juan – UNSJ, Argentina. We would also like to acknowledge cooperation from the RTS – Institute for Systems Engineering, University of Hannover, Germany, especially to Prof. Dr.-Eng. Bernardo Wagner.

References

1. R. Agarwal, *Difference Equations and Inequalities, Theory, Methods, and Applications* (Marcel Dekker, Inc., New York, 2000).
2. J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Rob. Automat.* **7**, 278–288 (1991).
3. R. Carelli, J. Santos-Victor, F. Roberti and S. Tosetti, "Direct visual tracking control of remote cellular robots," *Rob. Autonom. Syst.* **54**, 805–814 (2006).
4. J. H. Chung, B. J. Yi, W. K. Kim and H. Lee, "The Dynamic Modeling and Analysis for an Omnidirectional Mobile Robot with Three Caster Wheels," *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (2003) pp. 521–527.
5. C. Connolly, J. Burns and R. Weiss, "Path planning using Laplace's equation," *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio (1990) pp. 2102–2106.
6. C. De la Cruz and R. Carelli, "Dynamic model based formation control and obstacle avoidance of multi-robot systems," *Robotica* **26**, 345–356 (2008) (Cambridge).
7. F. Del Rio, G. Jiménez, J. Sevillano, C. Amaya and A. Balcells, "Error Adaptive Tracking for Mobile Robots," *Proceedings of 28th Annual Conference on IEEE Industrial Electronics Society*, Sevilla, Spain (2002) pp. 1–42415–2420.
8. K. Do and J. Pan, "Global output-feedback path tracking of unicycle-type mobile robots," *Rob. Comput.-Integr. Manuf.* **22**, 166–179 (2006).
9. W. Dong and Y. Guo, "Dynamic Tracking Control of Uncertain Mobile Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada (2005) pp. 2774–2779.
10. R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Network* **9**, 589–600 (1998).
11. R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," *J. Rob. Syst.* **4**, 149–163 (1997).
12. D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Rob. Automat. Mag.* **4**, 23–33 (1997).
13. T. Fraichard and H. Asama, "Inevitable Collision States: A Step Towards Safer Robots?," *IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, IROS (2003).
14. T. Fraichard and A. Scheuer, "Car-Like Robots and Moving Obstacles," *IEEE International Conference on Robotics and Automation*, San Diego, California, ICRA (1994).
15. T. Fukao, H. Nakagawa and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. Rob. Automat.* **16**, 609–615 (2000).
16. S. Ge and Y. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonom. Rob.* **13**, 207–222 (2002).
17. C. L. Hwang and L. J. Chang, "Trajectory tracking and obstacle avoidance of car-like mobile robots in an intelligent space using mixed H_2/H_∞ decentralized control," *Trans. Mechatron.* **12**, 345–352 (2007).
18. C. L. Hwang, S. Y. Han and L. J. Chang, "Trajectory Tracking of Car-Like Mobile Robots using Mixed H_2/H_∞ Decentralized Variable Structure Control," *International Conference on Mechatronics*, Chongqing, China (2005) pp. 520–525.
19. E. Jang, S. Jung and T. Hsia, "Collision Avoidance and Control of a Mobile Robot Using a Hybrid Force Control Algorithm", *30th Annual Conference on IEEE Industrial Electronics Society*, Busan, Korea (2004) pp. 413–418.
20. S. Jung, E. Jang and T. Hsia, "Collision Avoidance of a Mobile Robot Using Intelligent Hybrid Force Control Technique," *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 4418–4423.
21. Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot," *Proceedings of IEEE International Conference on Robotics and Automation*, Tsukuba, Japan (1990) pp. 384–389.
22. K. Kim, "Receding horizon tracking control for constrained linear continuous time-varying systems," *IEEE Proc. Control Theory Appl.* **150**, 534–538 (2003).
23. G. Klančar and I. Škrjank, "Tracking-error model-based predictive control for mobile robots in real time," *Rob. Autonom. Syst.* **55**, 460–469 (2007).
24. J. C. Latombe, *Robot Motion Planning*, vol. 0124 (Kluwer, Dordrecht, The Netherlands, 1991).

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press

25. S. Lee and J. H. Park, "Virtual Trajectory in Tracking Control of Mobile Robots," *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronic*, Kobe, Japan, AIM (2003).
26. S. Liu, H. Zhang, S. X. Yang and J. Yu, "Dynamic Control of a Mobile Robot Using an Adaptive Neurodynamics and Sliding Mode Strategy," *Proceedings of 5th Congress Intelligent Control and Automation*, Hangzhou, China (2004) pp. 5007–5011.
27. Z. Y. Liu, R. H. Jing, X. Q. Ding and J. H. Li, "Trajectory Tracking Control of Wheeled Mobile Robots Based on the Artificial Potential Field," *Fourth International Conference on Natural Computation*, Jinan, China (2008) pp. 382–387.
28. A. Masoud, "Using hybrid vector-harmonic potential fields for multi-robot, multi-target navigation in a stationary environment," *Proc. IEEE Int. Conf. Rob. Automat.* **4**, 3564–3571 (1996).
29. T. Myers and L. Vlacic, "Autonomous driving in a time-varying environment," *IEEE Adv. Rob. Soc. Impacts* **1**, 53–58 (2005).
30. Y. K. Nak and R. Simmons, "The Lane-Curvature Method for Local Obstacle Avoidance," *IEEE International Conference on Robotics and Automation*, Geiranger, Norway, ICRA (1998).
31. J. Normey-Rico, I. Alcalá, J. Gomez-Ortega and E. Camacho, "Mobile robot path tracking using PID controller," *Control Eng. Pract.* **9**, 1209–1214 (2001).
32. J. Normey-Rico, J. Gomez-Ortega and E. Camacho, "A Smith-predictor-based generalized predictive controller for mobile robot path-tracking," *Control Eng. Pract.* **7**, 729–740 (1999).
33. L. Ojeda and J. Borenstein, "Reduction of Odometry Errors in Over-constrained Mobile Robots," *Proceedings of the UGV Technology Conference at the SPIE AeroSense Symposium*, Orlando, Florida (2003) pp. 21–25.
34. E. Owen and L. Montano, "A Robocentric Motion Planner for Dynamic Environments Using the Velocity Space," *IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, IROS (2006).
35. A. Rosales, G. Scaglia, V. Mut and F. di Sciascio, "Controller Designed by Means of Numeric Methods for a Benchmark Problem: RTAC (Rotational Translational Actuator)," *IEEE – Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Mexico (2006) pp. 97–104.
36. G. Scaglia, V. Mut, A. Rosales and O. Quintero, "Tracking Control of a Mobile Robot using Linear Interpolation," *International Conference on Integrated Modeling and Analysis in Applied Control and Automation – IMAACA*, Buenos Aires, Argentina (2007).
37. G. Scaglia, O. Quintero, V. Mut and F. di Sciascio, *Numerical Methods Based Controller Design for Mobile Robots* (IFAC World Congress, 2008).
38. G. Scaglia, O. Quintero, V. Mut and F. di Sciascio, *Numerical Methods Based Controller design for Mobile Robots* (Robotica – Cambridge University Press, Cambridge, UK, 2008).
39. M. Seder, K. Macek and I. Petrovic, "An Integrated Approach to Real Time Mobile Robot Control in Partially Known Indoor Environments," *Proceedings of 31st Annual Conference of the IEEE Industrial Electronics Society*, Raleigh, North Carolina (2005).
40. S. Shuli, "Designing approach on trajectory-tracking control of mobile robot," *Rob. Comput.-Integr. Manuf.* **21**, 81–85 (2005).
41. R. Simmons, "The Curvature–Velocity Method for Local Obstacle Avoidance," *IEEE International Conference on Robotics and Automation*, Geiranger, Norway, ICRA (1996).
42. C. Stachniss and W. Burgard, "An integrated approach to goal directed obstacle avoidance under dynamic constraints for dynamic environments," *IEEE International Conference on Intelligent Robots and Systems*, EPFL, Switzerland, IROS (2002).
43. G. Strang, *Linear Algebra and Its Applications*, 3rd ed. (MIT Academic Press, New York, 1980).
44. J. Tian, M. Gao and E. Lu, "Dynamic Collision Avoidance Path Planning for Mobile Robot Based on Multi-sensor Data Fusion by Support Vector Machine," *International Conference on Mechatronics and Automation*, Harbin, China, ICMA07 (2007) pp. 2779–2783.
45. P. S. Tsai, L. S. Wang, F. R. Chang and T. F. Wu, "Systematic Backstepping Design for B-spline Trajectory Tracking Control of the Mobile Robot in Hierarchical Model," *International Conference on Networking, Sensing and Control*, Taipei, Taiwan (2004) pp. 713–718.
46. I. Ulrich and J. Borenstein, "VFH*: Local obstacle avoidance with look ahead verification," *Proc. IEEE Int. Conf. Rob. Automat.* **3**, 2505–2511 (2000).
47. I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," *Proc. IEEE Int. Conf. Rob. Automat.* **2**, 1572–1577 (1998).
48. S. Vougioukas, "Reactive Trajectory Tracking for Mobile Robots based on Non Linear Model Predictive Control," *International Conference on Robotics and Automation*, Torun, Poland (2007) pp. 3074–3079.
49. D. Wang, D. Liu and G. Dissanayake, "A Variable Speed Force Field Method for Multi-Robot Collaboration," *Proceedings of International Conference on Intelligent Robots and Systems*, China (2006) pp.2697–2702.
50. T. Y. Wang and C. C. Tsai, "Adaptive Trajectory Tracking Control of a Wheeled Mobile Robot via Lyapunov Techniques," *Annual Conference of IEEE Industrial Electronics Society*, Busan, Korea (2004) pp. 389–394.
51. J. M. Yang and J. H. Kim, "Sliding mode control for trajectory of nonholonomic wheeled mobile robots,", *IEEE Trans. Rob. Automat.* **15**, 578–587 (1999).
52. X. Yang, K. He, M. Guo and B. Zhang, "An Intelligent Predictive Control Approach to Path Tracking Problem of Autonomous Mobile Robot," *IEEE International Conference on Robotics and Automation*, Beijing, China (1998) pp. 3301–3306.
53. Huai-Xiang Zhang, Guo-Jun Dai and Hong Zeng, "A Trajectory Tracking Control Method for Nonholonomic Mobile Robots," *International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, China (2007) pp. 7–11.

https://doi.org/10.1017/S0263574709005402 Published online by Cambridge University Press