# Robust-control-based controller design for a mobile robot

**Gustavo J. E. Scaglia · Vicente A. Mut ·
Mario Jordan · Carlos Calvo · Lucia Quintero**

**Abstract**   The control of a mobile robot using a linear model with uncertainty for design purposes is investigated. The uncertainty arises from the variation of the operation point of the mobile robot. Robust Control Theory is used for the controller design, which allows dealing with systems whose parameters may vary between certain bounds. The proposed controller has shown, in experimentation tests, an acceptable performance and an easy and simple practical implementation. Also, an application of the proposed controller to a leader-following problem is shown; in it, the relative position between robots is obtained through a laser.

**Keywords**   Modeling uncertainty · Nonlinear systems · Robotics · Trajectory tracking

## 1 Introduction

The control of mobile robots has stirred the attention of many researchers around the world. However, the control of these systems is not simple. The main problem is focused on the control of three variables: Cartesian coordinates $(x, y)$ and the orientation angle $\theta$ with only two control inputs, the linear velocity $(V)$ and the angular velocity $(W)$ of the mobile robot as shown in Fig. 1; see [1]. Different representations of such controllers have been published in

G. J. E. Scaglia (✉) · V. A. Mut · L. Quintero
Instituto de Automática (INAUT), Universidad Nacional de San Juan, Av. San Martín 1109 (oeste),
San Juan J5400ARL, Argentina
e-mail: gscaglia@inaut.unsj.edu.ar

V. A. Mut
e-mail: vmut@inaut.unsj.edu.ar

M. Jordan
Instituto Argentino de Oceanografía (IADO-CONICET), Florida 8000, Complejo CRIBABB, Edificio E1,
Bahía Blanca B8000FWB, Argentina

C. Calvo
Departamento de Matemática, Universidad Nacional de San Juan, Av. San Martín 1109 (oeste), San Juan J5400ARL, Argentina
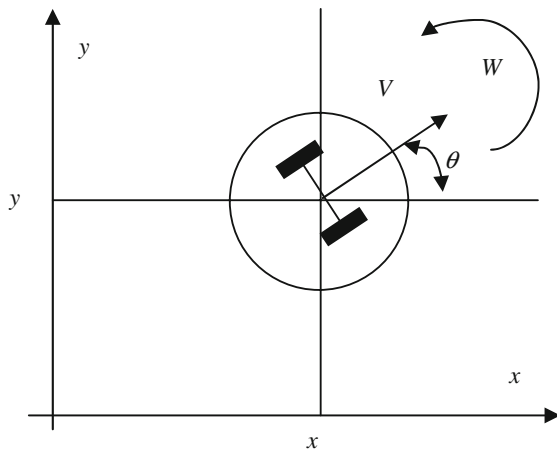e-mail: ccalvo@unsj.edu.ar

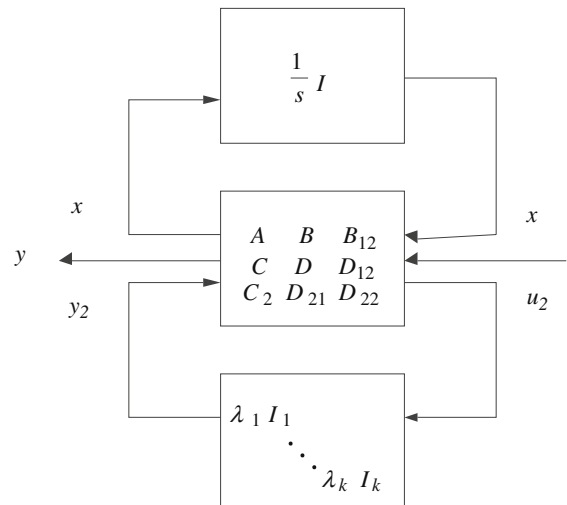**Fig. 1** Geometrical description of the mobile robot



**Fig. 2** Schematic representation of (18)

the literature, e.g. those based on predictive control [2], on robust adaptive control [3, 4], or on sliding-mode control [5], among others, which are not always easy to implement in real time.

The mobile robot considered in this work consists of a wheeled vehicle with autonomous movement, equipped with traction electrical motors that are controlled by an on-board computer. It is assumed that the mobile robot is composed of a rigid body with wheels that do not get deformed and moves on a horizontal plane. One of the main problems in the field of mobile robots is to make a vehicle follow a pre-established trajectory. In order to achieve that objective, control algorithms are designed so as to generate the appropriate commands applied to the traction motors controlling the mobile-robot movement. The environments where a mobile robot can move are classified into structured and partially structured. In the first case, the position of all the obstacles is known beforehand, so it is possible to generate a collision-free trajectory. In partially structured environments, only the position of some obstacles is known, so the mobile robot must have external sensors (ultrasound, laser, and camera) in order to determine the positions of such obstacles.

The use of path tracking in a navigation system is justified in structured workspaces as well as in partially structured workspaces, where 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, the algorithms used to avoid obstacles usually re-plan the trajectory in order to avoid a collision, generating a new reference trajectory from this point on. Besides, there exist algorithms that express the reference trajectory of the mobile robot as a function of a descriptor called $r$ [6] or $s$ (called "virtual time") [7] 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; otherwise, if the tracking error is small, the reference trajectory must tend to the original trajectory calculated by the global planner. In this way, the module of trajectory tracking will use the original path or the in-line recalculated path as reference to obtain the smallest error when the mobile robot follows the path [8]. Therefore, path tracking is always important independently of whether the reference trajectory has been generated by a global trajectory planner or a local one.

Various control strategies have been proposed for tracking trajectories, some of which are based on either kinematic or dynamic models of the mobile robot [9], depending on the operative speed and the precision of the dynamic model, respectively [10]. Different structures to control these systems have been developed as well. Tsuji et al. [11] used a time-varying feedback gain whose evolution can be modified through parameters that determine the convergence time and the behavior of the system. Fierro and Lewis [12] use the controller proposed by Kanayama et al. [13] to feed the inputs to a velocity controller, making the position error asymptotically stable. So, a controller to make the mobile-robot velocity follow the reference velocity is designed. The work of Fukao et al. [14] extends the

design proposed by Fierro and Lewis [12] and assumes that the model parameters are unknown. In [15] an adaptive controller that takes into account the parametric uncertainties and the robot's external perturbations is proposed to guarantee perfect velocity tracking. The reference for the velocity is obtained by using the controller proposed in [13]. In [16] two controllers are designed, which are called position controller and heading controller. The former ensures position tracking and the latter is activated when the tracking error is small enough and the tracking reference does not change its position. This reduces the error over the mobile-robot orientation at the end of the path. In [17] the posture controller is designed as a function of the posture error and, in this way, the reference velocities are generated on the basis of a specification set as: (i) if the distance to a reference position is relatively large, then the robot movement is quick, and the speed is reduced as the robot approaches the target; (ii) the robot should take the shortest amount of time to reach the desired position. Later, the reference velocities input a PID controller that generates the torque required by the desired speed. In [18] a controller for trajectory tracking is designed using the kinematic model of the mobile robot and a transformation matrix. Such a matrix is singular if the linear velocity of the mobile robot is zero. Therefore, the effectiveness of that controller is ensured only if the velocity is different from zero. Simulation results using a linear velocity different from zero as initial condition are shown in that paper. Sun [19] proposed a controller based on the error model of [13]. This controller is formed by two equations, which are switched depending on the value of the angular velocity of the mobile robot and its prescribed tolerance.

In contrast to previous papers where, in general, precise models are needed and commonly goodness of performance is achieved under special conditions, here a simple and universal approach is proposed for path following. To achieve this goal, it is assumed that mobile robots are represented by a family of linear models and, from robust control theory, a linear controller is designed. This controller is simple to implement and makes robot performance satisfactory around the operation point. Under this assumption, and knowing the desired state, a simple rotation and translation of coordinated axes is made to force the robot to follow a predefined path. The main contribution of this work is that the proposed methodology is based upon easily understandable concepts, and that there is no need for complex calculations to attain the control signal.

In this paper, the control scheme presented in [12, 14, 15, 17, 20] will be employed. Accordingly, a kinematic controller is designed first, which generates the reference velocity in order to reach the desired goal. Additionally, this reference is employed to input a velocity controller in the scheme. In our work, a PID controller is used as a velocity controller. The implemented controller will be placed on board an existing mobile robot in order to maintain its translational and rotational speeds at desired values; cf. [17, 20].

Besides, in our work it is not necessary to switch the controller as in [16] in cases when the position reference does not change and the tracking error is small. Our purpose is that, when this situation is detected, the desired orientation be changed, and then calculate the control signal by using the same expression. The controller developed by Sun and Cui, [18] has the disadvantage of having a linear velocity different from zero to be necessary for operating well. Furthermore, our controller does not need to change the control expression when the angular velocity is lower than a pre-established value in contrast to [19].

We propose to use a linearized model of a mobile robot, considering the variation on the linear model parameters, since the operating point is taken as an uncertainty. Thus, the objective is to attain a simpler linear model (a set of linear models) that will thoroughly describe the system's performance. Here the Theory of Robust Control is used to design the controller, in order to achieve a stable and robust performance for a set of linear models. The main result is an easily implemented controller that only requires the multiplication of matrices of numerical elements.

The paper is organized as follows. The mobile-robot model as a family of models around the chosen working point (linearization point) is given in Sect. 2. The design objectives are discussed in Sect. 3. Section 4 shows how to obtain the generalized system equations to apply the $H_\infty$ control theory. In Sect. 5, the proposed controller design is presented. The experiments on a laboratory mobile robot PIONEER 2DX are addressed in Sect. 6, including the leader-following problem where the relative position of both robots is obtained through a laser sensor. Finally in Sect. 7 the conclusions are given and an appendix with the baisc nomenclature used in this paper is presented.

## 2 System model

2.1 Model of the mobile robot

A nonlinear kinematic model for a mobile robot will be used, [21], as shown in Fig. 1, which is represented by,

$$\dot{x} = V\cos\theta, \quad \dot{y} = V\sin\theta, \quad \dot{\theta} = W. \tag{1}$$

We can write (1) as follows

$$\dot{x} = f(x, u), \quad y = Cx,$$

$$x = \begin{bmatrix} x & y & \theta \end{bmatrix}^T, \quad u = \begin{bmatrix} V & W \end{bmatrix}^T, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $V$ is the linear velocity of the mobile robot, $W$ is the angular velocity of the mobile robot, $(x, y)$ denotes the Cartesian position, $\theta$ is the orientation of the mobile robot, $(\dot{x}, \dot{y}, \dot{\theta})$ denotes the derivatives of $(x, y, \theta)$ with respect to time. Then, the aim is to find the values of $V$ and $W$ so that the mobile robot may follow a pre-determined trajectory. We assume that the mobile robot is moving on a horizontal plane without slip.

By developing (1) through Taylor series and considering small deviations in $x$ and $u$ with respect to a given point $\mathbf{x}_o$ and $\mathbf{u}_o$, [22, pp. 561–562], the higher-order terms can be neglected, and we obtain

$$\partial\dot{x} = \left.\frac{\partial f}{\partial x}\right|_{\mathbf{x}_o, \mathbf{u}_o} dx + \left.\frac{\partial f}{\partial u}\right|_{\mathbf{x}_o, \mathbf{u}_o} du, \tag{2}$$

$$\frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial u}du = \underbrace{\begin{bmatrix} 0 & 0 & -V\sin\theta \\ 0 & 0 & V\cos\theta \\ 0 & 0 & 0 \end{bmatrix}}_{A}\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} + \underbrace{\begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}}_{B}\begin{bmatrix} dV \\ dW \end{bmatrix}, \tag{3}$$

where $|_{\mathbf{x}_o, \mathbf{u}_o}$ means that the derivatives are evaluated in $\mathbf{x}_o$ and $\mathbf{u}_o$.

By rearranging the *controllability*[1] (contr) matrix of the system, we have

$$\text{contr} = \begin{bmatrix} A & AB & A^2B \end{bmatrix}. \tag{4}$$

By substituting (3) in (4), we obtain

$$\text{contr} = \begin{bmatrix} \cos\theta & 0 & 0 & -V\sin\theta & 0 & 0 \\ \sin\theta & 0 & 0 & V\cos\theta & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5}$$

It can be clearly noticed that the range of (5) is:

$$\text{rank(contr)} = 3 \text{ if } V \neq 0. \tag{6}$$

Since $C$ in (1) is the identity matrix, the system output is represented by the states, and the system becomes *observable*. From (6), it can be concluded that, since $V \neq 0$, the system represented by (3) is completely controllable. From (3) it can be noted that both $A$ and $B$ are dependent on $V$ and $\theta$. Then, by choosing two values of $V$ and $\theta$, for example:

$$V = 0.5\,\text{m/s}, \quad \theta = 45° \tag{7}$$

and by substituting (7) in (3), we have

---

[1] The dynamical system described by the equation $\dot{x} = Ax + Bu$, $y = Cx + Du$, $x(0) = x_0$, is said to be controllable if, for any initial state $x(0) = x_0$, $t_1 > 0$ and final state $x_1$, there exists a input $u(\cdot)$ such that $x$ satisfies $x(t_1) = x_1$. See Theorem 3.1 [23, p. 47] for controllability test.

$$A = \begin{bmatrix} 0 & 0 & -0.3536 \\ 0 & 0 & 0.3536 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.7071 & 0 \\ 0.7071 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{8}$$

Then, the mobile-robot performance is determined by,

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \tag{9}$$

and, when the robot moves in a small neighborhood of the selected values in (7), the linearized model of the mobile robot is obtained, as stated in Eq. 9.

2.2 Model variation

If, while moving, the mobile robot departs from the velocity and orientation values given by (7), the matrices that describe it, (3), will change as well. This change can be interpreted as a variation in the nominal model parameters in (9). This means, that the robot would be better and more precisely described if, instead of considering a single linear model, a family of linear models were stated. Zhou [23, pp. 261–264] shows that this family of models can be represented as follows:

$$G(s) = \begin{bmatrix} A + \sum_{i=1}^{k} \lambda_i A_i & B + \sum_{i=1}^{k} \lambda_i B_i \\ \hline C + \sum_{i=1}^{k} \lambda_i C_i & D + \sum_{i=1}^{k} \lambda_i D_i \end{bmatrix}, \tag{10}$$

where $A$, $B$, $C$, $D$ represent the nominal model, and parameter uncertainty is shown by scalars $\lambda_i \in (-1, 1)$. This kind of uncertainty can be represented by a *Linear Fractional Transformation* (*LTF*),[23, p. 264]. Then $G(s)$ can be expressed as,

$$G(s) = F_u \left( M, \frac{1}{s} \right), \tag{11}$$

where $M$ is determined by (12):

$$M = \begin{bmatrix} A + \sum_i \lambda_i A_i & B + \sum_i \lambda_i B_i \\ C + \sum_i \lambda_i C_i & D + \sum_i \lambda_i D_i \end{bmatrix}. \tag{12}$$

By defining the matrix, $P_i$ as:

$$P_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \tag{13}$$

where each $P_i$ can be stated as:

$$P_i = \begin{bmatrix} L_i \\ W_i \end{bmatrix} \begin{bmatrix} R_i \\ Z_i \end{bmatrix}^T, \tag{14}$$

where $L_i \in \mathbb{R}^{n \times q_i}$, $W_i \in \mathbb{R}^{n_y \times q_i}$, $R_i \in \mathbb{R}^{n \times q_i}$, $Z_i \in \mathbb{R}^{n_u \times q_i}$, and $q_i$ is the rank of $P_i$, $n$ is the order of $A$, $n_y$ is the output number; and $n_u$ is the input number. Then, we obtain,

$$\lambda_i P_i = \begin{bmatrix} L_i \\ W_i \end{bmatrix} [\lambda_i \ I q_i] \begin{bmatrix} R_i \\ Z_i \end{bmatrix}^T; \tag{15}$$

from this, $M$ can be described as,

$$M = \underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{M_{11}} + \underbrace{\begin{bmatrix} L_1 & \cdots & L_k \\ W_1 & \cdots & W_k \end{bmatrix}}_{M_{12}} \underbrace{\begin{bmatrix} \lambda_1 I_1 & & \\ & \ddots & \\ & & \lambda_k I_k \end{bmatrix}}_{\Delta P} \underbrace{\begin{bmatrix} R_1^* & Z_1^* \\ \vdots & \vdots \\ R_k^* & Z_k^* \end{bmatrix}}_{M_{21}}, \tag{16}$$

$$M = F_l \left( \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & 0 \end{bmatrix}, \Delta P \right). \tag{17}$$

By substituting (17) in (11), we obtain,

$$G(s) = F_u \left( F_l \left( \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & 0 \end{bmatrix}, \Delta P \right), \frac{1}{s} \right). \tag{18}$$

Equation (18) can be graphically interpreted as in Fig. 2.

From Fig. 2 and (16), (17) and (18), it can be inferred that [23, pp. 261–264],

$$B_2 = \begin{bmatrix} L_1 & \dots & L_k \end{bmatrix}, \; D_{12} = \begin{bmatrix} W_1 & \dots & W_k \end{bmatrix},$$
$$C_2 = \begin{bmatrix} R_1 & \dots & R_k \end{bmatrix}^*, \; D_{21} = \begin{bmatrix} Z_1 & \dots & Z_k \end{bmatrix}^*, \; D_{22} = 0. \tag{19}$$

After further elaboration of (18), we obtain,

$$G(s) = G_{11}(s) + \underbrace{G_{12}(s)\Delta P(I - G_{22}(s)\Delta P)^{-1}G_{21}(s)}_{\Delta G(s)}, \tag{20}$$

where the state matrices of $G_{ij}(s)$ are given by:

$$G_{11}(s) = \{A, B, C, D\}, \quad G_{12}(s) = \{A, B_2, C, D_{12}\},$$
$$G_{21}(s) = \{A, B, C_2, D_{21}\}, \quad G_{22}(s) = \{A, B_2, C_2, D_{22}\}. \tag{21}$$

Equation 19 reveals that the family $G(s)$ can be stated as a nominal or central model $G_{11}(s)$, plus a term $\Delta G(s)$ that shows how much a family's element has departed from the nominal model.

The present paper takes into account variations in a robot's linear velocity at a rate of $\pm 0.4$ m/s; it considers a single parameter, $\lambda_i$, as well. This is to say that $G(s)$ from (10) is,

$$G(s) = \left[ \begin{array}{c|c} A + \lambda A_1 & B + \lambda B_1 \\ \hline C + \lambda C_1 & D + \lambda D_1 \end{array} \right], \tag{22}$$

where:

$$A_1 = \begin{bmatrix} 0 & 0 & -0.4\sin(45) \\ 0 & 0 & 0.4\cos(45) \\ 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0.007 & 0 \\ 0.007 & 0 \\ 0 & 0.001 \end{bmatrix}, \quad C_1 = 0.001 \, I_3, \quad D_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{23}$$

It was observed that matrices $B_1$ and $C_1 \neq 0$, in order to make the models of the state variables of the family described in (20) both completely controllable and observable.

## 3 Design objectives and specifications

### 3.1 Design objectives

From (19) it can be noted that the behavior of the mobile robot is not described by a single linear model but through a family of time-invariant linear models. By considering the diagram of Fig. 3, the objective pursued is to design a controller $K(s)$, to ensure not only the stability of the system and the good performance when using a mobile robot's nominal model, but also when using any of the models of the family $G(s)$. That is, both robust stability and performance should be guaranteed.

During the first stage of the controller design it is assumed that the system adequately responds to step-like signals, both on coordinates $x$ and $y$ as a function of time, i.e., the response should not present significant overshoots.
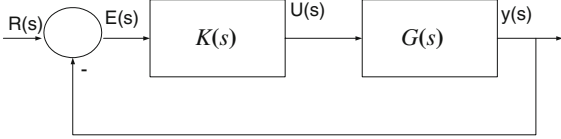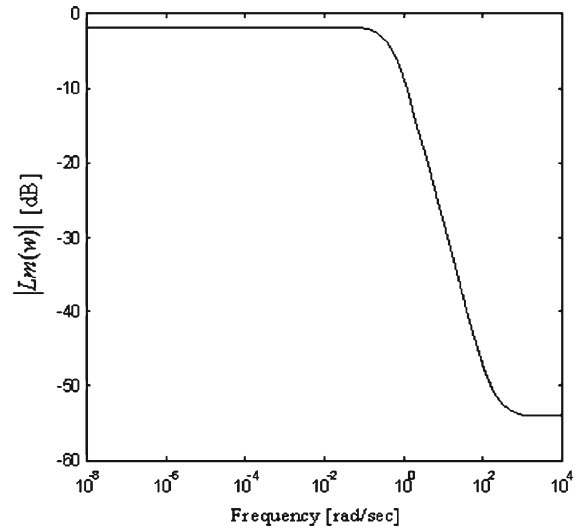
**Fig. 3** Simplified control scheme



**Fig. 4** $|L_m(w)|$ as a function of frequency

### 3.2 Robust stability

The difference between the nominal model and any model belonging to $G(s)$ can be represented in many ways. One of the simplest ways is written as follows [24, Chap. 11]:

$$G(s) = G_{11}(s) + A_a(s), \tag{24}$$

where $A_a$ stands for an additive perturbation and $G_{11}$ is the nominal model.

It can also be represented in an arrangement as a multiplicative uncertainty, as shown in (25),

$$G(s) = (I + A_m)G_{11}(s), \tag{25}$$

where $A_m$ is a multiplicative perturbation. The uncertainties are related to each other through the expression:

$$A_a = A_m \times G_{11}(s). \tag{26}$$

Morari et al. [24, p. 236] state that the magnitude of the perturbation can be measured in terms of a bound on $\overline{\sigma}(A_a)$ and $\overline{\sigma}(A_m)$; that is to say,

$$\overline{\sigma}(A_a) \leq L_a(w), \quad \forall w \tag{27}$$

$$\overline{\sigma}(A_m) \leq L_m(w), \quad \forall w, \tag{28}$$

where $\overline{\sigma}[\cdot]$ is the *maximum singular value*, $L_a(w)$ and $L_m(w)$ are scalar bounds on $\overline{\sigma}(A_a)$ and $\overline{\sigma}(A_m)$, respectively. The conditions for robust stability when using additive and multiplicative uncertainty to represent the family $G(s)$, can be found in detail in [23, Chap. 9, 24, Chap. 11, 25, Chap. 2] with some brief quotations below. The kind of uncertainties considered will be described as follows.

#### 3.2.1 Multiplicative uncertainty

The system is stable for all perturbations $A_m$ $(\overline{\sigma}(A_m) < L_m(w))$ if and only if

$$\overline{\sigma}\left[G_{11}K(I + G_{11}K)^{-1}\right]L_m < 1, \quad \forall w$$

$$\left\|L_m G_{11}K(I + G_{11}K)^{-1}\right\|_{\infty} < 1. \tag{29}$$

### 3.2.2 Additive uncertainty

The system is stable for all perturbations $A_a(\overline{\sigma}(A_a) < L_a(w))$ if and only if

$$\overline{\sigma}\left[K\left(I + G_{11}K\right)^{-1}\right] L_a < 1, \quad \forall w$$

$$\left\|L_a K\left(I + G_{11}K\right)^{-1}\right\|_{\infty} < 1. \tag{30}$$

From (26), we have:

$$\overline{\sigma}(A_a) \leq \overline{\sigma}(A_m)\overline{\sigma}(G_{11}), \quad \overline{\sigma}(A_m) \geq \frac{\overline{\sigma}(A_a)}{\overline{\sigma}(G_{11})}. \tag{31}$$

Then, from (28) and (31) $L_m$ is proposed to be defined as:

$$L_m(w) = \frac{\overline{\sigma}(A_a)}{\overline{\sigma}(G_{11})}. \tag{32}$$

Figure 4 shows the variation of $L_m(w)$ as a function of frequency.

### 3.3 Robust performance

This section presents a brief reference to some well-known definitions of Control Theory, but applied to the system of Fig. 2, in order to use them in the design of a robust controller for a mobile robot. Sensibility ($S$) and Complementary Sensibility ($T$) can be defined as,

$$S = (I + GK)^{-1}, \tag{33}$$
$$T = GK(I + GK)^{-1}. \tag{34}$$

It can also be shown that:

$$S + T = I. \tag{35}$$

If one considers that the system is single input-single output, the sensibility $S$ is given by

$$\lim_{\Delta P \to 0} \frac{\Delta T/T}{\Delta P/P} = S. \tag{36}$$

This means that the sensibility function sets the limit when $\Delta P \to 0$ of the percent variation of the closed-loop transfer function, when a percent variation arises in the plant to be controlled. $S$ is also the transfer function between the error signal and the reference signal of the system. It should obviously be expected that $S$ remains small enough within the operating frequency range.

This condition for a small $S$ can be stated as a limitation on the infinite norm of $HS$, where $H$ is a function that focuses on the frequency range where $S$ is desired to be *small*; that is:

$$\|HS\|_{\infty} < 1. \tag{37}$$

In (29) it can be noted that the *smaller the* value of $T$, the greater the robust stability will be whereas, from (37), it can be inferred that, for smaller $S$, both greater perturbation rejection and smaller sensibility are attained for the closed-loop transfer function when facing plant changes. It should be noted that $T$ and $S$ *cannot simultaneously be small* (35).

To obtain a solution from (29) and (37), the problem may be stated by seeking to find a controller that both ensures the internal stability and satisfies:

$$\left\|\begin{array}{c} HS \\ L_m T \end{array}\right\|_{\infty} < 1, \tag{38}$$

which guarantees both a good resistance to perturbation and a good stability for the family of models $G(s)$. This problem is called *mixed sensibility minimization* [26, p. 214, 27].
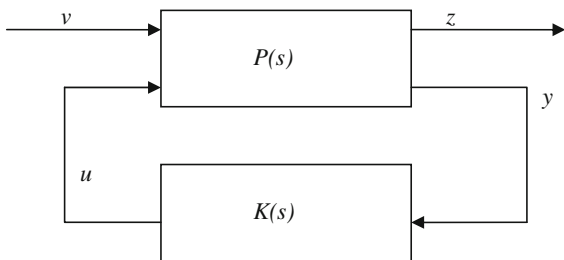
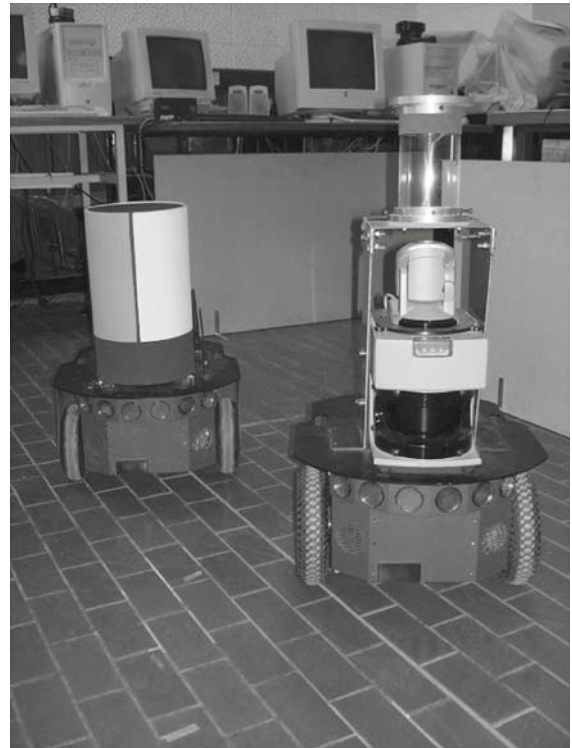**Fig. 5** Standard configuration for plant-controller



**Fig. 6** Pioneer 2DX mobile robot and its environment

## 4 Reduction to a standard form

The key problem in $H_\infty$ Control Theory is to design a controller that ensures the internal stability of the system shown in Fig. 5:

Here $v$ contains all exogenous inputs, $u$ is the controller output, $y$ is the controller input and $z$ is the objective [25, Chap. 8]. In addition, it should satisfy:

$$\|R_{zv}\|_\infty < \gamma, \tag{39}$$

where $R_{zv}$ is the transference function between $z$ and $v$, with $\gamma > 0$.

If $P(s)$ of Fig. 5 is split into four sub-matrices of adequate dimensions, then $R_{zv}$ may be stated as:

$$R_{zv} = F_l(P, K), \quad R_{zv} = P_{11} + P_{12} K(I - P_{22}K)P_{21}. \tag{40}$$

Equation 39 should be expressed as in Fig. 4, considering $\gamma = 1$, that is,

$$HS = H(I + G_{11}K(I - G_{11}K)^{-1}), \tag{41}$$

$$\begin{bmatrix} HS \\ L_m T \end{bmatrix} = \begin{bmatrix} H \\ 0 \end{bmatrix} + \begin{bmatrix} HG_{11} \\ L_m G_{11} \end{bmatrix} K(I - G_{11}K)^{-1}. \tag{42}$$

From (40) and (42) it follows that:

$$P(s) = \begin{bmatrix} \begin{bmatrix} H \\ 0 \end{bmatrix} & \begin{bmatrix} HG_{11} \\ L_m G_{11} \end{bmatrix} \\ I & G_{11} \end{bmatrix}, \quad P(s) = \begin{bmatrix} A_p & B_{1p} & B_{2p} \\ C_{1p} & D_{11p} & D_{12p} \\ C_{2p} & D_{21p} & D_{22p} \end{bmatrix}. \tag{43, 44}$$

The resulting functions $H(s)$ and $L_m(s)$, used in the design of the controller to be applied to the mobile robot in this paper, are shown by (45) and (46)

$$H = \begin{bmatrix} \dfrac{0.17}{0.2s+1} & 0 & 0 \\ 0 & \dfrac{0.17}{0.2s+1} & 0 \\ 0 & 0 & \dfrac{0.17}{0.06667s+1} \end{bmatrix}, \quad L_m = \dfrac{0.004018s+0.8036}{2s+1}I_3. \tag{45, 46}$$

## 5 Design of the robust controller

This section presents the design of a controller proposed for the mobile robot Pioneer 2DX available at INAUT [28, 29]. Figure 6 shows the Pioneer 2DX and the laboratory facilities where the experiments were carried out.

The plant is assumed to be described by the state equations:

$$\dot{x} = Ax + B_1 v + B_2 u, \quad z = C_1 x + D_{12} u, \quad y = C_2 x + D_{21} v. \tag{47}$$

The dimensions of $x$, $v$, $u$, $y$ and $z$ are $n$, $l$, $m$, $q$ and $p$, respectively. It is also compatible with:

$$D_{12}^T D_{12} = I_m, \quad D_{21} D_{21}^T = I_q. \tag{48}$$

The transformations needed to go from (44) to (47) and to comply with (48) are developed in [25, Chap. 4]. The following assumptions are made:

1. $(A, B_2)$ is stabilizable and that $(A, C_2)$ is detectable;
2. Matrices $D_{12}$ and $D_{21}$ satisfy (48);
3. rank $\begin{bmatrix} A - jwI & B_2 \\ C_1 & D_{12} \end{bmatrix} = n + m$, $\forall w$ real;
4. rank $\begin{bmatrix} A - jwI & B_1 \\ C_2 & D_{21} \end{bmatrix} = n + q$, $\forall w$ real.

The subsections below show the values for conditions 1, 2, 3 and 4 for the problem at hand.

***Condition 1:***

Controllable modes $= \{-15, 0, -5, -5, -0.5, -0.5, 0, 0\}$;
Non-controllable modes $= \{-0.5\}$;
Observable modes $= \{0, 0, 0\}$;
Non-observable modes $= \{-5, -5, -15, -0.5, -0.5, -0.5\}$.

***Condition 2:*** After the transformations, the matrices $D_{12}$ and $D_{21}$ become,

$$D_{12}^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad D_{21} = I_3. \tag{49, 50}$$

***Condition 3:*** See Fig.7a.

***Condition 4:*** See Fig. 7b.

Figure 7b shows that condition 4 is not attained. This condition states that the closed-loop system $F_l(P, K)$ does not have non-controllable modes on the imaginary axis. A proposal is to modify by considering a greater number of external inputs $w$, to fulfill condition 4. That is:

$$v_{\text{new}} = \begin{bmatrix} v & v_{\text{extra}} \end{bmatrix}^T, \quad z = R_{zv\text{new}} \quad v_{\text{new}} = \begin{bmatrix} R_{zv} & R_{zvn} \end{bmatrix} \begin{bmatrix} v \\ v_{\text{extra}} \end{bmatrix}, \tag{51}$$

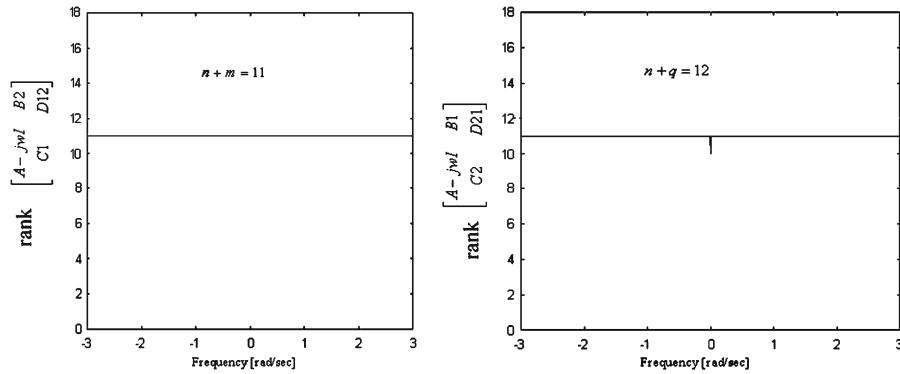$$\|R_{zv}\|_\infty \leq \|R_{zv\text{new}}\|_\infty. \tag{52}$$

**Fig. 7** **a** Condition 3; **b** Condition 4

If it is verified that it $\|R_{zv\text{new}}\|_\infty < 1$, then $\|Rzv\|_\infty < 1$ will be achieved as well. Matrix $B_{1p\text{new}}$ resulting from this procedure is

$$
B_{1p}\text{new} = \begin{bmatrix}
0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.7 \\
0.0 & 0.0 & 0.0 & 0.7 & 0.0 \\
1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 2.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 3.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
\underbrace{\phantom{0.0 \quad 0.0 \quad 0.0}}_{B_{1p}} & & & \underbrace{\phantom{0.0 \quad 0.0}}_{B_{1pn}} &
\end{bmatrix}. \tag{53}
$$

With this new matrix, *condition 4 is thoroughly met* by applying the theorem that allows finding the proposed controller $K(s)$ [25, Chap. 8]. The controller has a matrix $A$ of order $9 \times 9$. Then, to decrease the order of $K(s)$, a balanced realization was found. The Hankel singular values for the controller are [23, pp. 72–78],

$$
\sum = \{4.7136, 3.0157, 0.4951, 0.1554, 0.0891, 0.0173, 0.0075, 0.0053, 0.0011\}.
$$

Only three states were considered. The resulting controller is now called $K_r(s)$. Figure 8 shows the singular values for $K(s)$ and $K_r(s)$. The results obtained in simulations using $K_r(s)$ instead of $K(s)$ were slightly different. The controller finally implemented was obtained by making a conversion into a discrete controller of $K_r(s)$ which will be called $K_{rd}$. In the next section, experimental results applying this controller to the Pioneer 2DX mobile robot will be presented.

The reference path to be followed by the mobile robot will be denoted by the values of $x\text{ref}(t)$, $y\text{ref}(t)$ and $\theta\text{ref}(t)$, and the real trajectory depicted by the PIONEER 2DX mobile robot will be called $x(t)$, $y(t)$ and $\theta(t)$. The values for position $x(t)$ in discrete time $nT_0$, when $T_0$ is the simple time and $n \in \{0, 1, 2, \ldots\}$, will be denoted as $x_n$; the same notation will be used to reference any function in each sample time.

The controller presents a good performance when the mobile robot orientation is $45° \pm 20°$, which is a very restrictive condition for common operation in a mobile robot. Then, a translation and a rotation are proposed, so the controller assumes that the mobile robot moves at $45°$ while it follows the reference trajectory established by the trajectory global planner. The used control scheme can be seen in Fig. 9.

Here $\theta m$ represents the orientation of the mobile-robot model. The desired values used by the controller to calculate the control signal are,

$$
\begin{aligned}
x\text{d}_n &= \sqrt{(x\text{ref}_n - x_n)^2 + (y\text{ref}_n - y_n)^2} \cos(45°), \\
y\text{d}_n &= \sqrt{(x\text{ref}_n - x_n)^2 + (y\text{ref}_n - y_n)^2} \sin(45°).
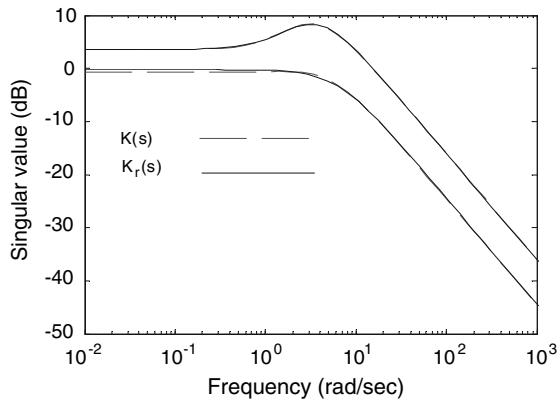\end{aligned} \tag{54}
$$

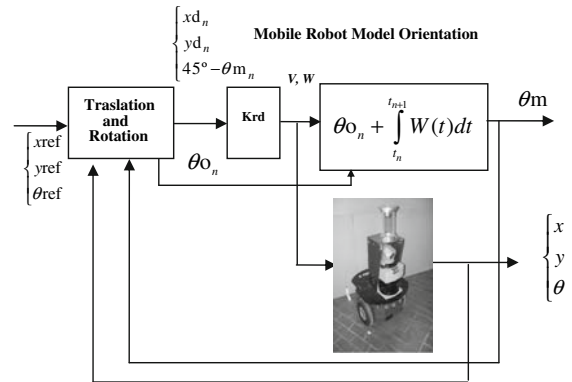**Fig. 8** Singular values of $K(s)$ and $K_r(s)$



**Fig. 9** Control scheme used

The mobile robot should have the reference trajectory orientation

$$\arctan \frac{y\text{ref}_{n+1} - y_n}{x\text{ref}_{n+1} - x_n}$$

to reach the desired trajectory. If this orientation is taken as $45°$, the orientation of the mobile robot each sample time is,

$$\theta o_n = 45 - \left( \arctan \frac{y\text{ref}_{n+1} - y_n}{x\text{ref}_{n+1} - x_n} - \theta_n \right), \tag{55}$$

where $\theta o_n$ represents the initial orientation of the mobile-robot model used at each sample time.

## 6 Experimental and simulation results

Experiments to test the performance of the proposed controller were carried out using a PIONEER 2DX mobile robot. Figure 6 shows the Pioneer 2DX and the laboratory facilities where the experiments were carried out. The PIONEER 2DX mobile robot includes an estimation system based on odometry, which adds accumulative errors to the system. From this, updating the data through external sensors is necessary. This problem is separated from the strategy of trajectory tracking and is not considered in this paper [8, 30]. The PIONEER 2DX has a PID velocity controller, used to maintain the velocities of the mobile robot at the desired values; cf. [17, 20].

To observe the performance of the proposed controller, a trajectory whose orientation changes abruptly was used (see Figs. 10, 11). It can also be noticed that, when the trajectory direction suddenly changes, the error increases, but it decreases afterwards, with a maximum error of about 100 mm. Besides, the error is not too large when compared with the size of the PIONEER 2DX, considering the demanding nature of the desired trajectory. Figure 11 shows how the mobile robot follows the reference trajectory arriving at the end of the trajectory in a precise way and it remains in that position without undesirable oscillatory motions. 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 (330 mm.). In other kinds of reference trajectories, in which the orientation changes smoothly, the controller performance will be better, as depicted in Fig. 12.

Another typical benchmark reference trajectory, e.g. a sinusoid, was used to test controller performance; in this case Fig. 12 shows the trajectory followed by the PIONEER 2DX mobile robot on the $xy$-plane, for an initial position of the mobile robot given by $x = 0$ m, $y = 0$ m but the reference trajectory starting from $x = 1$ m and $y = 1$ m. It can be seen from Fig. 12 that the mobile robot tends to the desired trajectory and then follows it in a precise way.

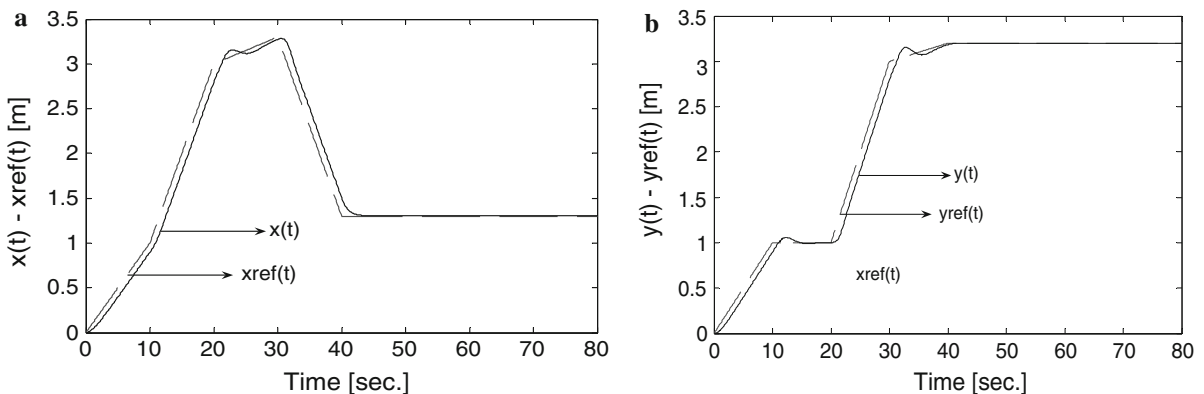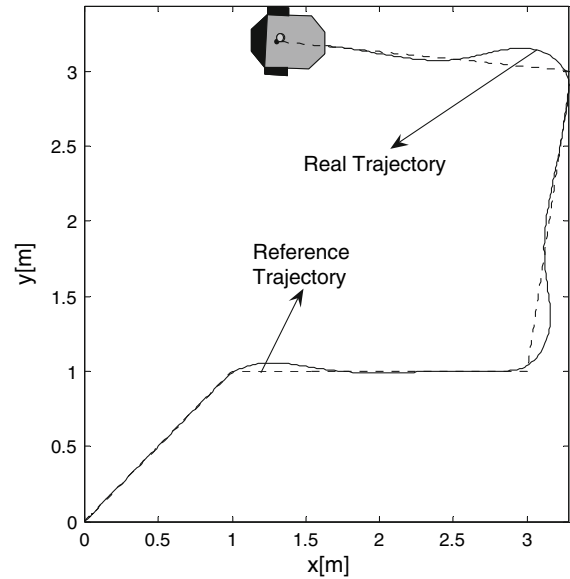**Fig. 10** Experimental results: real and reference trajectories



**Fig. 11** Experimental results: **a** Time evolution of $x$ref($t$) and $x(t)$, **b** time evolution of $y$ref($t$) and $y(t)$

## 6.1 Leader–robot following

An example of the application of the proposed methodology is presented, the control objective being to make a robot follow a leader, keeping a predetermined distance. Moreover, it can evidently be seen in this example that there is no dependency between the control scheme proposed and the odometry-based internal system. In leader following, the relative position between both robots is obtained through an external laser sensor. As leader, a PIONNER 2DX mobile robot was used.

From Fig. 13a the trajectory followed by both the leader robot and the follower in the $xy$-plane can be seen. Also, it can be observed that the follower robot quickly approaches the leader, and then follows it; the time evolution of the distance between both robots can be seen in Fig. 13b. The relative position between both robots was obtained by means of a laser sensor (see Fig.6).
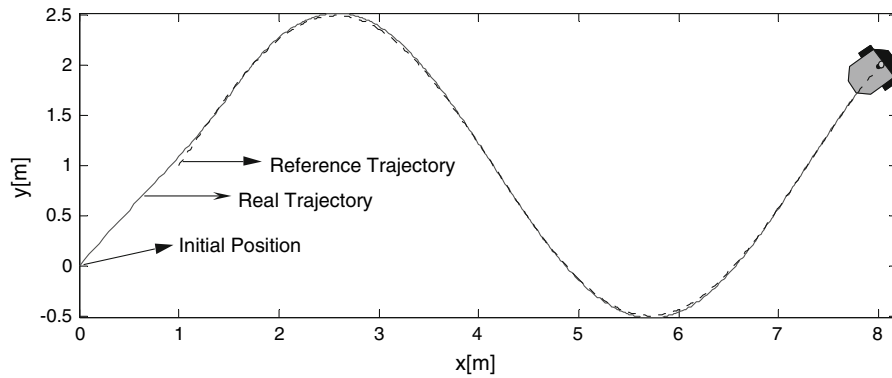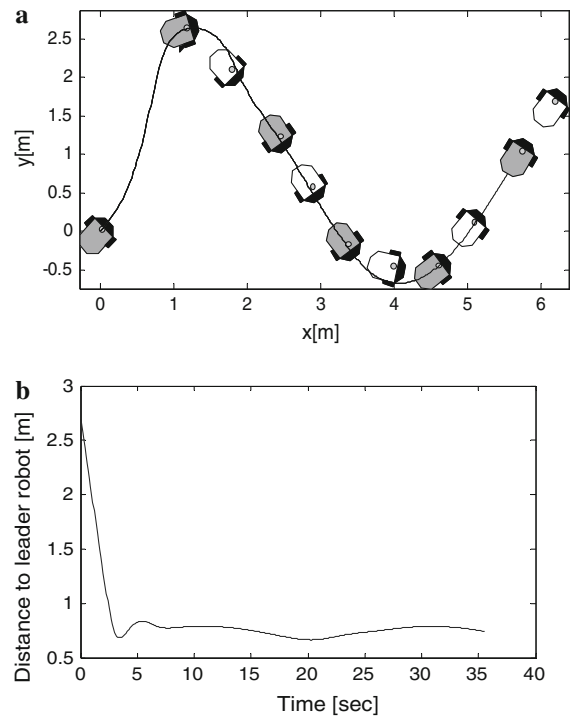
**Fig. 12** Experimental results: real and reference trajectories

**Fig. 13** Experimental results: **a** Trajectory followed by robots in the $xy$-plane, **b** time evolution of the relative distance between the leader and the follower robot



## 7 Conclusions

In this work a robust controller to control a nonlinear multivariable system (mobile robot in this case) has been proposed. The mobile robot can be described by a family of linear models, so a controller is designed by using robust control theory so that the mobile robot shows a very good performance around the linearization point. Then, a rotation and translation are used to extend the obtained results for an established orientation to any possible orientation of the mobile robot. The trajectory to be followed can be defined beforehand by the global planner (Figs. 10, 11, 12) or it can be obtained through the use of an external laser sensor which allows obtaining the relative position of the object to be followed (Fig. 13). From this it can be observed that the desired trajectory can be obtained in different ways and several objectives can be achieved. In all cases, the same control algorithm is used, which shows good performance. This demonstrates the importance of designing a controller that allows the robot to follow a pre-established trajectory irrespective of how it was generated.

The above control structures can be designed and implemented without great difficulty. Experimental results of the developed controllers on a PIONEER 2DX mobile robot have also been discussed. From the analysis of these experiments, it can be concluded that the trajectory error between the desired and the real trajectory of the mobile robot is very small. The effectiveness and feasibility are tested in practice through experiments.

As future work, the formulation of algorithms based on this technique will be applied to avoid unexpected obstacles and in multi-robot systems, where the control of a formation of mobile robots (the trajectory for the formation of the mobile robots as well as the relative position of each is established) will be analyzed. In these systems, various robots are used to carry out some complex tasks as for example vigilance, rescue, and heavy loads transportation, to name a few.

## Appendix. Nomenclature

| Symbol | Description | Unit |
|---|---|---|
| $\mathbb{R}$ | Field of real number | |
| $\mathbb{R}^{n \times m}$ | Rectangular $m \times n$ matrix | |
| $x, y$ | Cartesian coordinates of the mobile robot | m |
| $\theta$ | Orientation of the mobile robot | rad |
| $x\text{ref}, y\text{ref}$ | Reference Cartesian coordinates of the mobile robot | m |
| $\theta\text{ref}$ | Reference orientation of the mobile robot | rad |
| $x\text{d}, y\text{d}$ | Desired Cartesian coordinates of the mobile robot | m |
| $\theta\text{d}$ | Desired orientation of the mobile robot | rad |
| $\dot{x}, \dot{y}$ | Time derivatives of the Cartesian coordinates of the mobile robot | m/s |
| $\dot{\theta}$ | Time derivative of the orientation of the mobile robot | rad/s |
| $\theta\text{m}$ | Orientation of the mobile-robot model | rad |
| $\theta\text{o}$ | Initial orientation of the mobile-robot model | rad |
| $T_0$ | Sample time | s |
| $V$ | Linear or translational velocity of the mobile robot | m/s |
| $W$ | Angular or rotational velocity of the mobile robot | rad/s |
| Contr | Controllability[1] matrix | – |
| Observ | Observability[2] matrix | – |
| Rank | Rank of a matrix | – |
| $I$ | Identity matrix | |
| $G(s)$ | Transfer function of the system | |
| $K(s)$ | Transfer function of the controller | |
| $K_r(s)$ | Transfer function of the reduced controller | |
| $K_{rd}$ | Discretized transfer function of the reduced controller | |
| $\{A, B, C, D\}$ | Shorthand for state space realization $C (sI - A)^{-1} B + D$ | – |
| $G(s) = \left[ \begin{array}{c\|c} A & B \\ \hline C & D \end{array} \right]$ | Shorthand for state space realization $C (sI - A)^{-1} B + D$ | |
| $F_l(M, \Delta); M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ | Shorthand for $M_{11} + M_{12}\Delta(I - M_{22}\Delta)^{-1}M_{21}$ | |
| $F_u(M, \Delta); M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ | Shorthand for $M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}$ | |
| $A_a$ | Additive perturbation | |
| $A_m$ | Multiplicative perturbation | |
| $\bar{\sigma}[\cdot]$ | Maximum singular value | |
| $L_a$ | Scalar bound of the additive perturbation | |
| $L_m$ | Scalar bound of the multiplicative perturbation | |
| $\Sigma$ | Hankel singular values | |
| $w$ | Angular frequency ($w = 2\pi f$) | (rad/s) |
| $S$ | Sensibility | |

| Symbol | Description | Unit |
|--------|-------------|------|
| $T$ | Complementary sensibility | |
| $jw$ | Imaginary number | |

# References

1. Lee T-C, Song K-T, Lee C-H, Teng C-C (1999) Tracking control of mobile robots using saturation feedback controller. Rob Autom 4:2639–2644
2. Gómez Ortega J, Camacho EF (1996) Mobile robot navigation in a partially structured static environment, using neural predictive control. Control Eng Pract 4(12):1669–1679. doi:10.1016/S0967-0661(96)00184-0
3. Kim MS, Shin JH, Lee JJ (2000) Design of a robust adaptive controller for a mobile robot. In: Proceedings of the IEEE/RSJ international conference intelligent robots and systems (IROS), vol 3, pp 1816–1821
4. Park KH, Cho SB, Lee YW (2001) Optimal tracking control of a nonholonomic mobile robot. In: Proceedings of the ISIE 2001 international symposium on industrial electronics, vol 3, pp 2073–2076
5. Yang J-M, Kim J-H (1999) Control of nonholonomic mobile robots. IEEE Control Syst Mag 19:15–23. doi:10.1109/37.753931
6. Del Rio F, Jiménez G, Sevillano J, Arnaya C, Balcells A (2002) Error adaptive tracking for mobile robots. In: Proceedings of the 2002 28th annual conference of the IEEE industrial electronics society, Saville, Spain, pp 2415–2420
7. Lee S, Park JH (2003) Virtual trajectory in tracking control of mobile robots. In: Proceedings of IEEE/ASME international conference on advance intelligent mechatronics, pp 35–39
8. Normey-Rico J, Gomez-Ortega J, Camacho E (1999) A Smith-predictor-based generalized predictive controller for mobile robot path-tracking. Control Eng Pract 7:729–740. doi:10.1016/S0967-0661(99)00025-8
9. Lee T, Song K, Lee C, Teng C (2001) Tracking control of unicicle-modeled mobile robots using a saturation feedback controller. IEEE Trans Control Syst Technol 9(2):305–318. doi:10.1109/87.911382
10. Do KD, Pan J (2006) Global output-feedback path tracking of unicycle-type mobile robots. Rob Comput Integr Manuf 22:166–179. doi:10.1016/j.rcim.2005.03.002
11. Tsuji T, Morasso P, Kaneko M (1995) Feedback control of nonholonomic mobile robots using time base generator. In: Proceedings of the 34th IEEE conference on robotics and automation, vol 2. Nagoya, Japan, pp 1385–1390
12. Fierro R, Lewis F (1995) Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In: Proceedings of the 34th IEEE conference on decision and control, vol 4. New Orleans, pp 3805–3810
13. Kanayama Y, Kimura Y, Miyazaki F, Noguchi T (1990) A stable tracking control method for an autonomous mobile robot. In: Proceedings of the IEEE international conference on robotics and automation, pp 384–389
14. Fukao T, Nakagawa H, Adachi N (2000) Adaptive tracking control of a nonholonomic mobile robot. IEEE Trans Rob Autom 16(5):609–615
15. Kim Y-H, Ha I-J (2000) Asymptotic state tracking in a class of nonlinear systems via learning-based inversion. IEEE Trans Autom Control 45:2011–2027. doi:10.1109/9.887624
16. Chwa D (2004) Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. IEEE Trans Control Syst Technol 12(4):637–644. doi:10.1109/TCST.2004.824953
17. Shim H, Sung Y (2004) Stability and four-posture control for nonholonomic mobile robots. IEEE Trans Rob Autom 20(1):148–154. doi:10.1109/TRA.2003.819730
18. Sun S, Cui P (2004) Path tracking and a practical point stabilization of mobile robot. Rob Comput Int Manuf 20:29–34
19. Shuli S (2005) Designing approach on trajectory—tracking control of mobile robot. Rob Comput Int Manuf 21(1):81–85
20. Cruz D, McClintock J, Perteet B, Orqueda O, Cao Y, Fierro R (2007) Decentralized cooperative control—a multivehicle platform for research in networked embedded systems. IEEE Control Syst Mag 27(3):58–78
21. Campion G, Bastin G, d'Andrea-Novel B (1996) Structural properties and clasification of kinematic and dynamic models of wheeled mobile robots. IEEE Trans Rob Autom 12(1):47–62. doi:10.1109/70.481750
22. Brogan W (1991) Modern control theory. 3rd ed. Prentice Hall, Englewood Cliffs
23. Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Englewood Cliffs
24. Morari M, Zafiriou E (1989) Robust process control. Prentice-Hall, Englewood Cliffs
25. Green M, Limebeer D (1995) Linear robust control. Prentice-Hall, Englewood Cliffs
26. Chandrasekharan P (1996) Robust control of linear dynamical systems. Academic Press, london
27. Chen SB, Fan YH, Zhang FE (1997) Case study of robust linear quadratic design and mixed-sensitivity H∞ control. In: IEE Proceedings: Control theory applications, vol 14, pp 476–480
28. Mut VA, Postigo JF, Slawiñski E, Kuchen B (2001) Bilateral teleoperation of mobile robots. In: World multi-conference systems, cybernatics informatics, vol IX. Orlando, Miami, USA, pp 54–59
29. Secchi H (1998) Control de vehículos autoguiados con realimentación sensorial. Master Thesis (in Spanish). Edit. Fundación Universidad Nacional de San Juan, ISBN 950-605-190-9, San Juan, Argentina
30. Normey-Rico J, Alcalá I, Gomez-Ortega J, Camacho E (2001) Mobile robot path tracking using PID controller. Control Eng Pract 9:1209–1214. doi:10.1016/S0967-0661(01)00066-1