

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,100

Open access books available

116,000

International authors and editors

120M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Motion Control and Velocity-Based Dynamic Compensation for Mobile Robots

Felipe Nascimento Martins and
Alexandre Santos Brandão

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.79397>

Abstract

The design of motion controllers for wheeled mobile robots is often based only on the robot's kinematics. However, to reduce tracking error it is important to also consider the robot dynamics, especially when high-speed movements and/or heavy load transportation are required. Commercial mobile robots usually have internal controllers that accept velocity commands, but the control signals generated by most dynamic controllers in the literature are torques or voltages. In this chapter, we present a velocity-based dynamic model for differential-drive mobile robots that also includes the dynamics of the robot actuators. Such model can be used to design controllers that generate velocity commands, while compensating for the robot dynamics. We present an explanation on how to obtain the parameters of the dynamic model and show that motion controllers designed for the robot's kinematics can be easily integrated with the velocity-based dynamic compensation controller. We conclude the chapter with experimental results of a trajectory tracking controller that show a reduction of up to 50% in tracking error index *IAE* due to the application of the dynamic compensation controller.

Keywords: velocity-based dynamic model, dynamic modeling, dynamic compensation, motion control, tracking control

1. Introduction

A common configuration for mobile robots is the differential drive, which has two independently driven parallel wheels and one (or more) unpowered wheel to balance the structure [1]. For several years differential-drive mobile robots (DDMR) have been widely used in many applications because of their simple configuration and good mobility. Some applications of

DDMR are surveillance [2], floor cleaning [3], industrial load transportation [4], autonomous wheelchairs [5], and others.

In the literature, most of the motion controllers for DDMR are based only on its kinematics. The main reasons for that are: (a) the kinematic model is simpler than the dynamic model, therefore the resulting controllers are less complex and simpler to tune; (b) the accuracy of the dynamic model depends on several parameters that might change or are difficult to measure, like the robot's mass and moment of inertia; and (c) dynamic controllers usually generate torque or voltage commands, while mobile robots frequently have internal velocity controllers that take velocity as input [6]. However, the robot's low-level velocity control loops do not guarantee perfect velocity tracking, especially when high-speed movements and/or heavy load transportation are required. In such cases, to reduce tracking error, it becomes essential to consider the robot dynamics as well, as shown in [7].

A possible solution to overcome the problem described above is to design a controller that compensates for the robot's dynamics. Commercial mobile robots usually have internal controllers that accept velocity commands, like the Pioneer 3 from Adept Mobile Robots, the Khepera from K-Team Corporation, and the robuLAB-10 from Robosoft Inc. However, the control signals generated by most dynamic controllers in the literature are torques or voltages, as in [8–14]. Because of that, some researchers have proposed dynamic controllers that generate linear and angular velocities as commands [15, 16]. In some works, the dynamic model is divided in to two parts, allowing the design of independent controllers for the robot kinematics and dynamics [17–20]. Finally, to reduce performance degradation in applications in which the robot dynamic parameters may vary (such as load transportation) or when the knowledge of the dynamic parameters is imprecise, adaptive controllers can also be considered [7, 21].

The above-mentioned works applied a dynamic model that has linear and angular velocities as inputs, which illustrates the interest on such kind of dynamic model. In such context, this chapter explains the velocity-based dynamic model and its mathematical properties, which are useful for the design of controllers that compensate for the robot dynamics. It also illustrates how to design a trajectory tracking motion controller based on the robot's kinematics, and how to integrate it with a velocity-based dynamic compensation controller.

2. Dynamic model

The classical equation to represent the dynamics of mobile robots can be obtained via Lagrangian formulation, resulting in [22].

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}_m(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_m(\dot{\mathbf{q}}) + \mathbf{G}_m(\mathbf{q}) + \tau_d = \mathbf{B}(\mathbf{q})\tau - \mathbf{A}^T(\mathbf{q})\lambda, \quad (1)$$

where $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T$ is the vector of generalized coordinates of the system with n degrees of freedom, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the matrix of inertia, $\mathbf{V}_m(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is the matrix of Coriolis and centrifugal forces, $\mathbf{F}_m(\dot{\mathbf{q}}) \in \mathbb{R}^{n \times 1}$ is the vector that represents viscous friction, $\mathbf{G}_m(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ is the vector of gravitational torques, $\tau_d \in \mathbb{R}^{n \times 1}$ is the disturbance vector,

$\tau \in \mathbb{R}^{r \times 1}$ is the vector of input torques, where r is the number of inputs, $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{n \times r}$ is the input transformation matrix, $\lambda \in \mathbb{R}^{m \times 1}$ is the vector that represents restriction forces, and $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ is the matrix associated to such restrictions. Two well known properties of such model are [9, 22]:

1. $\mathbf{M}(\mathbf{q})$ is a symmetric and positive definite matrix, that is, $\mathbf{M}(\mathbf{q}) = \mathbf{M}(\mathbf{q})^T > 0$;
2. $(\dot{\mathbf{M}} - 2\mathbf{V}_m)$ is antisymmetric.

The above properties are widely used on the development and stability analysis of controllers for mobile robots, as shown in [8, 9, 22, 23]. But, such controllers generate torque commands, not velocities, as usually accepted by commercial robots. The conversion from torque to velocity commands requires knowledge of the actuation system of the robot (model of its motors and its speed controllers). On the other hand, a controller designed from a velocity-based dynamic model generates linear and angular velocities that can be directly applied as commands for mobile robots.

In such a context, now the dynamic model for the DDMR proposed in [16] is reviewed. For convenience, we first present its equations again. Then, the dynamic model is written in such a way that it becomes similar to the classical dynamic equation based on torques. **Figure 1** depicts a DDMR with the variables of interest. There, u and ω are, respectively, the linear and angular velocities, G is the center of mass, h is the point of interest (whose position should be controlled)

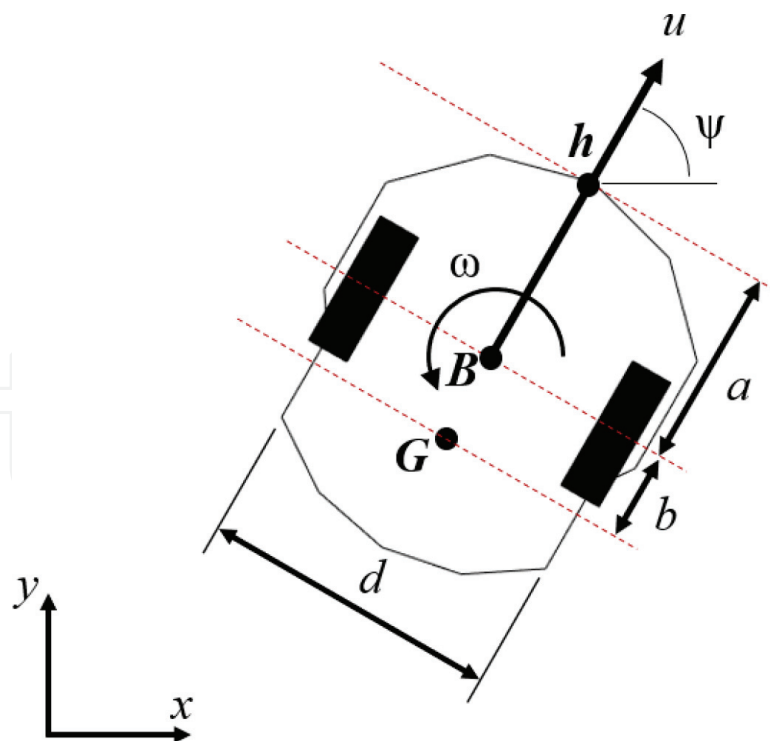


Figure 1. The differential drive mobile robot (DDMR). u and ω are, respectively, the linear and angular velocities, G is the center of mass, h is the point of interest with coordinates x and y in the XY plane, ψ is the robot orientation, a is the distance from the point of interest to the point in the middle of the virtual axle that links the traction wheels (point B), b is the distance between points G and B , and d is the distance between the points of contact of the traction wheels to the floor.

with coordinates x and y in the XY plane, ψ is the robot orientation, a is the distance from the point of interest to the point in the middle of the virtual axle that links the traction wheels (point B), b is the distance between points G and B , and d is the distance between the points of contact of the traction wheels to the floor. The complete mathematical model is written as [16].

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

where $\theta = [\theta_1, \dots, \theta_6]^T$ is the vector of identified parameters and $\delta = [\delta_x, \delta_y, 0, \delta_u, \delta_\omega]^T$ is the vector of parametric uncertainties associated to the mobile robot. The equations describing the parameters θ are presented in Section 3. The model is split into kinematic and dynamic parts. The kinematic model is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \end{bmatrix}, \quad (3)$$

which is a modified approach to describe the robot kinematics. The classical unicycle model is obtained when $a = 0$ in (3), but here we consider the case in which $a \neq 0$, which means that the (x, y) position described by the model is not in the center of the line between the traction wheels, but at a distance a from it (see point h in **Figure 1**). We use this model because it is useful on the design of the trajectory tracking controller, as shown in Section 4.

The part of the equation that represents the dynamics is given by

$$\begin{bmatrix} \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\theta_3}{\theta_1}\omega^2 - \frac{\theta_4}{\theta_1}u \\ -\frac{\theta_5}{\theta_2}u\omega - \frac{\theta_6}{\theta_2}\omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\theta_1} & 0 \\ 0 & \frac{1}{\theta_2} \end{bmatrix} \begin{bmatrix} u_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} \delta_u \\ \delta_\omega \end{bmatrix}. \quad (4)$$

As shown in [21], by rearranging its terms the Eq. (4) can be written as

$$\begin{bmatrix} -\theta_1 & 0 \\ 0 & -\theta_2 \end{bmatrix} \begin{bmatrix} \delta_u \\ \delta_\omega \end{bmatrix} + \begin{bmatrix} \theta_1 & 0 \\ 0 & \theta_2 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \theta_4 & -\theta_3\omega \\ \theta_5\omega & \theta_6 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_r \\ \omega_r \end{bmatrix}, \quad (5)$$

or, in a compact form, as

$$\Delta + \mathbf{H}'\dot{\mathbf{v}} + \mathbf{c}(\mathbf{v})\mathbf{v} = \mathbf{v}_r, \quad (6)$$

where $\mathbf{v}_r = [u_r \ \omega_r]^T$ is the vector of reference velocities, $\mathbf{v} = [u \ \omega]^T$ is the vector containing the actual robot velocities, and the matrices \mathbf{H}' and $\mathbf{c}(\mathbf{v})$, and the vector Δ are given by

$$\mathbf{H}' = \begin{bmatrix} \theta_1 & 0 \\ 0 & \theta_2 \end{bmatrix}, \quad \mathbf{c}(\mathbf{v}) = \begin{bmatrix} \theta_4 & -\theta_3\omega \\ \theta_5\omega & \theta_6 \end{bmatrix} \quad \text{and} \quad \Delta = \begin{bmatrix} -\theta_1 & 0 \\ 0 & -\theta_2 \end{bmatrix} \begin{bmatrix} \delta_u \\ \delta_\omega \end{bmatrix}. \quad (7)$$

Let us rewrite $\mathbf{c}(\mathbf{v})$ by adding and subtracting the term $i\theta_3u$ to its fourth element (where $i = 1\text{rad}^2/\text{s}$), such that

$$\mathbf{c}(\mathbf{v}) = \begin{bmatrix} \theta_4 & -\theta_3\omega \\ \theta_5\omega & \theta_6 + (i\theta_3 - i\theta_3)u \end{bmatrix}, \quad (8)$$

so that the term $\mathbf{c}(\mathbf{v})\mathbf{v}$ can be written as

$$\begin{bmatrix} 0 & -\theta_3\omega \\ \theta_3\omega & 0 \end{bmatrix} \begin{bmatrix} iu \\ \omega \end{bmatrix} + \begin{bmatrix} \theta_4 & 0 \\ 0 & \theta_6 + (\theta_5 - i\theta_3)u \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}. \quad (9)$$

The role of the term $i = 1\text{rad}^2/\text{s}$ is to make the units consistent to allow us to split $\mathbf{c}(\mathbf{v})$ into two matrices, while keeping the numerical values unchanged. Now, let us define $\mathbf{v}' = [iu \ \omega]^T$ as the vector of modified velocities, so that

$$\mathbf{v}' = \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}. \quad (10)$$

The terms in the vector of modified velocities are numerically equal to the terms in the vector of actual velocities \mathbf{v} , only its dimensions are different. By rewriting the model equation, the following matrices are defined:

$$\mathbf{H} = \begin{bmatrix} \theta_1/i & 0 \\ 0 & \theta_2 \end{bmatrix}, \quad \mathbf{F}(\mathbf{v}') = \begin{bmatrix} \theta_4/i & 0 \\ 0 & \theta_6 + (\theta_5/i - \theta_3)iu \end{bmatrix}, \quad \text{and} \quad \mathbf{C}(\mathbf{v}') = \begin{bmatrix} 0 & -\theta_3\omega \\ \theta_3\omega & 0 \end{bmatrix}. \quad (11)$$

Finally, the dynamic model of a DDMR can be represented by

$$\mathbf{v}_r = \mathbf{H}\dot{\mathbf{v}}' + \mathbf{C}(\mathbf{v}')\mathbf{v}' + \mathbf{F}(\mathbf{v}')\mathbf{v}' + \Delta, \quad (12)$$

or

$$\begin{bmatrix} u_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} \theta_1/i & 0 \\ 0 & \theta_2 \end{bmatrix} \begin{bmatrix} i\dot{u} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0 & -\theta_3\omega \\ \theta_3\omega & 0 \end{bmatrix} \begin{bmatrix} iu \\ \omega \end{bmatrix} + \begin{bmatrix} \theta_4/i & 0 \\ 0 & \theta_6 + (\theta_5/i - \theta_3)iu \end{bmatrix} \begin{bmatrix} iu \\ \omega \end{bmatrix} + \begin{bmatrix} \delta_u \\ \delta_\omega \end{bmatrix}. \quad (13)$$

Notice that $\mathbf{c}(\mathbf{v})\mathbf{v} = \mathbf{C}(\mathbf{v}')\mathbf{v}' + \mathbf{F}(\mathbf{v}')\mathbf{v}'$ and $\mathbf{H}'\dot{\mathbf{v}} = \mathbf{H}\dot{\mathbf{v}}'$, that is, the dimensions of the resulting vector \mathbf{v}_r are kept unchanged.

The model represented by Eq. (13) is mathematically equivalent to the one proposed in [16] and used in [7], where it was validated via simulation and experiments. Nevertheless, the model presented here is written in such a way that some mathematical properties arise. Such

properties, which are presented and discussed in the next session, can be applied on the design and stability analysis of dynamic controllers.

3. Dynamic parameters and model properties

To calculate the dynamic parameters of the vector θ , one has to know physical parameters of the robot, like its mass, its moment of inertia, friction coefficient of its motors, etc. The equations describing each one of the parameters θ_i are

$$\begin{aligned}
 \theta_1 &= \left[\frac{R_a}{k_a} (mr^2 + 2I_e) + 2rk_{DT} \right] \frac{1}{(2rk_{PT})} \quad [s], \\
 \theta_2 &= \left[\frac{R_a}{k_a} (I_e d^2 + 2r^2(I_z + mb^2)) + 2rdk_{DR} \right] \frac{1}{(2rdk_{PR})} \quad [s], \\
 \theta_3 &= \frac{R_a mbr}{k_a 2k_{PT}} \quad [sm/rad^2], \\
 \theta_4 &= \frac{R_a}{k_a} \left(\frac{k_a k_b}{R_a} + B_e \right) \frac{1}{rk_{PT}} + 1, \\
 \theta_5 &= \frac{R_a mbr}{k_a dk_{PR}} \quad [s/m], \text{ and} \\
 \theta_6 &= \frac{R_a}{k_a} \left(\frac{k_a k_b}{R_a} + B_e \right) \frac{d}{2rk_{PR}} + 1,
 \end{aligned} \tag{14}$$

where m is the mass of the robot, I_z is its moment of inertia at G , R_a , k_b and k_a are the electrical resistance, the electromotive constant, and the constant of torque of its motors, respectively, B_e is the coefficient of friction, I_e is the moment of inertia of each group rotor-reduction gear-wheel, r is the radius of each wheel, and b and d are distances defined in **Figure 1**. It is assumed that the internal motor controllers are of type PD (proportional-derivative) with proportional gains $k_{PT} > 0$ and $k_{PR} > 0$, and derivative gains $k_{DT} \geq 0$ and $k_{DR} \geq 0$. It is also assumed that the inductances of the motors are negligible, and both driving motors are identical.

Obtaining accurate values of all physical parameters of a robot might be difficult, or even not possible. Therefore, it is useful to discuss an identification procedure to directly obtain the values of the dynamic parameters θ . Such procedure is explained in Section 3.2.

It is interesting to point out that the dynamic model adopted here considers that the robot's center of mass G can be located anywhere along the line that crosses the center of the structure, as illustrated in **Figure 1**. This means that the formulation of the proposed dynamic model is adequate for robots that have a symmetrical weight distribution between their left and right sides. Because most differential drive robots have an approximately symmetrical weight distribution (with each motor and wheel on either left or right sides), such assumption does not introduce significant modeling errors on most cases. It should also be noticed that $\theta_i > 0$ for

$i = 1, 2, 4, 6$. The parameters θ_3 and θ_5 can be negative and will be null if, and only if, the center of mass G is exactly in the center of the virtual axle, that is, $b = 0$. Finally, in [21], it was shown that the model parameters θ_1 to θ_6 cannot be written as a linear combination of each other, that is, they are independent.

3.1. Model properties

The mathematical properties of the dynamic model (12) are:

1. The matrix \mathbf{H} is symmetric and positive definite, or $\mathbf{H} = \mathbf{H}^T > 0$;
2. The inverse of \mathbf{H} exists and is also positive definite, or $\exists \mathbf{H}^{-1} > 0$;
3. The matrix $\mathbf{F}(\mathbf{v}')$ is symmetric and positive definite, or $\mathbf{F}(\mathbf{v}') = \mathbf{F}^T > 0$, if $\theta_6 > -(\theta_5/i - \theta_3)iu$;
4. The matrix \mathbf{H} is constant if there is no change on the physical parameters of the robot;
5. The matrix $\mathbf{C}(\mathbf{v}')$ is skew symmetric;
6. The matrix $\mathbf{F}(\mathbf{v}')$ can be considered constant if $\theta_6 \gg |(\theta_5/i - \theta_3)iu|$ and there is no change on the physical parameters of the robot;
7. The mapping $\mathbf{v}_r \rightarrow \mathbf{v}'$ is strictly output passive if $\theta_6 > -(\theta_5/i - \theta_3)iu$ and $\Delta = 0$.

To analyze the above mathematical properties, first recall that $\theta_i > 0$ for $i = 1, 2, 4, 6$. Properties 1 and 2 can be confirmed by observing that \mathbf{H} is a diagonal square matrix formed by θ_1 and θ_2 . $\mathbf{F}(\mathbf{v}')$ is also a diagonal square matrix formed by θ_4 and $\theta_6 + (\theta_5/i - \theta_3)iu$. Property 3 holds if $\theta_6 > -(\theta_5/i - \theta_3)iu$. Property 4 holds if there is no change on the physical parameters of the robot (i.e., if there is no change on the robot's mass, moment of inertia, etc.). $\mathbf{C}(\mathbf{v}')$ is a square matrix formed by $\theta_3\omega$ and $-\theta_3\omega$, whose transpose is also its negative, which proves property 5. Property 6 holds if there is no change on the physical parameters of the robot and $\theta_6 \gg |(\theta_5/i - \theta_3)iu|$. Finally, the proof for property 7 is given in [21].

3.2. Identified parameters

The values of the dynamic parameters $\boldsymbol{\theta}$ can be estimated via an identification procedure, described as follows. Let a system be represented by the regression model

$$\mathbf{Y} = \mathbf{W}\boldsymbol{\theta}, \quad (15)$$

where $\boldsymbol{\theta}$ is the vector of parameters and \mathbf{Y} is the system output. The least squares estimate of $\boldsymbol{\theta}$ is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \mathbf{Y}, \quad (16)$$

where $\hat{\boldsymbol{\theta}}$ is the vector with the estimated values of $\boldsymbol{\theta}$ and \mathbf{W} is the regression matrix. By rearranging (4) and ignoring uncertainty, the dynamic model can be represented by

$$\begin{bmatrix} u_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} \dot{u} & 0 & -\omega^2 & u & 0 & 0 \\ 0 & \dot{\omega} & 0 & 0 & u\omega & \omega \end{bmatrix} [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6]^T, \quad (17)$$

where

$$\mathbf{Y} = \begin{bmatrix} u_r \\ \omega_r \end{bmatrix}, \mathbf{W} = \begin{bmatrix} \dot{u} & 0 & -\omega^2 & u & 0 & 0 \\ 0 & \dot{\omega} & 0 & 0 & u\omega & \omega \end{bmatrix}, \boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6]^T. \quad (18)$$

In order to obtain an estimate for the values of $\boldsymbol{\theta}$, each robot needs to be excited with speed reference signals (u_r, ω_r) , while the actual values of its velocities (u, ω) and accelerations $(\dot{u}, \dot{\omega})$ are measured and stored. In our case, the excitation signals consisted of a sum of six sine waves with different frequencies and amplitudes. All data were stored and the regression model was assembled so that the vector \mathbf{Y} and the matrix \mathbf{W} had all values obtained in each sampling instant. Subsequently, the value of $\boldsymbol{\theta}$ for each robot was calculated by least squares method.

In order to verify the assumptions that $\theta_6 \gg |(\theta_5/i - \theta_3)iu|$ and $\theta_6 > -(\theta_5/i - \theta_3)iu$, we have analyzed the dynamic parameters of five differential drive robots obtained via identification procedure. The analysis was done considering the parameters of the following robots: a Pioneer 3-DX with no extra equipment ($P3$), a Pioneer 3-DX with a LASER scanner and omnidirectional camera ($P3_{laser}$), a robotic wheelchair while carrying a 55 kg person (RW_{55}), a robotic wheelchair while carrying a 125kg person (RW_{125}), and a Khepera III ($KIII$). The Khepera III robot weighs 690 g, has a diameter of 13 cm and is 7 cm high. Its dynamic parameters were identified by Laut and were originally presented in [24]. By its turn, the Pioneer robots weigh about 9 kg, are 44 cm long, 38 cm wide, and 22 cm tall (without the LASER scanner). The LASER scanner weighs about 50% of the original robot weight, which produces an important change in the mass and moment of inertia of the structure. Finally, the robotic wheelchair presents an even greater difference in dynamics because of its own weight (about 70 kg) and the weight of the person that it is carrying. The dynamic parameters for the above-mentioned robots are presented in **Table 1**.

The value of u is limited to 0.5 m/s for the Khepera III robots, to 1.2 m/s for the Pioneer robots, and to 1.5 m/s for the robotic wheelchair. Therefore, using the values presented in **Table 1** one

	$P3$	$P3_{laser}$	RW_{55}	RW_{125}	$KIII$
$\theta_1 [s]$	0.5338	0.2604	0.3759	0.4263	0.0228
$\theta_2 [s]$	0.2168	0.2509	0.0188	0.0289	0.0568
$\theta_3 [sm/rad^2]$	-0.0134	-0.0005	0.0128	0.0058	-0.0001
θ_4	0.9560	0.9965	1.0027	0.9883	1.0030
$\theta_5 [s/m]$	-0.0843	0.0026	-0.0015	0.0134	0.0732
θ_6	1.0590	1.0768	0.9808	0.9931	0.9981

Table 1. Identified dynamic parameters of a Pioneer 3-DX with no extra equipment ($P3$), a Pioneer 3-DX with a LASER scanner ($P3_{laser}$), a robotic wheelchair while carrying a 55 kg person (RW_{55}), a robotic wheelchair while carrying a 125 kg person (RW_{125}), and a Khepera III ($KIII$).

can verify that the conditions of $\theta_6 > -(\theta_5/i - \theta_3)iu$ and $\theta_6 \gg |(\theta_5/i - \theta_3)iu|$ are valid for all sets of identified parameters. Therefore, the dynamic model of the above-mentioned robots can be represented as in (12), with properties 1–7 valid under the considered conditions.

4. Controller design

To illustrate the usefulness of the modified model and its properties, in this section we show the design of a trajectory tracking controller and a dynamic compensation controller. The controller design is split in two parts, as in [7]. The first part is based on the inverse kinematics and the second one compensates for the robot dynamics. The use of the dynamic model properties is shown on the second part.

The control structure is shown in **Figure 2**, where blocks **K**, **D**, and **R** represent the kinematic controller, the dynamic compensation controller, and the robot, respectively. **Figure 2** shows that the kinematic controller receives the desired values of position $\mathbf{h}_d = [x_d \ y_d]^T$ and velocity $\dot{\mathbf{h}}_d$ from the trajectory planner (which is not considered in this work). Then, based on those values and on the actual robot position $\mathbf{h} = [x \ y]^T$ and orientation ψ , the kinematic controller calculates the desired robot velocities $\mathbf{v}_d = [u_d \ \omega_d]^T$. The desired velocities \mathbf{v}_d and the actual robot velocities $\mathbf{v} = [u \ \omega]^T$ are fed into the dynamic controller. Such controller uses those values and the estimates of the robot parameters $\boldsymbol{\theta}$ to generate the velocity commands $\mathbf{v}_r = [u_r \ \omega_r]^T$ that are sent as references to the robot internal controller.

4.1. Kinematic controller

The same kinematic controller presented in [7, 21] is shown here. It is a trajectory tracking controller based on the inverse kinematics of the robot. If only the position of the point of interest $\mathbf{h} = [x \ y]^T$ is considered, the robot's inverse kinematics can be written as

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\frac{1}{a} \sin \psi & \frac{1}{a} \cos \psi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \quad (19)$$

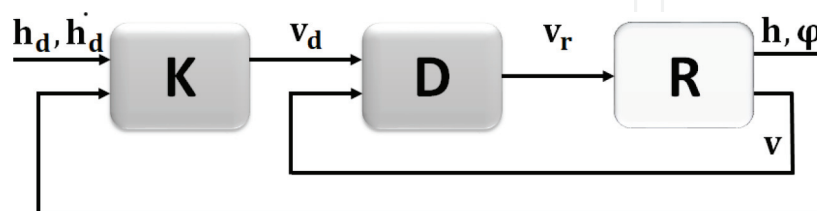


Figure 2. Structure of the control system. The kinematic controller **K** receives the desired values of position \mathbf{h}_d and velocity $\dot{\mathbf{h}}_d$, the actual robot position \mathbf{h} and its orientation ψ , and calculates the desired robot velocities \mathbf{v}_d . Those values and the actual robot velocities \mathbf{v} are fed into the dynamic compensation controller **D**, that generates the velocity commands \mathbf{v}_r that are sent as references to the robot **R**.

The inverse kinematics described by Eq. (19) is valid only for $a \neq 0$. This is the reason why we prefer to adopt this model instead of the classical unicycle model, as discussed earlier. Considering (19), the adopted control law is

$$\begin{bmatrix} u_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\frac{1}{a} \sin \psi & \frac{1}{a} \cos \psi \end{bmatrix} \begin{bmatrix} \dot{x}_d + l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) \\ \dot{y}_d + l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \end{bmatrix}, \quad (20)$$

for which $\mathbf{v}_d = [u_d \ \omega_d]^T$ is the vector of desired velocities given by the kinematic controller; $\mathbf{h} = [x \ y]^T$ and $\mathbf{h}_d = [x_d \ y_d]^T$ are the vectors of actual and desired coordinates of the point of interest h , respectively; $\tilde{\mathbf{h}} = [\tilde{x} \ \tilde{y}]^T$ is the vector of position errors given by $\mathbf{h}_d - \mathbf{h}$; $k_x > 0$ and $k_y > 0$ are the controller gains; $l_x, l_y \in \mathbb{R}$ are saturation constants; and $a > 0$. The \tanh terms are included to limit the values of the desired velocities \mathbf{v}_d to avoid saturation of the robot actuators in case the position errors $\tilde{\mathbf{h}}$ are too big, considering $\dot{\mathbf{h}}_d$ is appropriately bounded.

It is important to point out that the orientation of a DDMR is always tangent to the path being followed. Moreover, the desired trajectory defines the desired linear speed u_d , which means that the robot will be moving either forward or backwards. Therefore, it is not necessary for the controller to explicitly control the robot's orientation to make it successfully follow a trajectory with a desired orientation.

For the stability analysis of the kinematic controller, it is supposed a perfect velocity tracking, which allows equating (19) and (20) under the assumption of $u \equiv u_d$ and $\omega \equiv \omega_d$, which means that the dynamic effects are, at this moment, ignored. Then, the closed-loop equation is obtained in terms of the velocity errors, which is

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \end{bmatrix} + \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix} \begin{bmatrix} \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) \\ \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (21)$$

Now, the output error vector $\tilde{\mathbf{h}}$ (21) can be written as

$$\dot{\tilde{\mathbf{h}}} = - \begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) & l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \end{bmatrix}^T, \quad (22)$$

which has an unique equilibrium point at the origin. To conclude the stability analysis of such equilibrium, $V = \frac{1}{2} \tilde{\mathbf{h}}^T \tilde{\mathbf{h}} > 0$ is considered as the Lyapunov's candidate function. Its first time derivative is

$$\dot{V} = \tilde{\mathbf{h}}^T \dot{\tilde{\mathbf{h}}} = -\tilde{x} l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) - \tilde{y} l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) < 0, \forall \tilde{\mathbf{h}}. \quad (23)$$

Regarding these results, one can immediately conclude that the system characterized so far has a globally asymptotically stable equilibrium at the origin, which means that the position errors

$\tilde{x}(t) \rightarrow 0$ and $\tilde{y}(t) \rightarrow 0$ as $t \rightarrow \infty$. This result will be revisited later, after adding a dynamic controller to the system in order to implement the whole control scheme.

Remark. Considering the case in which the reference is a fixed destination point, instead of a trajectory, the robot reaches such a point and stops there. Assuming $u \equiv u_d$ and $\omega \equiv \omega_d$, Eq. (20) guarantees that $\omega = 0$ when $\tilde{x} = 0$ and $\tilde{y} = 0$, therefore $\psi(t) \rightarrow \psi_{\text{constant}}$.

4.2. Dynamic compensation controller

Now, the use of the proposed dynamic model and its properties is illustrated via the design of a dynamic compensation controller. It receives the desired velocities \mathbf{v}_d from the kinematic controller and generates a pair of linear and angular velocity references \mathbf{v}_r for the robot servos, as shown in **Figure 2**. First, let us define the vector of modified velocities \mathbf{v}'_d as

$$\mathbf{v}'_d = \begin{bmatrix} u'_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_d \\ \omega_d \end{bmatrix}, \quad (24)$$

and the vector of velocity errors is given by $\tilde{\mathbf{v}}' = \mathbf{v}'_d - \mathbf{v}'$.

Regarding parametric uncertainties, the proposed dynamic compensation control law is

$$\mathbf{v}_r = \hat{\mathbf{H}}(\dot{\mathbf{v}}'_d + \mathbf{T}(\tilde{\mathbf{v}}')) + \hat{\mathbf{C}}\mathbf{v}'_d + \hat{\mathbf{F}}\mathbf{v}'_d, \quad (25)$$

where $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{F}}$ are estimates of \mathbf{H} , \mathbf{C} , and \mathbf{F} , respectively, $\mathbf{T}(\tilde{\mathbf{v}}') = \begin{bmatrix} l_u & 0 \\ 0 & l_\omega \end{bmatrix} \begin{bmatrix} \tanh\left(\frac{k_u}{l_u} \tilde{u}\right) \\ \tanh\left(\frac{k_\omega}{l_\omega} \tilde{\omega}\right) \end{bmatrix}$, $k_u > 0$ and $k_\omega > 0$ are gain constants, $l_u \in \mathbb{R}$ and $l_\omega \in \mathbb{R}$ are saturation constants, and $\tilde{\omega} = \omega_d - \omega$ and $\tilde{u} = u_d - u$ are the current velocity errors. The term $\mathbf{T}(\tilde{\mathbf{v}}')$ provides a saturation in order to guarantee that the commands to be sent to the robot are always below the corresponding physical limits, considering that \mathbf{v}'_d and $\dot{\mathbf{v}}'_d$ are bounded to appropriate values.

In this chapter, we consider that the dynamic parameters are exactly known, that is, $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$. This means that $\hat{\mathbf{H}} = \mathbf{H}$, $\hat{\mathbf{C}} = \mathbf{C}$, and $\hat{\mathbf{F}} = \mathbf{F}$. The analysis considering parametric error is presented in [7, 21].

Using the Lyapunov candidate function $V = \frac{1}{2} \tilde{\mathbf{v}}'^T \mathbf{H} \tilde{\mathbf{v}}' > 0$, and considering that the dynamic parameters are constant, one has

$$\dot{V} = -\tilde{\mathbf{v}}'^T \mathbf{H} \mathbf{T}(\tilde{\mathbf{v}}') - \tilde{\mathbf{v}}'^T \mathbf{C} \tilde{\mathbf{v}}' - \tilde{\mathbf{v}}'^T \mathbf{F} \tilde{\mathbf{v}}'. \quad (26)$$

Observing property 5, of antisymmetry of \mathbf{C} , the derivative of the Lyapunov function can be written as

$$\dot{V} = -\tilde{\mathbf{v}}'^T \mathbf{H} \mathbf{T}(\tilde{\mathbf{v}}') - \tilde{\mathbf{v}}'^T \mathbf{F} \tilde{\mathbf{v}}'. \quad (27)$$

According to Property 1, \mathbf{H} is symmetric and positive definite. The terms of $\mathbf{T}(\tilde{\mathbf{v}}')$ have the same sign of the terms of $\tilde{\mathbf{v}}'$. Property 3 states that \mathbf{F} is symmetric and positive definite if $\theta_6 > -(\theta_5/I - \theta_3)lu$, condition that was shown to hold for our robot. Therefore, one can conclude that $\dot{V} < 0$, that is, $\tilde{\mathbf{v}}' \in L_\infty$ and $\tilde{\mathbf{v}}' \rightarrow \mathbf{0}$ with $t \rightarrow \infty$, and $\tilde{\mathbf{v}} \in L_\infty$ and $\tilde{\mathbf{v}} \rightarrow \mathbf{0}$ with $t \rightarrow \infty$.

Regarding the kinematic controller, it has been shown [7] that a sufficient condition for the asymptotic stability is

$$\|\tilde{\mathbf{h}}\| > \frac{\|\mathbf{A}\tilde{\mathbf{v}}\|}{\min}(k_x, k_y), \quad (28)$$

where $\mathbf{A} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \end{bmatrix}$. Because $\tilde{\mathbf{v}}(t) \rightarrow \mathbf{0}$, the condition (28) is asymptotically verified for any value of $\tilde{\mathbf{h}}$. Consequently, the tracking control error $\tilde{\mathbf{h}}(t) \rightarrow \mathbf{0}$, thus accomplishing the control objective.

To sum up, by using a control structure as shown in **Figure 2** with a dynamic compensation controller given by Eq. (25), different motion controllers can be applied. In our example, the trajectory tracking controller given by Eq. (20) was used. This is the system that we have implemented and for which we present some experimental results in Section 5.

5. Experimental results

In this section, we present some experimental results using a Pioneer 3-DX, from Adept Mobile Robots. In all experiments, the robot starts at position $(0.0, 0.0)m$ with orientation 0° , and should follow an 8-shape trajectory also starting at $(0.0, 0.0)m$. The trajectory to be followed by the robot is represented by a sequence of desired positions \mathbf{h}_d and velocities $\dot{\mathbf{h}}_d$, both varying in time. The reference path is illustrated in **Figure 3**.

We have implemented the control structure shown on **Figure 2** using the control laws given by Eqs. (20) and (25). In total, we have executed 10 experiments for each controller, from now on referred to as KC (kinematics controller) and DC (dynamic compensation). In the case of KC, the robot receives as commands the values \mathbf{v}_d calculated by the kinematics controller and there is no dynamic compensation. On the other hand, in the case of DC, the dynamic compensation controller is active and the robot receives as commands the values of \mathbf{v}_r calculated by the dynamic compensation controller. We have repeated the experiments for four cases: KC with load, KC without load, DC with load, and DC without load. The load consists of a weight of 24.8 kg placed on top of the robot, while the original weight of the robot is 10.4 kg.

The following parameters were used in all experiments: $a = 0.15$ m, sample time of 0.1 s (this is the sample time of the Pioneer 3-DX); controller gains $k_x = 0.1$, $k_y = 0.1$, $k_u = 4$, $k_w = 4$, and saturation constants $l_x = 0.1$, $l_y = 0.1$, $l_u = 1$, $l_w = 1$. The robot used in the experiments is a Pioneer 3-DX without LASER scanner, therefore the parameters used in the dynamic compensation controller are the ones in column P3 from **Table 1**.

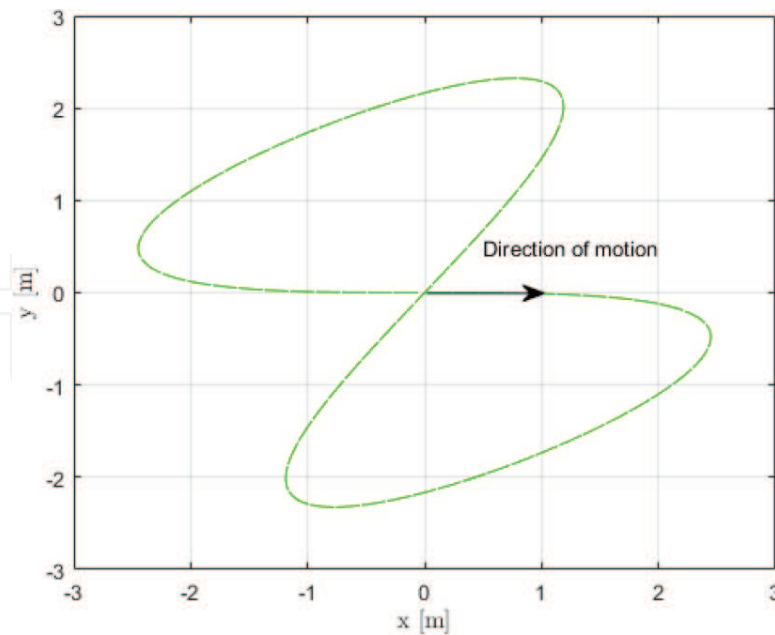


Figure 3. 8-shape reference path to be followed by the robot. Initial reference position is $(0.0, 0.0)m$ and the direction of motion is indicated in the figure. The robot starts at position $(0.0, 0.0)m$ with orientation 0° .

Figure 4 illustrates the results of 2 experiments, both without load. **Figure 4(a)** shows the 8-shape path followed by the robot without load when controlled by KC and DC. Robot path was recovered through its odometry. One can notice that the path followed by the robot is slightly different under KC or DC. The robot's linear and angular velocities also change along the path, as shown in **Figure 4(b)**.

A better visualization of the tracking error is given by **Figure 5**, which shows the evolution of the distance error during the experiments without load. The distance error is defined as the

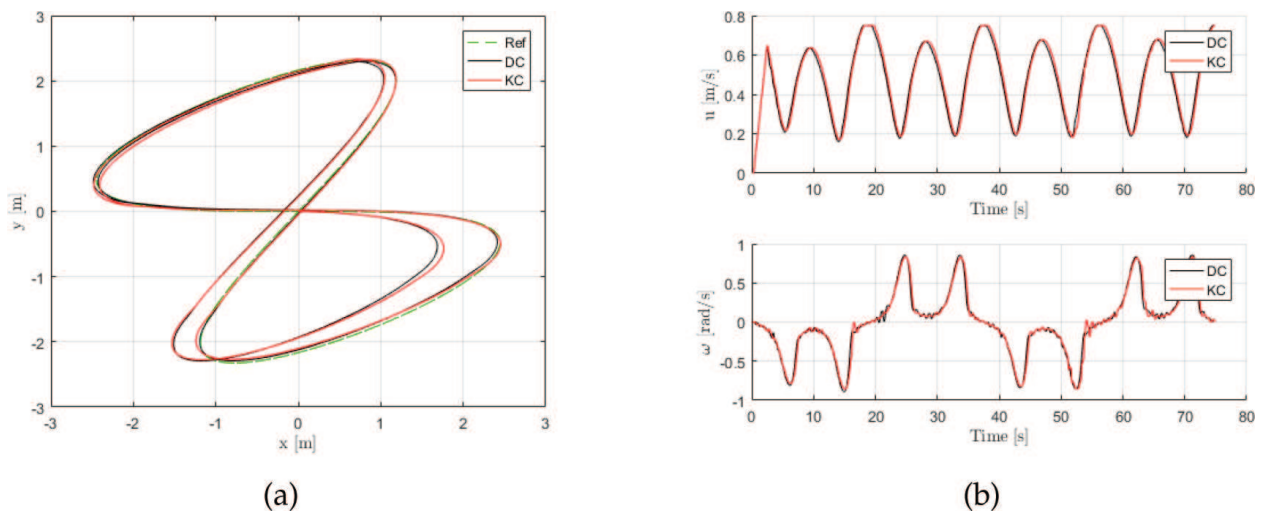


Figure 4. Experiments without load: (a) robot path; (b) linear and angular velocities. In all graphs, the black line represents the results for the case in which the dynamic compensation (DC) is active, while the red line represents the results for the kinematic controller (KC).

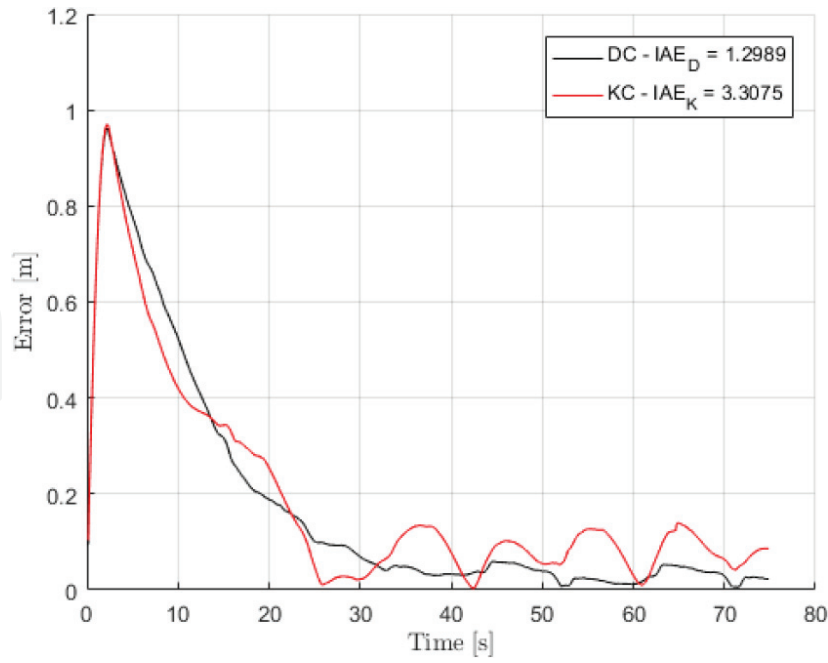


Figure 5. Evolution of tracking error without load. The black line represents the error for the case in which the dynamic compensation (DC) is active, while the red line represents the error for the kinematic controller (KC). The corresponding values of IAE_{30} for this experiment are also shown in the figure.

instantaneous distance between the desired position \mathbf{h}_d and the actual robot position \mathbf{h} . It can be noticed that the distance error is similar for KC and DC in the first part of the path. At the beginning of the experiment, the tracking error increases quite a lot, reaching almost 1.0 m. This happens because the robot needs to accelerate from zero to catch up with the reference trajectory. After a few seconds, the error starts to decrease and around 25 – 30 s, the robot follows the trajectory at normal speed. From this point on, it is clear that the average error is smaller when the DC is active.

Figure 6(a) shows the 8-shape path followed by the robot when carrying the load and controlled by KC and DC. One can notice that the path followed by the robot is slightly different under KC or DC, and there is more distortion in the path when compared to the case in which the robot carries no load. The robot's linear and angular velocities also change along the path, as shown in **Figure 6(b)**, and are very similar to the previous case.

The tracking error is given by **Figure 7**, which shows the evolution of the distance error during the experiments with load. As before, the robot needs to accelerate from zero to catch up with the reference trajectory, which causes the tracking error to increase in the first part of the experiments. But, in this case, the error in the first part of the experiment is actually higher for DC. This happens because the dynamic parameters used in the dynamic compensation controller remained unchanged during all experiments, with and without load. This means that the case in which the robot is carrying load is unfavorable for the dynamic compensation controller because the dynamics is not properly compensated, causing the error to increase. Even so, after about 30 s, the tracking error of DC gets smaller than the error for KC.

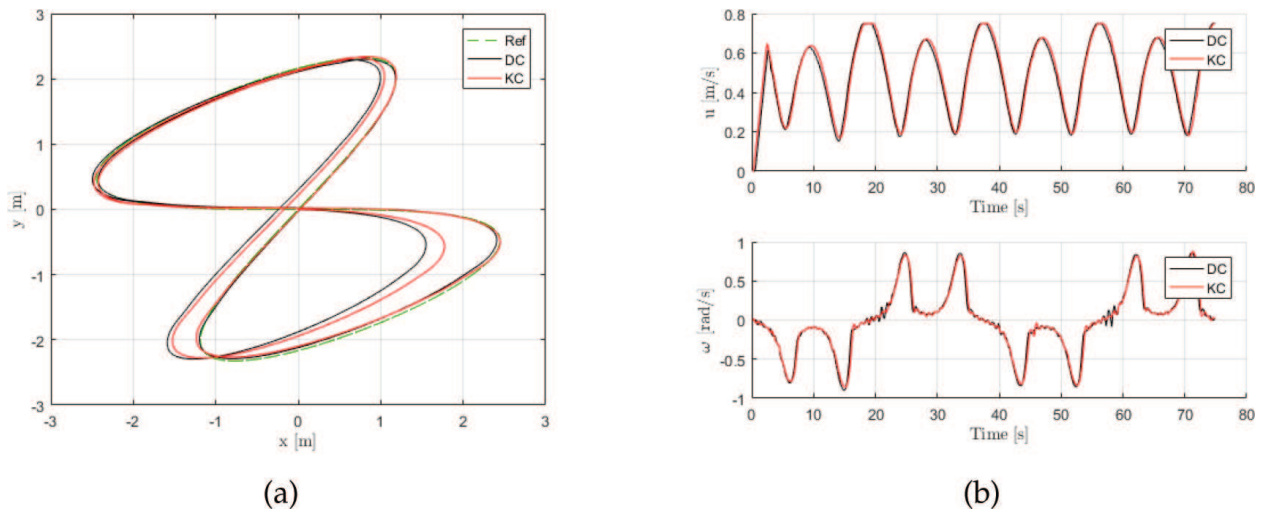


Figure 6. Experiments with load: (a) robot path; (b) linear and angular velocities. In all graphs the black line represents the results for the case in which the dynamic compensation (DC) is active, while the red line represents the results for the kinematic controller (KC).

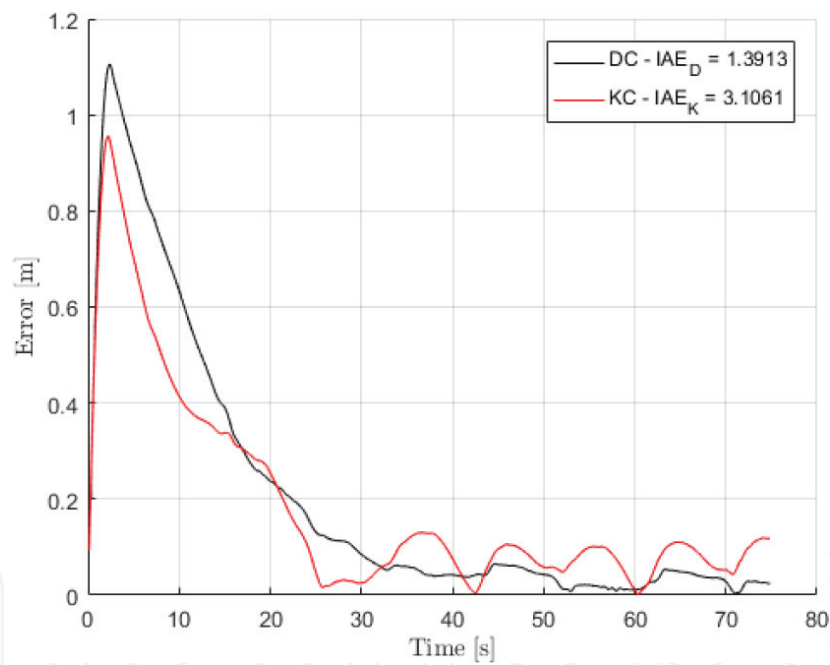


Figure 7. Evolution of tracking error with load. The black line represents the error for the case in which the dynamic compensation (DC) is active, while the red line represents the error for the kinematic controller (KC). The corresponding values of IAE_{30} for this experiment are also shown in the figure.

To evaluate the performance of the system we have calculated the IAE performance index, where $IAE = \int_{t_1}^{t_2} E(t)dt$, $E(t) = \sqrt{\hat{x}^2 + \hat{y}^2}$ is the instantaneous distance error and $t_2 - t_1$ is the period of integration. The average and standard deviation values of IAE for all experiments are reported in **Table 2**. There, IAE_{tot} was calculated considering $t_2 = 75$ s and $t_1 = 0$, that is, for the total period of each experiment. By its turn, the value of IAE_{30} was calculated only for the

	With load		Without load	
	IAE_{tot}	IAE_{30}	IAE_{tot}	IAE_{30}
Kinematic controller	14.30 ± 0.66	3.10 ± 0.05	14.08 ± 0.18	3.11 ± 0.12
Dynamic compensation	15.39 ± 0.42	1.41 ± 0.13	13.05 ± 0.07	1.29 ± 0.02

Here, $IAE = \int_{t_1}^{t_2} |E(t)| dt$, $E(t) = \sqrt{\tilde{x}^2 + \tilde{y}^2}$ is the instantaneous distance error, and $t_2 - t_1$ is the period of integration. For IAE_{tot} , $t_2 = 75$ s and $t_1 = 0$. For IAE_{30} , $t_2 = 75$ s and $t_1 = 35$ s.

Table 2. Average and standard deviation of IAE performance index calculated for experiments with and without load (lower value is better, highlighted in bold).

final 30 seconds of each experiment, that is, considering $t_2 = 75$ s and $t_1 = 35$ s. Therefore, IAE_{30} gives a good indication of the performance of the system after the error due to initial acceleration have faded out. From the results highlighted in bold in **Table 2**, it is clear that the performance of the system with the dynamic compensation controller is better in the long run because the correspondent values of IAE_{30} are about 50% of those for the kinematic controller. This is true even for the case in which the robot is carrying load.

It is important to emphasize that the dynamic parameters used in the dynamic compensation controller remained unchanged during all experiments, which means that the dynamics is not properly compensated when carrying load. This is illustrated by the fact that IAE_{tot} is bigger when the dynamic compensation is active and the robot is carrying load. Even so, in our experiments the performance was better in the long run when the dynamic compensation controller remained active.

One should notice that an increase in controller gains k_x and k_y could result in better performance (smaller tracking error), especially when the robot is carrying load. Nevertheless, we kept the same values of controller gains during all experiments to be able to compare the results.

6. Conclusion

In this chapter, we illustrate that the performance (in term of IAE) of a motion control system for a mobile robot can be up to 50% better under certain conditions when dynamic compensation is included. Such dynamic compensation can be implemented as shown in **Figure 2**, in which Eq. (25) is used with parameters identified via the procedure described in Section 3.2.

It is worth mentioning that the values of controller gains used in the experiments here reported were not optimum. The values of the gains were chosen empirically so that we could compare different cases. Optimization of controller gains can be executed to reduce tracking error, energy consumption, or a weighted combination of both, as shown in [25]. This means that the performance of the overall system could potentially be better than reported here.

We also presented a formulation of a dynamic model for differential-drive mobile robots, and discussed its mathematical properties. When compared to the classical dynamic model based on torques, the model used in this chapter has the advantages of accepting velocities as inputs,

and modeling the dynamics of the robot's actuators. We have shown that such model and its properties are useful on the design and stability analysis of a dynamic compensation controller for a differential-drive mobile robot. Moreover, because the mathematical structure of (12) is similar to the classical torque-based model, classical strategies for controller design [8, 26] can be adapted for designing controllers for mobile robots using the model presented in this chapter.

The dynamic model presented in this chapter can be used in connection with other kinematic controllers designed for commercial mobile robots, even in the context of coordinated control of multi-robot formations [27]. This integration requires no change on the original controller equations since the dynamic model accepts the same velocity commands as commercial robots. We invite the interested reader to download our toolbox for MATLAB/Simulink®, which include blocks to simulate the differential-drive kinematics and dynamics, a kinematic controller and two dynamic compensation controllers, one of which being the one presented in this chapter [28].

Acknowledgements

The authors thank the Institute of Engineering, Hanze University of Applied Sciences, for the partial financial support given for the publication of this chapter.

Author details

Felipe Nascimento Martins^{1*} and Alexandre Santos Brandão²

*Address all correspondence to: fe.nascimento.martins@pl.hanze.nl

1 Institute of Engineering, Hanze University of Applied Sciences, Assen, The Netherlands

2 Department of Electrical Engineering, Federal University of Viçosa, Viçosa, Brazil

References

- [1] Siegwart R, Nourbakhsh IR, Scaramuzza D. Introduction to Autonomous Mobile Robots. 2nd ed. London, England: MIT Press; 2011
- [2] Birk A, Kenn H. RoboGuard, a teleoperated mobile security robot. Control Engineering Practice. 2002;**10**:1259-1264
- [3] Prassler E et al. A short history of cleaning robots. Autonomous Robots. 2000;**9**(3):211-226
- [4] Stouten B, de Graaf AJ. Cooperative transportation of a large object-development of an industrial application. In: IEEE International Conference on Robotics and Automation. Vol. 3. 2004. pp. 2450-2455. New Orleans, LA, USA: IEEE; DOI: 10.1109/ROBOT.2004.1307428
- [5] Andaluz VH et al. Robust control with dynamic compensation for human-wheelchair system. In: Intelligent Robotics and Applications. Switzerland: Springer; 2014. pp. 376-389

- [6] Morin P, Samson C. Motion control of wheeled mobile robots. In: Springer Handbook of Robotics. Heidelberg, Germany: Springer; 2008. pp. 799-826
- [7] Martins FN et al. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*. 2008;**16**:1354-1363. DOI: 10.1016/j.conengprac.2008.03.004
- [8] Fierro R, Lewis FL. Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics. *Journal of Robotic Systems*. 1997;**14**(3):149-163
- [9] Das T, Kar IN. Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. *IEEE Transactions on Control Systems Technology*. 2006;**14**(3):501-510
- [10] Lapierre L, Zapata R, Lepinay P. Combined path-following and obstacle avoidance control of a wheeled robot. *The Int. Journal of Robotics Research*. 2007;**26**(4):361-375
- [11] Shojaei K, Mohammad Shahri A, Tarakameh A. Adaptive feedback linearizing control of nonholonomic wheeled mobile robots in presence of parametric and nonparametric uncertainties. *Robotics and Computer-Integrated Manufacturing*. 2011;**27**(1):194-204
- [12] Wang K. Near-optimal tracking control of a Nonholonomic mobile robot with uncertainties. *International Journal of Advanced Robotic Systems*. 2012;**9**(66):1-11
- [13] Do KD. Bounded controllers for global path tracking control of unicycle-type mobile robots. *Robotics and Autonomous Systems*. 2013;**61**(8):775-784. ISSN: 0921-8890. <https://doi.org/10.1016/j.robot.2013.04.014>
- [14] Onat A, Ozkan M. Dynamic adaptive trajectory tracking control of nonholonomic mobile robots using multiple models approach. *Advanced Robotics*. 2015;**29**(14):913-928
- [15] Antonini P, Ippoliti G, Longhi S. Learning control of mobile robots using a multiprocessor system. *Control Engineering Practice*. 2006;**14**:1279-1295
- [16] De La Cruz C, Carelli R. Dynamic model based formation control and obstacle avoidance of multi-robot systems. *Robotica*. 2008;**26**(03):345-356
- [17] Chen CY et al. Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots. *Mechatronics*. 2009;**19**(2):156-166. ISSN: 0957-4158
- [18] De La Cruz C, Celeste WC, Bastos-Filho TF. A robust navigation system for robotic wheelchairs. *Control Engineering Practice*. 2011;**19**(6):575-590
- [19] Rossomando FG, Soria C, Carelli R. Adaptive neural sliding mode compensator for a class of nonlinear systems with unmodeled uncertainties. *Engineering Applications of Artificial Intelligence*. 2013;**26**(10):2251-2259
- [20] Utstumo T, Berge TW, Gravdahl JT. Non-linear model predictive control for constrained robot navigation in row crops. In: *IEEE International Conference on Industrial Technology (ICIT 2015)*. Seville, Spain: IEEE; 2015. DOI: 10.1109/ICIT.2015.7125124

- [21] Martins FN, Sarcinelli-Filho M, Carelli R. A velocity-based dynamic model and its properties for differential drive mobile robots. *Journal of Intelligent & Robotic Systems*. 2017; **85**(2):277-292
- [22] Fukao T, Nakagawa H, Adachi N. Adaptive tracking control of a Nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*. 2000;**16**(5):609-615
- [23] Kim MS, Shin JH, Lee JJ. Design of a robust adaptive controller for a mobile robot. In: *Proc. of the IEEE/RSJ Int. Conf. On Intelligent Robots and Systems*. Vol. 3. Takamatsu, Japan: IEEE; 2000. pp. 1816-1821. DOI: 10.1109/IROS.2000.895235
- [24] Laut J. A dynamic parameter identification method for migrating control strategies between heterogeneous wheeled mobile robots. [PhD thesis]. Worcester Polytechnic Institute; 2011
- [25] Martins FN, Almeida GM. Tuning a velocity-based dynamic controller for unicycle mobile robots with genetic algorithm. In: *Jornadas Argentinas de Robótica*. Olavarría, Argentina: Universidad Nacional del Centro de la Provincia de Buenos Aires; 2012
- [26] Spong MW, Hutchinson S, Vidyasagar M. *Robot Modeling and Control*. Vol. 3. New York: Wiley; 2006
- [27] Brandao AS et al. A multi-layer control scheme for multi-robot formations with adaptive dynamic compensation. In: *IEEE International Conference on Mechatronics*, 2009. ICM 2009. Malaga, Spain: IEEE; 2009
- [28] F.N. Martins. Velocity-based Dynamic Model and Adaptive Controller for Differential Steered Mobile Robot. 2017. Available from: <http://www.mathworks.com/matlabcentral/fileexchange/44850>

