Tuning a Velocity-based Dynamic Controller for Unicycle Mobile Robots With Genetic Algorithm

Felipe Nascimento Martins and Gustavo Maia Almeida
NERA - Núcleo de Estudos em Robótica e Automacão
Federal Institute of Education, Science and Technology of Espirito Santo - IFES, Campus Serra
Rod. ES-010 km 6,5 - Manguinhos, Serra, ES, Brazil. 29173-087
{felipemartins, gmaia}@ifes.edu.br

***Abstract*— This paper addresses the application of genetic algorithm in the tuning problem of an adaptive controller for unicycle-like mobile robots. The motivation for this work comes from the fact that these robots are characterized by a complex dynamical model which makes the space of the possible controller gains difficult to search. First, the paper presents the design of the robot tracking controller that is based on a dynamic model that has linear and angular velocities as inputs, which is not usual in the literature for dynamic models of mobile robots. A brief explanation of the dynamic model and its properties is provided. Then, it shows the use of a genetic algorithm in the selection of the controller gains. Simulation results are presented and illustrate that the proposed system can be successfully applied to select controller gains for a robot that tracks a desired trajectory, either minimizing tracking error, energy consumption or a combination of both indexes.**

I. INTRODUCTION

Unicycle-like mobile robots are frequently adopted to accomplish several tasks, due to their good mobility and simple configuration. Nonlinear control for this type of robot has been studied for several years, and so far it is an important research field [1]. The unicycle structure has been used in various robotic applications, such as surveillance [2], floor cleaning [3], industrial load transportation using automated guided vehicles (AGVs) [4], and autonomous wheelchairs [5]. For this kind of robot, some authors have addressed the problem of trajectory tracking, a quite important functionality that allows a mobile robot to describe a desired trajectory when accomplishing a task. Some of the controllers designed so far are based only on the kinematics of the mobile robot. However, to perform tasks that require high speed movements and/or heavy load transportation, it is fundamental to consider the robot dynamics, besides its kinematics. Thus, there are some studies that present the design of controllers that compensate for the robot dynamics. As an example, [6] present a robust-adaptive controller based on neural networks that deals with disturbances and non-modeled dynamics and generate torque control law for nonholonomic mobile robots. In [7] it is proposed a fuzzy logic-based adaptive controller in which the system uncertainty is estimated by a fuzzy logic system and its parameters are tuned on-line. There, the dynamic model includes the actuator dynamics, and the commands generated are voltages for the robot motors. Other types of trajectory tracking controllers based on the robot dynamics are developed in [8]–[12].

The control signals generated by most of the dynamic controllers presented in the literature are torques or voltages for the robot motors, like in most of the above mentioned works, while commercial robots usually accept velocity commands. In such a context, [13] presented the development of a dynamic model that has the advantage of using linear and angular velocities as inputs. This model was used in the development of an adaptive dynamic controller presented in [14]. An evolution of such dynamic model was presented in [11], that also shows some important model properties and presents the design of an adaptive dynamic compensation controller for unicycle robots.

One of the problems regarding the above mentioned dynamic controllers is how to select a proper set of controller gains for the robot to have good performance. First of all, it is important to define what ”good performance” means, and how it can be computed, in order to make the selection of a proper set of controller gains. The use of evolutionary algorithms to solve optimization problems, including controller gain settings, is a well studied problem [15]. The problem of selecting a set of controller gains is addressed in [16], where the authors present three types of evolutionary optimization algorithms for PI controller tuning. Following the same idea, in [17] the authors present a tuning method for a multivariable predictive control of a hot rolling mill with genetic algorithm.

The above mentioned works have shown that system performance can be improved via evolutionary optimization algorithms. Therefore, in this paper we present the results of the application of a genetic algorithm used to select a set of controller gains for a mobile robot trajectory tracking dynamic controller. The controller we use is based on the approach to model the dynamics of an unicycle-like mobile robot presented in [11]. This dynamic model is not common in the literature, as it has linear and angular velocities as inputs. It is developed in such a way that important properties arise. Such properties are also summarized in the present paper, and some of them are used to design an adaptive compensation controller for the unicycle mobile robot, with proven stable performance. Selection of the controller gains is performed to minimize tracking error (IAE), energy consumption and a balanced combination of both indexes.

Simulation results illustrate system performance for some of the selected controller gains. Simulation files are available for download and can be freely used and modified [18].

The paper is hereinafter organized in the following sections: Section II, which presents the mathematical representation of the unicycle-like dynamic model and its properties; Section III, where the kinematic and dynamic controllers are developed; Section IV, that presents genetic algorithm fundamentals; Section V, which presents simulation results and a comparison of system performance for three different sets of controller gains; and, finally, Section VI, where some conclusions are highlighted.

II. DYNAMIC MODEL

The dynamic model of the unicycle-like mobile robot proposed in [11] is now reviewed. It's structure is similar to the classical dynamic equation of a manipulator. Fig. 1 depicts a unicycle-like mobile robot and the parameters and variables of interest. There, u and ω are, respectively, the linear and angular velocities, *G* is the center of mass, *C* is the position of the castor wheel, h is the point of interest with coordinates x and y in the XY plane, ψ is the robot orientation, and a is the distance from the point of interest to the point in the middle of the virtual axis linking the traction wheels (point *B*).

The complete robot mathematical model is written as a kinematic and a dynamic models, as shown in the sequel. The kinematic model is

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

while the dynamic model is

$$\begin{bmatrix}\dot{u}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}\frac{\theta_3}{\theta_1}\omega^2 - \frac{\theta_4}{\theta_1}u\\ -\frac{\theta_5}{\theta_2}u\omega - \frac{\theta_6}{\theta_2}\omega\end{bmatrix} + \begin{bmatrix}\frac{1}{\theta_1} & 0\\ 0 & \frac{1}{\theta_2}\end{bmatrix}\begin{bmatrix}u_{ref}\\ \omega_{ref}\end{bmatrix} + \begin{bmatrix}\delta_u\\ \delta_\omega\end{bmatrix}, \quad (2)$$

where $\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^T$ is the vector of model parameters (which are identified) and δ_x, δ_y, δ_u, and δ_ω are parametric uncertainties associated to the mobile robot.

The parameters included in the vector $\boldsymbol{\theta}$ are functions of some physical parameters of the robot, such as its mass m, its moment of inertia I_z at *G*, the electrical resistance R_a of its motors, the electromotive constant k_b of its motors, the constant of torque k_a of its motors, the coefficient of friction B_e, the moment of inertia I_e of each group rotor-reduction gear-wheel, the radius r of the wheels, and the distances b and d (see Fig. 1). It is assumed that the robot servos have PD controllers to control the velocities of each motor, 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 motors associated to both driven wheels have the same characteristics, and that their inductances are neglectable. The equations describing the parameters θ_i were firstly presented in [13], and are

$$\theta_1 = \left[\frac{R_a}{k_a}\left(mR_t r + 2I_e\right) + 2rk_{DT}\right]\frac{1}{2rk_{PT}} \ [s],$$

$$\theta_2 = \frac{\left[\frac{R_a}{k_a}\left(I_e d^2 + 2R_t r\left(I_z + mb^2\right)\right) + 2rdk_{DR}\right]}{(2rdk_{PR})} \ [s],$$

$$\theta_3 = \frac{R_a}{k_a}\frac{mbR_t}{2k_{PT}} \ [sm/rad^2],$$

$$\theta_4 = \frac{R_a}{k_a}\left(\frac{k_a k_b}{R_a} + B_e\right)\frac{1}{rk_{PT}} + 1 \ [1],$$

$$\theta_5 = \frac{R_a}{k_a}\frac{mbR_t}{dk_{PR}} \ [s/m],$$

$$\theta_6 = \frac{R_a}{k_a}\left(\frac{k_a k_b}{R_a} + B_e\right)\frac{d}{2rk_{PR}} + 1 \ [1].$$

It should be noticed that $\theta_i > 0$ for $i = 1, 2, 4, 6$. The parameters θ_3 and θ_5, by their turn, can be negative and will be null if, and only if, the center of mass *G* is exactly in the middle of the virtual axis linking the traction wheels (point *B*), i.e. $b = 0$. In this paper, however, it is assumed that $b \neq 0$.

Remark 1. The dynamics of the robot actuators is included in the presented model, which is not the case for the model that has torques as inputs.

By rearranging the terms, (2) can be written as

$$\boldsymbol{\Delta} + \mathbf{H}\dot{\mathbf{v}}' + \mathbf{C}(\mathbf{v}')\mathbf{v}' + \mathbf{F}(\mathbf{v}')\mathbf{v}' = \mathbf{v_r}, \qquad (3)$$

where $\mathbf{v_r} = [u_{ref} \quad \omega_{ref}]^T$ is the vector of reference velocities, $\mathbf{v} = [u \quad \omega]^T$ is the vector containing the current robot velocities, $\mathbf{v}' = [Iu \quad \omega]^T$ is the vector of modified velocities, given by

$$\mathbf{v}' = \begin{bmatrix}I & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u\\ \omega\end{bmatrix},$$

where $I = 1rad^2/s$, and the matrices $\mathbf{H}$, $\mathbf{C}(\mathbf{v}')$ and $\mathbf{F}(\mathbf{v}')$, and the vector $\boldsymbol{\Delta}$ are given by

$$\mathbf{H} = \begin{bmatrix}\theta_1/I & 0\\ 0 & \theta_2\end{bmatrix}, \quad \mathbf{F}(\mathbf{v}') = \begin{bmatrix}\theta_4/I & 0\\ 0 & \theta_6 + (\theta_5/I - \theta_3)Iu\end{bmatrix},$$

$$\mathbf{C}(\mathbf{v}') = \begin{bmatrix}0 & -\theta_3\omega\\ \theta_3\omega & 0\end{bmatrix}, \quad \boldsymbol{\Delta} = \begin{bmatrix}-\theta_1 & 0\\ 0 & -\theta_2\end{bmatrix}\begin{bmatrix}\delta_u\\ \delta_\omega\end{bmatrix}.$$

Please, refer to [11] for more detailed information on the development of the presented dynamic model.

Remark 2. It is important to remark that the products of matrices in (3) result in velocities, not in torques. Therefore,

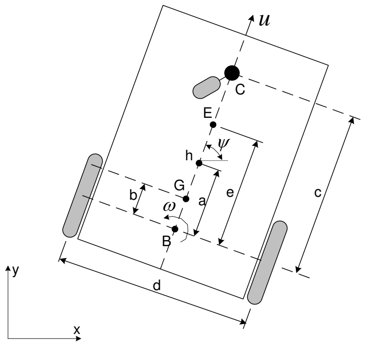

Fig. 1. The unicycle-like mobile robot.

the model here presented is suitable to be used in connection with commercial mobile robots, that usually have linear and angular velocities as reference signals. Some examples are the Pioneer robots from Mobile Robots, and the Khepera robots from K-Team Corporation.

Remark 3. The mathematical structure of the dynamic model represented by (3) is similar to the one that describes the dynamics of manipulators. So, strategies and techniques used to design controllers for manipulator robots can be adapted to design controllers for mobile robots.

A. Properties of the Dynamic Model

The model here presented is somewhat different from the classical dynamic model based on torques, and its properties deserve some analysis. They 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 $\mathbf{F}$ is symmetric and positive definite, or $\mathbf{F} = \mathbf{F}^T > 0$, 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 $\mathbf{\Delta} = \mathbf{0}$.

Properties 1, 2 and 3 can be proved by observing that $\mathbf{H}$ and $\mathbf{F}$ are diagonal and their terms are all positive. Property 4 is true under the assumption that there is no change in the robot parameters, i.e., the robot structure, mass, moment of inertia etc. do not change and the robot navigates on a horizontal plane. It is worth mentioning that $\mathbf{H}$ does not depend on the robot position if it navigates on a horizontal plane. Property 5 is straightforwardly proven by checking the terms of $\mathbf{C}(\mathbf{v})$. It is interesting to notice that $(\dot{\mathbf{H}} - \mathbf{2C})$ is skew symmetric because $\mathbf{H}$ is constant and, therefore, $\dot{\mathbf{H}} = \mathbf{0}$. Property 6 can be verified by observing that if $\theta_6 \gg |(\theta_5/I - \theta_3)Iu|$ the term $(\theta_5/I - \theta_3)Iu$ can be ignored, and $\mathbf{F}$ will depend only on θ_4 and θ_6, which are constant under the assumption of constant robot parameters. The condition $\theta_6 \gg |(\theta_5/I - \theta_3)Iu|$ was verified via experimental tests in different unicycle-like mobile robots.

Passivity is an important system property that can be used to design a controller for such a system [19]. The passivity property of the dynamic model (3) is presented by the following theorem.

Theorem 1: By considering $\mathbf{\Delta} = \mathbf{0}$ and $\theta_6 > -(\theta_5/I - \theta_3)Iu$, and assuming that $\mathbf{v_r} \in L_{2e}$ and $\mathbf{v}' \in L_{2e}$, the mapping $\mathbf{v_r} \rightarrow \mathbf{v}'$ of the dynamic model $\mathbf{H}\dot{\mathbf{v}}' + \mathbf{C}(\mathbf{v}')\mathbf{v}' + \mathbf{F}(\mathbf{v}')\mathbf{v}' = \mathbf{v_r}$ is strictly output passive.

Please refer to [11] for the theorem proof.

Parameter identification performed for three different unicycle-like mobile robots presented in [13], and for a unicycle-like robotic wheelchair, shows that the conditions $\theta_6 > -(\theta_5/I - \theta_3)Iu$ and $\theta_6 \gg |(\theta_5/I - \theta_3)Iu|$ are true for the robots whose parameters were identified. Those robots are two Pioneer 2-DX (one having an onboard computer and the other having not) and one Pioneer 3-DX having an onboard computer and a laser scanner mounted on it, all of them from Mobile Robots. Such robots have different weights and dynamics, specially the Pioneer 3-DX, because the laser sensor mounted on the top front of it weighs about 50% of the robot weight, which produces an important change in the robot mass and moment of inertia. The robotic wheelchair presents an even greater difference in the dynamic parameters, because of its own weight (about $70kg$). Parameter identification of the robotic wheelchair was performed when it was carrying a person who weighs $55kg$ and when it was carrying a person who weighs $125kg$. For both cases the above assumptions were verified.

As an example, the parameter values for the Pioneer 3-DX robot are

$$\theta_1 = 0.2604\, s, \quad \theta_2 = 0.2509\, s, \quad \theta_3 = -0.0005\, sm/rad^2,$$
$$\theta_4 = 0.9965, \quad \theta_5 = 0.00263\, s/m, \quad \theta_6 = 1.0768,$$

and for the robotic wheelchair carrying a person with $125kg$ they are

$$\theta_1 = 0.4263\ s, \quad \theta_2 = 0.0289\ s, \quad \theta_3 = 0.0058\ sm/rad^2,$$
$$\theta_4 = 0.9883, \quad \theta_5 = 0.0134\ s/m, \quad \theta_6 = 0.9931.$$

As u is limited to $1.2m/s$ for the Pioneer robots and to $2m/s$ for the robotic wheelchair, the assumptions of $\theta_6 > -(\theta_5/I - \theta_3)Iu$ and $\theta_6 \gg |(\theta_5/I - \theta_3)Iu|$ are valid. Then, the dynamic model of the above mentioned robots can be represented as in (3), with properties 1-7 valid.

III. CONTROLLER DESIGN

A trajectory tracking controller was designed based on the presented robot model. The controller was split in two parts: one is based on the inverse kinematics and the other compensates for the robot dynamics. The complete control structure is shown in Fig. 2, where $\mathbf{K}$, $\mathbf{D}$ and $\mathbf{R}$ represent the kinematic controller, the adaptive dynamic compensation controller, and the robot, respectively.

A. The Kinematic Controller

The kinematic controller is based on the kinematic model of the robot, which is given by (1). The kinematic control law here adopted is

$$\begin{bmatrix} u^c_{ref} \\ \omega^c_{ref} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\frac{1}{a} sin\psi & \frac{1}{a}\cos\psi \end{bmatrix} \begin{bmatrix} \dot{x}_d + l_x \tanh(\frac{k_x}{l_x}\tilde{x}) \\ \dot{y}_d + l_y \tanh(\frac{k_y}{l_y}\tilde{y}) \end{bmatrix}, \quad (4)$$

for which $a > 0$; $\mathbf{v_d} = [u^c_{ref} \quad \omega^c_{ref}]^T$ is the output of the kinematic controller; $\mathbf{h} = [x \ y]^T$ and $\mathbf{h_d} = [x_d \ y_d]^T$ are the vectors of the current and the desired coordinates of the point of interest *h*, respectively; $\tilde{\mathbf{h}} = \mathbf{h_d} - \mathbf{h}$ is the vector of position errors; $k_x > 0$ and $k_y > 0$ are the controller gains; and l_x, $l_y \in \mathbb{R}$ are saturation constants.

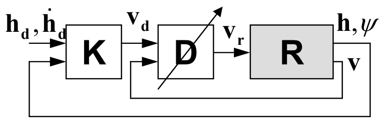

Fig. 2. The structure of the proposed controller.

Stability analysis considering this controller is presented in [14]. There, using the Lyapunov candidate function $V = \frac{1}{2}\tilde{\mathbf{h}}^T\tilde{\mathbf{h}} > 0$, it is shown that the system has a globally asymptotically stable equilibrium at the origin, which means that the position errors $\tilde{x}(t) \to 0$ and $\tilde{y}(t) \to 0$ as $t \to \infty$.

B. The Adaptive Dynamic Compensation Controller

The adaptive dynamic compensation controller receives the references for linear and angular velocities $\mathbf{v_d}$ from the kinematic controller, and generates another pair of linear and angular velocities commands $\mathbf{v_r}$ for the robot servos, as shown in Figure 2. The vector of modified desired velocities $\mathbf{v'_d}$ is defined as

$$\mathbf{v'_d} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_d \\ \omega_d \end{bmatrix},$$

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

To design the dynamic controller, (3) will be written in its linear parametrization form, as

$$\mathbf{v_r} = \mathbf{G}'\boldsymbol{\theta} = \begin{bmatrix} \dot{u} & 0 & -\omega^2 & u & 0 & 0 \\ 0 & \dot{\omega} & 0 & 0 & u\omega & \omega \end{bmatrix} \boldsymbol{\theta}, \tag{5}$$

where the uncertainties vector was neglected. Regarding parametric uncertainties, the proposed control law is

$$\mathbf{v_r} = \hat{\mathbf{H}}(\dot{\mathbf{v}}'_\mathbf{d} + \mathbf{T}(\tilde{\mathbf{v}}')) + \hat{\mathbf{C}}\mathbf{v'_d} + \hat{\mathbf{F}}\mathbf{v'_d}, \tag{6}$$

where $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{F}}$ are estimates of $\mathbf{H}$, $\mathbf{C}$, and $\mathbf{F}$, respectively, $\mathbf{T}(\tilde{\mathbf{v}}') = \begin{bmatrix} l_u & 0 \\ 0 & l_\omega \end{bmatrix} \begin{bmatrix} \tanh(\frac{k_u}{l_u} I\tilde{u}) \\ \tanh(\frac{k_\omega}{l_\omega}\tilde{\omega}) \end{bmatrix}$, $k_u > 0$ and $k_\omega > 0$ are gain constants, $l_u \in \mathbb{R}$ and $l_\omega \in \mathbb{R}$ are saturation constants, and $\tilde{\omega} = \omega_d - \omega$, $\tilde{u} = u_d - u$ 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 within the corresponding physical limits, considering that $\mathbf{v'_d}$ and $\dot{\mathbf{v}}'_\mathbf{d}$ are bounded to appropriate values.

An updating control law is designed for the case in which the dynamic parameters are not correctly identified, or may change from task to task. To design the updating law, the control law is written in its linear parametrization format, or

$$\mathbf{v_r} = \mathbf{G}\hat{\boldsymbol{\theta}} = \begin{bmatrix} \sigma_1 & 0 & -\omega_d\omega & u_d & 0 & 0 \\ 0 & \sigma_2 & (Iu_d\omega - Iu\omega_d) & 0 & u\omega_d & \omega_d \end{bmatrix} \hat{\boldsymbol{\theta}}, \tag{7}$$

where $\sigma_1 = \dot{u}_d + l_u \tanh(\frac{k_u}{l_u}\tilde{u})$, $\sigma_2 = \dot{\omega}_d + l_\omega \tanh(\frac{k_\omega}{l_\omega}\tilde{\omega})$. By defining the vector of parametric errors $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$, where $\hat{\boldsymbol{\theta}}$ is the vector of parameter estimates, (7) can be written as $\mathbf{v_r} = \mathbf{G}\boldsymbol{\theta} + \mathbf{G}\tilde{\boldsymbol{\theta}}$, or

$$\mathbf{v_r} = \mathbf{H}\boldsymbol{\sigma} + \mathbf{C}\mathbf{v'_d} + \mathbf{F}\mathbf{v'_d} + \mathbf{G}\tilde{\boldsymbol{\theta}}, \tag{8}$$

where $\boldsymbol{\sigma} = \dot{\mathbf{v}}'_\mathbf{d} + \mathbf{T}(\tilde{\mathbf{v}}')$.

The following equation is used as a parameter updating law. To prevent parameter drifting, a Leakage term, or a σ-modification term [20], was included in the equation. Then, the robust updating law

$$\dot{\hat{\boldsymbol{\theta}}} = \boldsymbol{\gamma}\mathbf{G^T}\tilde{\mathbf{v}}' - \boldsymbol{\gamma}\boldsymbol{\Gamma}\hat{\boldsymbol{\theta}} \tag{9}$$

is obtained, where $\boldsymbol{\Gamma} \in \mathbb{R}^{6\times 6}$ is a diagonal positive gain matrix.

A complete stability analysis considering the adaptive dynamic compensation controller is presented in [11]. There, it is shown that if the disturbance bounded, the velocity error is also bounded. Stability of the equilibrium is guaranteed if the disturbance is limited, and the robustness to disturbances is reduced as the velocity error reduces. This means that when the velocity error is very small, a disturbance of small intensity might provoke an increase in error. This fact leads to the conclusion that, in practice, $\tilde{\mathbf{v}}'$ will not be null and, therefore, there always will be some velocity error $\tilde{\mathbf{v}} \neq \mathbf{0}$. It should be noticed that an increase in k_u and k_ω gains increases the robustness to disturbances.

Remark 4. It should be pointed out that the proposed controller does not guarantee that $\tilde{\boldsymbol{\theta}} \to \mathbf{0}$ when $t \to \infty$. In other words, parameter estimates might converge to values that does not correspond to the physical parameters. Nevertheless, it is not required that $\tilde{\boldsymbol{\theta}} \to \mathbf{0}$ in order to make $\tilde{\mathbf{v}}$ converge to a bounded value.

Remark 5. It is important to point out that a nonholonomic mobile robot must be oriented according to the tangent of the trajectory path to track a trajectory with small error. Otherwise, the control errors would increase. This is true because the nonholonomic platform restricts the direction of the linear velocity developed by the robot. So, if the robot orientation is not tangent to the trajectory, the distance to the desired position at each instant will increase. The fact that the control errors converge to a bounded value shows that robot orientation does not need to be explicitly controlled, and will be tangent to the trajectory path while the control errors remain small.

Remark 6. The updating law adjusts parameter estimates when the velocity error is different than zero. In other words, here we consider that the velocity tracking error is all caused by a parameter estimation error. Although this is not entirely true, the proposed updating law is capable of providing a reduction in the velocity tracking error $\tilde{\mathbf{v}}$ as will be illustrated.

Considering the complete controller, some of the parameters, like l_x, l_y, l_u and l_ω, must be determined according to the physical limitations of the robot. But others, like k_x, k_y, k_u and k_ω can be chosen by the user. An important question that arises is: ”How to select a proper set of controller gains for the robot to have good performance?”. One possible way of solving this problem is to select a set of gains to minimize some cost function (like energy consumption or tracking error). In the next section we discuss the fundamentals of genetic algorithms, that will be used to select the controller gains according to some functions we define.

IV. GENETIC ALGORITHMS

Genetic algorithms (GA) are search algorithms based on the mechanics of natural selection of Darwin and natural genetics of Mendel. They combine survival of the fittest among string structures with a structures yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to search new points with expected improved performance.

GAs were first presented by [21] and have been used in many diverse areas such as function optimization, image processing, signal processing and system identification [22] [23]. A GA is a parallel global search technique that emulates natural genetic operators and works on a population representing different parameter vectors whose optimal value with respect to some (fitness) criterion is searched. This technique includes operations such as reproduction, crossover and mutation. These operators work with a number of artificial creatures called generation. By exchanging information from each individual in a population. GAs preserve a better individual and yield higher fitness generation by generation such that the performance can be improved. Next, we will briefly describe the basic operators in a GA.

A. Reproduction

Reproduction is a process in which a new generation of population is formed by selecting individuals from an existing population, according to their fitness. This process results in individuals with higher fitness values obtaining one or more copies in the next generation, while low fitness individuals may have none. Note, however, that reproduction does not generate new individuals but only favours the percentage of fit individuals in a population of given size.

B. Crossover

This operation provides a mechanism for individual to exchange information via probabilistic process. This operation takes two ”parents” individuals and produces two ”offspring” who are new individual whose characteristics are a combination of those of their parents.

C. Mutation

Mutation is an operation where some characteristics of an individuals are randomly modified, yielding a new individual. Here, the operation simply consists in randomly changing the value of one bit of the string representing an individual.

V. TUNNING AND SIMULATION PROCESS

In this section, a genetic algorithm is used to optimize parameters k_x, k_y, k_u and k_ω of the tracking controller described in Section III. Our GA creates a population of 200 individuals that contain the parameters necessary to minimize the objective function. The individuals of the GA are

$$[k_x, \quad k_u, \quad k_\omega], \tag{10}$$

and are coded as real values. We have considered that $k_x = k_y$, so it was necessary to select only one gain for the kinematic controller. This can be explained by the fact that there is no preferable direction for the robot to follow. After that, the GA algorithm computes the fitness of each individual in the population and selects the best individuals of each generation. The fitness of each individual is calculated as

$$Fit(k_x, k_u, k_\omega) = IAE_{gain} IAE + En_{gain} Energy, \tag{11}$$

where $IAE = \int_0^T |E(t)|dt$ is the integral of the tracking error, $E(t) = \sqrt{\tilde{x}^2 + \tilde{y}^2}$ is the instantaneous distance error, $Energy = \int_0^T (u^2(t) + \omega^2(t))dt$ represents the energy consumption, and T is the total simulation period. IAE_{gain} and En_{gain} are constants that can be altered to balance the importance of each factor (IAE or Energy). After that, the genetic operators (reproduction, crossover and mutation) are carried out through the population to create a new generation. Mutation rate is 0.1% and crossover rate is 80%. Individual selection is made by standard roulette wheel. GA runs iteratively for 15 generations, presenting at the end of the procedure the best individual, that is, the set of gains that results in smaller $Fit(k_x, k_u, k_\omega)$. We have repeated this procedure for over 70 different combinations of IAE_{gain} and En_{gain}.

After applying each resulting individual into the controller gains, we have simulated the hole system considering the Pioneer 3DX complete dynamic model, including its speed and acceleration limitations. White noise was added to the position and velocities signals sent to the controllers. The period of each simulation was $T = 250s$, in which the robot should follow an 8-shape trajectory (varying linear and angular speeds). We have made all MATLAB scripts and Simulink models available for download in [18]. The files include the robot dynamic model, the controllers, the genetic algorithm and some scripts to run the whole simulation.

VI. RESULTS AND CONCLUSIONS

Some of the resulting gains selected by the GA, associated with the corresponding IAE and Energy indexes, are shown in table I as an example. Line (a) represents the set of gains selected for $IAE_{gain} = 1$ and $En_{gain} = 0$. Line (b) represents the set of gains selected for $IAE_{gain} = 0$ and $En_{gain} = 1$. Lines (c-e) are resulting of different combinations of both gains. It can be seen that the smallest IAE value is achieved by the set of gains shown in line (a), while the smallest Energy index is obtained by the individual in line (b), as expected.

To select a proper set of controller gains one should consider other performance issues that are not shown in the values presented in table I. For instance, controller gains in line (a) will result in small tracking error, as shown in

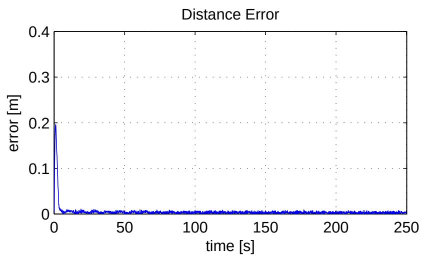

Fig. 3. Distance error for minimal IAE.

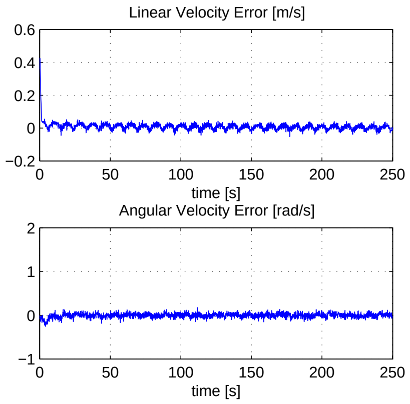

Fig. 4. Velocity error for minimal IAE.

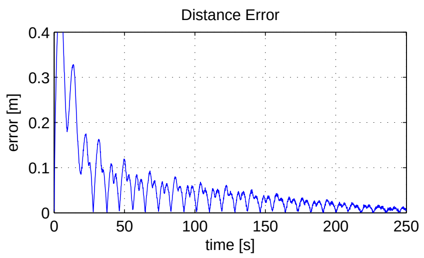

Fig. 5. Distance error for minimal energy.

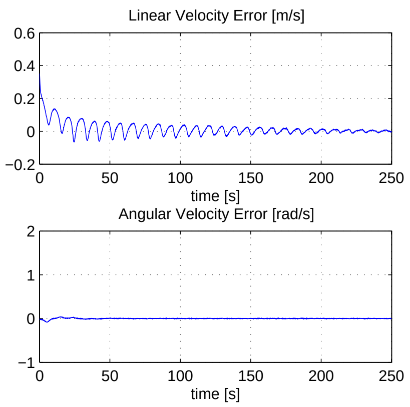

Fig. 6. Velocity error for minimal energy.

Figure 3, but the robot behaviour will not be smooth. Figure 4 illustrates that fact by showing that the velocity errors for minimal IAE have some fast oscillation. On the other hand, controller gains in line (b) will result in much less energy consumption and smoother behaviour (Figure 6), but the tracking error will be higher, as illustrated in Figure 5.

The presented simulation results show that a genetic algorithm can be successfully applied to select controller gains that result in minimizing energy consumption, tracking error or a combination of both indexes. Different sets of controller gains can be selected according to some defined criteria. This allows the user to make a proper selection of controller gains according to the task that is going to be carried out by the robot, considering a compromise between tracking error and battery consumption.

A limitation of the proposed tuning method is that it relies on the knowledge of the robot dynamic model. Therefore, the quality of the tuning results depends on the quality of the robot model. In our case, we have used a dynamic model of the Pioneer 3-DX that has been validated, as shown in previous works [11], [14]. MATLAB/Simulink simulation files are available for download and can be modified by the user to adapt the system to different robot models and conditions.

References

[1] B. Lantos and L. Márton, *Nonlinear Control of Vehicles and Robots*. Springer, 2011.

[2] S. Patel, R. Sanyal, and T. Sobh, "RISCBOT: A WWW-Enabled Mobile Surveillance and Identification Robot," *Journal of Intelligent Robotic Systems*, vol. 45, no. 1, pp. 15–30, 2006.

[3] E. Prassler, A. Ritter, C. Schaeffer, and P. Fiorini, "A Short History of Cleaning Robots," *Autonomous Robots*, vol. 9, no. 3, pp. 211–226, 2000.

[4] B. Stouten and A. de Graaf, "Cooperative transportation of a large object-development of an industrial application," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 2004, pp. 2450–2455.

[5] C. De La Cruz, T. Bastos, and R. Carelli, "Adaptive motion control law of a robotic wheelchair," *Control Engineering Practice*, vol. 19, no. 2, pp. 113–125, 2011.

[6] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 149–163, 1997.

[7] T. Das and I. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 501–510, 2006.

[8] M. Kim, J. Shin, and J. Lee, "Design of a robust adaptive controller for a mobile robot," *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1816–1821, 2000.

[9] S. Liu, H. Zhang, S. Yang, and J. Yu, "Dynamic control of a mobile robot using an adaptive neurodynamics and sliding mode strategy," in *Proc. pf the 5th World Congress on Intelligent Control and Automation*, vol. 6, 2004, pp. 5007–5011.

TABLE I

SET OF GAINS AND PERFORMANCE

	k_x=k_y	k_u	k_ω	IAE	Energy
(a)	9.172	9.329	9.727	0.948	56.42
(b)	0.301	1.140	8.404	13.54	51.35
(c)	5.951	8.090	1.059	1.04	54.39
(d)	6.872	8.527	0.523	1.01	54.45
(e)	6.777	9.736	0.594	0.99	54.43

[10] W. Dong and Y. Guo, "Dynamic tracking control of uncertain nonholonomic mobile robots," *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 2774–2779, 2005.

[11] F. Martins, M. Sarcinelli-Filho, T. Bastos, and R. Carelli, "Dynamic modeling and adaptive dynamic compensation for unicycle-like mobile robots," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.

[12] C. Resende, F. Espinosa, I. Bravo, M. Sarcinelli-Filho, and T. Bastos-Filho, "A trajectory tracking controller with dynamic gains for mobile robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3746–3751.

[13] C. De La Cruz and R. Carelli, "Dynamic modeling and centralized formation control of mobile robots," in *32nd IEEE Conference on Industrial Electronics*, November 2006, pp. 3880–3885.

[14] F. Martins, W. Celeste, R. Carelli, M. Sarcinelli-Filho, and T. Bastos-Filho, "An Adaptive Dynamic Controller for Autonomous Mobile Robot Trajectory Tracking," *Control Engineering Practice*, vol. 16, pp. 1354–1363, 2008.

[15] P. Fleming and R. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control engineering practice*, vol. 10, no. 11, pp. 1223–1241, 2002.

[16] R. Radu-CodruţDavid, S. Preitl, J. Tar, and J. Fodor, "Three evolutionary optimization algorithms in pi controller tuning," in *Applied Computational Intelligence in Engineering and Information Technology: Revised and Selected Papers from the 6th IEEE Int. Symp. on Applied Computational Intelligence and Informatics 2011*, vol. 1. Springer-Verlag New York Incorporated, 2012, p. 95.

[17] G. Almeida, J. Salles, J. Denti-Filho, and F. Rossomando, "Tuning a multivariable predictive control of a hot rolling mill with genetic algorithm," in *Proceedings of the Brazilian Symposium of Intelligent Automation*, 2009.

[18] NERA - Nucleo de Estudos em Robotica e Automacao, "Arquivos para simulacao de robo movel uniciclo," 2012. [Online]. Available: http://nera.sr.ifes.edu.br/?p=157

[19] H. Khalil, *Nonlinear systems*. Prentice Hall, NJ, 2002.

[20] S. Boyd and S. Sastry, *Adaptive control: stability, convergence, and robustness*. Prentice Hall, USA, 1989.

[21] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Boston, MA: MIT Press, 1992.

[22] K. Dejong, "An analysis of the behavior of a class of genetic adaptive systems." University of Michigan, Ann Arbor,Michigan" PHD Dissertation, 1975.

[23] R. Das and D. E. Goldberg, "Discrete-time parameters estimation with genetic algorithm," in *Proceedings of the 19th Annual Pittsburg Conference on Modeling and Simulation*, California, USA, 1988.