Rev. Ivoir. Sci. Technol., 13 (2009) 57 – 76 57
ISSN 1813-3290

A ROBUST ADAPTIVE CONTROLLER FOR A WHEELED MOBILE ROBOT

Donatien NGANGA-KOUYA[1*] et A. Francis OKOU[2]

[1]Department of mechanical Engineering, ENSET Libreville, Gabon
[2]Department of Electrical and Computer Engineering Royal Military College of Canada Kingston, Canada

(Reçu le 27 Janvier 2009, accepté le 03 Avril 2009)

* Correspondance et tirés à part, e-mail: *Ngad1109@yahoo.com*

ABSTRACT

This paper proposes a robust adaptive nonlinear controller to stabilize a wheeled mobile robot. The controller equations are obtained using a recursive backstepping design method. The robot model is divided into two parts: a state space model with intermediate control inputs and algebraic nonlinear equations that relate the true and the intermediate control inputs. The robot parameters are assumed unknown for the design. First, a suitable change of variable is applied to the initial robot dynamics to expose the inherent strict feedback structure of this model. Next, a three-step robust adaptive backstepping control design method is applied to find the intermediate control input expressions. Finally the true control inputs are obtained solving the nonlinear equations that relates intermediate and true control inputs, iteratively. A direct adaptive algorithm based on the projection method is used to update the controller parameters online. The main advantage of this adaptation method is that estimated parameters convergence is guaranteed and they remain inside predefined domains. The proposed control design method is tested in simulation. Results show good tracking performances when system parameters are perfectly known and when they are assumed unknown with large abrupt variations.

Keywords : *Mobile Robot, Robust Adaptive Control, Backstepping Design Method, Newton-Raphson Method, Projection Method.*

RÉSUMÉ

Un correcteur robuste adaptatif pour un robot mobile équipé de roues.

Cet article propose un correcteur non linéaire robuste et adaptatif pour stabiliser un robot mobile équipé à roues. Les équations du correcteur sont obtenues en utilisant une méthode de conception récursive dite backstepping.

Le modèle du robot est divisé en deux parties : Une représentation dans l'espace d'état faisant intervenir des entrées de commande intermédiaires et une équation algébrique non linéaire qui relie les entrées de commande intermédiaires aux entrées exactes de commande du système. Les paramètres du robot sont supposés inconnus durant la phase de conception du correcteur. Dans un premier temps, un changement de variable approprié est appliqué au modèle initial du robot pour faire ressortir la structure interne du modèle. Dans un second temps, une méthode de conception robuste et adaptative en trois étapes est utilisée pour obtenir les expressions des entrées de commande intermédiaires. Enfin, les entrées exactes de commande du système sont obtenues en résolvant d'une manière itérative, les équations algébriques non linéaires qui relient les entrées de commande intermédiaires aux entrées exactes de commande. Un algorithme adaptatif direct basé sur la projection est utilisé pour mettre à jour en temps réel les paramètres du correcteur. Le principal avantage de cette méthode d'adaptation est que la convergence des paramètres estimés est garantie dans un domaine prédéfini. La méthode de conception proposée est évaluée en simulation. Les résultats montrent des bonnes performances de poursuite de trajectoires quand les paramètres du système commandé sont parfaitement connus et lorsqu'ils sont inconnus et sujettes à des grandes variations.

Mots-clés : *Robot mobile, commande robuste et adaptative, méthode de conception dite backstepping, méthode de newton-raphson, méthode par projection.*

I - INTRODUCTION

The importance of autonomous robots for domestic and military application is now well established [1,2]. The number of recent contributions testifies the high level of research activities in this domain. In [3], a new control rule for determining vehicle linear and rotational velocities are presented. The design is based on the assumption that a 'perfect velocity tracking' is achieved. The reference [4] removes the previous assumption and proposes a control structure that makes possible the integration of a kinematic controller and neural network computed-torque controller. In [5], a μ-synthesis robust controller of a four-wheel steering (4WS) vehicle is designed with the optimized weighting functions to attenuate the external disturbances while the yaw rate is chosen as the only feedback signal. The reference [6] studies robust steering and traction of four-wheel steering (4WS) vehicles with varying velocity, mass, moment of inertia, and road–tire interaction. A nonlinear input–output decoupling controller along with a robust control scheme is proposed. The H-infinity controller and observer gains are obtained by solving two new Riccati algebraic equations. In

most of the previous contributions, the mobile robot is represented either by kinematic equations or linearized model when dynamic equations are used.
This paper introduces an innovative approach to address the mobile robot tracking controller design problem. The basic idea is to divide this difficult problem into two straightforward problems. This idea is indeed the main advantage of our design approach. It is motivated by the observation that much of the nonlinearities of the robot model appear into actuator equations. Therefore, a model based design methodology is proposed. The highly nonlinear model of the mobile robot is spitted into two parts: A state space model in which three intermediate control inputs are voluntarily introduced, and three nonlinear algebraic equations involving true and intermediate control inputs. First, an adaptive backstepping control design method [7] is employed to synthesize the intermediate input equations that solve the trajectory tracking problem. Next, the Newton-Raphson method [8] is used to solve nonlinear algebraic equations to obtain the true control inputs. A projection based adaptation method is proposed to update controller parameters when robot parameters change. Two scenarios are used to assess the controller performances. First, the robot parameters are assumed known therefore the adaptation algorithms are switched off. In the second case, robot parameters experiment large abrupt variations and the adaptation module along with the robust nonlinear terms are used to ensure good trajectory tracking performances.
The paper is partitioned as follows. The mobile robot state space model is discussed in Section II. The design objective is presented in Section III. Section IV exposes the backstepping control design method used to obtain the intermediate input equations along with the algorithm used to compute the true control inputs. Controller parameter adaptation laws are proposed in the same section. Simulations are performed and the results are presented in Section V. Two different trajectories are used to evaluate the control tracking performances. The paper ends with a conclusion.

II - FUNDAMENTAL CONCEPTS

II-1. the wheeled mobile robot model

The four wheeled autonomous vehicle studied in this paper is represented by the half vehicle model that is illustrated at ***Figure 1***. The mobile robot motion can be represented by the following state space equations [9, 10].

$$\dot{x} = V\cos(\theta + \beta) \quad (1.a)$$

$$\dot{y} = V\sin(\theta + \beta) \quad (1.b)$$

$$\dot{\theta} = \omega \quad (1.c)$$

$$\dot{V} = \frac{\cos(\beta)}{M} U_x + \frac{\sin(\beta)}{M} U_y \tag{1.d}$$

$$\dot{\beta} = \frac{-\sin(\beta)}{M \cdot V} U_x + \frac{\cos(\beta)}{M \cdot V} U_y - \omega \tag{1.e}$$

$$\dot{\omega} = \frac{1}{J} U_\omega \tag{1.f}$$

where

$$U_\omega = L^f \left\{ A^f \cos(\delta^f) + F^f \sin(\delta^f) \right\} - L^r \left\{ A^r \cos(\delta^r) + F^r \sin(\delta^f) \right\} - b \cdot \omega \tag{2.c}$$

$$U_x = -A^f \sin(\delta^f) + F^f \cos(\delta^f) - A^r \sin(\delta^r) + F^r \cos(\delta^r) - B \cdot V \cdot \cos(\beta) \tag{2.a}$$

$$U_y = A^f \cos(\delta^f) + F^f \sin(\delta^f) + A^r \cos(\delta^r) + F^r \sin(\delta^r) - B \cdot V \cdot \sin(\beta) \tag{2.b}$$

and

$$A^f \approx \mu_f (\delta^f - \beta - \frac{L^f \omega}{V}) \text{ and } A^r \approx \mu_r (\delta^r - \beta - \frac{L^r \omega}{V}) \tag{3}$$

State variables x, y, θ, V, β and ω are the x-position, the y-position, the orientation, the longitudinal velocity of center of mass, the skidding .angle and the angular velocity of the centre of mass, respectively. δ^f, δ^r, F^f and F^r are the control inputs and they represent the front wheel steering angle, the rear wheel steering angle, front wheel traction force and rear wheel traction force, respectively. Parameters M, J, L^f and L^r are the mass of the vehicle, the moment of inertia about the centre of mass, the distance between the front wheel and the centre of mass and the distance between the rear wheel and the center of mass, respectively. B, b, μ_f and μ_r stand for linear and angular viscous ratios, Front and rear wheel lateral friction coefficients, respectively. It is important to mention that this model is composed of three relationships (1.a, 1.b and 1.c) referred to as the 'kinematics' and three equations (1.d, 1.e and 1.f) usually called the 'dynamics'.

II-2. design strategy and objective

Our objective is to synthesize δ^f, δ^r, F^f and F^r such that position and orientation variables (x, y, θ) follow the desired trajectories $(x_{ref}, y_{ref}, \theta_{ref})$. The proposed design strategy consists in considering

(U_x, U_y, U_ω) that appears in equations (1) as intermediate control inputs, and in finding their equations to solve the trajectory tracking problem. Next, equation (2) will be used to solve online for the true control inputs $(\delta^f, \delta^r, F^f, F^r)$, iteratively. The Newton-Raphson method is suggested to solve the nonlinear equations. Integrators are voluntarily introduced into the control loop for practical consideration and to guarantee that steady state tracking errors will go to zero. To reveal the inherent strict feedback structure of the mobile robot model and to introduce these integrators into the control loop, the following new variables and change of variables are defined:

$$V_x = V\cos(\beta) \quad \text{and} \quad V_y = V\sin(\beta) \tag{4}$$

$$\Omega_x = \int_0^t (x - x_{ref})d\tau, \quad \Omega_y = \int_0^t (y - y_{ref})d\tau, \quad \Omega_\theta = \int_0^t (\theta - \theta_{ref})d\tau \tag{5}$$

The mobile robot augmented state space model has the following form:

$$\dot{\Omega}_x = x - x_{ref} \tag{6.a}$$

$$\dot{\Omega}_y = y - y_{ref} \tag{6.b}$$

$$\dot{\Omega}_\theta = \theta - \theta_{ref} \tag{6.c}$$

$$\dot{x} = \cos(\theta)V_x - \sin(\theta)V_y \tag{7.a}$$

$$\dot{y} = \sin(\theta)V_x + \cos(\theta)V_y \tag{7.b}$$

$$\dot{\theta} = \omega \tag{7.c}$$

When nominal value of the unknown robot parameters (M, J, B, b, μ_f and μ_r) are introduced into the dynamic parts, equations 1.d, 1.e and 1.f become:

$$\dot{V}_x = \frac{1}{M}U_{xn} + \omega V_y + p_1(t) \tag{8.a}$$

$$\dot{V}_y = \frac{1}{M}U_{yn} - \omega V_x + p_2(t) \tag{8.b}$$

$$\dot{\omega} = \frac{1}{J}U_{\omega n} + p_3(t) \tag{8.c}$$

where U_{xn}, U_{yn} and $U_{\omega n}$ represent the nominal intermediate control inputs. Their expressions are identical to equations 2 when robot parameters are

substituted by their nominal values. Parameters $p_1(t)$, $p_2(t)$ and $p_3(t)$ gather the terms that depend on unknown parameters. They are treated like disturbances in this work.

$$p_1(t) = -\Delta A^f \sin(\delta^f) - \Delta A^r \sin(\delta^r) - \Delta B \cos(\beta) V$$

$$p_2(t) = \Delta A^f \cos(\delta^f) + \Delta A^r \cos(\delta^r) - \Delta B \sin(\beta) V$$

$$p_3(t) = \Delta A^f L^f \cos(\delta^f) - \Delta A^r L^r \cos(\delta^r) - \Delta b \cdot \omega$$

where

$$\Delta A^f \approx \Delta\mu_f (\delta^f - \beta - \frac{L^f \omega}{V}),\ \Delta A^r \approx \Delta\mu_r (\delta^r - \beta - \frac{L^r \omega}{V})$$

$$\Delta\mu_f = \mu_f - \mu_{fn},\ \Delta\mu_r = \mu_r - \mu_{fn},\ \Delta B = B - B_n$$

Parameters B_n, b_n, μ_{fn} and μ_{rn} are nominal parameters of linear and angular viscous ratios, Front and rear wheel lateral friction coefficients, respectively. According to their expressions $p_1(t)$, $p_2(t)$ and $p_3(t)$ are obviously bounded.

The block diagram representation of equations (6,7 and 8) is shown at ***Figure. 2.*** One can easily see that the augmented state space model is composed of three cascaded subsystems. The state variables of the first subsystem are the inputs of the second subsystem, and so on. This structure is refereed to as a strict feedback form. It is very suitable for the application of a backstepping control design method to obtain the controller equation.

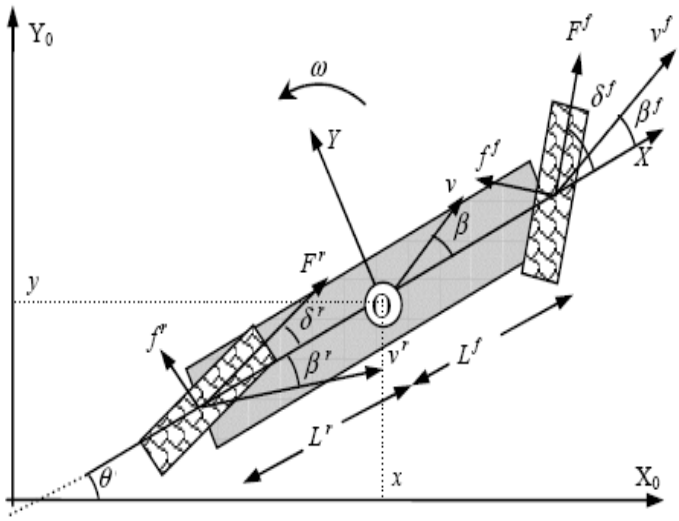


Figure 1 : *Geometrical parameters of the half vehicle model.*

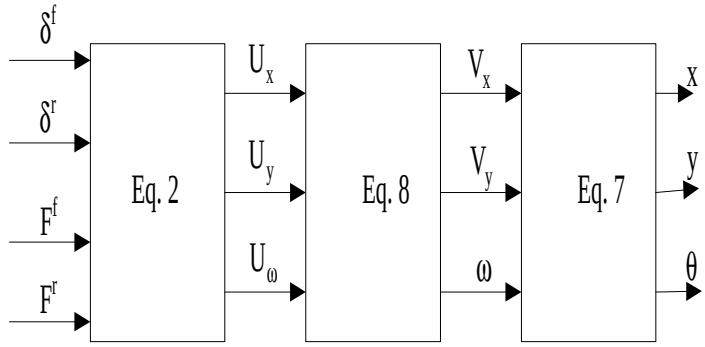


Figure 2 : *The intrinsic structure of the mobile robot model*

II-3. backstepping method for controller design

A systematic application of a recursive backstepping method to the system represented by equations (6 to 8) leads to the control structure depicted at figure 3. During the design, robot parameters such as M, J, B, b, μ_f and μ_r are assumed unknown. One can note that the proposed controller is composed of 3 sub controllers. The $\Omega_{xy\theta}$-controller is used to find the suitable value of the position and orientation state vector (x, y, θ) denoted (x^*, y^*, θ^*) that steers $(\Omega_x, \Omega_y, \Omega_\theta)$ to zero. The $xy\theta$-controller is used to obtain the 'ideal value' for (V_x, V_y, ω) denoted (V_x^*, V_y^*, ω^*) in order to drive (x, y, θ) to its 'ideal value', (x^*, y^*, θ^*). The last sub controller is the $V_x V_y \omega$-controller. It is used to find the intermediate control inputs equations in such a way that they steer the vector (V_x, V_y, ω) to its reference (V_x^*, V_y^*, ω^*). Sub controller equations are obtained as follows. First, we consider the sub-system represented by the equations (6). If (x, y, θ) were a vector of control inputs, it should be selected equals to,

$$x^* = x_{ref} - k_1 \Omega_x \quad (9.a)$$

$$y^* = y_{ref} - k_2 \Omega_y \quad (9.b)$$

$$\theta^* = \theta_{ref} - k_3\Omega_\theta \tag{9.c}$$

in order to drive $(\Omega_x \ , \Omega_y \ , \Omega_\theta)$ to zero. Equations (9) represent 'ideal values' or references of $(x \ , y \ , \theta)$. They are $\Omega_{xy\theta}$-controller equations. However, $(x \ , y \ , \theta)$ is not the vector of control inputs but a vector of state variables. It is therefore used to introduce new variables that represent the errors between position and orientation variables and their references.

$$\tilde{x} = x - x^* \tag{10.a}$$

$$\tilde{y} = y - y^* \tag{10.b}$$

$$\tilde{\theta} = \theta - \theta^* \tag{10.c}$$

Dynamics of $(\Omega_x \ , \Omega_y \ , \Omega_\theta)$ into the new coordinates have the following form, therefore:

$$\dot{\Omega}_x = -k_1\Omega_x + \tilde{x} \tag{11.a}$$

$$\dot{\Omega}_y = -k_2\Omega_y + \tilde{y} \tag{11.b}$$

$$\dot{\Omega}_\theta = -k_3\Omega_\theta + \tilde{\theta} \tag{11.c}$$

Next, references of $(V_x \ , V_y \ , \omega)$ will be selected to drive $(\tilde{x} \ , \tilde{y} \ , \tilde{\theta})$ to zero. Dynamics of $(\tilde{x} \ \ \tilde{y} \ \ \tilde{\theta})$ can be rewritten into the following form:

$$\dot{\tilde{x}} = \cos(\theta)V_x - \sin(\theta)V_y - \dot{x}_{ref} + k_1(x - x_{ref}) \tag{12.a}$$

$$\dot{\tilde{y}} = \sin(\theta)V_x + \cos(\theta)V_y - \dot{y}_{ref} + k_2(y - y_{ref}) \tag{12.b}$$

$$\dot{\tilde{\theta}} = \omega - \dot{\theta}_{ref} + k_3(\theta - \theta_{ref}) \tag{12.c}$$

If $(V_x \ , V_y \ , \omega)$ were the vector of control inputs, it should be selected equals to,

$$V_x^* = \cos(\theta)\left\{\dot{x}_{ref} + k_1x_{ref} - k_1x - k_4\tilde{x} - \tilde{\Omega}_x\right\} + \sin(\theta)\left\{\dot{y}_{ref} + k_1y_{ref} - k_2y - k_5\tilde{y} - \tilde{\Omega}_y\right\} \tag{13.a}$$

$$V_y^* = -\sin(\theta)\left\{\dot{x}_{ref} + k_1x_{ref} - k_1x - k_4\tilde{x} - \tilde{\Omega}_x\right\} + \cos(\theta)\left\{\dot{y}_{ref} + k_1y_{ref} - k_2y - k_5\tilde{y} - \tilde{\Omega}_y\right\} \tag{13.b}$$

$$\omega^* = \dot{\theta}_{ref} + k_3\theta_{ref} - k_3\theta - k_6\tilde{\theta} - \tilde{\Omega}_\theta \tag{13.c}$$

in order to force $(\tilde{x}\ ,\tilde{y}\ ,\tilde{\theta})$ to converge to zero. Since the vector $(V_x\ ,V_y\ ,\omega)$ is not the input, it is used to introduce a new vector of state variables.

$$\tilde{V}_x = V_x - V_x^* \tag{14.a}$$

$$\tilde{V}_y = V_y - V_y^* \tag{14.b}$$

$$\tilde{\omega} = \omega - \omega^* \tag{14.c}$$

Equations (12) into the new coordinates become:

$$\dot{\tilde{x}} = \cos(\theta)\tilde{V}_x - \sin(\theta)\tilde{V}_y - k_4\tilde{x} - \tilde{\Omega}_x \tag{15.a}$$

$$\dot{\tilde{y}} = \sin(\theta)\tilde{V}_x - \cos(\theta)\tilde{V}_y - k_5\tilde{y} - \tilde{\Omega}_y \tag{15.b}$$

$$\dot{\tilde{\theta}} = \tilde{\omega} - k_6\tilde{\theta} - \tilde{\Omega}_\theta \tag{15.c}$$

Equations (13) represent $xy\theta$-controller equations. Next, the intermediate control inputs will be found to drive $(\tilde{V}_x\ ,\tilde{V}_y\ ,\tilde{\omega})$ to zero. The dynamic equations of $(\tilde{V}_x\ ,\tilde{V}_y\ ,\tilde{\omega})$ are:

$$\dot{\tilde{V}}_x = \frac{1}{M}U_{xn} + \omega V_y - \dot{V}_x^* + p_1 \tag{16.a}$$

$$\dot{\tilde{V}}_y = \frac{1}{M}U_{yn} - \omega V_x - \dot{V}_y^* + p_2 \tag{16.b}$$

$$\dot{\tilde{\omega}} = \frac{1}{J}U_{\omega n} - \dot{\omega}^* + p_3 \tag{16.c}$$

Expressions of $(\dot{V}_x^*\ ,\dot{V}_y^*\ ,\dot{\omega}^*)$ in terms of other state variables can be easily obtained by differentiating equations (13) and substituting each derivative by its expression in terms of the state variables. To stabilize $(\tilde{V}_x\ ,\tilde{V}_y\ ,\tilde{\omega})$ to zero, it is sufficient to select $(U_{xn}\ ,U_{yn}\ ,U_{zn})$ equals to,

$$U_{xn} = \hat{M}u_1 - \frac{K}{4}u_1^2\tilde{V}_x \tag{17.a}$$

$$U_{yn} = \hat{M}u_2 - \frac{K}{4}u_2^2\tilde{V}_y \tag{17.b}$$

$$U_{\omega n} = \hat{J}u_3 - \frac{K}{4}u_3^2\tilde{\omega} \tag{17.c}$$

where

$$u_1 = -k_7\tilde{V}_x - \omega V_y + \dot{V}_x^* - \cos(\theta)\tilde{x} - \sin(\theta)\tilde{y} - \hat{p}_1(t) - \frac{K}{4}\tilde{V}_x$$

$$u_2 = -k_8\tilde{V}_y + \omega V_x + \dot{V}_y^* + \sin(\theta)\tilde{x} + \cos(\theta)\tilde{y} - \hat{p}_2(t) - \frac{K}{4}\tilde{V}_y$$

$$u_3 = -k_9\tilde{\omega} + \dot{\omega}^* - \tilde{\theta} - \hat{p}_3 - \frac{K}{4}\tilde{\omega}$$

The dynamic equations of (16) in closed loop become,

$$\dot{\tilde{V}}_x = -k_7\tilde{V}_x - \cos(\theta)\tilde{x} - \sin(\theta)\tilde{y} + \tilde{p}_1(t) - \frac{K}{4}\tilde{V}_x - \frac{\tilde{M}}{M}u_1 - \frac{K}{4}\frac{1}{M}u_1^2\tilde{V}_x \quad (18.a)$$

$$\dot{\tilde{V}}_y = -k_8\tilde{V}_y + \sin(\theta)\tilde{x} + \cos(\theta)\tilde{y} + \tilde{p}_2(t) - \frac{K}{4}\tilde{V}_y - \frac{\tilde{M}}{M}u_2 - \frac{K}{4}\frac{1}{M}u_2^2\tilde{V}_y \quad (18.b)$$

$$\dot{\tilde{\omega}} = -k_9\tilde{\omega} - \tilde{\theta} + \tilde{p}_3 - \frac{K}{4}\tilde{\omega} - \frac{\tilde{J}}{J}u_3 - \frac{K}{4}\frac{1}{J}u_3^2\tilde{\omega} \quad (18.c)$$

Equations (17) are $V_xV_y\omega$-controller equations. The output of the $V_xV_y\omega$-controller is sent to the module that solves the nonlinear equations (2) to obtain the true control inputs $\left(\delta^f \quad ,\delta^r \quad ,F^f \quad ,F^r\right)$. The following paragraph presents the algorithm used to compute the true control inputs knowing the intermediate inputs and the state variables. The set of nonlinear algebraic nonlinear equations has four unknowns represented by $\left(\delta^f \quad ,\delta^r \quad ,F^f \quad ,F^r\right)$. To guarantee a unique solution to this problem, the input F^r is set to zero. Let

$$H(x) = \begin{pmatrix} h_1(x) & h_2(x) & h_2(x) \end{pmatrix}^T \quad (19)$$

where

$$h_1(x) = -A^f\sin(\delta^f) + F^f\cos(\delta^f) - A^r\sin(\delta^r) + F^r\cos(\delta^r) - B\cdot V_x - U_x$$

$$h_2(x) = A^f\cos(\delta^f) + F^f\sin(\delta^f) + A^r\cos(\delta^r) + F^r\sin(\delta^r) - B\cdot V_y - U_y$$

$$h_3(x) = C^f\left\{A^f\cos(\delta^r) + F^f\sin(\delta^f)\right\} - b\cdot\omega - U_z - C^r\left\{A^r\cos(\delta^r) + F^r\sin(\delta^f)\right\}$$

$$x = \left(\delta^f \quad ,\delta^r \quad ,F^f \quad ,F^r\right), \text{ with } F^r = 0.$$

We suggest taking the controller output equals to the third iteration value of the Newton-Raphson algorithm given by equation (19). Simulations have revealed that only three iterations are necessary to reach the convergence.

$$x(k+1) = x(k) - \left(\frac{\partial H}{\partial x}\right)_{x=x(k)}\cdot H(x(k)) \quad (20)$$

The initial value $x(k=0)$ is equal to the previous controller output. In other words, $\left(\delta^f \quad ,\delta^r \quad ,F^f\right) = x(k=3)$. $(\frac{\partial H}{\partial x})_{x=x(k)}$ represents the Jacobian matrix of H(x) at x = x(k).

To be able to compensate for mass, moment of inertia, viscous ratios and friction coefficient variations, the proposed controller is provided with an adaptation module. That module is responsible for computing parameters $\hat{M}$, $\hat{J}$ and $\hat{p}_i$, $i=1,2,3$. The proposed algorithm is based on the projection method [11] defined as follows:

$$\text{Proj}\left[y,\hat{\theta}_i \right] = y \text{ , if } \{\theta_{im} \le \hat{\theta}_i \le \theta_{iM}\} \text{ or } \{\hat{\theta}_i \le \theta_{im} \text{ and } y \ge 0\} \text{ or } \{\theta_{iM} \le \hat{\theta}_i \text{ and } y \le 0\}$$

$$\text{Proj}\left[y,\hat{\theta}_i \right] = y[1 - \frac{\theta_{im}^2 - \hat{\theta}_i^2}{\theta_{im}^2 - (\theta_{im} - \rho_i)^2}] \text{ , } \{\hat{\theta}_i \le \theta_{im} \text{ and } y < 0\}$$

$$\text{Proj}\left[y,\hat{\theta}_i \right] = y[1 - \frac{\theta_i^2 - \hat{\theta}_{iM}^2}{(\theta_{iM} + \rho_i)^2 - \theta_{iM}^2}] \text{ , } \{\theta_{iM} \le \hat{\theta}_i \text{ and } y > 0\}$$

where θ_{im}, θ_{iM} and ρ_i are real numbers. y is the function that is being projected.

The definition indicates that the projection of the function y is equal to itself as far as $\hat{\theta}_i$ is inside or is moving towards the interval $[\theta_{im} \quad \theta_{iM}]$. If $\hat{\theta}_i$ is outside the interval and is moving in the wrong direction, the projection of y is equal to a fraction of y.The variable at the right hand side of the equality will represent $\dot{\hat{\theta}}_i$. Therefore, when $\hat{\theta}_i$ will be leaving the interval $[\theta_{im} \quad \theta_{iM}]$, the algorithm will reduce $\dot{\hat{\theta}}_i$, immediately. The adaptation strategy based on the projection method will guarantee that estimated parameters remain and converge inside the predefined domain $[\theta_{im} + \rho_i \quad \theta_{iM} - \rho_i]$. Predefined lower and upper bounds of the estimated parameter $\hat{\theta}_i$ are represented by θ_{im} and θ_{iM}. The positive real number ρ_i is chosen by the designer. The adaptation equations could be obtained from the stability study of the closed loop system describes by equations (12, 15 and 18). However, for the sake of simplicity, an alternative method that gives the same results is preferred in this paper. The equations are obtained by inspection using the following rule of thumb: For instance, to find the adaptation law for $\hat{M}$, (i.e., $\dot{\hat{M}}$), we use the equation (18.a) and the equation (18.b) since $\tilde{M}$ appears in both of them. The function

y of the projection function is equal to $\tilde{V}_x$ (since it is the state variable of equation (18.a)) multiplied by the factor of $\tilde{M}$ in the equation (18.a) (i.e., $-u_1$ in this case) plus $\tilde{V}_y$ (since we are now considering equation (18.b)) multiplied by $-u_2$ (the factor of $\tilde{M}$ in the equation (18.b)). The adaptation laws for the proposed controller are:

$$\dot{\hat{M}} = \alpha \, Proj\left\{-u_1 \tilde{V}_x - u_2 \tilde{V}_y, \quad \hat{M}\right\} \tag{21}$$

$$\dot{\hat{J}} = \beta \, Proj\left\{-u_3 \tilde{V}_z, \quad \hat{J}\right\} \tag{22}$$

$$\dot{\hat{p}}_1 = \gamma_1 \, Proj\left\{\tilde{V}_x, \quad \hat{p}_1\right\} \tag{23}$$

$$\dot{\hat{p}}_2 = \gamma_2 \, Proj\left\{\tilde{V}_y, \quad \hat{p}_2\right\} \tag{24}$$

$$\dot{\hat{p}}_3 = \gamma_3 \, Proj\left\{\tilde{V}_\omega, \quad \hat{p}_3\right\} \tag{25}$$

Parameters α, γ_1, γ_2 and γ_3 are positive gains. It is possible to show that the closed loop system described by equations (11-15-18) along with the adaptation module represented by equations 21 to 25 is stable and converges to its equilibrium point. The proof is given in the appendix section.

III - RESULTS AND DISCUSSION

Simulations are now performed to assess the proposed design method effectiveness. The robot nominal parameters are given at ***Table 1***. The proposed controller is tested in two different situations. The first case assumed that the controller knows the robot parameters, perfectly. The adaptation module is therefore not used. The robot parameters that appear in the controller equations are equal to their nominal values. The robot is used to follow the circular path represented by the following equations.

$$\ddot{x}_{ref} = -\left(\frac{\pi}{10}\right)^2 \cos\left(\frac{\pi}{10} t\right), \quad \dot{x}_{ref}(t=0) = x_{ref}(t=0) = 0$$

$$\ddot{y}_{ref} = -\left(\frac{\pi}{10}\right)^2 \sin\left(\frac{\pi}{10} t\right), \; \dot{y}_{ref}(t=0) = \frac{\pi}{10}, \; y_{ref}(t=0) = 0$$

$$\theta_{ref}(t) = \arctan\left(\frac{\dot{y}_{ref}}{\dot{x}_{ref}}\right)$$

Controller gains are given at ***Table 1***. ***Figure 4*** illustrates the tracking performance of the controller. The initial position of the robot is x = 0.1m, y = 0 and the $\theta = -15^0$. One can see that a perfect tracking performance is achieved. The robot is able to closely follow the desired trajectory. ***Figure 5*** shows the vehicle speed. After a peak at the very begin of the trajectory, a steady state speed of about 0.3 m/s is reached at t = 0.2 second. ***Figures 6*** and ***7*** show the other state variables of the system. They are bounded. All of them converge to their steady state values after 1 second. Controller outputs are illustrated at ***Figures 8.a*** and ***8.b***. Once again, these signals are all bounded and their steady state values are realistic.

Table 1 : *Model parameters*

$M = 1kg$	$L^r = 2m$	$\mu_r = 1$
$J = 1N.m$	$L^f = 2m$	$\mu_f = 1$

Table 2 : *Control gains*

$k_1 = 5$	$k_2 = 5$	$k_3 = 10$
$k_4 = 100$	$k_5 = 100$	$k_6 = 10$
$k_7 = 100$	$k_8 = 100$	$k_9 = 10$

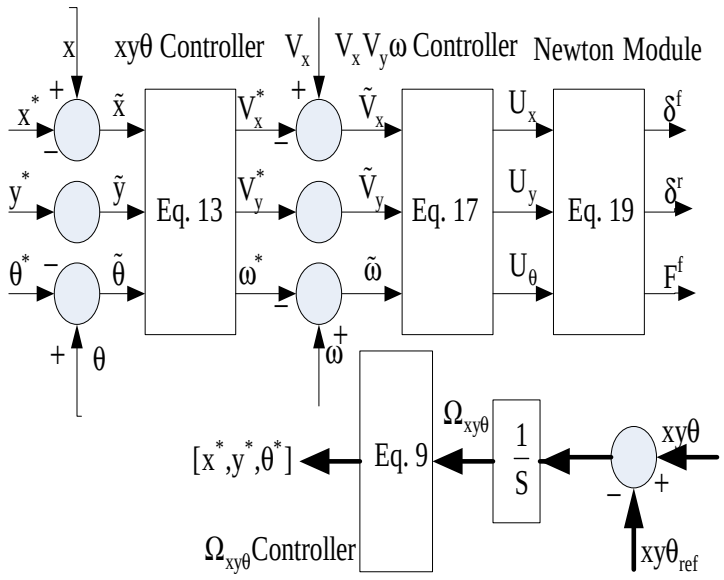


Figure 3 : *Controller structure*

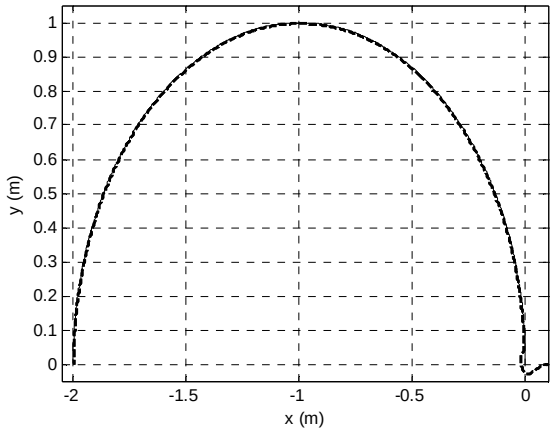


Figure 4 : *Robot motion in the x-y plane: dash-line: actual trajectory, continuous line: desired trajectory.*

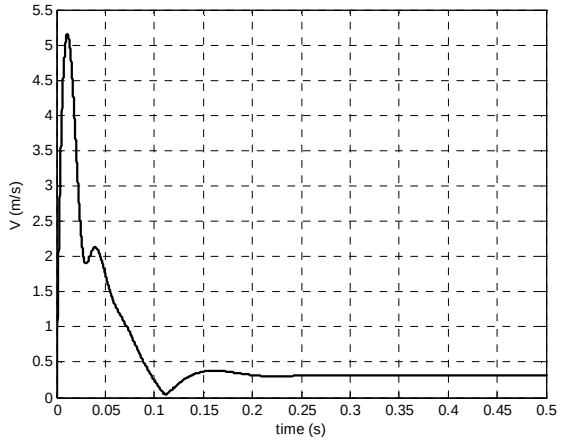


Figure 5 : *Speed V*

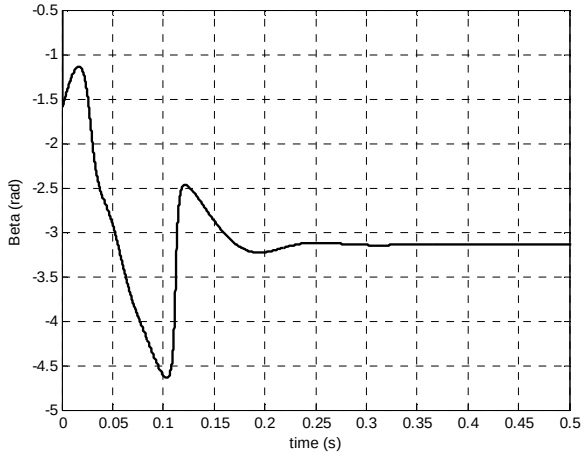


Figure 6 : *Angle beta*

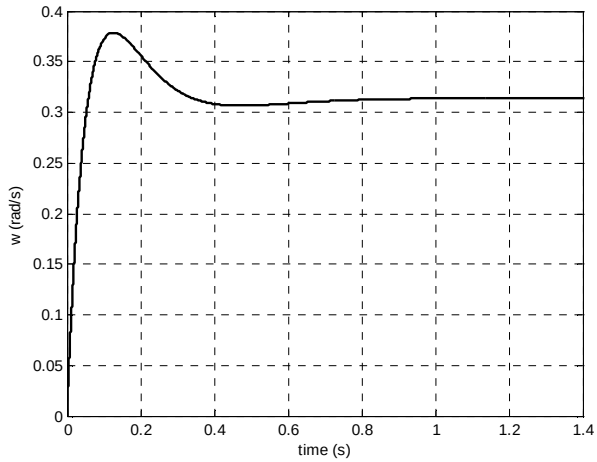
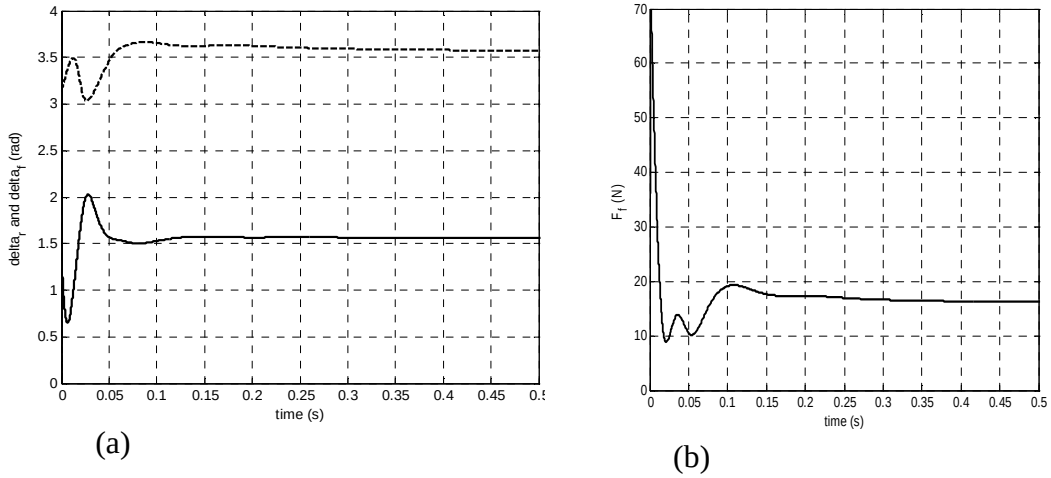


Figure 7 : *Angular speed*



(a) (b)

Figure 8 : *Control efforts*

In the second situation, the robot parameters are abruptly changed from their nominal values. The wheeled mobile robot and the controller parameters are given at ***Table 3***. The robot is used to follow the path described by the following equations:

$$x_{ref}(t) = t \qquad \theta_{ref}(t) = \arctan\left(\frac{\dot{y}_{ref}}{\dot{x}_{ref}}\right)$$

$$\ddot{y}_{ref} = 4\sin(2t)\ ,\ \dot{y}_{ref}(t=0) = \frac{\pi}{10},\ y_{ref}(t=0) = 0$$

The initial position of the robot is x = 0.1 m, y = 0 and the $\theta = -15^0$. Robot mass and moment of inertia increase five times at t = 3 second. Controller performances when the robust gain K = 0 and the adaptation module is off are compared to the performances when the gain K is equal to 10 and the adaptation module is used. ***Figure 9*** illustrates the controller's tracking performances in the x-y plan when K = 10 and the adaptation module is on. We can see that parameter variations have negligible effects on the controller capability to follow the desired trajectory. ***Figure 10*** shows angles $\theta(t)$ and $\theta_{ref}(t)$ waveforms. One can note that a good tracking performance is achieved. The error $\theta(t) - \theta_{ref}(t)$ is highlighted at ***Figure 11***. ***Figure 12*** shows the controller performance when K = 10 and the adaptation module is switch off. We see that the performance degraded considerably when robot parameters change. The error $\theta(t) - \theta_{ref}(t)$ is highlighted at Fig 13. Simulation results clearly demonstrate the benefit of the robust adaptive characteristic.

Table 3 : *Model and controller parameters*

$k_4 = 100$	$M = 5kg$	$L^r = 2m$	$\mu_r = 1$	$\alpha = 10$
$k_5 = 100$	$J = 5N.m$	$L^f = 2m$	$\mu_f = 1$	$\gamma_1 = 2$
$k_6 = 10$	$k_1 = 5$	$k_2 = 5$	$k_3 = 10$	$\gamma_2 = 2$
$k_7 = 100$	$k_8 = 100$	$k_9 = 10$	$\gamma_3 = 2$	

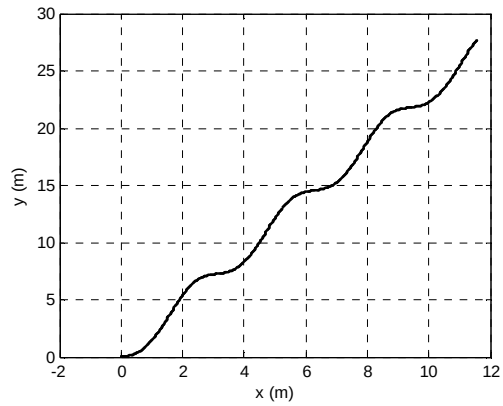


Figure 9 : *Robot motion in the x-y plane: discontinous line: actual trajectory, continuous line: desired trajectory*

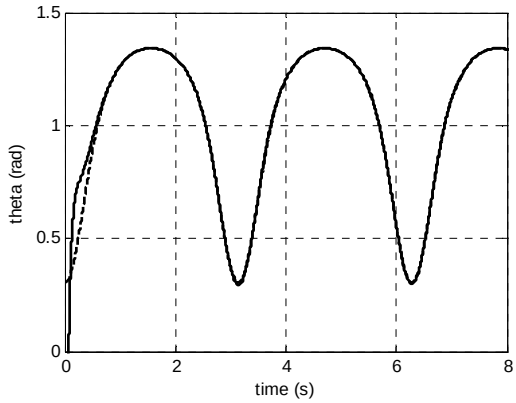


Figure 10 : *Orientation and its reference when K=1 0 and adaptation module is on. discontinous line: reference, continuous line: orientation*

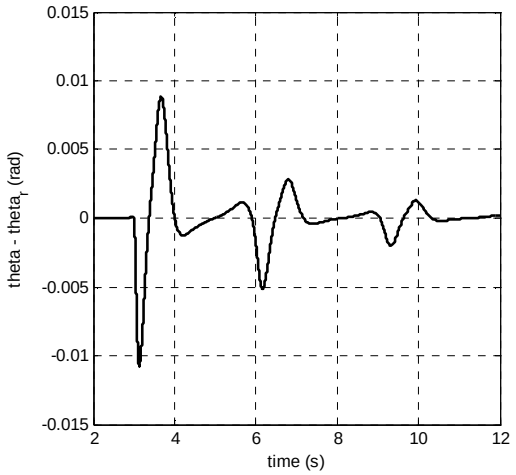


Figure 11 : *Orientation error when K= 10 and adaptation module is on*

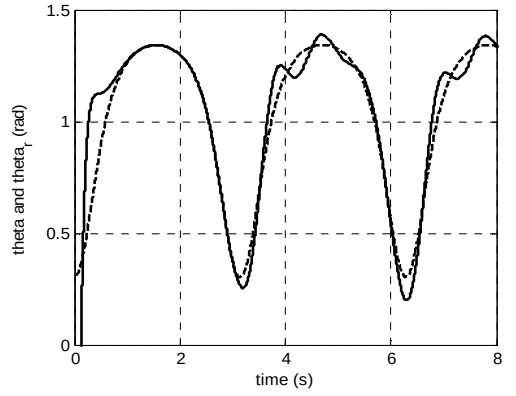


Figure 12 : *Orientation and its reference when K= 0 and adaptation module is off. discontinous line: reference, continuous line: orientation.*

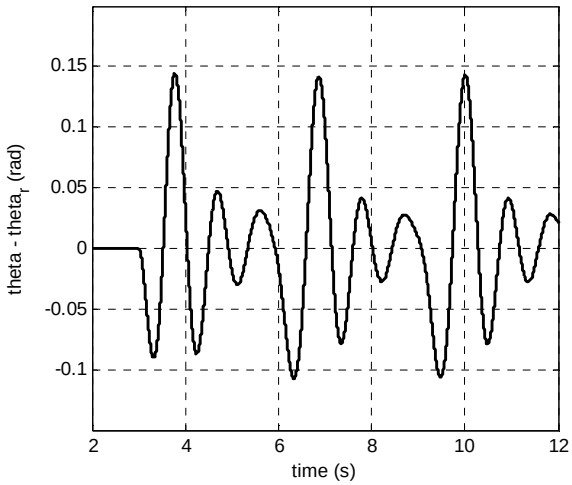


Figure 13 : *Orientation error when K= 0 and adaptation module is off*

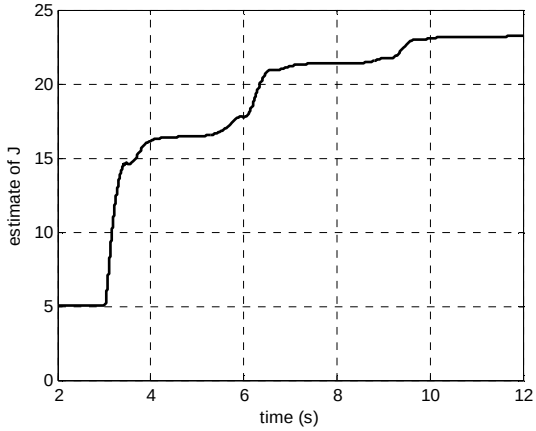


Figure 14 : *Estimated inertia when K= 10 and adaptation module is on*

V - CONCLUSION

An innovative approach is proposed to design a trajectory tracking controller for a wheeled mobile robot which parameters are assumed unknown. It combines the robust adaptive backsteeping nonlinear control design method and the well-known Newton-Raphson method, a nonlinear algebraic equation resolution method. Simulations are used to evaluate the controller performance. The results demonstrate the effectiveness of the new control design method. The control is able to follow perfectly a desired trajectory even if robot parameters change on route.

Appendix

This appendix shows the stability of the closed loop mobile robot system. Let us consider the following candidate Lyapunov function

$$V=\frac{1}{2}\sum_{j=x}^{\theta}\tilde{\Omega}_j^2+\frac{1}{2}\sum_{j=x}^{\theta}\tilde{V}_j^2+\frac{1}{2}(\tilde{x}^2+\tilde{y}^2+\tilde{\theta}^2)+\frac{1}{2}\sum_{j=1}^{3}\frac{1}{\gamma_j}\tilde{p}_j^2+\frac{1}{2}(\frac{M}{\alpha}\tilde{M}+\frac{J}{\beta}\tilde{J}) \quad (26)$$

Differentiating V and replacing the derivative of each variable in the result, and using the following properties enjoyed by the projection algorithm [11], $\left|\text{Proj}\left[y,\hat{\theta}_i\right]\right| \le |y|$ and $\tilde{\theta}_i\left|\text{Proj}\left[y,\hat{\theta}_i\right]\right| \ge \tilde{\theta}_i|y|$; yields,

$$\dot{V} \le -(k_1\tilde{\Omega}_x^2+k_2\tilde{\Omega}_y^2+k_2\tilde{\Omega}_\theta^2)-(k_4\tilde{x}^2+k_5\tilde{y}^2+k_6\tilde{\theta}^2)-(k_7\tilde{V}_x^2+k_8\tilde{V}_y^2+k_9\tilde{\omega}^2) \quad (27)$$

Integrating (27), one obtains

$$\lim_{t\to\infty}\int_{t_0}^{t} Z d\tau \le \lim_{t\to\infty}\int_{t_0}^{t} -\dot{V} d\tau \le V(0)-\lim_{t\to\infty} V(t) \le \infty \quad (28)$$

where

$$Z=(k_1\tilde{\Omega}_x^2+k_2\tilde{\Omega}_y^2+k_2\tilde{\Omega}_\theta^2)+(k_4\tilde{x}^2+k_5\tilde{y}^2+k_6\tilde{\theta}^2)+(k_7\tilde{V}_x^2+k_8\tilde{V}_y^2+k_9\tilde{\omega}^2)$$

By virtue of Barbalat's lemma [11], we can conclude that the system converges exponentially.

RÉFÉRENCES

[1] - Bishop R., 'Intelligent vehicle applications worldwide', *IEEE Intelligent Systems and Their Applications*, vol 15, no. 1, pp 78 – 81, Jan.-Feb. 2000.

[2] - Garcia, E.; Jimenez, M.A.; De Santos, P.G.; Armada, M., 'The evolution of robotics research'. *IEEE Robotics & Automation Magazine*, vol 14, no. 1, pp. 90 – 103, March 2007.

[3] - Kanayama Y., Kimura Y., Miyazaki F., and Nogichi T., 'A Stable Tracking Control Method for a Autonomous Mobile Robot', *Proceedings of the IEEE International Conference on Robotics and Automation*, vol.1 pp. 384 – 389, 13-18 May 1990.

[4] - Fierro, R.; Lewis, F.L., 'Control of a nonholonomic mobile robot using neural networks' *IEEE Transactions on Neural Networks*, vol 9, no. 4, pp. 589 – 600, July 1998.

[5] - Yin G., Chen N., and Li P., 'Improving Handling Stability Performance of Four-Wheel Steering Vehicle via μ-Synthesis Robust Control', *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2432-2439, Sep. 2007.

[6] - Jia Y., 'Robust control with decoupling performance for steering and traction of 4WS vehicles under velocity-varying motion**';** *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 554 – 569, May 2000.

[7] - M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. , Wiley, New York, 1995.

[8] - M. W. Hirsch and S. Smale, "On algorithms for solving f(x) = 0," Commun. *Pure Appl. Math*., vol. 32, pp. 281–312, 1979.

[9] - Jurgen Ackermann, *Robust Control: Systems with Uncertain Physical Parameters* , Springer-Verlag, London Limited, 1993.

[10] - Khatir, M.E.; Davidson, E.J.; 'Decentralized control of a large platoon of vehicles operating on a plane with steering dynamics', *Proceedings of the 2005 American Control Conference*, vol. 3 pp. 2159 – 2165, 8-10 June 2005.

[11] - Marino R. and Tomei P.,'Robust Adaptive State Feedback Tracking Control for Nonlinear Systems', *IEEE Trans. On Automatic Control*, vol. 43, pp. 84-89, 1999.