MOBILE ROBOT CONTROL USING SLIDING MODE AND NEURAL NETWORK

Vinícius Menezes de Oliveira *
Walter Fetter Lages** Edson Roberto de Pieri***

* Federal University of Rio Grande - Brazil
** Federal University of Rio Grande do Sul - Brazil
*** Federal University of Santa Catarina - Brazil
{vinicius,w.fetter}@ieee.org, edson@das.ufsc.br

Abstract: The complete model of a mobile robot can be seen as the cascade of a dynamic and a kinematic sub-systems. By taking advantage from this fact, a combined controller using a sliding mode controller to controle the kinematics sub-system and a neural network computed-torque control to control the dynamics is proposed. The proof of stability is based on Lyapunov theory. Since the kinematic controller conforms to the restrictions of Brockett’s theorem, the robot can be stabilized to desired posture. Experimental real-time results are presented. Copyright © 2003 IFAC

Keywords: Nonholonomic systems, mobile robot control, sliding mode, neural network

1. INTRODUCTION

Mobile robots can be used in a large variety of applications. Among the most known applications are bomb disposal, spatial exploration and underwater activities (e.g., oil exploration, replacement of divers in dangerous tasks). Besides that, there are some applications of autonomous guided vehicles in the reduction of traffic congest, pollution and accidents caused by human inability.

Due to this wide variety of mobile robots applications, the control of mobile robots has received much attention in technical literature. Contrarily to manipulator robots, mobile robots are, usually, nonholonomic systems, i.e., they have constraints that cannot be integrated. For mobile robots with differential drive it is possible to consider the kinematics and the dynamics as a cascade system. A number of papers consider only the control of the kinematics of the system, since the nonholonomic constraints are kinematic constraints (Goddavn and Egeland 1997). Some techniques used to control mobile robots kinematics are: discontinuous control, time-variant state feedback control (Morin and Samson 2000). Since the dynamical model of a mobile robot is a non-linear one, some techniques to control the dynamics of the system are linearizing feedback, adaptive linearizing control (Lages and Hemery 1998) and neural networks (Fierro and Lewis 1998, de Oliveira et al. 2000).

Differently of (Chwa et al. 2002), this work presents a kinematics control loop based on sliding modes and a dynamics control loop based on a neural network. The task imposed to the robot is to move toward a reference posture (origin of the system), and the control law conforms to the restrictions shown by Brockett (Brockett 1982). The neural network chosen due to its capability to learn and to the fact that the accurate measurement of the parameters of the dynamics of the mobile robot is very difficult.
2. MOBILE ROBOT MODELLING

The mobile robot used in this work (see figure 1) is a circular platform with 4 wheels, with 2 of them mounted on the same axis with a DC motor attached to each one. This robot is a differential-drive robot. See (Campion et al. 1996) for some insights on structural aspects of this class of robot.

A mobile robot system having an n-dimensional configuration space $\mathcal{C}$ with generalized coordinates $q = [q_1, \ldots, q_n]^T$ and with $m$ constraints can be described as follows (Yamamoto and Yun 1994):

$$\dot{q} = \mathbf{M}(q) \ddot{q} + \mathbf{V}(q) \dot{q} + \mathbf{F}(q) + \mathbf{\tau}_d = \mathbf{B}(q) \tau - \mathbf{A}(q) \lambda$$

(1)

where $\mathbf{M}(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix (symmetric and positive definite), $\mathbf{V}(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the centripetal and coriolis terms vector, $\mathbf{F}(q) \in \mathbb{R}^{n \times 1}$ is the friction terms, $\mathbf{\tau}_d$ denotes bounded unknown disturbances including unstructured unmodeled dynamics; $\mathbf{\tau}$ is the input vector.

Which can be rewritten as:

$$\mathbf{M}(q) \ddot{q} + \mathbf{V}(q, \dot{q}) \dot{q} + \mathbf{F}(q) + \mathbf{\tau}_d = \mathbf{B}(q) \tau - \mathbf{A}(q) \lambda$$

(6)

where $\mathbf{M}(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix (symmetric and positive definite), $\mathbf{V}(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the centripetal and coriolis terms vector, $\mathbf{F}(q) \in \mathbb{R}^{n \times 1}$ is the friction terms, $\mathbf{\tau}_d$ denotes bounded unknown disturbances including unstructured unmodeled dynamics; $\mathbf{\tau}$ is the input vector.

3. SLIDING MODE CONTROLLER

The purpose of the control law obtained by sliding mode technique is to drive the trajectory of the system to a pre-specified surface (defined by the designer) in the state space and maintain within this surface for all subsequent time. This surface is called switching surface.

Such surface is also known as sliding manifold because, at least in theory, once this surface is intercepted by the system trajectory, the control law would impose to the system trajectory to track the surface for all the future time (the trajectory will slide over the surface).

The dynamics of the process limited to this surface denotes the behavior of the controlled system. The first step is to design the sliding surface according to the desired behavior of the closed-loop system, such as convergence to the origin and parametric variation robustness (Bloch and Drakunov 1994).

It is important to note that to design such a sliding surface one has to consider that the control law should be able to switch from one value to the other in an infinitesimal time (Gao and Hung 1993). However, since in the real world it is not possible to have a null switching time, the chattering (Hsu and Costa 1996) is present, usually as a high frequency oscillation around the equilibrium point, and can excite high frequency modes of the dynamics.

3.1 Kinematic Control Loop

The controller proposed in this section has a simple functional structure. The system inputs are the reference posture and the state vector of the kinematic model. In this control scheme a Lyapunov function $V$ is designed to navigate the system to the origin of the work space $\mathcal{T} \subset \mathbb{R}^n$. The robot navigation to the origin of the configuration space $\mathcal{C}$ is guaranteed by the associated gradient $\varepsilon = \nabla V$. Once this gradient posses some necessary properties, the robot navigation will happen.

The control law is designed to keep the system trajectory along the $\varepsilon(x, y)$ gradient. The objective is to keep the linear velocity vector of the
vehicle collinear to the gradient; the velocity of the movement along the gradient can be determined independently. The trajectory is obtained by solving the following equation:

\[
\frac{dy}{dx} = \frac{\varepsilon_y(x, y)}{\varepsilon_x(x, y)}
\]  

(7)

and it must be smooth and continuous, so the first derivatives of the gradient associated to the Lyapunov function \( \frac{\partial \varepsilon}{\partial x}, \frac{\partial \varepsilon}{\partial y}, \frac{\partial \varepsilon_y}{\partial x} \) and \( \frac{\partial \varepsilon_y}{\partial y} \) are limited.

Let the orientation error be given by:

\[
\Delta \theta = \theta_e - \theta
\]  

(8)

The error dynamics is obtained by taking the first time-derivative of the equation 8, resulting in:

\[
\Delta \dot{\theta} = F(x, y, \theta) v - \omega
\]  

(9)

where

\[
F = \left[ \frac{\varepsilon_x \frac{\partial \varepsilon_x}{\partial x} + \varepsilon_y \frac{\partial \varepsilon_y}{\partial x} - \varepsilon_x \frac{\partial \varepsilon_x}{\partial y} - \varepsilon_y \frac{\partial \varepsilon_y}{\partial y}}{||\varepsilon||^3 \cos(\theta)} \right]
\]  

(10)

Defining the control input \( \omega \) as:

\[
\omega = F(x, y, \theta) v + \xi \text{sign}(\Delta \theta) \sqrt{||\Delta \theta||}
\]  

(11)

ensures the convergence of \( \Delta \theta \) to zero will be in a finite time, with \( \xi \) as a positive and finite scalar constant (Guldner and Utkin 1994).

To ensure the existence of the sliding mode with a finite control is necessary that the term \( F(x, y, \theta) v \) in 11, be finite. The velocity control used to give a finite \( F(x, y, \theta) v \) is defined as follow:

\[
v(t) \triangleq -||\varepsilon|| \tilde{v}(t)
\]  

(12)

where \( \tilde{v}(t) \) is a limited auxiliary control input. It is possible to observe that \( \varepsilon(0, 0) \) and, consequently, \( v(t) \) and \( F(x, y, \theta) v \) also converge to zero at the origin. When the sliding mode occurs along \( \Delta \theta = 0 \), the gradient field \( \varepsilon \) is followed by the system, reducing the kinematic system 4, under the control 11 and 12, to the following system:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \begin{bmatrix}
\varepsilon_x \\
\varepsilon_y
\end{bmatrix} \begin{bmatrix}
v(t) \\
\tilde{v}(t)
\end{bmatrix} = -\begin{bmatrix}
\varepsilon_x \\
\varepsilon_y
\end{bmatrix} \begin{bmatrix}
v(t) \\
\tilde{v}(t)
\end{bmatrix}
\]  

(13)

At first sight is possible to think the reduced system is still under the restrictions of Brockett’s theorem (Brockett 1982), since the system has a state dimension higher than control input dimension. But, in a more detailed analysis, one can observe that due to the sliding mode technique, the gradient of the Lyapunov function is exactly tracked, reducing the order of the system by the restriction of the movement to the resulting manifold of 7.

### 3.2 Control Law

Let \( V(x, y) \) be candidate to the Lyapunov function, expressed by:

\[
V(x, y) = \frac{1}{2} \left( \frac{x^2}{a} + y^2 \right) > 0
\]  

(14)

where \( a \) is a positive scalar constant and \( V \) a positive definite function. Let the associated gradient be given as:

\[
\varepsilon(x, y) = -\nabla V = \begin{bmatrix}\frac{x}{a} \\
-y
\end{bmatrix}
\]  

(15)

According to the expression 7 and with the associated gradient 15, we have trajectories of the type:

\[
y = \gamma |x|^a
\]  

(16)

where \( \gamma \) depends on the initial conditions, without importance to the control.

The control input responsible for the co-linear orientation of the robot to the gradient \( \varepsilon(x, y) \) is obtained replacing equation 15 in 11, resulting in:

\[
\omega = \frac{-x \sin \theta - y \cos \theta}{\sqrt{x^2 + (ay)^2}} |\Delta \theta| \sqrt{||\Delta \theta||}
\]  

(17)

with the orientation error expressed by equation 8 and velocity control given by:

\[
v(t) = -\sqrt{x^2 + (ay)^2} \tilde{v}(t)
\]  

(18)

While the sliding mode exists, we have \( \theta = \theta_e \) and the robot movement is determined by the reduced system 13 with the restriction 7. Taking:

\[
\tilde{v}(t) = -v_0
\]  

(19)

the position error has an exponential convergence. The convergence of the orientation \( \theta \) to zero can be determined by analysing:

\[
\theta = \theta_e = \arctan\left(\frac{ay}{x}\right) = \arctan(ax^{a-1})
\]  

(20)

during the time evolution of \( x \) (equation 13), with control 18 and \( a > 1 \).

Considering the region around the origin and the approximation \( \tan(\alpha) \approx \alpha \), to small values of \( \alpha \), is possible to conclude that \( \theta \) converges exponentially to zero.

### 4. Dynamic Controller

In this section is adopted an strategy based on the application of artificial neural networks to control the mobile robot dynamics (Fierro and Lewis 1998).

The neural network used in this work has 6 neurons in the input layer, 8 neurons in the hidden layer and, finally, 2 neurons in the output layer. Thus, the output signal of the neural net is given by the following equation:

\[
y(x) = W^T \sigma(V^T x)
\]  

(21)

where \( x \in \mathbb{R}^{6 \times 1} \) is the input vector of the neural network, \( V \in \mathbb{R}^{6 \times 8} \) is the weight matrix
between the input layer and the intermediate layer and \( \mathbf{W} \in \mathbb{R}^{k \times 2} \) is the weight matrix between the intermediate layer and the output layer. The function \( \sigma(\cdot) \) is the so called activation function and in this work we decided to use the following sigmoidal function:

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]

(22)

The application of neural networks in the control of the dynamics of a mobile robot is intuituated from the intrinsic ability of neural networks to map unknown nonlinear functions. Based on this feature above and considering \( \Gamma(x) \) a continuous function \( \mathbb{R}^n \to \mathbb{R}^m \) is possible to show that, making \( x \) restricted to a compact subset \( U_n \subset \mathbb{R}^n \), for a given number \( N \) of neurons in the hidden layer, there is a configuration of the neural network such that:

\[
\Gamma(x) = \mathbf{W}^T \sigma(\mathbf{V}^T x) + \epsilon
\]

(23)

where \( \epsilon \) is the neural network approximation error.

Thus, an estimative of \( \Gamma(x) \) is given by:

\[
\hat{\Gamma}(x) = \hat{\mathbf{W}}^T \sigma(\hat{\mathbf{V}}^T x) + \epsilon
\]

(24)

where \( \hat{\mathbf{V}} \) and \( \hat{\mathbf{W}} \) are estimatives of the ideal weight matrices.

In this work we use the on-line training technique, which has only one phase, because the weights are adjusted during the execution phase. The weights are adjusted according to the following equations:

\[
\Delta \hat{\mathbf{W}} = \mathbf{F}_c \sigma(\hat{\mathbf{V}}^T x) \mathbf{e}_c^T - \mathbf{F}_c \sigma'(\hat{\mathbf{V}}^T x) \hat{\mathbf{V}}^T x \mathbf{e}_c^T
\]

\[
- k \mathbf{F} ||\mathbf{e}_c|| \hat{\mathbf{W}}
\]

\[
\Delta \hat{\mathbf{V}} = \mathbf{G}_c \sigma'(\hat{\mathbf{V}}^T x) \hat{\mathbf{W}} \mathbf{e}_c^T
\]

\[
- k \mathbf{G} ||\mathbf{e}_c|| \hat{\mathbf{V}}
\]

(25)

(26)

where the design parameters \( \mathbf{F} \) and \( \mathbf{G} \) are positive definite matrices and \( k > 0 \).

5. CONTROL STRUCTURE OF THE ROBOT

In this section we present, besides the overall structure of the controller, the proof of stability of the neural network and, as a consequence, the stability of the controller.

Once the dynamic model of the mobile robot is defined, we present in figure 2 the block diagram of the control scheme proposed. As already mentioned, we have a block related to the control of the kinematics and another block related to the control of the dynamics, where the neural network is involved.

The velocity tracking error is given by

\[
\mathbf{e}_c = \mathbf{v}_c - \mathbf{v}
\]

(27)

Deriving equation 27 and substituting it in equation 6, the dynamics of the robot can be described by:

\[
\mathbf{M} \mathbf{e}_c = -\nabla \mathbf{e}_c - \mathbf{r} + f(x) + \mathbf{r}_d
\]

(28)

where the nonlinear function \( f(x) \) of the robot is given by:

\[
f(x) = \mathbf{M} \mathbf{v}_c + \nabla \mathbf{v}_c + \mathbf{F}
\]

(29)

The input vector of the neural network is defined as follows:

\[
x = \left[ \begin{array}{c} u^T \\ v_c^T \\ \dot{v}_c^T \end{array} \right]^T
\]

(30)

Continuing, we obtain a control law expressed by:

\[
\mathbf{r} = \hat{\mathbf{f}} + \mathbf{K}_4 \mathbf{e}_c - \gamma
\]

(31)

where \( \mathbf{K}_4 \) is a positive definite matrix of gains and \( \hat{\mathbf{f}}(x) \) is an estimative of function \( f(x) \) of the robot, which is performed by the neural network. The \( \gamma \) signal is to guarantee the control law robustness to unstructured unmodeled disturbances (Fierro and Lewis 1998).

Applying this control in equation 28, the closed loop system can be described as:

\[
\mathbf{M} \mathbf{e}_c = - (\mathbf{K}_4 + \nabla_m) \mathbf{e}_c + \hat{\mathbf{f}}(x) + \mathbf{r}_d + \gamma
\]

(32)

with \( \hat{\mathbf{f}} = f - \dot{\mathbf{f}} \).

Consider the following Lyapunov function candidate:

\[
\mathbf{V} = \mathbf{V}_1 + \mathbf{V}_2
\]

(33)

where \( \mathbf{V}_1 \) is defined in expression 14, with \( \mathbf{V}_1 > 0 \) and \( \mathbf{V}_1 < 0 \), and

\[
\mathbf{V}_2 = \frac{1}{2} \left[ \mathbf{e}_c^T \mathbf{M} \mathbf{e}_c + tr \{ \hat{\mathbf{W}}^T \mathbf{F}^{-1} \hat{\mathbf{W}} \} + tr \{ \hat{\mathbf{V}}^T \mathbf{G}^{-1} \hat{\mathbf{V}} \} \right]
\]

(34)

The first time-derivative of 33 results:

\[
\dot{\mathbf{V}}_2 = \mathbf{e}_c^T \mathbf{M} \mathbf{e}_c + \mathbf{e}_c^T \hat{\mathbf{W}} \mathbf{e}_c + tr \{ \hat{\mathbf{W}}^T \mathbf{F}^{-1} \hat{\mathbf{W}}^T \} + tr \{ \hat{\mathbf{V}}^T \mathbf{G}^{-1} \hat{\mathbf{V}}^T \}
\]

(35)

then by replacing 32 in 34 and using the skew-symmetry property gives:

\[
\dot{\mathbf{V}}_2 = - \mathbf{e}_c^T \mathbf{K}_4 \mathbf{e}_c + \mathbf{e}_c^T (\delta + \gamma) +
+k || \mathbf{e}_c || tr \{ \hat{\mathbf{Z}} (\mathbf{Z} - \hat{\mathbf{Z}}) \}
\]

(36)

where \( \delta \) is the disturbance term expressed as:

\[
\delta = \mathbf{W} \dot{\mathbf{e}}^T x + \mathbf{W}^T \mathbf{O}(\mathbf{V}^T x) + \epsilon + \mathbf{r}_d
\]

Now considering that

\[
\mathbf{O}(\mathbf{V}^T x)
\]

denotes the higher order terms in the Taylor series
Fig. 2. Block diagram of the controller.

$$tr(\dot{Z}(Z - \bar{Z})) = \langle \dot{Z}, Z > - ||\dot{Z}||^2$$

$$\leq ||\dot{Z}|| ||Z|| - ||\dot{Z}||$$  \hspace{1cm} (37)

then expression (36) can be rewritten as:

$$\dot{V}_2 \leq -e_c^T e_c - ||e_c|| \left[ k ||\dot{Z}|| \left( ||\dot{Z}|| - Z_M \right) + K_4 ||e_c|| - C_0 - C_1 ||\dot{Z}|| \right]$$  \hspace{1cm} (38)

Let the auxiliary constant \( C_3 \) be given by:

$$C_3 \triangleq \frac{1}{2} \left( Z_M + \frac{C_1}{k} \right)$$  \hspace{1cm} (39)

Manipulating the terms in the brackets in equation 38, substituting 39 and square completing we obtain:

$$\dot{V}_2 \leq -||e_c|| \left[ K_4 ||e_c|| + k(||\dot{Z}|| - C_3)^2 - kC_3^2 - C_0 \right] - e_c^T e_c$$  \hspace{1cm} (40)

For the function \( \dot{V}_2 \) to be negative, we must guarantee that

$$||e_c|| > \frac{k_2C_3^2 + C_0}{K_4}$$  \hspace{1cm} (41)

or

$$||\dot{Z}|| > C_3 + \sqrt{C_3^2 + \frac{C_0}{k}}$$  \hspace{1cm} (42)

Bearing in mind the Lyapunov theory and LaSalle theorem we show that \( ||e_c|| \) and \( ||\dot{Z}|| \) is uniformly locally stable (according to the figure 3).

6. EXPERIMENTAL RESULTS

Finally we present the experimental results. The objective is to drive the robot to the origin of the system. The initial position of the robot is \( x_0 = 1.0(m), y_0 = 1.0(m) \) and \( \theta_0 = 0.0(\text{rd}) \). The gains of the kinematic controller are \( a = 2.0, \xi = 10.0 \) and \( v_0 = 0.5(m/s) \). To the dynamic controller we have \( K_4 = 50 \, I, k_z = 0.001, k = 0.01, F = 3.0 \) and \( G = 4.0, \) where \( I \) is the identity matrix with appropriate dimensions.

In figure 4 we have the trajectory described by \( x \) coordinate of the robot during the movement to the origin of coordinate system. Next, in figure 5 is described the \( y \) coordinate trajectory. The orientation of the robot during the approximation to the origin can be visualized in figure 6.

Fig. 3. Region of stability to the neural network based controller.

Fig. 4. Position \( x \) of the robot during motion.

The planar trajectory executed by the robot to achieve the origin \((0,0,0)\) is in figure 7. The torques provided by the right (solid line) and left (dotted line) motors are depicted in figure 8.
As mentioned in section 3, it is possible to observe in figures 6 and 8 the presence of the chattering effect, but this effect did not cause any undesirable effect in the control of the robot, as can be seen in the trajectory executed by the mobile robot when converged to the origin (0, 0, 0).

REFERENCES


