ADAPTIVE LINEARIZING CONTROL OF MOBILE ROBOTS

Walter Fetter Lages* Elder M. Hemerly**

* Fundação Universidade do Rio Grande
Departamento de Física
Av. Itália, Km. 8, 96201-900 Rio Grande - RS, Brazil
w.fetter@ieee.org

** Instituto Tecnológico de Aeronáutica
Departamento de Controle e Conversão de Energia
CTA - ITA - IEEE, 12228-900 São José dos Campos - SP, Brazil
hemerly@ele.ita.cta.br

Abstract: Controllers for mobile robots are often designed by taking into account only the kinematics of the robot. The difficulties associated with the dynamic effects such as mass and inertia moments arise from the nonlinearities of the robot model and from the uncertain about these parameters. This paper proposes a controller based on the certainty equivalence principle that considers the dynamic effects on the robot. The proposed controller deals with the nonlinearities by means of feedback linearization, while the unknown parameters are estimated by a modified least squares algorithm. Simulated and experimental results are presented. Copyright ©1998 IFAC

Keywords: Adaptive control, Mobile robots, Linearizable systems, Autonomous vehicles, Nonlinear control

1. INTRODUCTION

While adaptive control has been used for control of manipulator robots for some time, its use for mobile robot control is not widespread. Of course, there are several works using less traditional variants of adaptive control, such as neural networks behavior control and other soft computing approaches. These techniques seem to be more useful for higher control levels. For lower control levels, more traditional control methods should be used, because a failure here could deliver uncontrolled power, thereby compromising the robot and the environment physical security. Formal proofs of stability are thus required.

The lack of papers on adaptive control of mobile robots is in part due to the fact that many works use only the kinematic model. The parameters needed by this kind of model are related to the geometry of the robot only (Campion et al. 1996). Hence, as they can be determined by calibration procedures (Borenstein et al. 1996), the use of adaptive control can provide no benefits.

The dynamic model of the robot includes parameters such as mass and inertia moments. Contrary to geometric parameters, dynamic parameters can vary by a reasonable amount, even during a single mission of the robot. They can vary, for example, due to variations on payload mass, fuel consumption, payload spatial distribution, etc. Therefore the values for dynamic parameters can not be obtained by calibration, thus justifying the use of adaptive control.

Although a general theory for adaptive control of nonlinear systems is not available, under some mild restrictive conditions (Pereira 1995), it is
possible to develop adaptive controllers for a representative class of nonlinear systems.

In this work it is shown that a mobile robot belongs to such a class and a linearizing adaptive control law is designed. In particular, model reference adaptive control is employed. The proposed adaptation strategy is based on a recursive least squares formulation, where the cost to minimize includes the parameter estimation error and the output error.

When a system is linearized by feedback, the higher order terms are not neglected, as in Taylor linearization, but canceled by adequate system inputs (Isidori 1995). The main drawback of this approach is that if the parameters and state of the system are not well known, the higher order terms will not be exactly canceled, thereby hampering system performance and stability. The problem of parameter uncertain is addressed here by means of adaptive control. The state determination problem can be solved by data fusion methods. For details on this second topic, see Borenstein et al. (1996).

2. MODELING

This work considers a differential drive mobile robot.

The dynamic model of the mobile robot (Fig. 1) can be obtained using the Lagrangian formalism (Yamamoto and Yun 1994), and in state space has the form

$$\dot{x} = \begin{bmatrix} S \nu \\ f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (S^TMS)^{-1} \end{bmatrix} \tau$$

(1)

where the state vector is defined as

$$x = \begin{bmatrix} q \\ \nu \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \phi \\ \theta_t \\ \theta_t \hat{\theta}_t \end{bmatrix}^T$$

(2)

with

$$q = \begin{bmatrix} x_c \\ y_c \phi \\ \theta_t \\ \theta_t \hat{\theta}_t \end{bmatrix}^T$$

(3)

$$\nu = \begin{bmatrix} \theta_t \\ \hat{\theta}_t \end{bmatrix}^T$$

(4)

The input vector, composed by the torque on each wheel, is

$$\tau = \begin{bmatrix} \tau_t \\ \tau_t \end{bmatrix}^T$$

(5)

and

$$f_2 = (S^TMS)^{-1} \left( -S^T M \dot{\phi} - S^T V \right)$$

(6)

with

$$M(q) = \begin{bmatrix} m & 0 & m_c d \sin \phi & 0 \\ 0 & m & -m_c d \cos \phi & 0 \\ m_c d \sin \phi & -m_c d \cos \phi & I_c & 0 \\ 0 & 0 & 0 & I_w \end{bmatrix}$$

(7)

$$V(q, \dot{q}) = \begin{bmatrix} m_c d \dot{\phi}^2 \sin \phi \\ m_c d \dot{\phi}^2 \cos \phi \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(8)

$$S(q) = \begin{bmatrix} c (b \cos \phi + d \sin \phi) & c (b \cos \phi - d \sin \phi) \\ c (b \sin \phi - d \cos \phi) & c (b \sin \phi + d \cos \phi) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(9)

where

$$P_o = \text{intersection point between the robot simetry axis and the wheel axis;}$$

$$P_c = \text{center of mass;}$$

$$d = \text{distance from } P_o \text{ to } P_c;$$

$$b = \text{distance from wheels to simetry axis;}$$

$$r = \text{wheels radius;}$$

$$m_c = \text{robot mass without wheels and actuators;}$$

$$m_w = \text{mass of each wheel/actuator set;}$$

$$I = \text{moment generated by } m_c \text{ and } m_w \text{ with respect to the vertical axis trough } P_o;$$

$$I_w = \text{moment generated by the wheel/actuator set with respect to the wheel axis;}$$

$$\theta_t, \hat{\theta}_t = \text{angular displacement of the tracking wheels}$$

$$c = r/(2b)$$

3. FEEDBACK LINEARIZATION

Consider the class of multivariable nonlinear systems with the same number of inputs and outputs, which can be written (Isidori 1995) as

$$\dot{x} = f(x) + g_1(x)u_1 + \cdots + g_p(x)u_p$$

(10)

$$y_1 = h_1(x)$$

(11)

$$y_p = h_p(x)$$

with \(x \in \mathbb{R}^n, \ u \in \mathbb{R}^p\) and \(y \in \mathbb{R}^p\) and \(f(\cdot), g(\cdot)\) and \(h(\cdot)\) are smooth functions.
The time differential of the output $y_j$ is

$$
\dot{y}_j = L_t h_j(x) + \sum_{i=1}^{P} L_{gi} h_j(x) u_p \quad (12)
$$

where $L_t h_j(x)$ and $L_{gi} h_j(x)$ are the Lie differentials of the output function $h_j(x)$ with respect to $f(x)$ and $g(x)$, respectively, given by

$$
L_t h_j(x) = \sum_{k=1}^{n} \frac{\partial h_j(x)}{\partial x_k} \dot{x}_{k}\tag{13}
$$

$$
L_{gi} h_j(x) = \sum_{k=1}^{n} \frac{\partial h_j(x)}{\partial x_k} \dot{x}_{kg}\tag{14}
$$

with $\dot{x}_{k}$ and $\dot{x}_{kg}$ representing the portions due to $f(x)$ and $g(x)$ of the k-th state equation of the system described by (10) (11).

Regarding (12), if each $L_{gi} h_j(x) = 0$, then no input variable should appear on the time differential of the system output. Let $\gamma_1$ be the lowest integer such that at least one of the input variables appear explicitly on $y_j^{\gamma_1}$, i.e.

$$
y_j^{\gamma_1} = L_{gi}^{\gamma_1} h_j(x) + \sum_{k=1}^{P} L_{gi} \left( L_{gi}^{\gamma_1-1} h_j(x) \right) u_p \quad (15)
$$

where

$$
L_{gi}^{\gamma_1} h_j(x) = \sum_{k=1}^{n} \frac{\partial L_{gi}^{\gamma_1-1} h_j(x)}{\partial x_k} \dot{x}_{k}\tag{16}
$$

$$
L_{gi} L_{gi}^{\gamma_1-1} h_j(x) = \sum_{k=1}^{n} \frac{\partial L_{gi} L_{gi}^{\gamma_1-1} h_j(x)}{\partial x_k} \dot{x}_{kg}\tag{17}
$$

with at least one term $L_{gi} L_{gi}^{\gamma_1-1} h_j(x) \neq 0$ for all $x$ in the region where the linearization is valid.

By defining

$$
E(x) = \begin{bmatrix}
L_{gi} L_{gi}^{\gamma_1-1} h_1 & \cdots & L_{gi} L_{gi}^{\gamma_1-1} h_1 \\
\vdots & \ddots & \vdots \\
L_{gi} L_{gi}^{\gamma_1-1} h_p & \cdots & L_{gi} L_{gi}^{\gamma_1-1} h_p
\end{bmatrix}\quad (18)
$$

expression (15) can be written in the form

$$
\begin{bmatrix}
\dot{y}_1^{\gamma_1} \\
\vdots \\
\dot{y}_p^{\gamma_1}
\end{bmatrix} = \begin{bmatrix}
L_{gi}^{\gamma_1} h_1 \\
\vdots \\
L_{gi}^{\gamma_1} h_p
\end{bmatrix} + E(x) \begin{bmatrix}
u_1 \\
\vdots \\
u_p
\end{bmatrix}\quad (19)
$$

Therefore, if the inverse matrix of $E(x)$ exists for all $x$ belonging to the region of interest, under the state feedback law

$$
u(x) = -E^{-1}(x) \begin{bmatrix}
L_{gi}^{\gamma_1} h_1 \\
\vdots \\
L_{gi}^{\gamma_1} h_p
\end{bmatrix} + E^{-1}(x)v \quad (20)
$$

the system (10-11) becomes, in closed loop

$$
\begin{bmatrix}
y_1^{\gamma_1} \\
\vdots \\
y_p^{\gamma_1}
\end{bmatrix} = \begin{bmatrix}
v_1 \\
\vdots \\
v_p
\end{bmatrix}\quad (21)
$$

which is linear under the input-output point of view.

3.1 Output Equation

In a similar way to Yamamoto and Yun (1994), an arbitrary reference point $P_r$ is chosen. The robot should be controlled to force the point $P_r$ to follow the desired trajectory. The position of $P_r$ with respect to the global reference frame $\{X_0,Y_0\}$ is given by

$$
x_r = x_c + x_r^e \cos \phi - y_r^e \sin \phi\quad (22)
$$

$$
y_r = y_c + x_r^e \sin \phi + y_r^e \cos \phi\quad (23)
$$

where $(x_r^e, y_r^e)$ is the position of $P_r$ with respect to the $\{X_c, Y_c\}$ frame.

The expressions (22-23) are the output equations of the system described by (1), i.e.

$$
y = h(q) = \begin{bmatrix} x_r \ y_r \end{bmatrix}^T\quad (24)
$$

which, by twice differentiating (24) with respect to time gives

$$
\ddot{y} = \alpha + \beta f_2 + \beta \left(S^T M S\right)^{-1} \tau\quad (25)
$$

where $\alpha$ and $\beta$ are computed from the system state and known geometric parameters.

Expression (25) can be represented in the same form of (19) with

$$
E(x) = \beta \left(S^T M S\right)^{-1}\quad (26)
$$

and

$$
L_{gi} h(x) = \alpha + \beta f_2\quad (27)
$$

Therefore, according to (20), the linearizing control input should be

$$
\tau = -E^{-1} L_{gi} h + E^{-1} v\quad (28)
$$

3.2 Equivalent Input

The equivalent input vector is generated by reference model. More specifically, the equivalent input vector is given by

$$
v = \begin{bmatrix} v_1 \ v_2 \end{bmatrix}^T\quad (29)
$$

with

$$
v_1 = \dot{y}_m + \alpha_2 (\dot{y}_m - \ddot{x}_r) + \alpha_1 (y_m - x_r)\quad (30)
$$
\[ v_2 = \tilde{y} m_2 + \alpha_4 (\tilde{y} m_2 - \tilde{y}) + \alpha_3 (y m_2 - y) \tag{31} \]

where \( y m_1 \) and \( y m_2 \) are the output of the reference model associated to each component of (24), described by transfer functions with the following form

\[ G_m(s) = \frac{\alpha_1}{s^2 + \alpha_2 s + \alpha_1} \tag{32} \]

with \( \alpha_1 \) being design parameters that will shape the dynamics of the robot.

4. ADAPTIVE FEEDBACK LINEARIZATION

As discussed in section 1, if there are uncertain associated to the nonlinear functions \( f(x) \) and \( g(x) \) in (10), the cancellation will not be exact, resulting in a nonlinear input-output relation. To overcome these difficulties it is proposed here, based on Sastry and Isidori (1989) a linearizing controller with parameter adaptation capabilities.

Consider the feedback linearization law given by (28), which in the absence of parameter uncertain is completely computable by means of (26-27). If there are parameter uncertain, (28) is modified to an adaptive form

\[ \tau = -\hat{E}^{-1} \hat{L}_i^2 h + \hat{E}^{-1} v \tag{33} \]

where \( \hat{E} \) and \( \hat{L}_i^2 h \) are the estimates of \( E \) and \( L_i^2 h \).

From (33) it is easy to see that

\[ v = \hat{L}_i^2 h + \hat{E} \tau \tag{34} \]

Hence, by appropriately rewriting (21), the following expression can be obtained

\[ \tilde{y} = v + L_i^2 h - \hat{L}_i^2 h + E \tau - \hat{E} \tau \tag{35} \]

which, by assuming that \( \hat{E} \) and \( \hat{L}_i^2 h \) are linearly parameterizable in the unknown parameter \( \theta \), can be recast as

\[ \tilde{y} = v + W \Phi \tag{36} \]

where \( \Phi = \theta - \hat{\theta} \) is the vector of parameter estimation errors and \( W \) is the regression matrix.

The Recursive Least Squares (RLS) estimation algorithm used here is based on the following cost function (Pereira 1995)

\[
J(\Phi, e) = \int_0^t L^{-1}(s) W \Phi T W^T L^{-1}(s) \, dt + \int_0^t ee^T \, d\tau \tag{37}
\]

where \( e(t) \) represents the tracking error vector between the system output vector and the model reference output vector, \( W \Phi \) represents the parameter estimation error and \( L(s) \) is the polynomial defined by

\[ L(s) = s^2 + \alpha_2 s + \alpha_1 \tag{38} \]

with \( \alpha_1 \) being the coefficients of the characteristic polynomial of the reference model. The notation \( L^{-1}(s) W \Phi \) denotes the convolution of the inverse Laplace transform of \( L^{-1}(s) \) with \( W \Phi \).

The minimization of the cost function (37) with respect to \( \hat{\theta} \) results the parameter adaptation law

\[
\dot{\Phi} = -PW_i^T \left[ I + W_i PW_i^T + W_i W_i^T \right]^{-1} W_i \Phi
- \alpha_0 \Phi P X^T \left[ I + XP X^T + XX^T \right]^{-1} X \Phi \tag{39}
\]

\[
\dot{P} = -PW_i^T \left[ I + W_i PW_i^T + W_i W_i^T \right]^{-1} W_i P \tag{40}
\]

where \( W_i \) is the regression matrix filtered by \( F(s) \), \( \alpha_0 \) is the adaptation gain and \( X \) is the regression matrix filtered by the reference model \( G(s) \).

The filter \( F(s) \) has the form

\[ F(s) = \frac{1}{(\varepsilon s + 1)} \tag{41} \]

with \( \varepsilon \in \mathbb{R}^+ \), a project parameter based on the time constants of the reference model.

Note that while \( \Phi \) can not be computed in real-time, the variables \( W_i \Phi \) and \( X \Phi \) can be computed as

\[
W_i \Phi = \bar{e}_i + \alpha_0 \bar{e}_i + \alpha_1 \bar{e}_i + F(s) \left( W \bar{\theta} \right) + W_i \dot{\theta} \tag{42}
\]

and

\[
X \Phi = e + L^{-1}(s) \left( W \bar{\theta} \right)
- \left( L^{-1}(s) W \right) \bar{\theta} \tag{43}
\]

where \( e_i \), \( \bar{e}_i \) and \( \bar{e}_i \) are the output error vector \( e \) filtered through \( F(S) \), \( sF(s) \) and \( s^2 F(s) \), respectively. Note that these filters are proper by contruction.

5. MODEL PARAMETRIZATION

The Twil mobile robot, built in our labs will be used as an example in this section, but any mobile robot could be used. It is assumed that the mass and inertia parameters \( (m_c, m_w, I, I_w) \) - see
section 2) are unknown. The geometric parameters are assumed to be known. In particular the distance \( d \) from \( P_r \) to \( P_c \) is known to be zero (Fig. 1).

From (35) and (36) it can be seen that
\[
W \Phi = L_2^2 h - \hat{L}_2^2 h + E_T - \hat{E}_T
\]  \( (44) \)
and by replacing \( E \) and \( L_2^2 h \) from (26) and (27) results in
\[
W \Phi = \beta \left( f_2 - \hat{f}_2 \right) \\
+ \beta \left( (S^T MS)^{-1} \tau - (S^T \hat{M} S)^{-1} \tau \right) \tag{45}
\]

Now, by recalling that for Twil \( d = 0 \), it can be concluded from (6) (7) and (8) that \( f_2 \) and \( \hat{f}_2 \) are also zero. Also, it can be shown by literally computing \((S^T MS)^{-1}\) that it has the following structure
\[
(S^T MS)^{-1} = \begin{bmatrix} K_p & K_s \\ K_s & K_p \end{bmatrix} \tag{46}
\]
where \( K_p \) and \( K_s \) are constants depending on the unknown robot parameters.

Recalling that \( \Phi \) is the vector of parameter estimation errors and that \( \tau = [\tau_r \ \tau_I] \), expression (45) can be written as
\[
W \Phi = \beta \begin{bmatrix} \tau_r & \tau_I \end{bmatrix} \begin{bmatrix} K_p - \hat{K}_p \\ K_s - \hat{K}_s \end{bmatrix} \tag{47}
\]
hence,
\[
W = \beta \begin{bmatrix} \tau_r & \tau_I \end{bmatrix} \tag{48}
\]
\[
\Phi = \begin{bmatrix} K_p - \hat{K}_p \\ K_s - \hat{K}_s \end{bmatrix}^T \tag{49}
\]
which means that
\[
\hat{\theta} = [\hat{K}_p \ \hat{K}_s]^T \tag{50}
\]

Since the unknown parameters \( \theta \) are assumed to be constant, it is obvious that \( \hat{\theta} = -\beta \). Therefore, given initial estimates for \( \hat{\theta} \), it can be updated recursively by using (40) and (40). As \( f_2 = \hat{f}_2 = 0 \) and \( \alpha \) does not depend on unknown parameters, it follows from (27) and (44) that \( L_2^2 h = \hat{L}_2^2 h = \alpha \). Also, from (26) and (46) and from the fact that \( \beta \) does not depend on unknown parameters too, it can be written
\[
\hat{E} = \beta (S^T \hat{M} S)^{-1} = \beta \begin{bmatrix} \hat{K}_p & \hat{K}_s \\ \hat{K}_s & \hat{K}_p \end{bmatrix} \tag{51}
\]
and from (33) the input vector \( \tau \) can be computed as
\[
\tau = \begin{bmatrix} \hat{K}_p & \hat{K}_s \\ \hat{K}_s & \hat{K}_p \end{bmatrix}^{-1} \beta^{-1} (-\alpha + \nu) \tag{52}
\]

\[\text{Fig. 2. Reference model output (dashed) and robot position (solid) in } X_0 \text{ direction.}\]

\[\text{Fig. 3. Reference model output (dashed) and robot position (solid) in } Y_0 \text{ direction.}\]

Note that even if \( f_2 \neq 0 \), the same type of parametrization would be possible, with larger \( \Phi \) and \( W \) due to the unknown parameters in \( f_2 \).

6. SIMULATION RESULTS

Simulated results for Twil mobile robot are shown in Figs. 2, 3 and 4. The reference model was
\[
G_m(\hat{s}) = \frac{10}{s^2 + 1.4\sqrt{10}s + 10} \tag{53}
\]

The initial values for the estimated parameters were chosen to be 1.5 times the actual ones. The covariance matrix was initialized with \( P = 100I \), and null initial conditions were used on all filters. A simulated sampling period of 10ms was used.

As anticipated, the tracking performance of the controller increases with time due to the tuning of the unknown parameters. Also, the estimated parameters converge to the actual ones. After 10s of simulated time, their values were \( \hat{\theta}_1 = 5.09242 \) and \( \hat{\theta}_2 = 2.58819 \), which is very close to the true ones: \( \theta_1 = 5.03886 \) and \( \theta_2 = 2.42035 \).
7. EXPERIMENTAL RESULTS

Experimental results for the adaptive and the non-adaptive version of the proposed controller are shown in Figs. 5 and 6. The unknown parameters were initialized with their nominal values. As can be seen, initially the performance of both controllers is almost the same, but as time goes by, the performance of the adaptive controller is increased due to parameter tuning.

8. CONCLUSIONS

In this work a mobile robot controller using the full dynamic model of robot was presented. While for small robots the dynamics can be neglected, for large autonomous vehicles developing high velocities and carrying significant payloads these effects can hinder the controller performance. Since the dynamic model structure is well known, the main problem preventing its use for controller development is knowledge of the model parameters. Here this problem is dealt with by means of adaptive control. The proposed controller effectiveness was validated by simulation and experimental results. Further research directions include the development of an adaptive version of the polar coordinates controller proposed in Lages and Hemery (1998).

9. REFERENCES