Integrated Backstepping Mobile Robot Controller Design by Applying Sum of Squares Approach

Shun-Hung CHEN * and Jyh-Ching JUANG *

Abstract: The paper presents a sum of squares (SOS) based backstepping control design method for a three wheels omni-directional mobile robot. The characteristic of the strict-feedback system associated with mobile robots is considered to construct the backstepping controller. To account for the saturation problem in the mobile robot, the SOS conditions are developed based on the backstepping controller framework to achieve stability and enlarge the guaranteed region of convergence. Computer simulations for mobile robots demonstrate the effectiveness of the proposed SOS-based backstepping controller design method.

Key Words: backstepping control, sum of squares programming, input constraint, mobile robots.

1. Introduction

Wheel mobile robots have been demonstrated to be capable of performing high mobility in many robotics applications [1],[2]. In particular, the use of Mecanum wheels [3] has made it possible to achieve a full mobility in the plane without reorienting the robot. As a result, the omni-directional mobile robot is popular for indoor robot service applications due to its maneuverability and agility. In [4], Watanabe discussed several omni-directional wheels and investigated control strategies including resolved acceleration control, PID control, adaptive PI control for the kinematic model of the robot. As the omni-directional robots are nonlinear systems, it is desired to synthesize a control law that is capable of addressing the nonlinear characteristics while maintaining simplicity in designing and implementing the path tracking controller. In this paper, the backstepping technique is adopted for the design of such a controller. The backstepping technique [5]–[7] is a recursive application of the selection of virtual control variables and the design of Lyapunov functions in systematically constructing a stabilizing controller [8]. However, it is known that a major concern with the backstepping approach is that certain functions must be “linear in the unknown parameters” and some very tedious analyses are needed to determine the so-called “regression matrices”. To account for the computation of the regression matrices, several attempts have been made to integrate the backstepping control method with other control algorithms to achieve stability and performance [9],[10].

Recall that the backstepping method relies on the use of Lyapunov functions to ensure stability and performance, there is, unfortunately, a lack of systematic approach in synthesizing the Lyapunov functions.

In this paper, a systematic design procedure of seeking for virtual control and Lyapunov functions is presented. Based on the recursive structure of backstepping method, Lyapunov functions and virtual controls in the recursion are expressed in terms of polynomials. Thus, the associated parameter determination problem is formulated as a convex program or sum-of-squares (SOS) program. The paper proposes an SOS-based backstepping controller design approach which has the following significances. The SOS technique [11],[12] is utilized to provide a systematic manner in synthesizing the Lyapunov function and controller. In addition, the backstepping controller design framework is exploited so that stability can be assured and the numerical solution in SOS programming can be easily obtained. To apply the proposed approach to omni-directional mobile robots, the controller saturation issue is addressed. It is shown that the tradeoffs between stability, performance, and controller saturation can be considered at the same time in the SOS-based backstepping controller design.

The following notations are used. An $m$-dimensional vector space is denoted by $\mathbb{R}^m$. The notation $\Sigma_n$ stands for the set of continuous polynomials that can be expressed as a sum of squares.

The paper is organized as follows. In Section 2, the dynamic model of a three wheels omni-directional mobile robot is described. In Section 3, the backstepping control methodology is presented. The SOS programming technique is used to determine the Lyapunov function and address the controller saturation issue. Section 4 applies the SOS-based backstepping control design method to the omni-directional mobile robot. Finally, Section 5 provides some concluding remarks.

2. Problem Formulation

In the paper, the controller design of a three wheels omni-directional mobile robot is considered. The omni-directional mobile robot which is depicted in Fig. 1 consists of three Mecanum wheels that are 120 degrees apart. As each wheel comprises eight spindles like rollers that are perpendicular to the rotating axis, the wheel can be controlled to rotate around their rotation axis as well as to slide compliantly in the lateral direction. As a result, the omni-directional mobile robot has a good maneuverability and agility, especially for tuning control. In the following, for simplicity, both the slide effect upon slipping on the ground and the friction of the active wheels are
ignored. According to Newton’s law of motion, the dynamic model of the omni-directional mobile robot can be described as [1],[13]:

$$\begin{align*}
\dot{x} &= -\frac{3b}{2m} x + a \sin \theta u_1 + a \sin \left(\frac{\pi}{3} - \theta\right) u_2 + a \sin \left(\frac{\pi}{3} + \theta\right) u_3 \\
\dot{y} &= -\frac{3b}{2m} y + a \cos \theta u_1 + a \cos \left(\frac{\pi}{3} - \theta\right) u_2 + a \cos \left(\frac{\pi}{3} + \theta\right) u_3 \\
\dot{\theta} &= -\frac{3b}{J} L \dot{x} + a L u_1 + a L u_2 + a L u_3
\end{align*}$$

(1)

where $x$ and $y$ represent the position of the center of mass of the mobile platform with respect to the Cartesian space, $\theta$ is the orientation of the vehicle, $u_i$ is the control input to the wheel, $L$ is the distance between the wheel and the center of mass of the mobile platform, $J$ represents the moment of inertia, $m$ is the mass of the mobile robot, and $a$ and $b$ are parameters that are related to the dynamics and control characteristics of the robot. Let the pose vector be $x_1 = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$, velocity vector be $x_2 = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$, and control input vector be $u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$, the model (1) can be rewritten as

$$\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= Ax_2 + B(\theta)u
\end{align*}$$

(2)

where $A = \text{diag} \left[ \frac{-3b}{2m}, \frac{-3b}{2m}, \frac{-3bL^2}{J} \right]$ and

$$B(\theta) = \begin{bmatrix}
\frac{a}{m} \sin \theta & -\frac{a}{m} \sin \left(\frac{\pi}{3} - \theta\right) & \frac{a}{m} \sin \left(\frac{\pi}{3} + \theta\right) \\
\frac{a}{m} \cos \theta & -\frac{a}{m} \cos \left(\frac{\pi}{3} - \theta\right) & -\frac{a}{m} \cos \left(\frac{\pi}{3} + \theta\right) \\
aL & aL & aL
\end{bmatrix}$$

The input matrix $B(\theta)$ depends on the orientation $\theta$ nonlinearly. It is also noted that the input matrix is nonsingular for any $\theta$. Indeed, the inverse of $B(\theta)$ can be evaluated as

$$B^{-1}(\theta) = \begin{bmatrix}
-\frac{2m}{3a} \sin \theta & 2m \cos \theta & 1 \\
-\frac{2m}{3a} \sin \left(\frac{\pi}{3} - \theta\right) & -2m \cos \left(\frac{\pi}{3} - \theta\right) & 1 \\
\frac{2m}{3a} \sin \left(\frac{\pi}{3} + \theta\right) & -\frac{2m}{3a} \cos \left(\frac{\pi}{3} + \theta\right) & 1
\end{bmatrix} \begin{bmatrix}
\frac{3aL}{2} \\
\frac{3aL}{2} \\
\frac{3aL}{2}
\end{bmatrix}$$

(3)

The design objective is to synthesize a control input so that the vehicle trajectory follows a given reference trajectory $x_r$. In addition, it is required that the each control input $u_i$ is subject to saturation, that is, there exists a positive $\bar{u}$ such that $|u_i| \leq \bar{u}$ for $i = 1, 2, 3$.

3. SOS-Based Backstepping Design

In this section, an SOS-based backstepping control design approach is developed for stability and path control of the omni-directional mobile robot. The backstepping methodology is a recursive design algorithm that has been applied to solve many engineering problems. The approach formulates the system as a strict-feedback system and applies Lyapunov direct method to stabilize the feedback loop. The SOS-based backstepping approach attempts to determine a polynomial Lyapunov function so that stability and trajectory tracking can be achieved. The issue of controller saturation can also be accounted for in the SOS formulation.

3.1 Backstepping Control Design

The tracking error is defined as the difference between the pose vector $x_1$ and its reference $x_r$, i.e.,

$$e_1 = x_1 - x_r$$

(4)

Differentiating (4) results in

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_r = x_2 - \dot{x}_r$$

(5)

Regarding $x_2$ as a virtual control and let $x_2$ be

$$x_2 = \varphi(e_1) = -K_1 e_1 + \ddot{x}_r$$

(6)

for a first stage control gain matrix $K_1$. Substituting the virtual control (6) into (5) yields

$$\dot{e}_1 = -K_1 e_1 + \ddot{x}_r - \dot{x}_r = -K_1 e_1$$

(7)

The dynamic subsystem in (7) can be easily shown to be stable provided that the matrix $K_1$ is positive definite. In applying the backstepping technique, define the backstepping error vector $e_2$ as

$$e_2 = x_2 - \varphi(e_1) = x_2 + K_1 e_1 - \dot{x}_r$$

(8)

Therefore, (7) could be rewritten as

$$\dot{e}_1 = e_2 - K_1 e_1$$

(9)

Differentiating (8) leads to

$$\dot{e}_2 = Ax_2 - K_1^2 e_1 - \ddot{x}_r + B(x)u$$

(10)

The control law of the feedback system can be then designed as

$$u = \left( B(x) \right)^{-1} \left( -Ax_2 + \ddot{x}_r - K_2 e_2 \right)$$

(11)

where $K_2$ is the feedback gain to be designed. Substituting (11) into (10) results in

$$\dot{e}_2 = -K_1^2 e_1 + (K_1 - K_2) e_2$$

(12)

Let $e = \begin{bmatrix} e_1^T & e_2^T \end{bmatrix}^T$. Consider the following Lyapunov function candidate of the feedback system

$$V_{fb}(e) = e_1^T K_1^2 e_1 + e_2^T e_2$$

(13)

The derivative of (13) along the state trajectory is

$$\dot{V}_{fb}(e) = 2e_1^T K_1^2 e_1 + 2e_2^T e_2$$

$$= -2e_1^T K_1^2 e_1 + 2e_2^T (K_1 - K_2) e_2$$

(14)

Clearly, the Lyapunov function (13) can be used to guarantee the stability of the system when $K_2 > K_1 > 0$. Here, the notation $>$ is interpreted in the sense of matrix definiteness.
3.2 Input Constraint Consideration

For the omni-directional mobile robot, three electrical motors are equipped to drive the fuselage. However, the control input of electrical motor is subject to saturation. Suppose one motor is saturated, the couplings among different wheels as revealed in \( B(\theta) \) on the resulting dynamics may lead to erroneous trajectory and orientation. Therefore, although the backstepping control design law (11) ensures global stability when the matrices \( K_1 \) and \( K_2 \) are selected appropriately, the design must also be examined to avoid controller saturation. Note that from (8) and (12), the system dynamics in terms of the error can be expressed as

\[
\begin{align*}
\dot{e}_1 &= -K_1 e_1 + e_2 \\
\dot{e}_2 &= -(AK_1 + K_2^T) e_1 + (A + K_1) e_2 + Ax_r - \bar{x}_r + \nu 
\end{align*}
\]  

(15)

Here the new control vector

\[ v = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T \]

is related to the original control input vector

\[ u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T \]

via \( u = B^{-1}(\theta) v \). As each control signal is bounded in magnitude \( |u| \leq \bar{u} \), from (3), it is not difficult to show that the requirement \( |u| \leq \bar{u} \) can be recast as

\[ \left( \frac{2m}{3a} \right)^2 (v_1^2 + v_2^2) + \left( \frac{1}{3aL} \right)^2 v_3^2 \leq \bar{u}^2 \]

or

\[ v^T Q v \leq \bar{u}^2 \]

(16)

where \( Q = \text{diag} \left( \left( \frac{2m}{3a} \right)^2 , \left( \frac{2m}{3a} \right)^2 , \left( \frac{1}{3aL} \right)^2 \right) \). For the backstepping controller (11), the control input is given by

\[ v = -A \dot{x}_r + \ddot{x}_r + K_{by} e \]

(17)

where the state feedback control gain \( K_{by} \) is given as

\[ K_{by} = \begin{bmatrix} AK_1 & -A - K_2 \end{bmatrix} \]

(18)

Neglecting the reference commands, the system dynamics (15) can be expressed as

\[ \dot{e} = \begin{bmatrix} -K_1 & I \\ -(AK_1 + K_2^T) & (A + K_1) \end{bmatrix} e + \begin{bmatrix} 0 \\ I \end{bmatrix} \nu \]

(19)

For the system (19) and the control law \( v = K_{by} e \), the region of convergence can be assessed by using the Lyapunov function (13). Indeed, via the \( S \)-procedure [14], when the inputs are subject to saturations, the admissible region of convergence can be found to be

\[
\left\{ e \in \mathbb{R}^n \mid e^T \begin{bmatrix} K_1^T & 0 \\ 0 & I \end{bmatrix} e \leq \eta \right\}
\]

where \( \eta \) is the largest scalar such that

\[ \bar{u}^2 \begin{bmatrix} K_1^T & 0 \\ 0 & I \end{bmatrix} - \eta K_{by}^T Q K_{by} > 0 \]

(20)

It is desired to maximize \( \eta \) so that the region of convergence is enlarged.

3.3 SOS-Based Design

In this section, a tracking controller will be designed based on the framework established by the backstepping approach. For the system in (15), the following control law is to be designed

\[ v = -A \dot{x}_r + \ddot{x}_r + h(e) \]

(21)

where \( h(e) \) is the feedback control law. In terms of the arguments in SOS programming, the design problem can be formulated as the determination of a Lyapunov function candidate \( V(e) \) so that the following conditions are satisfied.

\[
\begin{align*}
V(e) &> 0 \forall e \in \mathbb{R}^n \setminus \{0\} \\
V(0) &> 0 \\
\{ e \in \mathbb{R}^n \mid p(e) \leq \beta \} &\subseteq \{ e \in \mathbb{R}^n \mid V(e) \leq 1 \} \\
\{ e \in \mathbb{R}^n \mid V(e) \leq 1 \} &\subseteq \{ e \in \mathbb{R}^n \mid h^T(e) Q h(e) \leq \bar{u}^2 \}
\end{align*}
\]

The first condition (22) implies that the Lyapunov function candidate \( V(e) \) is positive definite. The second condition (23) serves the purpose of bounding the level set of \( V(e) \) through the positive definite function \( p(e) \) and positive \( \beta \). Once such a condition is met, the region of convergence is known to be \( \{ e \in \mathbb{R}^n \mid p(e) \leq \beta \} \). Clearly, it is desired that the scalar \( \beta \) to be as large as possible so that the guaranteed region of convergence is enlarged. The third condition (24) is to ensure that the derivative of the Lyapunov function candidate is negative definite in the region of interest (except when \( e = 0 \)). The last condition (25) is relevant to the boundedness of the control input. It implies that in the region of convergence, the controller amplitude is bounded.

Applying P-satz [11] and after some simplifications and manipulations, it can be shown that a set of sufficient conditions can be derived as follows

\[ \max \beta \]

(26)

over \( V, h \), and SOS polynomials \( s_1, s_2, s_3, \) and \( s_4 \) such that

\[ V - l_1 \in \Sigma_n \]

(27)

\[ -((\beta - \rho)s_1 + (V - 1)) \in \Sigma_n \]

(28)

\[ -\left( (1 - V)s_2 + \frac{\partial V}{\partial e}(\bar{A} e + \bar{B} h(e))s_3 + l_2 \right) \in \Sigma_n \]

(29)

and

\[ \left( \bar{u}^2 - h^T(e) Q h(e) \right) - (1 - V) s_4 \in \Sigma_n \]

(30)

where \( l_1 \) and \( l_2 \) are positive definite polynomials. The derivation of (27)-(29) can be found in [11,15]. The condition (30), on the other hand, is unique to the proposed SOS-based design method. As the SOS-based design formulation relies on the backstepping controller design framework, the polynomial is \( p(e) \) can indeed be selected as \( V_{by}(e) \) in (13). In this case, if the optimal \( \beta \) is greater than the optimal \( \eta \) in (20), the guaranteed region of convergence is enlarged.

Some remarks on the solving of the unknown polynomials \( V, h, s_1, s_2, s_3, s_4, \) and the optimal \( \beta \) to the problem (26) are
then made. As the conditions involve the product terms, the solving of the problem requires iterations. Typically, the degrees of the polynomials are specified and the coefficients are then determined. The polynomials are determined alternatively until a convergence condition is satisfied. In the iteration, the Lyapunov function and control law of the backstepping design are used as the initial estimates of $V$ and $h$, respectively, that is, $V_0 = V_{h_0}(e)$ and $h_0 = K_{h_0}e$. The polynomials $V_i$ and $h_i$ at the $i$-th iteration are then determined alternatively. Each iteration loop may contain the following three steps.

**Step 1.** With $V_i$ and $h_i$ given, the step determines the SOS polynomials $s_1$, $s_2$, $s_3$, and $s_4$ such that the scalar $\beta$ is maximized such to (28)-(30). This step is a standard SOS programming.

**Step 2.** The step updates the control polynomial $h$. As the condition (30) depends on the term $h$ quadratically, a perturbation approach is adopted. More precisely, it is assumed that $h = h_1 + \tilde{h}_1$ for some $\tilde{h}_1$. The polynomials $s_1$, $s_2$, $s_3$, and $\tilde{h}_1$ are determined so that (28), (29), and the following equation are satisfied.

$$\left(\tilde{a}^2 - h_1^2(e)Qh_1(e) - 2h_1^2(e)\tilde{Q}_{h_1}(e) - (1 - V)s_4\right) \in \Sigma_n$$

The Lyapunov function candidate $V_i$ and the solved polynomial $s_3$ in the previous step are used in the SOS program. Afterwards, the control law is updated as $h_{i+1} = h_1 + \tilde{h}_1$.

**Step 3.** With the solved $h_{i+1}$, $s_2$, $s_3$, and $s_4$ in the previous step, this step determines $V$ so that (27)-(30) are satisfied and the scalar $\beta$ is optimized. Once the problem is feasible, the Lyapunov function candidate $V_{i+1}$ is updated. Further, the control law $h_{i+1}$ solved in the previous step is a legitimate controller. The operation can then proceed with another iteration. On the other hand, if the problem is not feasible due to the added nonlinear term $h_1^2(e)\tilde{Q}_{h_1}(e)$ to (30), the operation returns to Step 2 with an adjusted $\beta$.

The termination condition depends on the monitoring of the progress in the optimization of $\beta$. When the progress in an iteration loop is less than a pre-specified threshold, the computation is stopped.

In the problem (30), only the saturation constraint on $h(e)$ is considered. In practice, the saturation constraint should be imposed on $v$ which is a combination of the feedback control $h(e)$ and the reference trajectory command as in (21). Recall that

$$v^TQv = (-Ax + \bar{x} + h(e))^T Q (-Ax + \bar{x} + h(e)) \leq 2(-Ax + \bar{x})^T Q (-Ax + \bar{x}) + 2h_1^2(e)\tilde{Q}_{h_1}(e)$$

Thus, once the reference trajectory command is given, an upper bound on $(-Ax + \bar{x})^T Q (-Ax + \bar{x})$ is computed, say, $\varepsilon_1 \geq (-Ax + \bar{x})^T Q (-Ax + \bar{x})$ for all time, and the condition (30) is replaced by

$$\left(\frac{\tilde{a}^2 - 2\varepsilon_1}{2} - h_1^2(e)\tilde{Q}_{h_1}(e) - (1 - V)s_4\right) \in \Sigma_n$$

(31)

### 4. Simulation Results

In this section, the proposed SOS-based backstepping controller design approach is applied to design a path tracking controller for an omni-directional robot. The model of the three wheels omni-directional mobile robot is described in (1). Some parameters of the omni-directional mobile robot are given in Table 1. As the backstepping methodology is capable of ensuring system stability and eliminating the tracking error, it is employed to design of a path planning controller. In this case, the controller is given in (11). Suppose that the two matrices $K_1$ and $K_2$ are selected as $K_1 = \text{diag}(1, 1, 0.5)$ and $K_2 = \text{diag}(2, 2, 1)$, the feedback gain matrix becomes

$$K_{bh} = \begin{bmatrix} -0.024 & 0 & 0 & -1.976 & 0 & 0 \\ 0 & -0.024 & 0 & 0 & -1.976 & 0 \\ 0 & 0 & -0.024 & 0 & -0.952 & 0 \end{bmatrix}$$

The backstepping controller will ensure stability and asymptotic tracking. When the control input magnitude is bounded by $\bar{u} = 2$, the guaranteed region of convergence in (19) can be found by solving a linear matrix inequality (20). The scalar $\eta$, in this case, is 0.571.

It is desired to enlarge the guaranteed region of convergence. To this end, the SOS-based design approach is applied to determine a nonlinear feedback control law $h(e)$. In the design, the positive definite polynomial $p(e)$ is set as $p(e) = e^T \begin{bmatrix} K_1^2 & 0 \\ 0 & 1 \end{bmatrix} e$ so that the resulting controller can be directly compared with the backstepping controller. The solving of the SOS program requires some design iterations as remarked in the previous section. The two polynomials $l_1$ and $l_2$ are selected as 0.001$e^2$ and the Lyapunov function candidate $V(e)$, feedback control law $h(e)$, and auxiliary SOS polynomials $s_1$, $s_2$, $s_3$, and $s_4$ are determined alternatively so that (27)-(30) are satisfied while the scalar $\beta$ is being optimized. In this optimization process, the SOSTOOLS [16], which is a free, third party MATLAB toolbox for solving the problem of SOS programs, is used. In the SOS program, the Lyapunov function candidate and the feedback controller are assumed to be quadratic polynomials. After some iterations, the SOS-based approach leads to the following Lyapunov function $V(e)$ and feedback controller

$$h(e) = \begin{bmatrix} h_1(e) & h_2(e) & h_3(e) \end{bmatrix}^T :$$

$$V(e) = e_1^T \begin{bmatrix} 0.118 & 0.641 \times 10^{-5} & 0.228 \times 10^{-3} \\ 0.641 \times 10^{-5} & 0.118 & -0.828 \times 10^{-3} \\ 0.228 \times 10^{-3} & -0.828 \times 10^{-3} & 0.465 \times 10^{-1} \end{bmatrix} e_1 + e_2^T \begin{bmatrix} 0.238 & -0.544 \times 10^{-4} & -0.307 \times 10^{-4} \\ -0.544 \times 10^{-4} & 0.238 & -0.544 \times 10^{-4} \\ -0.307 \times 10^{-4} & -0.544 \times 10^{-4} & 0.433 \end{bmatrix} e_2 + e_3^T \begin{bmatrix} -0.308 & 0.137 \times 10^{-4} & 0.436 \times 10^{-3} \\ -0.201 \times 10^{-5} & -0.309 & -0.132 \times 10^{-2} \\ -0.745 \times 10^{-3} & 0.268 \times 10^{-2} & -0.201 \end{bmatrix} e_3$$

and

<table>
<thead>
<tr>
<th>Table 1 Parameters of the omni-directional mobile robot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The distance from wheel to the center of mass</td>
</tr>
<tr>
<td>Mass</td>
</tr>
<tr>
<td>Moment of inertia</td>
</tr>
<tr>
<td>Parameter $a$</td>
</tr>
<tr>
<td>Parameter $b$</td>
</tr>
</tbody>
</table>
Note that the controller is designed to rotate at a constant rate. The reference trajectory is commanded to follow a circle trajectory while its orientation is specified. This resulted in a linear design model, the controller being determined at the expense of design and implementation complexity. For instance, the guaranteed region of convergence can still be enlarged significantly larger than $\eta$ which bounds the guaranteed region of convergence is found to be $1.832$ which is significantly larger than $\eta$. It is noted that when higher order Lyapunov function candidate and feedback controller are considered, the guaranteed region of convergence can still be enlarged at the expense of design and implementation complexity. For the omni-directional robot, the backstepping design has resulted in a linear design model, the controller being determined through the SOS programming can be approximated by a linear controller. Nevertheless, the capability of the SOS-based design approach is demonstrated.

To assess the performance of the two controllers, the robot is commanded to follow a circle trajectory while its orientation is controlled to rotate at a constant rate. The reference trajectory command is thus $x(t) = \begin{bmatrix} \alpha \cos \omega t & \alpha \sin \omega t \end{bmatrix}^T$ where $\alpha$ is the radius of the circle, $\omega$ is the angular rate of the robot along the circular trajectory, and $\Omega$ is the rotation rate of the orientation of the robot. When $\alpha = 12$, $\omega = 0.2$, and $\Omega = 4$ and the initial conditions are $e_1(0) = \begin{bmatrix} -12 & 0 & 0 \end{bmatrix}^T$ and $e_2(0) = \begin{bmatrix} 12 & 0 & 0 \end{bmatrix}^T$, the tracking errors and control signals of the two controllers are depicted in Figs. 2 and 3, respectively. Both controllers are capable of tracking the reference command asymptotically. Due to the initial error and the demanding reference command, both controllers experience saturation initially. The SOS-based controller stabilizes the robot at a faster rate and the errors are less pronounced. The effectiveness of the proposed SOS-based backstepping controller design approach is verified.

5. Conclusions

In the paper, an SOS-based backstepping controller design method for omni-directional mobile robots has been presented. The backstepping approach has been applied to achieve stability and command tracking by recursively synthesizing virtual control variables. Under the framework, the SOS technique has been applied to account for the controller saturation issue. A systematic approach for the synthesis of a polynomial control law and a Lyapunov function has been provided so that not only stability and trajectory tracking can be achieved but also controller saturation can be addressed. The proposed approach has been verified by computer simulations. In the future, the proposed approach will be applied to other nonlinear systems to assess its capabilities.

Acknowledgments

This work was supported by the National Science Council, Taiwan under Grant NSC 98-2218-E-006-256.

References


Shun-Hung Chen

He received his B.S. and M.S. degrees from I-Shou University, Kaohsiung, Taiwan, in 2001 and 2004, respectively. He is currently a Ph.D. student at the Department of Electrical Engineering, National Cheng Kung University. His research interests include nonlinear controller design, fuzzy systems, and navigation.

Jyh-Ching Juang

He received the B.S. and M.S. degrees from National Chiao-Tung University, Hsinchu, Taiwan, in 1980 and 1982, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1987. He was with Lockheed Aeronautical System Company, Burbank before he joined the faculty of the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan in 1993. His research interests include sensor networks, GNSS signal processing, and software-based receivers.