Appl. Math. and Comp. Sci., *1998, Vol.8, No.1, 73–88*

A SYMBOLIC COMPUTATION TOOLBOX FOR THE DESIGN OF DYNAMICAL ADAPTIVE NONLINEAR CONTROLLERS

E. Miguel RIOS-BOLÍVAR*, Alan S.I. ZINOBER**

We describe a new symbolic computation toolbox BACKDSMC developed using the MATLAB Symbolic Toolbox and intended for the design of dynamical adaptive nonlinear controllers for regulation and tracking tasks of a class of observable minimum phase uncertain nonlinear systems. This toolbox also allows us to design non-adaptive controllers for systems without uncertainty, and adaptive sliding mode controllers (SMC) to provide robustness in the presence of disturbances. The design procedure employs the basic ideas of the adaptive backstepping algorithm with tuning functions via input-output linearization, and is applicable to both triangular and nontriangular systems.

1. Introduction

The computer technology advances of recent years have allowed the development of a number of computer software systems intended for numerical and symbolic computation, such as MACSYMA, MAPLE and MATHEMATICA. The availability of these packages has allowed the development of useful toolboxes for the systematic analysis and design of feedback control systems. For instance, some of the toolboxes developed so far include analysis and control design for affine and non-affine systems (de Jager, 1996; Glumineau and Graciani, 1996), modelling and nonlinear control design (Blankenship *et al.*, 1995), and analysis and design based on flatness (Rothfuss and Zeitz, 1996). These toolboxes simplify the use of systematic and recursive control design methods so that the design of stabilizing controllers may be carried out more efficiently.

The various backstepping control design algorithms (Jiang and Praly, 1991; Kanellakopoulos *et al.*, 1991; Krstić *et al.*, 1992) recently compiled in Krstić *et al.* (1995), provide a systematic framework for the design of tracking and regulation strategies suitable for large classes of nonlinear systems. The adaptive backstepping algorithm has enlarged the class of nonlinear systems controlled via a Lyapunov-based control law to uncertain systems transformable into the *parametric strict feedback* (PSF) form or the *parametric pure feedback* (PPF) form. In general, local stability is

* Departamento de Sistemas de Control, Facultad de Ingeniería, Universidad de Los Andes, Mérida 5101, Venezuela, e-mail: riosm@ing.ula.ve.
** Department of Applied Mathematics, University of Sheffield, Sheffield S10 2TN, U.K., e-mail: A.Zinober@sheffield.ac.uk.

achieved for systems in the PPF form, whilst global stability is guaranteed for systems in the PSF form (Kanellakopoulos *et al.*, 1991). These two forms can be seen as special structural *triangular* forms of nonlinear systems which are adaptively input-output linearizable with the linearizing output $y = x_1$. A more general algorithm has been developed by Rios-Bolívar *et al.* (1995), which makes it possible to design dynamical adaptive controllers following an input-output linearization procedure based upon the backstepping approach with tuning functions (Krstić *et al.*, 1992), and is applicable to both triangular and nontriangular uncertain nonlinear systems.

This paper is organized as follows: Section 2 outlines the generalized backstepping algorithm. Some features of the symbolic toolbox BACKDSMC are given in Section 3. Section 4 presents examples of application of the dynamical adaptive backstepping algorithm using the symbolic toolbox, and some conclusions are presented in Section 5.

2. Dynamical Adaptive Control Design

The Dynamical Adaptive Backstepping (DAB) algorithm proposed in Rios-Bolívar *et al.* (1995) is based upon a combination of dynamical input-output linearization and the adaptive backstepping algorithm with tuning functions (Krstić *et al.*, 1992). Since it has been developed in a general context, without the use of canonical forms, its applicability to both triangular (PSF and PPF forms) and nontriangular systems is guaranteed, but it requires that the controlled plant be *observable* and *minimum phase*. The observability condition is required to guarantee the existence of a local nonlinear mapping which transforms the plant into a convenient form of the error system, as shown below. The role of the minimum phase property is to allow the applicability of the systematic algorithm presented here and to guarantee stability of the closed-loop system. This general algorithm includes as a particular case the adaptive backstepping algorithm with tuning functions (Krstić *et al.*, 1992) developed for systems in PSF and PPF forms.

Consider a single-input single-output nonlinear system with linearly parameterized uncertainty

$$\begin{aligned} \dot{x} &= f_0(x) + \Phi(x)\theta + \Big(g_0(x) + \Psi(x)\theta\Big)u \\ y &= h(x) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state; $u, y \in \mathbb{R}$ denote the input and output, respectively; and $\theta = [\theta_1, \dots, \theta_p]^T$ is a vector of unknown parameters. Here f_0, g_0 and the columns of the matrices $\Phi, \Psi \in \mathbb{R}^{n \times p}$ are smooth vector fields in a neighbourhood R_0 of the origin $x = 0$ with $f_0(0) = 0$, $g_0(0) \neq 0$; and h is a smooth scalar function also defined in R_0.

The steps leading to the the design of the dynamical adaptive compensator follow an input-output linearization procedure in which, at each step, a control dependent nonlinear mapping and a tuning function are constructed. The parameter update law and the dynamical adaptive control law which stabilize the controlled plant are designed at the final step. In order to characterize the class of nonlinear systems for which this procedure is applicable, we set up a nonlinear mapping by considering the output $y(t)$ and its first $n-1$ time derivatives as follows:

$$\dot{y} = \frac{\partial h}{\partial x}\dot{x} = \frac{\partial h}{\partial x}\Big[f_0(x) + \Phi(x)\theta + \Big(g_0(x) + \Psi(x)\theta\Big)u\Big] \tag{2}$$

Due to the presence of the unknown parameter vector θ, we rewrite (2) as

$$\dot{y} = \mathcal{L}_h^1(x,\hat{\theta},u) = \frac{\partial h}{\partial x}\Big[f_0(x) + \Phi(x)\hat{\theta} + \Big(g_0(x) + \Psi(x)\hat{\theta}\Big)u\Big] + \omega_1(\theta - \hat{\theta}) \tag{3}$$

where $\hat{\theta}$ is an estimate of θ, and the vector ω_1 is defined as

$$\omega_1 = \frac{\partial h}{\partial x}\Big(\Phi(x) + u\Psi(x)\Big) \tag{4}$$

In other words, (3) may be rewritten as

$$\dot{y} = \mathcal{L}_h^1(x,\hat{\theta},u) = \widehat{\mathcal{L}}_h^1(x,\hat{\theta},u) + \omega_1(\theta - \hat{\theta}) \tag{5}$$

with

$$\widehat{\mathcal{L}}_h^1(x,\hat{\theta},u) := \frac{\partial h}{\partial x}\Big[f_0(x) + \Phi(x)\hat{\theta} + \Big(g_0(x) + \Psi(x)\hat{\theta}\Big)u\Big] \tag{6}$$

The second time derivative of the output is

$$\begin{aligned} \ddot{y} &= \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial x}\dot{x} + \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial\hat{\theta}}\dot{\hat{\theta}} + \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial u}\dot{u} \\ &= \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial x}\Big[f_0(x) + \Phi(x)\theta + \Big(g_0(x) + \Psi(x)\theta\Big)u\Big] + \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial\hat{\theta}}\dot{\hat{\theta}} + \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial u}\dot{u} \end{aligned} \tag{7}$$

which can be rewritten as

$$\ddot{y} = \mathcal{L}_h^2(x,\hat{\theta},u,\dot{u}) = \widehat{\mathcal{L}}_h^2(x,\hat{\theta},u,\dot{u}) + \omega_2(\theta - \hat{\theta}) \tag{8}$$

with

$$\widehat{\mathcal{L}}_h^2 := \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial x}\Big[f_0(x) + \Phi(x)\hat{\theta} + \Big(g_0(x) + \Psi(x)\hat{\theta}\Big)u\Big] + \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial\hat{\theta}}\dot{\hat{\theta}} + \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial u}\dot{u} \tag{9}$$

and

$$\omega_2 = \frac{\partial\left(\mathcal{L}_h^1\right)}{\partial x}\Big(\Phi(x) + u\Psi(x)\Big) \tag{10}$$

By proceeding successively in this manner, we obtain the j-th time derivative of the output

$$y^{(j)} = \mathcal{L}_h^j\left(x,\hat{\theta},u,\dot{u},\dots,u^{(j-1)}\right) = \widehat{\mathcal{L}}_h^j\left(x,\hat{\theta},u,\dot{u},\dots,u^{(j-1)}\right) + \omega_j(\theta - \hat{\theta}) \tag{11}$$

with

$$\widehat{\mathcal{L}_h^j} := \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial x}\left[f_0(x) + \Phi(x)\hat{\theta} + \left(g_0(x) + \Psi(x)\hat{\theta}\right)u\right] + \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial\hat{\theta}}\dot{\hat{\theta}} + \sum_{k=0}^{j-2}\frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial u^{(k)}}u^{(k+1)} \tag{12}$$

and

$$\omega_j = \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial x}\left(\Phi(x) + u\Psi(x)\right) \tag{13}$$

The expression (11) is valid if the relative degree is one. The general expression for systems with well-defined relative degree, i.e. $1 \leq \rho \leq n$, has the form

$$y^{(j)} = \mathcal{L}_h^j\left(x, \hat{\theta}, u, \dot{u}, \ldots, u^{(j-\rho)}\right) = \widehat{\mathcal{L}_h^j}\left(x, \hat{\theta}, u, \dot{u}, \ldots, u^{(j-\rho)}\right) + \omega_j(\theta - \hat{\theta}) \tag{14}$$

with

$$\widehat{\mathcal{L}_h^j} := \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial x}\left[f_0(x) + \Phi(x)\hat{\theta} + \left(g_0(x) + \Psi(x)\hat{\theta}\right)u\right] + \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial\hat{\theta}}\dot{\hat{\theta}} + \sum_{k=0}^{j-\rho-1}\frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial u^{(k)}}u^{(k+1)} \tag{15}$$

In other words, the time derivatives of the output are obtained by the application of the following recursively defined operator:

$$\begin{aligned}\mathcal{L}_h^0 &= h(x)\\ \mathcal{L}_h^j &:= \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial x}\left[f_0(x) + \Phi(x)\theta + \left(g_0(x) + \Psi(x)\theta\right)u\right]\\ &+ \frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial\hat{\theta}}\dot{\hat{\theta}} + \sum_{k=0}^{j-\rho-1}\frac{\partial\left(\mathcal{L}_h^{j-1}\right)}{\partial u^{(k)}}u^{(k+1)}\end{aligned} \tag{16}$$

which also characterizes the control dependent nonlinear mapping

$$z = \Xi\left(x, \hat{\theta}, u, \ldots, u^{(n-\rho-1)}\right) = \begin{bmatrix} y \\ y^{(1)} \\ \vdots \\ y^{(n-1)} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_h^0 \\ \mathcal{L}_h^1 \\ \vdots \\ \mathcal{L}_h^{n-1} \end{bmatrix} \tag{17}$$

Assumption 1. The system (1) is locally *observable*, i.e. the mapping (17) satisfies the rank condition

$$\text{rank}\frac{\partial \Xi(\cdot)}{\partial x} = n \tag{18}$$

in a subspace $R_1 \subset R_0 \subset \mathbb{R}^n$.

Assumption 2. The system (1) is *minimum phase* in $R_1 \subset R_0 \subset \mathbb{R}^n$.

For observable minimum phase nonlinear systems of the form (1), the general problem of adaptively tracking a bounded desired reference signal $y_r(t)$ with smooth and bounded derivatives can be solved through the DAB algorithm summarized as follows:

DAB Algorithm

Coordinate transformation

$$\begin{aligned} z_1 &:= y - y_r(t) = h^{(0)}(x) - y_r(t) \\ z_k &:= \hat{h}^{(k-1)}(\cdot) - y_r^{(k-1)}(t) + \alpha_{k-1}(\cdot), \quad 2 \le k \le n \end{aligned} \tag{19}$$

with

$$\begin{aligned} \hat{h}^{(k)} &= \frac{\partial \hat{h}^{(k-1)}}{\partial \hat{\theta}}\tau_k + \frac{\partial \hat{h}^{(k-1)}}{\partial x}\left[f_0 + \Phi\hat{\theta} + (g_0 + \Psi\hat{\theta})v_1\right] \\ &+ \sum_{i=1}^{k-\rho-1} \frac{\partial \hat{h}^{(k-1)}}{\partial v_i} v_{i+1} + \frac{\partial \hat{h}^{(k-1)}}{\partial t} \end{aligned} \tag{20}$$

$$\omega_k = \left(\frac{\partial \hat{h}^{(k-1)}}{\partial x} + \frac{\partial \alpha_{k-1}}{\partial x}\right)\left(\Phi(x) + u\Psi(x)\right) \tag{21}$$

$$\begin{aligned} \alpha_k &= z_{k-1} + \left(\sum_{i=2}^{k-1} z_i \frac{\partial \hat{h}^{(i-1)}}{\partial \hat{\theta}} + \sum_{i=3}^{k-1} z_i \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}}\right)\Gamma\omega_k^T \\ &+ \sum_{i=1}^{k-\rho-1} \frac{\partial \alpha_{k-1}}{\partial v_i} v_{i+1} + \frac{\partial \alpha_{k-1}}{\partial \hat{\theta}}\tau_k + \frac{\partial \alpha_{k-1}}{\partial t} \\ &+ \frac{\partial \alpha_{k-1}}{\partial x}\left[f_0 + \Phi\hat{\theta} + (g_0 + \Psi\hat{\theta})v_1\right] + c_k z_k \end{aligned} \tag{22}$$

$$\tau_k = \Gamma \sum_{i=1}^{k} \omega_k^T z_k \tag{23}$$

Parameter update law

$$\dot{\hat{\theta}} = \tau_n = \Gamma W^T z = \Gamma\left[\omega_1^T \; \omega_2^T \ldots \; \omega_n^T\right] z \tag{24}$$

DAB Algorithm (cont.)
Dynamical adaptive compensator

$$
\begin{aligned}
\dot{v}_1 &= v_2 \\
\dot{v}_2 &= v_3 \\
&\vdots \\
\dot{v}_{n-\rho} &= \frac{1}{\left(\frac{\partial \hat{h}^{(n-1)}}{\partial v_{n-\rho}} + \frac{\partial \alpha_{n-1}}{\partial v_{n-\rho}}\right)} \Bigg[- z_{n-1} + y_r^{(n)} - \left(\frac{\partial \hat{h}^{(n-1)}}{\partial x} + \frac{\partial \alpha_{n-1}}{\partial x}\right) \\
&\quad \times \left[f_0 + \Phi\hat{\theta} + (g_0 + \Psi\hat{\theta})v_1 \right] - \frac{\partial \hat{h}^{(n-1)}}{\partial t} - \frac{\partial \alpha_{n-1}}{\partial t} \\
&\quad - \left(\frac{\partial \hat{h}^{(n-1)}}{\partial \hat{\theta}} + \frac{\partial \alpha_{n-1}}{\partial \hat{\theta}}\right)\tau_n - \sum_{i=2}^{n-1} \left(\frac{\partial \hat{h}^{(n-1)}}{\partial \hat{\theta}} + \frac{\partial \alpha_{n-1}}{\partial \hat{\theta}}\right) z_i \Gamma \omega_n^T \\
&\quad - \sum_{i=1}^{n-\rho-1} \left(\frac{\partial \hat{h}^{(n-1)}}{\partial v_i} + \frac{\partial \alpha_{n-1}}{\partial v_i}\right) v_{i+1} - c_n z_n \Bigg] \qquad (25)
\end{aligned}
$$

with

$$v_1 = u$$

where the c_i's are constant design parameters and $\Gamma = \Gamma^T > 0$ is the adaptation gain matrix. The control u is obtained implicitly as the solution of the nonlinear time-varying differential equation (25). The overall closed-loop *error system* has the form

$$\dot{z} = A_z z + W(\theta - \hat{\theta}) \qquad (26)$$

$$\dot{\hat{\theta}} = \Gamma W^T z \qquad (27)$$

where the matrix A_z has the following skew-symmetric form:

$$
A_z = \begin{bmatrix}
-c_1 & 1 & 0 & \cdots & 0 \\
-1 & -c_2 & 1 + \varrho_{2,3} & \cdots & \varrho_{2,n} \\
0 & -1 - \varrho_{2,3} & -c_3 & \cdots & \varrho_{3,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & -\varrho_{2,n-1} & -\varrho_{3,n-1} & \cdots & 1 + \varrho_{n-1,n} \\
0 & -\varrho_{2,n} & -\varrho_{3,n} & \cdots & -c_n
\end{bmatrix}.
$$

with

$$\varrho_{i,j} = \left(\frac{\partial \hat{h}^{(i-1)}}{\partial \hat{\theta}} + \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}}\right) \Gamma \omega_j^T \qquad (28)$$

The skew-symmetric form of A_z is important for the stability of the system (26)–(27), since the relation

$$A_z + A_z^T = -2 \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_n \end{bmatrix} \tag{29}$$

yields

$$\dot{V} = -\sum_{i=1}^{n} c_i z_i^2 \tag{30}$$

with the quadratic Lyapunov function

$$V = \frac{1}{2} z^T z \tag{31}$$

It has been proved in (Rios-Bolívar, 1997; Rios-Bolívar *et al.*, 1995) that the stability of the overall system is guaranteed and also asymptotic tracking is achieved. These facts are summarized in the following theorem (Rios-Bolívar, 1997).

Theorem 1. *The closed-loop adaptive system consisting of the plant (1), the dynamical controller defined by (25) and the update law (24), has a locally stable equilibrium at* $(z, \hat{\theta} - \theta) = (0, 0)$ *and* $\lim_{t\to\infty} z(t) = 0$, *which means that asymptotic tracking is achieved, i.e.*

$$\lim_{t\to\infty} \Big[y(t) - y_r(t) \Big] = 0 \tag{32}$$

Moreover, if $\lim_{t\to\infty} y_r^{(i)}(t) = 0$, $i = 1, \ldots, n$ *and* $[\Phi(0) + u\Psi(0)] = 0$, *then* $\lim_{t\to\infty} x(t) = 0$.

Theorem 1 guarantees local asymptotic tracking in general. Nevertheless, global asymptotic tracking can be achieved if the relative degree is defined globally, and also Assumptions 1 and 2 are satisfied globally.

2.1. Dynamical Adaptive Sliding Mode Control

A particularly important aspect in regulation and tracking tasks for uncertain systems is robustness in the presence of disturbances and unmodelled dynamics. In Rios-Bolívar *et al.* (1996) a solution to this problem has been proposed, which is based on the combination of the above adaptive input-output linearization algorithm and sliding mode control (SMC). It permits to design dynamical adaptive sliding mode tracking controllers. The resulting control law achieves robust asymptotic stability with considerably reduced *chattering.*

To provide robustness, the DAB algorithm can be modified for the design of dynamical adaptive output tracking controllers (see (Rios-Bolívar *et al.*, 1996) for details). The modification is carried out at the final step of the algorithm by incorporating the following *sliding surface* defined in terms of the error coordinates:

$$\sigma = k_1 z_1 + k_2 z_2 + \cdots + k_{n-1} z_{n-1} + z_n = 0 \tag{33}$$

where the scalar coefficients $k_i > 0$, $i = 1, \ldots, n-1$, are chosen such that the polynomial

$$p(s) = k_1 + k_2 s + \cdots + k_{n-1} s^{n-2} + s^{n-1} \tag{34}$$

in the complex variable s is Hurwitz. Additionally, the Lyapunov function is modified as follows:

$$V = \frac{1}{2}\sum_{i=1}^{n-1} z_i^2 + \frac{1}{2}\sigma^2 + \frac{1}{2}(\theta - \hat{\theta})^T \Gamma^{-1} (\theta - \hat{\theta}) \tag{35}$$

Differentiating (35), we can obtain the update law

$$\dot{\hat{\theta}} = \tau_n = \tau_{n-1} + \Gamma\sigma\Big(\omega_n + \sum_{i=1}^{n-1} k_i \omega_i\Big) = \Gamma\left(\sum_{i=1}^{n-1} z_i \omega_i + \sigma\Big(\omega_n + \sum_{i=1}^{n-1} k_i \omega_i\Big)\right) \tag{36}$$

and the dynamical adaptive sliding mode output tracking controller

$$\begin{aligned} \hat{h}^{(n)}(\cdot) &- y_r^{(n)}(t) + \alpha_n(\cdot) + \left(\sum_{i=2}^{n-1} z_i \frac{\partial \hat{h}^{(i-1)}}{\partial \hat{\theta}} + \sum_{i=3}^{n-1} z_i \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}}\right) \\ &\times \Gamma\left(\omega_n + \sum_{i=1}^{n-1} k_i \omega_i\right) - \sum_{i=1}^{n-1} k_i \left(\sum_{j=2}^{i-1} z_j \frac{\partial \hat{h}^{(j-1)}}{\partial \hat{\theta}} + \sum_{j=3}^{i-1} z_j \frac{\partial \alpha_{j-1}}{\partial \hat{\theta}}\right) \Gamma \omega_i \\ &+ \sum_{i=1}^{n-1} k_i \left(\frac{\partial \hat{h}^{(i-1)}}{\partial \hat{\theta}} + \frac{\partial \alpha_{i-1}}{\partial \hat{\theta}}\right)(\tau_n - \tau_i) + \sum_{i=1}^{n-1} k_i(-z_{i-1} - c_i z_i + z_{i+1}) \\ &= -\kappa\big(\sigma + \beta \operatorname{sign}(\sigma)\big) \end{aligned} \tag{37}$$

with $\kappa > 0$, $\beta > 0$ and α_n defined by

$$\begin{aligned} \alpha_n(\cdot) &= \frac{\partial \alpha_{n-1}}{\partial \hat{\theta}} \tau_n + \frac{\partial \alpha_{n-1}}{\partial x}\left[f_0 + \Phi\hat{\theta} + (g_0 + \Psi\hat{\theta})u\right] \\ &+ \sum_{i=1}^{n-\rho-1} \frac{\partial \alpha_{n-1}}{\partial u^{(i-1)}} u^{(i)} + \frac{\partial \alpha_{n-1}}{\partial t} \end{aligned} \tag{38}$$

This dynamical adaptive sliding mode control yields

$$\dot{V} = -\sum_{i=1}^{n-1} c_i z_i^2 + z_{n-1} z_n - \kappa\sigma^2 - \kappa W |\sigma| \tag{39}$$

which guarantees asymptotic stability for suitably chosen design parameters (Rios-Bolívar *et al.*, 1996).

3. Symbolic Computation Toolbox

Since the DAB algorithm has been developed in a general context, its implementation via the Symbolic Algebra MATLAB toolbox allows us to deal with various classes of systems. The most general class corresponds to observable minimum phase systems which may be in triangular or nontriangular form. Triangular systems in PSF or PPF forms are particular subclasses of linearizable observable systems. On the other hand, the DAB algorithm may be used for the design of non-adaptive controllers for nonlinear systems without uncertainty by specifying null matrices Φ and Ψ. Also, the combined DAB-SMC algorithm of Section 2.1 may be implemented via a symbolic computation toolbox.

We have developed the symbolic toolbox BACKDSMC which implements both the DAB and DAB-SMC algorithms, for the synthesis of tracking and regulating adaptive (and non-adaptive) controllers, while requiring a minimum effort by the user. It has the following features (Rios-Bolívar, 1997):

- it automatises the backstepping control design procedure,
- it does not use transformations into canonical forms,
- it allows for the design of a number of adaptive and deterministic controllers,
- it does not require the user to have expert knowledge of the backstepping design technique,
- it generates automatically MATLAB code programs for computer simulation of the closed-loop systems.

The types of controllers designed by BACKDSMC for regulation and tracking problems include:

- static and dynamical non-adaptive linearizing controllers for deterministic systems,
- static and dynamical adaptive backstepping controllers for uncertain systems,
- robust static and dynamical combined backstepping Sliding Mode Control (SMC) for uncertain systems (Rios-Bolívar *et al.*, 1996).

The outputs generated by BACKDSMC consist of the *feedback control law*, the *coordinate transformation* placing the system into the error coordinates, the *parameter update law* for uncertain systems, the *sliding surface* for the combined backstepping-SMC design, and the MATLAB *code programs* for simulation purposes.

The user has to provide the nonlinear functions of the mathematical model of the system written in the general form (1), and the symbolic desired output. Depending on the nature of y_r, the problem to be solved is either a regulation or a tracking one. Thus, when y_r is constant, the designed controller is for regulation, otherwise y_r is a time-dependent function and the controller is designed for tracking tasks.

4. Examples

The following three different examples illustrate the use of the symbolic toolbox BACKDSMC corresponding to various classes of systems. The results are presented in the form of m-files (MATLAB code programs) which can be directly used for numerical integration.

4.1. I/O Linearizable System

Consider the third order system without uncertainties

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} x_2 + x_1 x_3 \\ x_3 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u \tag{40}$$

Marino and Tomei (1995) have shown that (40) is locally input-output linearizable. The linearizing output is

$$y = x_1 \exp(-x_2) \tag{41}$$

The symbolic expressions which characterize the mathematical model (40) are defined by the following series of MATLAB commands:

```
f=sym('[x2+x1*x3;x3;0]')   % f(x)
g=sym('[0;0;1]')           % g(x)
phi=sym('[0;0;0]')         % Phi(x)
psi=sym('[0;0;0]')         % Psi(x)
h='x1*exp(-x2)'            % h(x)
yd='0'                     % desired output yd
```

The linearizing static controller is obtained by the command line

```
[c,tau,z]=backdsmc(f,g,h,yd,phi,psi,'iolineam','iolinear')
```

which also generates automatically the m-files 'iolineam.m' and 'iolinear.m'. The m-file 'iolineam.m' contains the system equations and the control law as shown below:

```
function xdot=iolineam(t,x);
global c;
%
%    Control law
%
u1=(-c(2)*c(1)/exp(x(2))*x(2)-2/exp(x(2))*x(2)+x(3)*c(1)/exp(x(2))*x(2)-...
x(3)*c(1)/exp(x(2))-x(3)^2/exp(x(2))*x(2)+2*x(3)^2/exp(x(2))+x(3)*c(2)/...
exp(x(2))*x(2)-x(3)*c(2)/exp(x(2))-c(3)*c(1)/exp(x(2))*x(2)+c(3)*x(3)/...
exp(x(2))*x(2)-c(3)*x(3)/exp(x(2))-c(3)*c(2)/exp(x(2))*x(2)-c(3)*c(2)*...
```

```
c(1)*x(1)/exp(x(2))-c(3)*x(1)/exp(x(2))-c(1)*x(1)/exp(x(2)))/...
(-exp(-x(2))*x(2)+exp(-x(2)));
%
%    System equations
%
xdot(1)=x(2)+x(1)*x(3);
xdot(2)=x(3);
xdot(3)=u1;
```

As shown in Marino and Tomei (1995), the value $x_2 = 1$ is a singular value for the above linearizing feedback control.

The m-file 'iolinear.m' is generated to run the numerical simulation of the closed-loop system in 'iolineam.m' and allows us to specify the initial conditions, the parameter design values and the initial and final times for simulations, as follows:

```
%
% This program runs iolineam.m
%
global c;
%
%    Parameter values
%
c=[;;];
t0=;      % Initial time
tf=;      % Final time
%
%     Initial conditions
%
x0=[,,];
[t,x]=ode23('iolineam',t0,tf,x0);
```

4.2. Uncertain PSF System

Consider the third order nonlinear system

$$\begin{aligned} \dot{x}_1 &= x_2 + \theta x_1^2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= u \end{aligned} \tag{42}$$

where θ is an unknown scalar parameter. This system is already in the PSF form and therefore global adaptive regulation is achieved. The results obtained by BACKDSMC include the coordinate transformation z, the update law $\dot{\hat{\theta}}$ for the unknown parameter, and the control law u.

The symbolic expressions which characterize the mathematical model (42) are defined by the following series of MATLAB commands:

```
f=sym('[x2;x3;0]')        % f(x)
g=sym('[0;0;1]')          % g(x)
phi=sym('[x1^2;0;0]')     % Phi(x)
psi=sym('[0;0;0]')        % Psi(x)
h='x1'                    % h(x)
yd='0'                    % desired output yd
```

The regulating adaptive controller is obtained by the command line

```
[c,tau,z]=backdsmc(f,g,h,yd,phi,psi,'psfmod')
```

which generates the m-file 'psfmod.m' for simulation purposes:

```
function xdot=psfmod(t,x);
global c ad theta;
%
%  Auxiliary variables
%
th1=x(4);
%
%  Update law
%
tau1=x(1)^3*ad(1)+(x(2)+th1*x(1)^2+c(1)*x(1))*ad(1)*x(1)^2*(2*th1*x(1)+...
c(1))+(2*th1*x(1)*x(2)+2*th1^2*x(1)^3+c(1)*x(2)+c(1)*th1*x(1)^2+x(3)+...
x(1)^2*(x(1)^3*ad(1)+(x(2)+th1*x(1)^2+c(1)*x(1))*ad(1)*x(1)^2*(2*th1*...
x(1)+c(1)))+c(2)*(x(2)+th1*x(1)^2+c(1)*x(1))+x(1))*ad(1)*x(1)^2*...
(2*th1*x(2)+6*th1^2*x(1)^2+2*c(1)*th1*x(1)+5*x(1)^4*ad(1)+10*x(1)^4*...
ad(1)*th1*x(2)+4*x(1)^3*ad(1)*c(1)*x(2)+14*x(1)^6*ad(1)*th1^2+18*...
x(1)^5*ad(1)*c(1)*th1+5*x(1)^4*ad(1)*c(1)^2+2*c(2)*th1*x(1)+c(2)*c(1)+1);
%
%   Control law
%
u1=-8*th1^2*x(2)*x(1)^2-2*c(1)*th1*x(1)*x(2)-2*th1*x(1)^2-2*x(2)-c(1)*...
x(1)-10*x(1)^4*ad(1)*th1*x(2)^2-(2*x(1)*x(2)+4*th1*x(1)^3+c(1)*x(1)^2+...
x(1)^2*(x(1)^4*ad(1)*(2*th1*x(1)+c(1))+2*(x(2)+th1*x(1)^2+c(1)*x(1))*...
ad(1)*x(1)^3)+c(2)*x(1)^2)*(x(1)^3*ad(1)+(x(2)+th1*x(1)^2+c(1)*x(1))*...
ad(1)*x(1)^2*(2*th1*x(1)+c(1))+(2*th1*x(1)*x(2)+2*th1^2*x(1)^3+c(1)*...
x(2)+c(1)*th1*x(1)^2+x(3)+x(1)^2*(x(1)^3*ad(1)+(x(2)+th1*x(1)^2+c(1)*...
x(1))*ad(1)*x(1)^2*(2*th1*x(1)+c(1)))+c(2)*(x(2)+th1*x(1)^2+c(1)*x(1))+...
x(1))*ad(1)*x(1)^2*(2*th1*x(2)+6*th1^2*x(1)^2+2*c(1)*th1*x(1)+5*x(1)^4...
*ad(1)+10*x(1)^4*ad(1)*th1*x(2)+4*x(1)^3*ad(1)*c(1)*x(2)+14*x(1)^6*...
```

```
ad(1)*th1^2+18*x(1)^5*ad(1)*c(1)*th1+5*x(1)^4*ad(1)*c(1)^2+2*c(2)*th1*...
x(1)+c(2)*c(1)+1))-2*x(3)*th1*x(1)-5*x(1)^4*ad(1)*x(2)-c(2)*c(1)*x(2)-...
2*c(1)*th1^2*x(1)^3-5*x(1)^6*ad(1)*th1-14*x(1)^8*ad(1)*th1^3-2*c(2)*...
th1^2*x(1)^3-(x(2)+th1*x(1)^2+c(1)*x(1))*x(1)^4*ad(1)*(2*th1*x(2)+6*...
th1^2*x(1)^2+2*c(1)*th1*x(1)+5*x(1)^4*ad(1)+10*x(1)^4*ad(1)*th1*x(2)+...
4*x(1)^3*ad(1)*c(1)*x(2)+14*x(1)^6*ad(1)*th1^2+18*x(1)^5*ad(1)*c(1)*...
th1+5*x(1)^4*ad(1)*c(1)^2+2*c(2)*th1*x(1)+c(2)*c(1)+1)-c(3)*(2*th1*...
x(1)*x(2)+2*th1^2*x(1)^3+c(1)*x(2)+c(1)*th1*x(1)^2+x(3)+x(1)^2*...
(x(1)^3*ad(1)+(x(2)+th1*x(1)^2+c(1)*x(1))*ad(1)*x(1)^2*(2*th1*x(1)+...
c(1)))+c(2)*(x(2)+th1*x(1)^2+c(1)*x(1))+x(1))-2*th1*x(2)^2-6*th1^3*...
x(1)^4-x(3)*c(1)-x(3)*c(2)-24*x(1)^6*ad(1)*th1^2*x(2)-4*x(1)^3*ad(1)*...
c(1)*x(2)^2-22*x(1)^5*ad(1)*c(1)*x(2)*th1-18*x(1)^7*ad(1)*c(1)*th1^2-5*...
x(1)^6*ad(1)*c(1)^2*th1-c(2)*c(1)*th1*x(1)^2-5*x(1)^4*ad(1)*c(1)^2*...
x(2)-2*c(2)*th1*x(1)*x(2)-2*x(3)*x(1)^5*ad(1)*th1-x(3)*x(1)^4*ad(1)*c(1);
%
%    System equations
%
xdot(1)=x(2)+theta(1)*x(1)^2;
xdot(2)=x(3);
xdot(3)=u1;
%
%    Parameter estimate equations
%
xdot(4)=tau1;
```

Note that th1 and ad(1) in the expressions obtained by BACKDSMC correspond to the parameter estimate $\hat{\theta}$ and the adaptation gain, respectively.

4.3. Uncertain Nontriangular System

Consider the third order uncertain nontriangular system

$$\begin{aligned} \dot{x}_1 &= -x_1 + \theta(x_2 x_3 + 1) \\ \dot{x}_2 &= x_3 + u \\ \dot{x}_3 &= -x_3 - x_1 + \theta x_2^2 \end{aligned} \tag{43}$$

where θ is an unknown scalar parameter. This system is not transformable into the PSF or the PPF form. Nevertheless, it is globally stabilizable to the equilibrium point $(x, \hat{\theta}) = (\theta, 0, -\theta, \theta)$ by choosing $y = x_2$ as the output. This output ensures that Assumptions 1 and 2 are satisfied globally. Therefore *global* stabilization of the equilibrium point is achieved. The dynamical adaptive controller is obtained by

[c,tau,z]=backdsmc(f,g,h,yd,phi,psi,'nontrian')

which generates the m-file 'nontrian.m' for simulation purposes:

```
function xdot=nontrian(t,x);
global c ad theta;
%
%  Auxiliary variables
%
u1=x(5);
u2=x(6);
th1=x(4);
%
%  Update law
%
tau1=(x(3)+u1+c(1)*x(2))*ad(1)*x(2)^2+(c(1)*x(3)+c(1)*u1-x(3)-x(1)+...
th1*x(2)^2+u2+c(2)*(x(3)+u1+c(1)*x(2))+x(2))*ad(1)*(-x(2)*x(3)-1+...
x(2)^2*c(1)-x(2)^2+x(2)^2*c(2));
%
%   Control law
%
control=-2*x(1)-th1*x(2)*x(3)+th1-2*th1*x(2)*u1-c(2)*c(1)*x(3)-c(2)*...
c(1)*u1-3*x(3)-2*u1+c(1)*x(3)+c(1)*x(1)-c(1)*th1*x(2)^2+th1*x(2)^2+...
c(2)*x(3)+c(2)*x(1)-c(2)*th1*x(2)^2-x(2)^2*((x(3)+u1+c(1)*x(2))*ad(1)*...
x(2)^2+(c(1)*x(3)+c(1)*u1-x(3)-x(1)+th1*x(2)^2+u2+c(2)*(x(3)+u1+c(1)*...
x(2))+x(2))*ad(1)*(-x(2)*x(3)-1+x(2)^2*c(1)-x(2)^2+x(2)^2*c(2)))-c(3)...
*(c(1)*x(3)+c(1)*u1-x(3)-x(1)+th1*x(2)^2+u2+c(2)*(x(3)+u1+c(1)*x(2))+...
x(2))-c(1)*x(2);
%
%   System equations
%
xdot(1)=-x(1)+theta(1)*(x(2)*x(3)+1);
xdot(2)=x(3)+u1;
xdot(3)=-x(3)-x(1)+theta(1)*x(2)^2;
%
%   Parameter estimate equations
%
xdot(4)=tau1;
%
%  Dynamic control equations
%
xdot(5)=x(6);
xdot(6)=control;
```

Numerical simulations were carried out for this dynamically controlled system for a nominal 'unknown' parameter value $\theta = 1$. Figure 1 shows the global asymptotic stabilization achieved with the design parameters $c_1 = 5$, $c_2 = 2$, $c_3 = 3$ and $\gamma = \mathrm{ad}(1) = 2$.

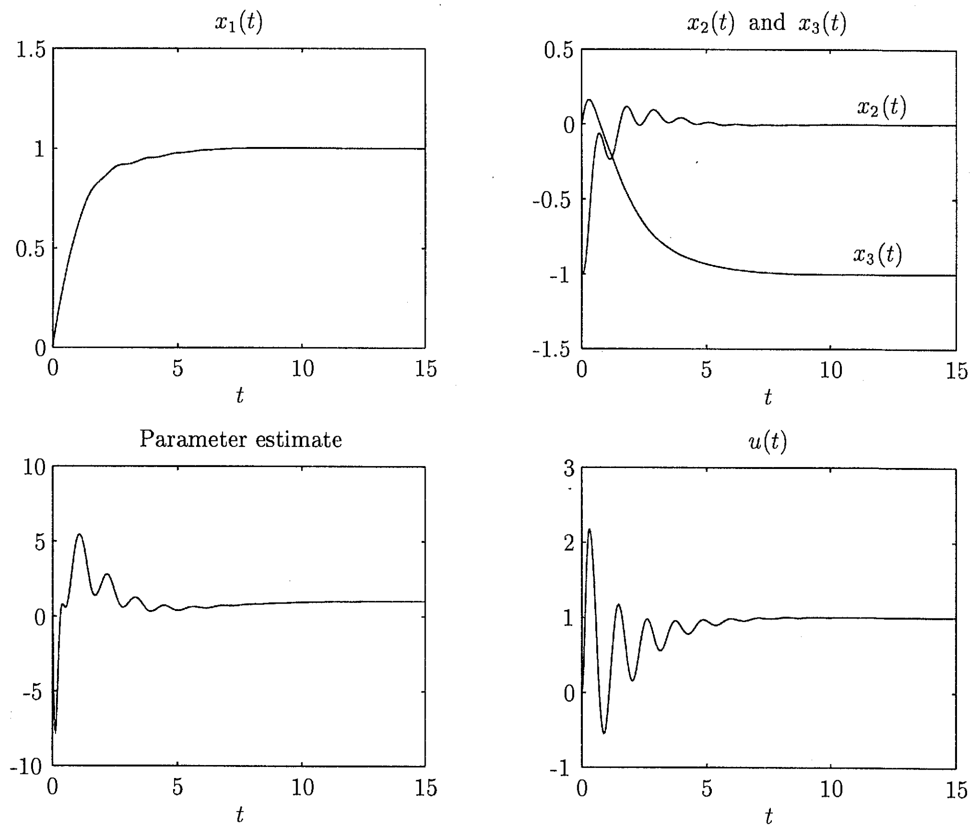

Fig. 1. Controlled state variables, parameter estimate and control input of a nontriangular system.

5. Conclusion

The use of a symbolic toolbox for the design of both static and dynamic adaptive backstepping controllers has been presented. The symbolic toolbox BACKDSMC has been developed using the MATLAB Symbolic Toolbox and implements a general adaptive backstepping algorithm with tuning functions. It is applicable to a large class of observable minimum phase systems in triangular or nontriangular forms.

References

Blankenship G.L., Ghanadan R., Kwatny H.G., La Vigna C. and Polyakov V. (1995): *Tools for integrated modelling, design, and nonlinear control.* — IEEE Contr. Syst. Mag., Vol.15, pp.65–79.

de Jager B. (1996): *Symbolic analysis and design for (nonaffine) nonlinear control systems.* — Proc. 13th *IFAC World Congress*, San Francisco, Vol.F, pp.289–294.

Glumineau A. and Graciani L. (1996): *Symbolic nonlinear analysis and control package.* — Proc. 13th *IFAC World Congress*, San Francisco, Vol.F, pp.295–298.

Jiang Z.P. and Praly L. (1991): *Iterative designs of adaptive controllers for systems with nonlinear integrators.* — Proc. 30th IEEE Conf. *Decision and Control*, Brighton, UK, pp.2482–2487.

Kanellakopoulos I., Kokotović P.V. and Morse A.S. (1991): *Systematic design of adaptive controllers for feedback linearizable systems.* — IEEE Trans. Automat. Contr., Vol.36, pp.1241–1253.

Krstić M., Kanellakopoulos I. and Kokotović P.V. (1992): *Adaptive nonlinear control without overparametrization.* — Syst. Contr. Lett., Vol.19, pp.177–185.

Krstić M., Kanellakopoulos I. and Kokotović P.V. (1995): *Nonlinear and Adaptive Control Design.* — New York: John Wiley & Sons.

Marino R. and Tomei P. (1995): *Nonlinear Control Design.* — London: Prentice-Hall.

Rios-Bolívar M. (1997): *Adaptive backstepping and sliding mode control of uncertain nonlinear systems.* — Ph.D. Thesis, Applied Math. Dept., University of Sheffield.

Rios-Bolívar M., Sira-Ramírez H. and Zinober A.S.I. (1995): *Output tracking control via adaptive input-output linearization: A backstepping approach.* — Proc. 34th IEEE Conf. *Decision and Control*, New Orleans, Vol.2, pp.1579–1584.

Rios-Bolívar M., Zinober A.S.I. and Sira-Ramírez H. (1996): *Dynamical sliding mode control via adaptive input-output linearization: A backstepping approach*, In: Robust Control via Variable Structure and Lyapunov Techniques (F. Garofalo and L. Glielmo, Eds.). — Berlin: Springer-Verlag, pp.15–35.

Rothfuss R. and Zeitz M. (1996): *A toolbox for symbolic nonlinear feedback design.* — Proc. 13th *IFAC World Congress*, San Francisco (CA), USA, Vol.F, pp.283–288.