Using Symbolic Computation for the Computer Aided Design of Nonlinear (Adaptive) Control Systems

Juan Carlos Gómez *

Department of Electrical and Computer Engineering
University of Newcastle
Newcastle, N.S.W. 2308
Australia
Phone: +61 +49 - 215963 – Fax: +61 +49 - 601712
e-mail: juan@ee.newcastle.edu.au
September, 1994

Abstract

This paper presents **NONACODE** (**NON**linear **A**daptive **CO**ntrol **DE**sign), a software package for the Computer Aided Design of Nonlinear (Adaptive) Control Systems, based on Differential Geometric Methods ([2]).

In Nonlinear Control System Design (NLCSD) with the Differential Geometric Approach, tedious symbolic calculations have to be carried out before any numerical techniques available on many CACSD packages can be used for the controller design. The use of symbolic computation systems allows to do such calculations avoiding elementary mistakes, such as wrong signs, missing brackets, wrong derivatives, etc. In addition, the use of computer algebra programming methods (in this case, the program **Mathematica** [3]) can make available to the design engineer some analytical procedures that relay on more sophisticated mathematical tools.

1 Introduction

NONACODE ([1]) is a software package for the design of Nonlinear (Adaptive) Control Systems with the Differential Geometric Approach ([2]) The package has been developed using Computer Algebra Programming Methods (namely the program *Mathematica*^{TM} [3]), and is a set of functions collected in four modules.

*This work has been partially developed at the Departamento de Electrónica - Facultad de Ciencias Exactas, Ingeniería y Agrimensura - Universidad Nacional de Rosario - Argentina, under a grant of CONICET (National Research Council)

NONACODE can be loaded in the workspace of $Mathematica^{TM}$, like any other Package, with the << operator, or with the *Needs* command, as follows:

$$In[1] :=<< Miscellaneous/nonacode.m \tag{1}$$

or

$$In[2] := Needs[\text{``}Miscellaneous\text{`}nonacode\text{`}''] \tag{2}$$

The four Modules are then available for the user.
A hint on how to use **NONACODE** is given by typing:

$$In[3] :=?NONACODE \tag{3}$$

while a *help* of each function in **NONACODE**, with the command:

$$In[4] :=?function-name \tag{4}$$

An important feature of **NONACODE** is that it is an open system, in the sense that can be improved by adding functions to deal with other nonlinear control problems, such as disturbance decoupling, output tracking for MIMO systems, etc..
The paper is organized as follows. **Modules** 1, 2, 3, and 4 are described in Sections 2, 3, 4 and 5, respectively. An observer based controller design for the output tracking of a single link robot arm is presented in Section 6, as an application example. Finally, some conclusions are given in Section 7.

2 Description of Module 1

Module 1 is compound by functions that perform some straightforward calculations of differential geometry. The following functions have been implemented in **Module 1**:

- **JACOB**$[f, x]$: computes the Jacobian matrix of the vector field $f(x)$.
- **LIEDEV**$[f, h, x]$: computes $L_f h(x)$, that is, the Lie derivative of the real valued function $h(x)$ along the direction defined by the vector field $f(x)$.
- **KLIEDEV**$[f, h, k, x]$: computes $L_f^k h(x)$, that is, the k-th Lie derivative of the real valued function $h(x)$ along the direction defined by the vector field $f(x)$.
- **LIE**$[f, g, x]$: computes $[f, g]$, that is, the Lie brackets of the vector fields $f(x)$ and $g(x)$.
- **ADJ**$[f, g, k, x]$: computes $ad_f^k g(x)$, that is the k-th adjoint of the vector fields $f(x)$ and $g(x)$.

3 Description of Module 2

Module 2 contains more sophisticated functions mainly addressed to the solution of the Output Tracking Problem for SISO systems that belong to the linear analytic class, that is of the form:

$$\dot{x} = f(x) + g(x)u \tag{5}$$

$$y = h(x) \tag{6}$$

with $x \in \Re^n$, $y, u \in \Re$, $f(x)$, $g(x)$ smooth vector fields, and $h(x)$ a smooth real valued function.
The following functions have been implemented in **Module 2**:

- **RELDEG**$[f, g, h, x]$: computes the *relative degree* (c.f.[2] for the definition) of system (5)-(6).
- **NORMAL**$[f, g, h, x]$: computes a coordinate transformation that takes system (5)-(6) to *normal form.*
- **EXACTLIN**$[f, g, h, x]$: computes a state feedback control law that exactly input-output linearize sytem (5)-(6).
  The state feedback control law is given by:

  $$u(x) = \frac{-L_f^r h(x) + v}{L_g L_f^{r-1} h(x)} \tag{7}$$

  where $r \leq n$ is the relative degree of the system.
- **OUTRACK**$[f, g, h, x, y_{ref}(t), \lambda]$: computes a state feedback control law that makes the output of system (5)-(6) to asymptotically track the reference signal $y_{ref}(t)$. The eigenvalues of the output error are given by the first r components of vector λ, being r the relative degree of the system.
  The state feedback control law is given by:

  $$u(x) = \frac{-L_f^r h(x) + y_{ref}^{(r)} - \sum_{i=1}^{r} c_{i-1} \left(L_f^{i-1} h(x) - y_{ref}^{(i-1)} \right)}{L_g L_f^{r-1} h(x)} \tag{8}$$

  where c_{i-1} with $i = 1, r$, are the coefficients of the error equation:

  $$e^{(r)} + c_{r-1} e^{(r-1)} + \ldots + c_1 e^{(1)} + c_0 e = 0 \tag{9}$$

4 Description of Module 3

Module 3 is compound by functions for the design of adaptive control of nonlinear systems using both Lyapunov and Hyperstability criteria for the parameter update laws. We consider system (5)-(6), but now assuming that $f(x)$ and $g(x)$ have parametric uncertainties of the form:

$$f(x) = \sum_{i=1}^{n1} k_{1i} f_i(x) \tag{10}$$

$$g(x) = \sum_{j=1}^{n2} k_{2j} g_j(x) \tag{11}$$

where $f_i, (i = 1, n1)$ and $g_j, (j = 1, n2)$ are known smooth vector fields, and where $k_{1i}, (i = 1, n1)$ and $k_{2j}, (j = 1, n2)$ are unknown parameters. We also assume that the system have relative degree $r = 1$.
The following functions for the solution of the Adaptive Output Tracking Problem have been implemented in **Module 3**:

- **LYADAPTRACK**$[F, G, k_1, k_2, k_{1es}, k_{2es}, h, x, y_{ref}(t), \alpha, \gamma]$: computes a control law that solves the Adaptive Output Tracking Problem, using a parameter update law based on Lyapunov criterion ([6], [7], [5]).
  The control law is given by:

$$u = \frac{-L_{f_{es}} h + \dot{y}_{ref} + \alpha(y_{ref} - y)}{L_{g_{es}} h} \tag{12}$$

where $L_{f_{es}} h(x)$ and $L_{g_{es}} h(x)$ stand for the estimates of $L_f h(x)$ and $L_g h(x)$ respectively. The parameter update law is given by:

$$\dot{\phi} = -\gamma e W \tag{13}$$

where $\phi := \theta - \theta_{es}$ is the parameter estimation error, $e := y - y_{ref}$ is the output error, and γ is a positive coefficient that fix the estimation error dynamics, and where:

$$W = [W_1 W_2]^T \tag{14}$$

with:

$$W_1 = \begin{bmatrix} L_{f_1} h \\ \vdots \\ L_{f_{n1}} h \end{bmatrix} \tag{15}$$

and

$$W_2 = \begin{bmatrix} L_{g_1} h \\ \vdots \\ L_{g_{n2}} h \end{bmatrix} . \frac{-L_{f_{es}} h + \dot{y}_{ref} + \alpha(y_{ref} - y)}{L_{g_{es}} h} \tag{16}$$

The input arguments F, G, k_1, k_2, k_{1es}, k_{2es} are given by:

$$F = [f_1(x), f_2(x), \ldots, f_{n_1}(x)] \tag{17}$$

$$G = [g_1(x), g_2(x), \ldots, g_{n_2}(x)] \tag{18}$$

$$k_1 = [k_{11}, k_{12}, \ldots, k_{1n_1}] \tag{19}$$

$$k_2 = [k_{21}, k_{22}, \ldots, k_{2n_2}] \tag{20}$$

$$k_{1es} = [k_{11es}, k_{12es}, \ldots, k_{1n_1es}] \tag{21}$$

$$k_{2es} = [k_{21es}, k_{22es}, \ldots, k_{2n_2es}] \tag{22}$$

- **HYPADAPTRACK**$[F, G, k_1, k_2, k_{1es}, k_{2es}, h, x, y_{ref}(t), \gamma_1, \gamma_2]$: computes a control law that solves the Adaptive Output Tracking Problem using a parameter update law based on Hyperstability criterion ([8], [5]).
  The control law is given by equation (12), but here, the parameter update law is given by:
  $$\phi(t) = -\gamma_1 eW - \gamma_2 \int_{t_0}^{t} eW d\tau \tag{23}$$
  The input arguments F, G, k_1, k_2, k_{1es}, k_{2es} are given by equations (17), (18), (19), (20), (21), and (22) respectively, while γ_1 and γ_2 are positive coefficients representing the adaptation gains.

5 Description of Module 4

Module 4 contains functions for the design of Nonlinear Asymptotic Observers for both SISO and MIMO systems. The following functions have been implemented in **Module 4**:

- For SISO systems of the form (5)-(6).
  - **OBSMATRIX1**$[f, g, h, x]$: computes the *Observability Matrix* defined as:
    $$\mathcal{O}(x) := \frac{\partial}{\partial t}.\begin{bmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} \tag{24}$$
  - **OBSERVER1**$[f, g, h, x, u, \lambda]$: computes for system (5)-(6) a nonlinear observer ([4]) that estimates the states $x(t)$, in such a way that the estimation error tends to zero as t tends to ∞, with eigenvalues given by the first r components of vector λ, being $r \leq n$ the relative degree of the system.
    The observer equation is given by:
    $$\dot{x}_{es} = f(x_{es}) + g(x_{es}).u(t) + \mathcal{O}^{-1}(x_{es})K(y(t) - h(x_{es})) \tag{25}$$
    where $x_{es}(t)$ stands for the estimates of $x(t)$, and the column vector K is such that the eigenvalues of the estimation error are given by the first r components of vector λ.
  - **OBSERDYN1**$[\lambda, r]$: computes the column vector K appearing in the observer equation (25), such that the eigenvalues of the estimation error are given for the first r components of vector λ, being r the relative degree of the system.
- Let us consider the following MIMO system:
  $$\dot{x} = f(x) + \sum_{i=1}^{n} g_i(x) u_i \tag{26}$$

$$
\begin{aligned}
y_1 &= h_1(x) \\
&\vdots \\
y_n &= h_n(x)
\end{aligned}
\tag{27}
$$

We define:

$$y(x) := [y_1(x), \ldots, y_m(x)] \tag{28}$$

$$u(t) := [u_1(x), \ldots, u_m(x)] \tag{29}$$

$$G(x) := [g_1(x), \ldots, g_m(x)] \tag{30}$$

$$H(x) := [h_1(x), \ldots, h_m(x)] \tag{31}$$

- **VECRELDEG**$[f, G, H, x]$
  For the MIMO system (26)-(27), the function **VECRELDEG** computes its vector relative degree (c.f. [2] for the definition)
- **OBSMATRIXN**$[f, G, H, x]$:
  Assuming that system (26)-(27) has vector relative degree $[r_1, \ldots, r_q]$ such that $\sum_{i=1}^{q} r_i = n$, the function **OBSMATRIXN** computes the *Observability Matrix*, defined as follows:

$$
\mathcal{O}(x) := \frac{\partial}{\partial t}.\begin{bmatrix} h_1(x) \\ L_f h_1(x) \\ \vdots \\ L_f^{r_1-1} h_1(x) \\ h_2(x) \\ L_f h_2(x) \\ \vdots \\ L_f^{r_2-1} h_2(x) \\ \vdots \\ h_m(x) \\ L_f h_m(x) \\ \vdots \\ L_f^{r_m-1} h_m(x) \end{bmatrix}
\tag{32}
$$

- **OBSERVERN**$[f, G, H, x, x_{es}, u, \lambda]$
  Assuming that system (26)-(27) has vector relative degree $[r_1, \ldots, r_q]$ such that $\sum_{i=1}^{q} r_i = n$, the function **OBSERVERN** computes an asymptotic nonlinear observer, whose estimation error dynamics can be arbitrarily fixed by means of vector λ. The observer equation is given by:

$$\dot{x}_{es} = f(x_{es}) + G(x_{es}).u(t) + \mathcal{O}^{-1}(x_{es})K(y(t) - H(x_{es})) \tag{33}$$

  with:

$$
K := \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_m \end{bmatrix}
\tag{34}
$$

and where K_i with $i = 1, m$ are defined as follows:

$$K_1 = \begin{bmatrix} K_{11} & 0 & \dots & 0 \\ K_{12} & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ K_{1r_1} & 0 & \dots & 0 \end{bmatrix} \tag{35}$$

$$K_2 = \begin{bmatrix} 0 & K_{21} & \dots & 0 \\ 0 & K_{22} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & K_{2r_2} & \dots & 0 \end{bmatrix} \tag{36}$$

$$\vdots$$

$$K_m = \begin{bmatrix} 0 & 0 & \dots & K_{m1} \\ 0 & 0 & \dots & K_{m2} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & K_{mr_m} \end{bmatrix} \tag{37}$$

Matrix K is such that the eigenvalues of the estimation error are the components of vector λ.

6 Application Example

As an application example we consider a single link robot arm consisting of a rigid link coupled to the shaft of an armature controlled DC motor drive.

An schematic diagram of the system is shown in Fig. 1, and the model variables and parameters are listed in Table 1.

$\theta(t)$	:	angle of the link
$\omega(t)$	:	angular velocity of the link
J_m	:	rotational inertia of the motor
J_l	:	rotational inertia of the link
M	:	total mass of the link
l	:	distance from the joint axes to the link center of mass
b	:	dumping coefficient of the motor
L_a	:	armature inductance
R_a	:	armature resistance
ϵ	:	back emf
i_a	:	armature current
$u(t)$	:	armature voltage

Table 1: Model Variables and Parameters

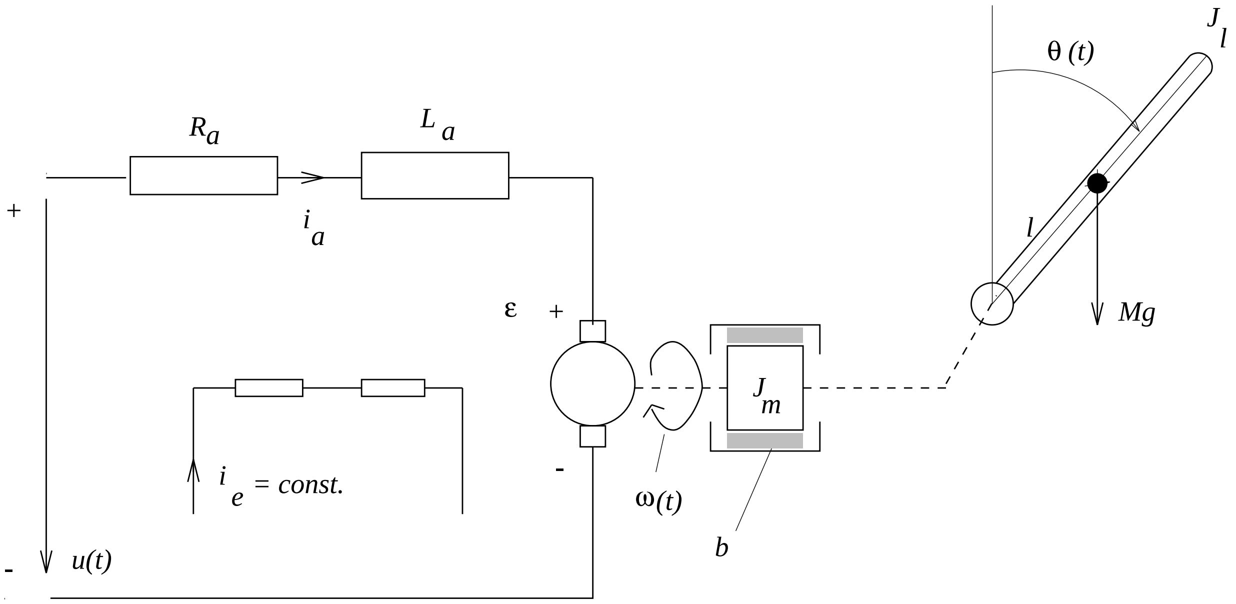

Figure 1: Single Link Robot Arm

The state variables of the system are supposed to be:

$$\begin{aligned} x_1(t) &:= \theta(t) \\ x_2(t) &:= \omega(t) \\ x_3(t) &:= i_a(t) \end{aligned} \tag{38}$$

The state equations of the system are:

$$\dot{x}_1(t) = x_2 \tag{39}$$

$$\dot{x}_2(t) = -\frac{b}{J}x_2 + \frac{K}{J}x_3 + \frac{Mgl}{J}sin(x_1) \tag{40}$$

$$\dot{x}_3(t) = -\frac{K}{L_a}x_2 - \frac{R_a}{L_a}x_3 + \frac{1}{L_a}u(t) \tag{41}$$

and the output equation:

$$y(t) = h(x) = \theta(t) = x_1(t) = [100]x \tag{42}$$

It can be seen that the equations are of the form (5)-(6), with:

$$f(x) = \begin{bmatrix} x_2 \\ -\frac{b}{J}x_2 + \frac{K}{J}x_3 + \frac{Mgl}{J}sin(x_1) \\ -\frac{K}{L_a}x_2 - \frac{R_a}{L_a}x_3 + \frac{1}{L_a}u(t) \end{bmatrix} \tag{43}$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} \tag{44}$$

$$h(x) = [100]x \tag{45}$$

Using function **RELDEG**, the relative degree of the system results:

$$In[1] := RELDEG[f, g, h, x] \tag{46}$$

$$Out[1] := 3 \tag{47}$$

that is $r = 3$. The control law for the solution of the output tracking problem is obtained calling function **OUTRACK**. In this case, we have chosen $\lambda_u = \{-10, -10, -10\}$, and the control law, results:

$$\begin{array}{l} u = \frac{-1000JL_a}{K}x_1 - \frac{L_a b^2}{JK}x_2 + Kx_2 + R_a x_3 + \frac{L_a b}{J}x_3 - \\ \frac{L_a Mgl}{K}x_2 cos(x_1) + \frac{L_a Mglb}{K}sin(x_1) + \frac{1000JL_a}{K}y_r + \\ \frac{300JL_a}{K}(-x_1^{(1)} + y_r^{(1)}) + \\ \frac{30JL_a}{K}(-x_1^{(2)} + y_r^{(2)}) + \frac{JL_a}{K}y_r^{(3)} \end{array} \tag{48}$$

The observer equation can be obtained by calling function **OBSERVER1**. Here, we have chosen $\lambda_o = \{-100, -100, -100\}$, and the observer equation is given by:

$$\dot{x}_{1es} = 300(x_1 - x_{1es}) + x_{2es} \tag{49}$$

$$\dot{x}_{2es} = 3000(x_1 - x_{1es}) - \frac{b}{J}x_{2es} + \frac{K}{J}x_{3es} + \frac{Mgl}{J}sin(x_{1es}) \tag{50}$$

$$\begin{array}{l} \dot{x}_{3es} = \frac{1}{L_a}u - \frac{K}{L_a}x_{2es} - \frac{R_a}{L_a}x_{3es} + \\ (x_1 - x_{1es}).\left[\frac{10^6 J}{K} + \frac{30000b}{K} - \frac{300Mgl}{K}cos(x_{1es})\right] \end{array} \tag{51}$$

Finally, the control law for output tracking is implemented with the estimates of the states variables.

For the digital simulation of the system, the following parameter values have been adopted:

$R_a = 1\Omega$ $\quad L_a = 0.002H$ $\quad K = 10$

$J = 4Nms^2$ $\quad b = 1Nms$ $\quad l = 1m$

$Mg = 20N$

Fig (2) shows the state $x_1(t)$, its estimate $x_{1es}(t)$, and the reference signal $y_{ref}(t)$. FIgures (3), (4) and (5) show the estimation error for the variables x_1, x_2 and x_3 respectively. Figure (6) shows the output error, and figure (7) represents the control variable $u(t)$.

7 Conclusions

A software package for the Computer Aided Design of Nonlinear (Adaptive) Control with the Differential Geometric Approach, has been developed using Computer Algebra. The package **NONACODE** allows to solve several standard nonlinear control problems, such as exact linearization via state feedback, output tracking, adaptive output tracking, and nonlinear observers for SISO and MIMO systems. The use of computer algebra programming methods makes the Differential Geometric Approach available to users who need not be familiar with this theory.

Since **NONACODE** is a collection of functions, it can be extended to include functions for the solution of other nonlinear control problems such as disturbance decoupling, and noninteractive control.

References

[1] Gómez, J.C. (1993): ***"NONACODE: A Software Package for Analysis and Design of Adaptive Control of Nonlinear Systems using Computer Algebra"***, Presented at SIAM 1993 Annual Meeting, Philadelphia, U.S.A., July 1993.

[2] Isidori, A. (1989): ***"Nonlinear Control Systems: An Introduction"***, 2nd. Edition. Springer-Verlag, Berlin.

[3] Wolfram, S. (1988): ***"Mathematica: A System for Doing Mathematics by Computer"***, Addison Wesley, Reading, Mass.

[4] D'Attellis, C. (1992): ***"Introducción a los Sistemas No Lineales de Control y sus Aplicaciones"***,(in Spanish), AADECA, Buenos Aires.

[5] Gómez, J. C.; Junco, S. (1992): ***"Linealización Exacta e Hiperestabilidad en el Diseño de Controladores Adaptables"***, (in Spanish), Proceedings of the XIII Simposio Nacional de Control Automático, Buenos Aires, Argentina.

[6] Sastry, S.S.; Isidori, A. (1989): ***"Adaptive Control of Linearizable Systems"***, IEEE Trans. Aut. Control 34-11, 1123-1131.

[7] Sastry, S.S.; Bodson, M. (1989): ***"Adaptive Control: Stability, Convergence, and Robustness"***, Prentice Hall Advance Reference Series, Information and System Sciences Series, Englewood Cliffs, N.J.

[8] Landau, Y.D. (1979): ***"Adaptive Control: The Model Reference Approach"***, Control and Systems Theory Series, Marcel Dekker, Inc., New York.

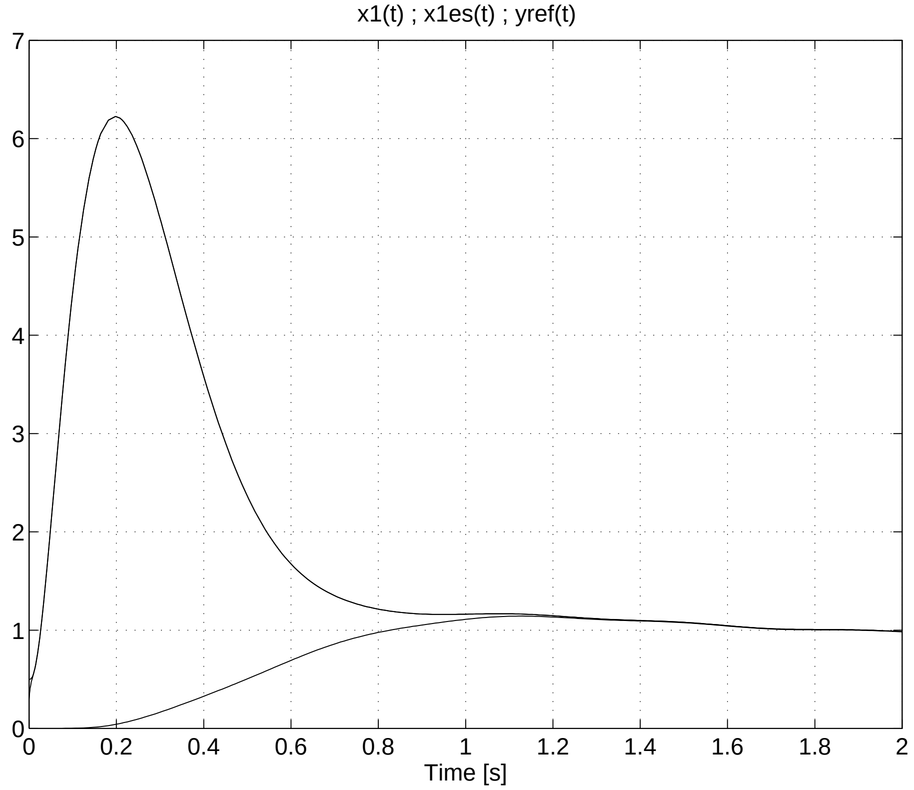

Figure 2:

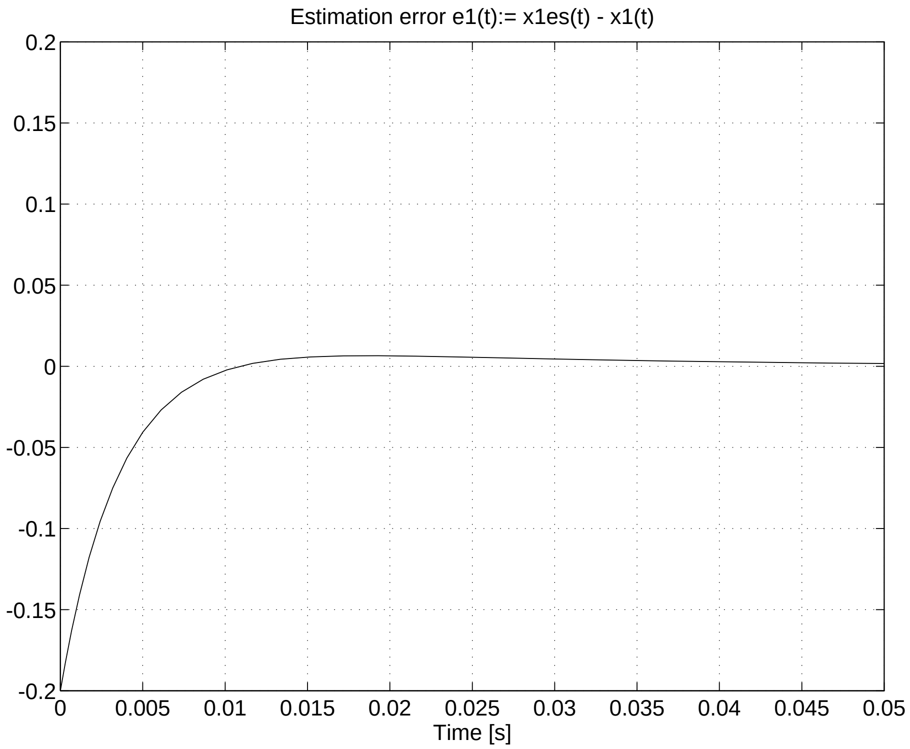

Figure 3:

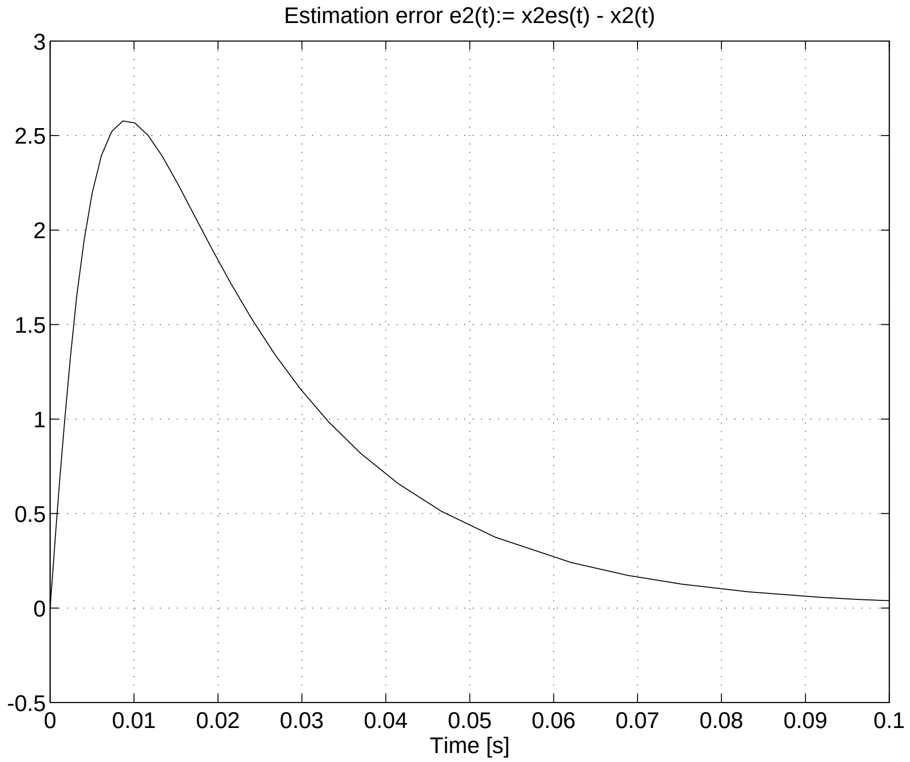

Figure 4:

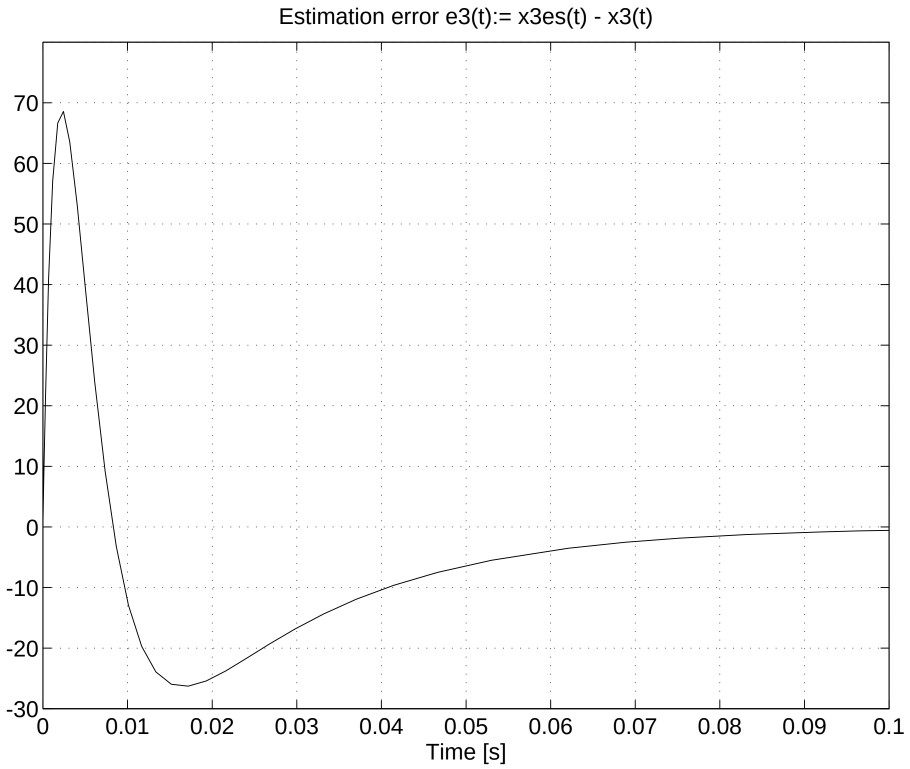

Figure 5:

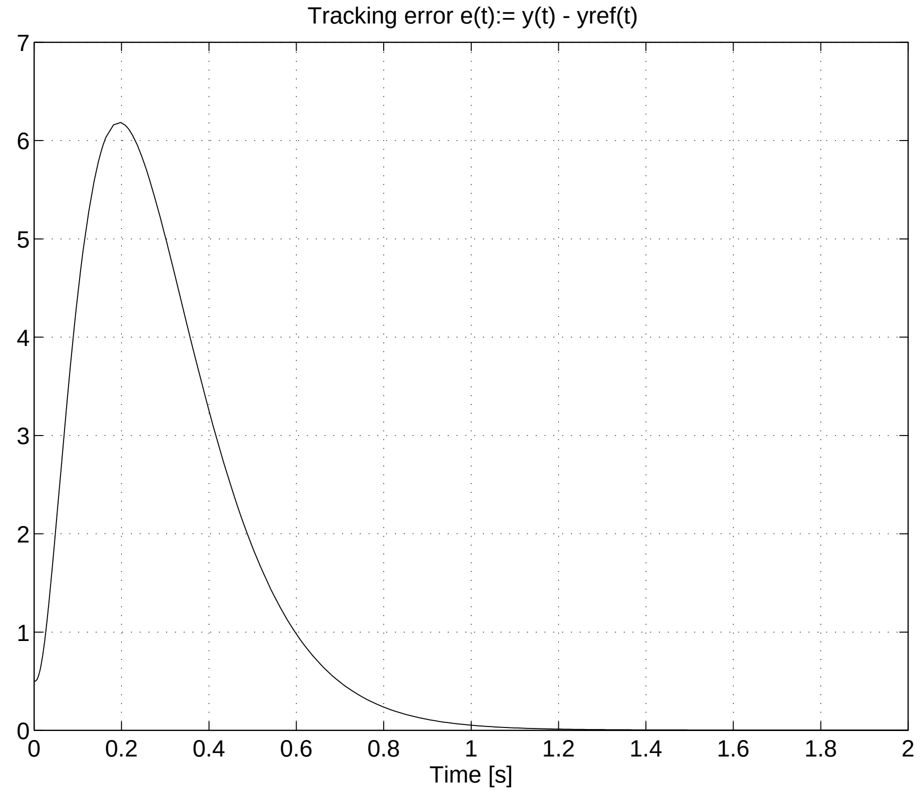

Figure 6:

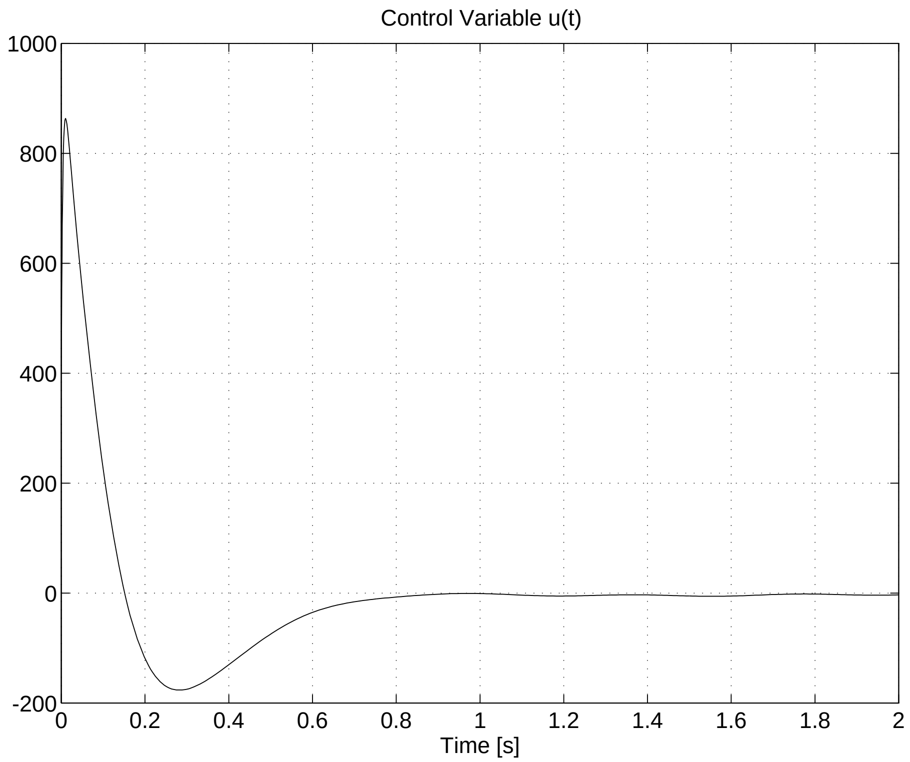

Figure 7: