



# Computational Approach to Control Laws of Strict-Feedback Nonlinear Systems

Rongdong Wang, Reza Ahangar

Department of Mathematics, Texas A & M University-Kingsville, Kingsville, TX, USA

Email: rongdong.wang@tamuk.edu, reza.ahangar@tamuk.edu

**How to cite this paper:** Wang, R.D. and Ahangar, R. (2016) Computational Approach to Control Laws of Strict-Feedback Nonlinear Systems. *Open Access Library Journal*, 3: e2938.

<http://dx.doi.org/10.4236/oalib.1102938>

**Received:** September 8, 2016

**Accepted:** October 6, 2016

**Published:** October 10, 2016

Copyright © 2016 by authors and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Maple is used to compute control laws of strict-feedback nonlinear systems. The stability of the system is also verified by Maple procedures. We show that computer algebra systems can play an important role in nonlinear system design, in research, and education of nonlinear systems.

## Subject Areas

Computer Engineering, Numerical Mathematics

## Keywords

Back-Stepping, Computer Algebra Systems (CAS), Strict-Feedback, Nonlinear Systems, Lyapunov Function, Maple Procedures

---

## 1. Introduction

Nonlinear system study has been advanced greatly in the last few decades. In particular, the back-stepping method was discovered and applied in the design of nonlinear control systems [1].

The complexity of the computation involved in the study of nonlinear systems makes the use of computer algebra inevitable. In fact, the nature of the computation of back-stepping, a recursive procedure, is symbolic.

Computer Algebra System, also called CAS for symbolic computation, has been advanced greatly in the last few decades as well. Powerful computer algebra systems now are capable to do numerical computation, symbolic computation, graphing, and programming. The rich built-in functions in a computer algebra system make the study of complicated systems handy.

Computer algebra has been used in the study of nonlinear systems in general (for

example, see [1]-[5]), and in the study of back-stepping in particular (for example, see [6]). However, for some reasons, computer algebra systems are not yet popular in the engineering society. In this paper, we show how computer algebra systems can be successfully used in computing a control law for strict-feedback systems.

The use of computer algebra systems allows researchers and students to save time from tedious and error prone computations and focus on the basic ideas and methods.

Therefore, it is not only important for industrial applications, but also for research and education. Maple is one of the most powerful computer algebra systems in the world. We use Maple to compute control laws of strict-feedback systems. We also use Maple to verify the stability and other behaviors of the systems.

In this paper, Maple procedures are listed and examples are given.

To familiarize the audience in the community of control systems with computer algebra systems is also a goal of this paper.

## 2. Back-Stepping for Strict-Feedback Systems

Assume a state vector  $x \in \mathbb{R}^n$ , we define nonlinear recursive and strict-feedback systems by the following:

$$\dot{x} = f(x) + g(x)z_1 \quad (1)$$

$$\begin{aligned} \dot{z}_1 &= h_1(x, z_1) + s_1(x, z_1)z_2 \\ \dot{z}_2 &= h_2(x, z_1, z_2) + s_2(x, z_1, z_2)z_3 \\ &\vdots \\ \dot{z}_{k-1} &= h_{k-1}(x, z_1, \dots, z_{k-1}) + s_{k-1}(x, z_1, \dots, z_{k-1})z_k \end{aligned} \quad (2)$$

The recurrence relation (2) can be briefly described by:

$$\dot{z}_k = h_k(x, z_1, \dots, z_k) + s_k(x, z_1, \dots, z_k)u \quad (3)$$

for  $k = 1, 2, 3, \dots, n$ , where  $x \in \mathbb{R}^n$ ,  $z_k \in \mathbb{R}$ , ( $k = 1, \dots, n$ ), and  $n$  is the vector field dimension. The value  $u \in \mathbb{R}$  is the plant input and the new state variable  $z_k$  is considered the plant output.

The functions  $f$  and  $g$  in relation (1) can be described by

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad f(x) = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{bmatrix}, \quad g(x) = \begin{bmatrix} g_1(x_1, \dots, x_n) \\ \vdots \\ g_n(x_1, \dots, x_n) \end{bmatrix}.$$

Lyapunov Stable Control Design:

Assume that for the system,  $\dot{x} = f(x) + g(x)u$ , there exists a continuously differentiable feedback control law  $u = \varphi(x)$  a smooth, positive definite, and radially unbounded function  $V(x)$  such that

$$\dot{V}(x) = \frac{\partial V}{\partial x}(f(x) + g(x)\varphi(x))$$

is negative definite, *i.e.*, the system

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

is asymptotically stable.

The following recursive procedure is used to compute a control law to stabilize the above strict-feedback system [7].

$$\begin{aligned}\alpha_1(x, z_1) &= \frac{1}{s_1(x, z_1)} \left( \varphi(x) - z_1 - \frac{\partial V}{\partial x} g(x) - h_1(x, z_1) + \frac{\partial \varphi}{\partial x} (f(x) + g(x)z_1) \right) \\ \alpha_2(x, z_1, z_2) &= \frac{1}{s_2(x, z_1, z_2)} \left( \alpha_1(x, z_1) - z_2 + (\varphi(x) - z_1)s_1(x, z_1) - h_2(x, z_1, z_2) \right. \\ &\quad \left. + \frac{\partial \alpha_1(x, z_1)}{\partial x} (f(x) + g(x)z_1) + \frac{\partial \alpha_1(x, z_1)}{\partial z_1} (h_1(x, z_1) + s_1(x, z_1)z_2) \right) \\ \alpha_j(x, z_1, \dots, z_j) &= \frac{1}{s_j(x, z_1, \dots, z_j)} \left( \alpha_{j-1}(x, z_1, \dots, z_{j-1}) - z_j \right. \\ &\quad \left. + (\alpha_{j-1}(x, z_1, \dots, z_{j-1}) - z_{j-1})s_{j-1}(x, z_1, \dots, z_{j-1}) \right. \\ &\quad \left. - h_j(x, z_1, \dots, z_j) + \frac{\partial \alpha_{j-1}(x, z_1, \dots, z_{j-1})}{\partial x} (f(x) + g(x)z_1) \right. \\ &\quad \left. + \sum_{i=1}^{j-1} \frac{\partial \alpha_{j-1}(x, z_1, \dots, z_{j-1})}{\partial z_i} (h_i(x, z_1, \dots, z_i) + s_i(x, z_1, \dots, z_i)z_{i+1}) \right)\end{aligned}$$

where  $j = 3, \dots, k$ . The control law can be described:  $u = \alpha_k(x, z_1, \dots, z_k)$ .

There are other control laws based on back-stepping. The above control law may not be the best choice for a particular control system with particular design concerns. However The program can be modified if some application oriented guidelines are given.

This paper is not for the detailed discussion of the back-stepping method; instead, it is for the computation of the method. From the above formulas one can see that the computation is very tedious and time-consuming.

Computer algebra can be used to compute the control laws.

### 3. Maple Procedures

The following is a Maple Procedure for computing the control law  $u$  of nonlinear strict-feedback systems. The first argument of the procedure is  $f(x)$ . The second argument of it is  $g(x)$ . The third and the fourth arguments are the known Lyapunov function,  $V(x)$ , and the control law,  $u = \varphi(x)$ , for system  $\dot{x} = f(x) + g(x)u$ .

The following program is the Maple Procedure designed to solve the dynamical control systems in (1), (2), and (3), in general to compute a control law to stabilize the strict-feedback system.

```
restart;
with(linalg);
strict:= proc (f, g, h, s, V, a)
    local z, alpha, n, k, i, j;
```

```

with(linalg); n := vectdim(f); k := vectdim(s);
for i to 1 do
    alpha[1] := simplify((-z[1]+a-add((diff(V, x[i]))*g[i], i = 1 .. n)-h[1]+(diff(a,
    x[1]))*(z[1]*g[1]+f[1]))/s[1]);
if k = 1 then break end if;
    alpha[2] := simplify((alpha[1]-z[2]-(z[1]-a)*s[1]-h[2]+add((diff(alpha[1],
    x[i]))*(z[1]*g[i]+f[i]), i = 1 .. n)+(diff(alpha[1],
    z[1]))*(z[2]*s[1]+h[1]))/s[2]);
if k = 2 then break end if;
for j from 3 to k do
    alpha[j] := simplify((alpha[j-1]-z[j]-(z[j-1]-alpha[j-2])*s[j-1]-h[j]+add
    ((diff(alpha[j-1], x[i]))*(z[1]*g[i]+f[i]), i = 1 .. n)+add((diff(alpha[j-1],
    z[i]))*(z[i+1]*s[i]+h[i]), i = 1 .. j-1))/s[j])
end do
end do
end proc;

```

Program (1): This table is a copy of the Maple procedure named “strict” that will be used to execute for a variety of Nonlinear Control Systems.

The following is the result of the execution of the Maple procedure to show phase portraits of second order nonlinear systems with arbitrary initial conditions. The arguments described in the above “strict” procedure are systems of equations and numbers of initial value problems (up to 20)

```

portrait:=proc(sys, m, tm, icr)
local r,n,s,sx,sx1,i,x,y,a,b;
with(plots):
x:=op(1,lhs(sys[1])); y:=op(1,lhs(sys[2]));
n:=m; r:=rand(-5..5); s:=array(1..n); sx:=array(1..n); sx1:=array(1..n);
for i from 1 to n do
s[i]:=dsolve({op(sys),subs(op(x)=0,x)=icr*(rand()/(2*10^(12)))r(),subs(op(y)=0,y)=i
cr*(rand()/(2*10^(12)))r()}},{x,y},numeric);
od;
for i from 1 to 1 do
sx[1]:=t->rhs(s[1](t)[2]); sx1[1]:=t->rhs(s[1](t)[3]):
if n=1 then break fi;sx[2]:=t->rhs(s[2](t)[2]); sx1[2]:=t->rhs(s[2](t)[3]):
if n=2 then break fi;sx[3]:=t->rhs(s[3](t)[2]);sx1[3]:=t->rhs(s[3](t)[3]):
if n=3 then break fi;sx[4]:=t->rhs(s[4](t)[2]);sx1[4]:=t->rhs(s[4](t)[3]):
if n=4 then break fi;sx[5]:=t->rhs(s[5](t)[2]);sx1[5]:=t->rhs(s[5](t)[3]):
if n=5 then break fi;sx[6]:=t->rhs(s[6](t)[2]); sx1[6]:=t->rhs(s[6](t)[3]):
if n=6 then break fi;sx[7]:=t->rhs(s[7](t)[2]); sx1[7]:=t->rhs(s[7](t)[3]):
if n=7 then break fi;sx[8]:=t->rhs(s[8](t)[2]);sx1[8]:=t->rhs(s[8](t)[3]):
if n=8 then break fi;sx[9]:=t->rhs(s[9](t)[2]);sx1[9]:=t->rhs(s[9](t)[3]):
if n=9 then break fi;sx[10]:=t->rhs(s[10](t)[2]);sx1[10]:=t->rhs(s[10](t)[3]):

```

```

if n=10 then break fi; sx[11]:=t->rhs(s[11](t)[2]): sxi1[11]:=t->rhs(s[11](t)[3]):
if n=11 then break fi; sx[12]:=t->rhs(s[12](t)[2]): sxi1[12]:=t->rhs(s[12](t)[3]):
if n=12 then break fi; sx[13]:=t->rhs(s[13](t)[2]): sxi1[13]:=t->rhs(s[13](t)[3]):
if n=13 then break fi; sx[14]:=t->rhs(s[14](t)[2]): sxi1[14]:=t->rhs(s[14](t)[3]):
if n=14 then break fi; sx[15]:=t->rhs(s[15](t)[2]): sxi1[15]:=t->rhs(s[15](t)[3]):
if n=15 then break fi; sx[16]:=t->rhs(s[16](t)[2]): sxi1[16]:=t->rhs(s[16](t)[3]):
if n=16 then break fi; sx[17]:=t->rhs(s[17](t)[2]): sxi1[17]:=t->rhs(s[17](t)[3]):
if n=17 then break fi; sx[18]:=t->rhs(s[18](t)[2]): sxi1[18]:=t->rhs(s[18](t)[3]):
if n=18 then break fi; sx[19]:=t->rhs(s[19](t)[2]): sxi1[19]:=t->rhs(s[19](t)[3]):
if n=19 then break fi; sx[20]:=t->rhs(s[20](t)[2]): sxi1[20]:=t->rhs(s[20](t)[3]):
if n=20 then break fi;
od;
a:=plot([seq([sx[i],sxi1[i],0..0.001],i=1..n)],style=point,labels=[x, y],caption="The
phase portrait with random initial conditions");
b:=plot([seq([sx[i],sxi1[i],0..tm],i=1..n)],labels=[x, y],caption="The phase por-
trait with random initial conditions"); print(display(a,b)); sx,sxi1:
end:

```

Program (2): This table represents the execution of the Maple procedure named “portrait” to produce a sequence of graphs (sgraph) for the strict-feedback nonlinear control systems.

Due to the paper size limitation, Maple procedure, and sgraphs to show the graphs of third order systems with arbitrary initial conditions are not listed. The arguments of the procedure are also systems of equations and numbers of initial value problems.

#### Examples

The following examples show that the above procedures work well.

**Example 1:** Given a dynamical system (see the reference [8], pp. 592-594)

$$\begin{cases} \dot{x}_1 = x_1^2 + x_1^3 + z_1 \\ \dot{z}_1 = u \end{cases} \quad (4)$$

The following are the Maple commands and results of the control solutions.

> f :=  $\langle x_1^2 - x_1^3 \rangle$ ; g :=  $\langle 1 \rangle$ ;

$$f := \begin{bmatrix} x_1^2 - x_1^3 \end{bmatrix}$$

$$g := \begin{bmatrix} 1 \end{bmatrix}$$

> V :=  $\frac{1}{2} \cdot x_1^2$ ;  $\phi := -x_1^2 - x_1$ ;

$$V := \frac{1}{2} x_1^2$$

$$\phi := -x_1^2 - x_1$$

> h :=  $\langle 0 \rangle$ ; s :=  $\langle 1 \rangle$ ;

$$h := \begin{bmatrix} 0 \end{bmatrix}$$

$$s := [ 1 ]$$

>  $u := \text{strict}(f, g, h, s, V, \varphi);$

$$u := -2z_1 - 2x_1^2 - 2x_1 - x_1^3 + 2x_1^4 - 2x_1z_1$$

We use our Maple procedure “portrait” to demonstrate the phase portraits of the system with random initial conditions. It can be seen that the system is globally asymptotically stable. Since the procedure “portrait” uses different variable names and needs for forming systems of differential equations in the Maple setting, we converted  $u$  to a suitable form first.

>  $u := \text{subs}(\{op(2, op(1, u)) = \xi I(t), x_1 = x(t)\}, u);$

$$u := -2\xi I(t) - 2x(t)^2 - 2x(t) - x(t)^3 + 2x(t)^4 - 2x(t)\xi I(t)$$

>  $\text{sys1} := \{diff(x(t), t) = x(t)^2 - x(t)^3 + \xi I(t), diff(\xi I(t), t) = u\};$

$$\text{sys1} := \left\{ \frac{d}{dt} x(t) = x(t)^2 - x(t)^3 + \xi I(t), \frac{d}{dt} \xi I(t) = -2\xi I(t) - 2x(t)^2 - 2x(t) - x(t)^3 + 2x(t)^4 - 2x(t)\xi I(t) \right\}$$

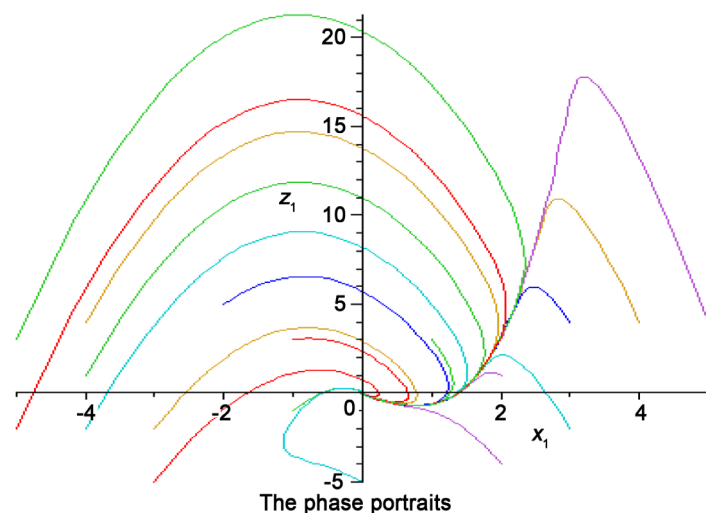
>  $\text{portrait}(\text{sys1}, 20);$

The portrait in **Figure 1** demonstrates the global stability of the strict feedback control systems with one level integrator  $z_1$  about the equilibrium point.

Now consider the nonlinear system (4) with two integrators  $z_1$  and  $z_2$ .

$$\begin{aligned} \dot{x}_1 &= x_1^2 + x_1^3 + z_1 \\ \dot{z}_1 &= z_2 \\ \dot{z}_2 &= u \end{aligned} \quad (5)$$

The following are the Maple commands and results.



**Figure 1.** Phase portrait of a nonlinear control system (4) with random initial conditions.

By executing the Maple procedure “strict” from the Program (1) and the “portrait” procedure from Program (2), we can generate a sequence of graphs that we labeled as a[1], a[2], and a[3].

>  $h := \langle 0, 0 \rangle; s := \langle 1, 1 \rangle;$

$$h := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$s := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

>  $u := \text{strict}(f, g, h, s, V, \varphi);$

$$u := -5z_1 - 5x_1^2 - 3x_1 - 3x_1^3 + 3x_1^4 - 6x_1z_1 - 3z_2 + 11x_1^5 - 5x_1^2z_1 - 8x_1^6 + 10x_1^3z_1 - 2z_1^2 - 2z_2x_1$$

>  $u := \text{subs}(\{x_1 = x(t), \text{op}(2, \text{op}(1, u)) = z1(t), \text{op}(2, \text{op}(13, u)) = z2(t)\}, u);$

$$u := -5z1(t) - 5x(t)^2 - 3x(t) - 3x(t)^3 + 3x(t)^4 - 6x(t)z1(t) - 3z2(t) + 11x(t)^5 - 5x(t)^2z1(t) - 8x(t)^6 + 10x(t)^3z1(t) - 2z1(t)^2 - 2z2(t)x(t)$$

>  $\text{sys} := \{\text{diff}(x(t), t) = x(t)^2 - x(t)^3 + z1(t), \text{diff}(z1(t), t) = z2(t), \text{diff}(z2(t), t) = u\};$

$$\text{sys} := \left\{ \frac{d}{dt} z2(t) = -5z1(t) - 5x(t)^2 - 3x(t) - 3x(t)^3 + 3x(t)^4 - 6x(t)z1(t) - 3z2(t) + 11x(t)^5 - 5x(t)^2z1(t) - 8x(t)^6 + 10x(t)^3z1(t) - 2z1(t)^2 - 2z2(t)x(t), \frac{d}{dt} x(t) = x(t)^2 - x(t)^3 + z1(t), \frac{d}{dt} z1(t) = z2(t) \right\}$$

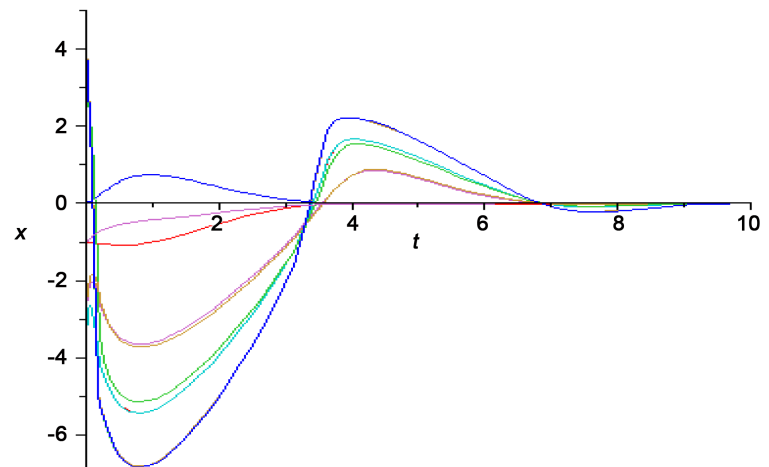
>  $a := \text{sgraph}(\text{sys}, 11);$

>  $a[1]; a[2]; a[3];$  (Figure 2).

**Example 2: The following example demonstrates three levels of integrators for back-stepping control** (see the reference [9], p. 35)

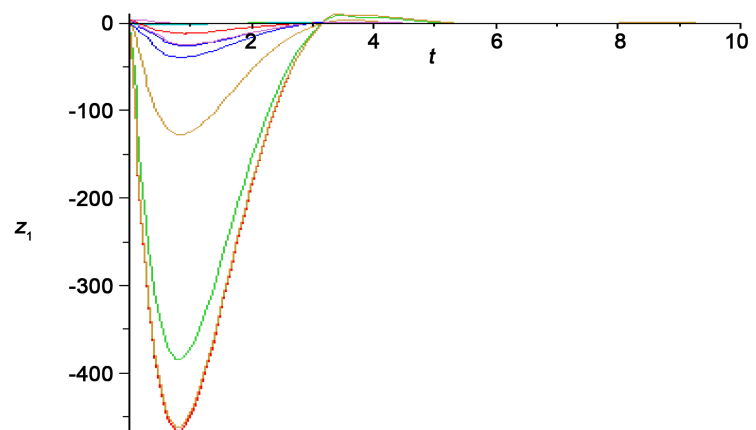
$$\begin{cases} \dot{x}_1 = x_1 z_1 \\ \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \dot{z}_3 = u \end{cases} \quad (6)$$

We initiate functions f and g as necessary by (1). We use the Lyapunov function V to execute the Maple procedure “strict”.



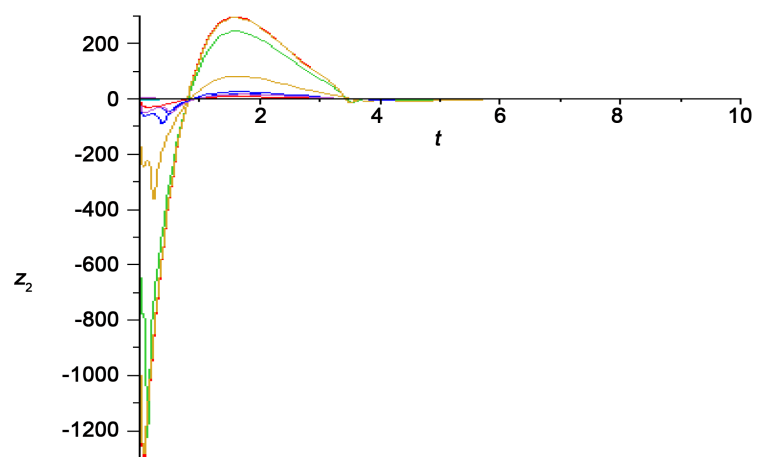
Solutions  $x_1(t)$

a[1]



Solutions  $z_1(t)$

a[2]



Solutions  $z_2(t)$

a[3]

**Figure 2.** a[1], a[2]), and a[3] are phase portraits of the nonlinear control system (5) in two levels using back-stepping algorithm.



$$> f := \langle 0 \rangle; g := \langle x_1 \rangle;$$

$$f := \begin{bmatrix} 0 \end{bmatrix}$$

$$g := \begin{bmatrix} x_1 \end{bmatrix}$$

$$> \phi := -x_1^2; V := \frac{1}{2} \cdot x_1^2;$$

$$\phi := -x_1^2$$

$$V := \frac{1}{2} x_1^2$$

$$> h := \langle 0, 0, 0 \rangle; s := \langle 1, 1, 1 \rangle;$$

$$h := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$s := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$> u := \text{strict}(f, g, h, s, V, \phi);$$

$$u := -3 z_1 - 5 x_1^2 - 14 x_1^2 z_1 - 5 z_2 - 16 x_1^2 z_1^2 - 8 z_2 x_1^2 - 3 z_3 - 8 x_1^2 z_1^3 - 12 x_1^2 z_1 z_2 - 2 z_3 x_1^2$$

**Example 3:** The following example is a nonlinear control system selected from the reference ([10], pp. 633-634):

$$\begin{cases} \dot{x}_1 = z_1 \\ \dot{z}_1 = x_1 - x_1^3 + z_2 \\ \dot{z}_2 = u \end{cases} \quad (7)$$

We use functions  $f$  and  $g$  from the system (7) to be used for Equation (1). Running the Maple procedure “strict” from the Program (1) produces the solution. The “portrait” procedure generates a sequence of graphs that we labeled  $b[1]$ ,  $b[2]$ , and  $b[3]$ .

$$> f := \langle 0 \rangle; g := \langle 1 \rangle;$$

$$f := \begin{bmatrix} 0 \end{bmatrix}$$

$$g := \begin{bmatrix} 1 \end{bmatrix}$$

$$> h := \langle x_1 - x_1^3, 0 \rangle; s := \langle 1, 1 \rangle;$$

$$h := \begin{bmatrix} x_1 - x_1^3 \\ 0 \end{bmatrix}$$

$$s := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$> V := \frac{1}{2} \cdot x_1^2; \phi := -x_1;$$

$$V := \frac{1}{2} x_1^2$$

$$\phi := -x_1$$

$$> u := \text{strict}(f, g, h, s, V, \phi);$$

$$u := -6 z_1 - 6 x_1 + 3 x_1^3 - 3 z_2 + 3 z_1 x_1^2$$

$$s := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$> V := \frac{1}{2} \cdot x_1^2; \phi := -x_1;$$

$$V := \frac{1}{2} x_1^2$$

$$\phi := -x_1$$

$$> u := \text{strict}(f, g, h, s, V, \phi);$$

$$u := -6 z_1 - 6 x_1 + 3 x_1^3 - 3 z_2 + 3 z_1 x_1^2$$

$$> u := \text{subs}(\{x_1 = x(t), z_1 = z1(t), z_2 = z2(t)\}, u);$$

$$u := -6 z1(t) - 6 x(t) + 3 x(t)^3 - 3 z2(t) + 3 x(t)^2 z1(t)$$

$$> \text{sys3} := \{ \text{diff}(x(t), t) = z1(t), \text{diff}(z1(t), t) = x(t) - x(t)^3 + z2(t), \\ \text{diff}(z2(t), t) = u \};$$

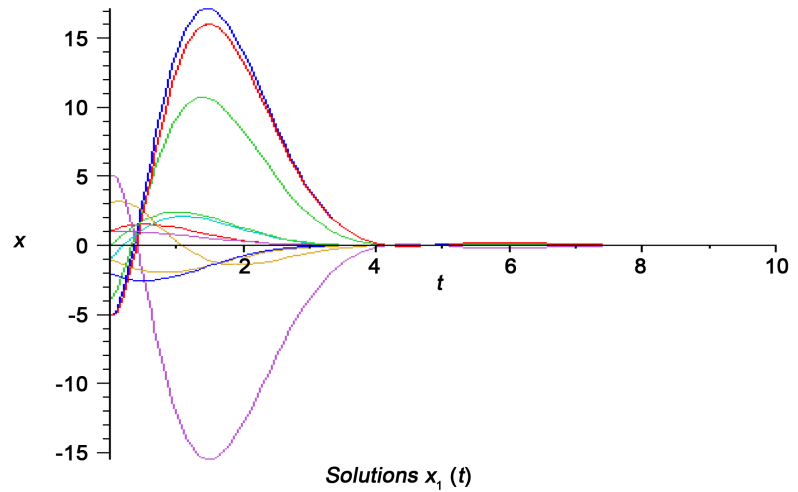
$$\text{sys3} := \left\{ \frac{d}{dt} z2(t) = -6 z1(t) - 6 x(t) + 3 x(t)^3 - 3 z2(t) \right. \\ \left. + 3 x(t)^2 z1(t), \frac{d}{dt} z1(t) = x(t) - x(t)^3 + z2(t), \frac{d}{dt} x(t) \right. \\ \left. = z1(t) \right\}$$

$$> b := \text{sgraph}(\text{sys3}, 11);$$

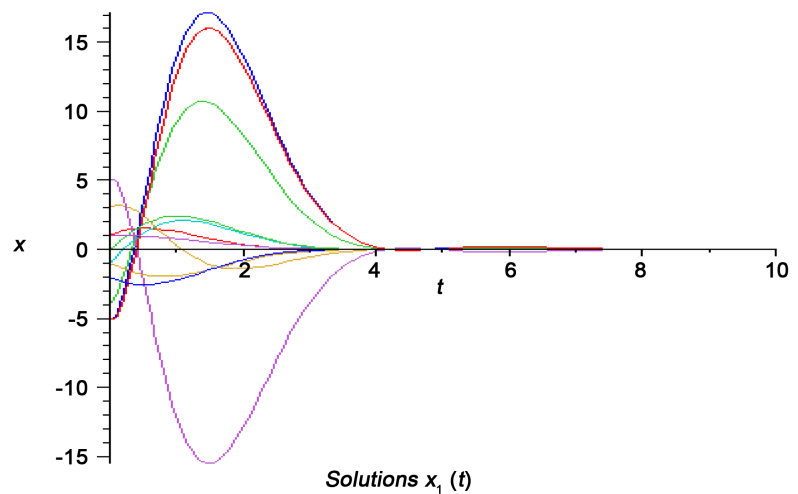
$$> b[1]; b[2]; b[3]; \text{ (Figure 3)}$$

#### 4. Conclusions

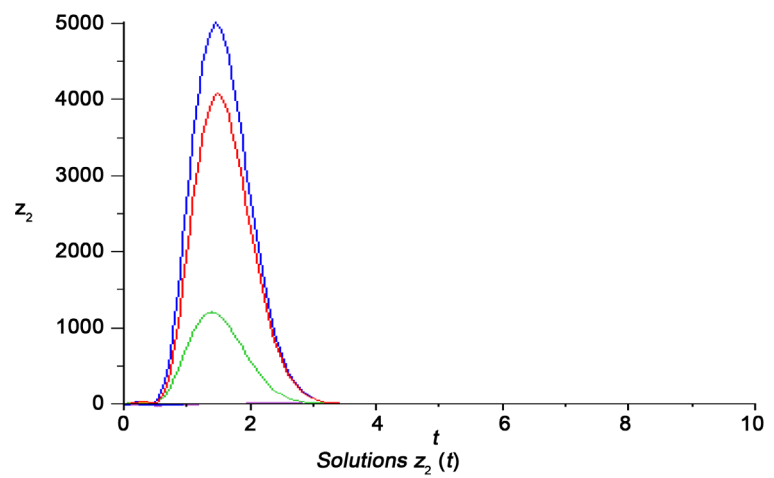
This paper shows the advantages of computer algebra systems like Maple in computation of nonlinear systems. Maple is successfully used to compute back-stepping control laws of nonlinear strict-feedback systems. The stability of systems with computed control laws is also verified by Maple procedures. As demonstrated by other researchers, computer algebra can be used for computing control laws of other systems with back-stepping. In general, it can be used to solve computation problems in science and engineering.



b[1]



b[2]



b[3]

**Figure 3.** The “portrait” procedure generates a sequence of graphs b[1], b[2], and b[3] for phase portrait of nonlinear control system (7).

We hope that more researchers, engineers, and students will take advantage of computer algebra systems as a tool in problem solving. Computer algebra systems should play more important roles in industry, research, and education.

## References

- [1] de Jager, B. (1995) The Use of Symbolic Computation in Nonlinear Control: Is It Viable? *IEEE Transactions on Automatic Control*, **40**, 84-89. <http://dx.doi.org/10.1109/9.362897>
- [2] Wang, R.D. (1993) Symbolic Computation Approach for Absolute Stability. *International Journal of Control*, **58**, 495-502. <http://dx.doi.org/10.1080/00207179308923013>
- [3] Wang, R., Woo, P. and Kwong, M.K. (1992) Using Maple for Symbolic Computations in Robotics, with. *International Journal of Robotics and Automation*, **7**, 41-49.
- [4] Yanami, H. and Anai, H. (2007) The Maple Package SyNRAC and Its Application to Robust Control Design. *Future Generation Computer Systems*, **23**, 721-726. <http://dx.doi.org/10.1016/j.future.2006.10.009>
- [5] Ahangar, R.R. and Wang, R. (2010) Symbolic Computation for Integrator Back-Stepping Control Laws. *The Journal of Combinatorial Mathematics and Combinatorial Computing (JCMCC)*, **74**, 33-42.
- [6] Pozo, F. and Ikhouane, F. (2006) Computing Adaptive Back-Stepping Control Law Using Computer Algebra. *Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design*, Munich, 4-6 October 2006, 403-407.
- [7] Kokotovic, P. and Sussmann, H. (1989) A Positive Real Condition for Global Stabilization of Nonlinear Systems. *Systems & Control Letters*, **13**, 125-133. [http://dx.doi.org/10.1016/0167-6911\(89\)90029-7](http://dx.doi.org/10.1016/0167-6911(89)90029-7)
- [8] Khalil, H.K. (2002) *Nonlinear Systems*. 3rd Edition, Prentice Hall, Upper Saddle River.
- [9] Krstic, M., Kanellakopoulos, I. and Kokotovic, P. (1995) *Nonlinear and Adaptive Control Design*. John Wiley, New York.
- [10] Kanellakopoulos, I., Kokotovic, P. and Morse, A. (1992) A Toolkit for Nonlinear Feedback Design. *Systems & Control Letters*, **18**, 83-92. [http://dx.doi.org/10.1016/0167-6911\(92\)90012-H](http://dx.doi.org/10.1016/0167-6911(92)90012-H)



Open Access Library

**Submit or recommend next manuscript to OALib Journal and we will provide best service for you:**

- Publication frequency: Monthly
- 9 [subject areas](#) of science, technology and medicine
- Fair and rigorous peer-review system
- Fast publication process
- Article promotion in various social networking sites (LinkedIn, Facebook, Twitter, etc.)
- Maximum dissemination of your research work

Submit Your Paper Online: [Click Here to Submit](#)

Or Contact [service@oalib.com](mailto:service@oalib.com)