EE5104 Adaptive Control Systems

AY2021/2022 Semester 2

Continuous Assessment 2

Philippe Andrin Brigger
A0248834U

Contents

1 Introduction

This *CA2* report is part of the spring 2022 module EE5104 Adaptive Control Systems at the National University of Singapore (NUS). Sliding control, also known as variable structure control, is a nonlinear control method that can be used to control nonlinear systems. The approach makes a system converge to a switching surface. The system then slides along this surface to converge to zero. Some advantages of sliding control are that it is insensitive to uncertain parameters and external disturbances and converges in finite time. One problem which carefully must be dealt with when using sliding control is the chattering effect. The first part of the report details the design process of the sliding control system. In particular, it is shown that the output error of the system converges to zero and that the system is insensitive to uncertain parameters. Next, the report discusses the simulation results of the proposed controller. Finally, the appendix contains the code used to run the simulations. In general, all vectors are printed in bold while scalars and matrices are printed as normal text.

2 Sliding Control Design

In this continuous assignment, a nonlinear sliding controller is designed. A sliding controller is capable of controlling nonlinear systems. In particular, the controller makes the system converge to zero in finite time and is insensitive to uncertain parameters. In the following, these properties will be shown.

2.1 Switching Surface

The system to be controlled is

$$\begin{aligned}\dot{x}_1 &= ax_1 + bu + d \\ \dot{x}_2 &= x_1\end{aligned} \tag{1}$$

The two states of the system are $x_1 = \frac{dy}{dt}$ and $x_2 = y$. The goal of the controller is to make the state vector $\boldsymbol{x}$ converge to zero asymptotically stable from any initial conditions. To do this, the system defined in eq. (1) is rewritten in state space form.

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}a & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}b\\ 0\end{bmatrix}u + \begin{bmatrix}1\\ 0\end{bmatrix}d \\[2ex] \boldsymbol{y} &= \begin{bmatrix}\dot{y}\\ y\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \\[2ex] \dot{\boldsymbol{x}} &= A\boldsymbol{x} + Bu + Dd \\ \dot{\boldsymbol{y}} &= C\boldsymbol{x}\end{aligned} \tag{2}$$

Additionally, the switching surface $\sigma(x)$ is defined as

$$\sigma(\boldsymbol{x}) = c_1x_1 + c_2x_2 = c_1\frac{dy}{dt} + c_2y \tag{3}$$

The controller design must ensure that $\boldsymbol{x} = 0$ is an asymptotically stable equilibrium point. Therefore, the switching surface $\sigma(\boldsymbol{x}) = 0$ should be asymptotically stable. The state vector will then converge to zero if the switching surface also converges to zero. The switching surface $\sigma(\boldsymbol{x}) = 0$ is asymptotically stable when the polynomial $C(s) = c_1s + c_2 = 0$ has all its roots in the left-half plane. This is the case if c_1 and c_2 have the same sign

- $C(s)$ asymptotically stable: $Re(s) = \text{-}Re(\frac{c_2}{c_1}) < 0$ iff $sign(c_1) = sign(c_2)$

To make sure the state vector $\boldsymbol{x}$ converges to zero, it must be shown that the switching surface σ also converges to zero.

2.2 Lyapunov's Direct Method

One approach to prove that an equilibrium point is asymptotically stable is with Lyapunov's Direct Method. Lyapunov's Direct Method says that if a scalar valued function $V(\boldsymbol{x}, t)$ of a system's state variables $\boldsymbol{x}$ satisfies certain conditions, then the equilibrium point of the (nonlinear) system will be globally asymptotically stable[1]. In other words, if a Lyapunov function $V(\sigma, t)$ can be found, then the switching surface σ will have an asymptotically stable equilibrium point at $\sigma = 0$ and converge to zero. By choosing the coefficients of eq. (3) to make the polynomial $\sigma(\boldsymbol{x}) = 0$ asymptotically stable, if σ converges to zero then the state vector $\boldsymbol{x}$ will converge to zero too. The convergence will happen regardless of the initial conditions since both

σ has a globally asymptotically stable equilibrium point at zero and the polynomial $\sigma(\boldsymbol{x}) = 0$ is asymptotically stable. The four conditions that a Lyapunov function $V(\sigma, t)$ must satisfy for an equilibrium point to be globally asymptotically stable are:

- $V(\sigma, t)$ is positive-definite
- $V(\sigma, t)$ is decrescent
- $V(\sigma, t)$ is radially unbounded
- $\dot{V}(\sigma, t)$ is negative definite

The proposed Lyapunov function is

$$V(\sigma, t) = \frac{\sigma^2}{2} \tag{4}$$

The Lyapunov function defined in eq. (4) is positive definite as σ^2 gives a positive scalar

- $V(\sigma), t)$ positive-definite: $V(\sigma, t) > 0 \,\forall\, \sigma(\boldsymbol{x}), t$ because $\sigma^2(\boldsymbol{x}) > 0 \,\forall\, \boldsymbol{x}, t$

The function is also radially unbounded as

- $V(\sigma, t)$ radially unbounded: $V(\sigma) \to \infty$ as $\|\sigma(\boldsymbol{x})\| \to \infty$

The time derivative of $V(\sigma, t)$ is calculated to determine whether $\dot{V}(\sigma, t)$ is negative definite

$$\frac{dV}{dt} = \sigma(x)\dot{\sigma}(x) = \sigma(x)\boldsymbol{p}^T\dot{\boldsymbol{x}} = \sigma(\boldsymbol{x})(\boldsymbol{p}^T A\boldsymbol{x} + \boldsymbol{p}^T Bu(t))$$

$\dot{V}(\sigma, t)$ will be negative definite if the control input $u(t)$ is chosen appropriately

$$u(t) = -\frac{\boldsymbol{p}^T A\boldsymbol{x}}{\boldsymbol{p}^T B} - \frac{\mu}{\boldsymbol{p}^T B} sign(\sigma(\boldsymbol{x})) \tag{5}$$

Inserting eq. (5) into $\dot{V}(\sigma, t)$ yields

- $\dot{V}(\sigma, t)$ negative definite: $\dot{V}(\sigma, t) = -\mu\sigma(\boldsymbol{x}) sign(\sigma(\boldsymbol{x})) < 0 \,\forall\, \boldsymbol{x}, t$

If the control gain μ is chosen high enough, the Lyapunov function will stay negative definite even if the estimated system parameters $\hat{A}$ and $\hat{B}$ don't perfectly match the true system parameters A and B, as the summation $\boldsymbol{p}^T A\boldsymbol{x} + \boldsymbol{p}^T Bu(t)$ will remain negative

$$\boldsymbol{p}^T A\boldsymbol{x} + \boldsymbol{p}^T Bu(t) = \frac{\boldsymbol{p}^T(A\boldsymbol{x}\hat{B}^T - \hat{A}\boldsymbol{x}B^T)}{\boldsymbol{p}^T\hat{B}} - \mu\frac{\boldsymbol{p}^T B}{\boldsymbol{p}^T\hat{B}} sign(\sigma) < 0 \text{ if } \mu \text{ large}$$

Since the Lyapunov function is negative definite, the function will never be larger than its starting value V_0. Therefore, the function implicitly must be decrescent

- $V(\sigma, t)$ decrescent: $V(\sigma, t) \leq V_0 \;\forall\, \boldsymbol{x}, t$

As a consequence, $\|\sigma(\boldsymbol{x})\|$ and $\|\boldsymbol{x}\|$ remain bounded.

2.3 Conclusion

To conclude, since the function in eq. (4) is a Lyapunov function, the switching surface will converge to zero regardless of the initial conditions. Furthermore, since the polynomial $\sigma(\boldsymbol{x}) = 0$ is designed as an asymptotically stable polynomial, the state vector $\boldsymbol{x}$ will converge to zero once the switching surface has been reached. The controller will also work with uncertain parameters if μ is chosen large enough. To summarize, the sliding controller proposed in this section fulfills the requirements to our system

- Zero steady state error
- $\boldsymbol{x} = 0$ is a globally asymptotically stable equilibrium point
- Control law $u(t)$ is insensitive to uncertain parameters
- Control input: $u(t) = -\frac{\boldsymbol{p}^T A\boldsymbol{x}}{\boldsymbol{p}^T B} - \frac{\mu}{\boldsymbol{p}^T B} sign(\sigma(\boldsymbol{x}))$

Since the control law proposed in eq. (5) is insensitive to uncertain parameters, this also means that the error convergence of the system will not be affected by an external disturbance d.
One drawback of the control law proposed in eq. (5) is that the $sign()$ function is prone to chatter. If for instance a motor is being controlled, this could mean that the motor wildly changes from spinning forwards and backwards because the $sign()$ function continuously jumps back and forth due to the chatter. This behavior is highly undesirable and may lead to damaging or destroying the hardware prematurely. One way to deal with this problem is to use a smoother relay function. One such function is the saturation function

$$sat(\sigma, \epsilon) = \begin{cases} 1 & \sigma > \epsilon \\ \frac{\sigma}{\epsilon} & -\epsilon \leq \sigma \leq \epsilon \\ -1 & \sigma < -\epsilon \end{cases} \tag{6}$$

The control input with a saturation relay function will be

$$u(t) = -\frac{\boldsymbol{p}^T A\boldsymbol{x}}{\boldsymbol{p}^T B} - \frac{\mu}{\boldsymbol{p}^T B} sat(\sigma(\boldsymbol{x}), \epsilon) \tag{7}$$

The choices of the control gain μ, the coefficients of the switching surface c_1, c_2 and the initial system states x_1^0, x_2^0 determine how quickly the system will converge to zero. The system will converge in two steps. First, the system will converge to the switching surface ($\sigma(\boldsymbol{x}) = 0$) and then slide along the switching surface to the origin ($\boldsymbol{x} = 0$). The convergence of the system to the switching surface t_σ is

$$t_\sigma = \frac{\|\sigma_0\|}{\mu} = \frac{\left\|c_1 x_1^0 + c_2 x_2^0\right\|}{\mu} \tag{8}$$

Once the switching surface is reached, the system slides along this surface to converge to the origin. Therefore, after t_σ, $\sigma(\boldsymbol{x})$ remains zero at all times. As a result, $sign(\sigma(\boldsymbol{x}))$ and $sat(\sigma(\boldsymbol{x}), \epsilon)$ remain zero. Thus, the convergence time from $\sigma(\boldsymbol{x}) = 0$

to $\boldsymbol{x} = 0$ may be calculated as follows.

$$
\begin{aligned}
\dot{x}_1 &= ax_1 + bu \\
&= ax_1 - b\frac{\boldsymbol{c}^T A\boldsymbol{x}}{\boldsymbol{c}^T B} \\
&= ax_1 - b\frac{\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} a & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}{\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} b \\ 0 \end{bmatrix}} \\
&= ax_1 - b\frac{\begin{bmatrix} ac_1 + c_2 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}{c_1 b} \\
&= ax_1 - ax_1 - \frac{c_2}{c_1}x_1 \\
\Rightarrow \dot{x}_1 &= -\frac{c_2}{c_1}x_1 \quad \text{for } t > t_\sigma
\end{aligned} \tag{9}
$$

Applying the Laplace transform and then inverse Laplace transform on eq. (9) yields

$$
\begin{aligned}
se^{t_\sigma s}X_1(s) - x_1(t_\sigma) &= -e^{t_\sigma s}\frac{c_2}{c_1}x_1 \\
\Rightarrow X_1(s) &= \frac{x_1(t_\sigma)e^{-t_\sigma}}{s+1} \\
\Rightarrow x_1(t) &= -x_1(t_\sigma)e^{-(t-t_\sigma)}
\end{aligned} \tag{10}
$$

Equation (10) and eq. (2) can be used to find the convergence of $x_2(t)$ once the switching surface has been reached

$$
\begin{aligned}
\dot{x}_2 &= x_1 \\
\Rightarrow x_2(t) &= x_1(t_\sigma)e^{-(t-t_\sigma)} + C
\end{aligned} \tag{11}
$$

To find C, the value of $x_2(t_\sigma)$ and $x_1(t_\sigma)$ must be known. These values are defined by the convergence of the states to the switching surface. The dynamics are found by applying the Laplace transform to the state equations from eq. (2).

$$
\begin{aligned}
sX_1(s) - x_1(0) &= -\frac{c_2}{c_1}X_1(s) - \frac{\mu}{\boldsymbol{p}^T B}\frac{sign(\sigma(\boldsymbol{x}))}{s} \\
sX_1(s) - x_1(0) &= -\frac{c_2}{c_1}X_1(s) - \frac{\mu}{c_1 bs}sign(\sigma(\boldsymbol{x})) \\
X_1(s) &= \frac{x_1(0)}{s + \frac{c_2}{c_1}} - \frac{\frac{\mu}{c_1 b}sign(\sigma(\boldsymbol{x}))}{s(s + \frac{c_2}{c_1})}
\end{aligned}
$$

Applying the Laplace transform to the above yields

$$
x_1(t) = x_1(0)e^{-\frac{c_2}{c_1}t} - \frac{\mu}{c_1 b}sign(\sigma(\boldsymbol{x}))\big(\frac{c_1}{c_2} - \frac{c_1}{c_2}e^{-\frac{c_2}{c_1}t}\big) \tag{12}
$$

From here, $x_2(t)$ can be retrieved

$$
\begin{aligned}
x_2(t) &= \int_0^t x_1(t)dt \\
&= -x_1(0)\frac{c_1}{c_2}e^{-\frac{c_2}{c_1}t} - \frac{\mu}{c_1 b}\frac{c_1}{c_2}sign(\sigma(\boldsymbol{x}))\big(t + \frac{c_1}{c_2}e^{-t}\big) \\
&\quad + (x_2(0) + x_1(0) + \frac{\mu}{c_1 b}\big(\frac{c_1}{c_2}\big)^2 sign(\sigma(\boldsymbol{x})))
\end{aligned} \tag{13}
$$

To determine the constant C from eq. (11), t_σ can be inserted in the above equation. As can be seen from eq. (10) and eq. (11), the convergence of the state vector to zero depends on the convergence time to the switching surface. Therefore, the state vector convergence also depends on the choice of μ, c_1, c_2 and initial states x_1^0, x_2^0. Therefore, if the system is perturbed less (initial states close to zero) and the control gain μ is high, the system will converge to zero quicker.

3 Simulation Results

The following parameters were used to define the system, switching surface and controller outlined in section 2.

- $a = 2$, $b = 1$
- $c_1 = 1$, $c_2 = 1$
- $d = 0.9 \cdot \sin(628t)$
- $\mu = 0.5$, $\epsilon = 0.01$

The simulation results for initial states $x_1^0 = 5$ and $x_2^0 = -3$ are shown in fig. 1. It can be observed that both states converge to zero. In a first phase, the states converge to the switching surface. The switching surface, where $\sigma(\boldsymbol{x}) = 0$, is reached after $t_\sigma = \frac{\|\sigma_0\|}{\mu} = 4\,\mathrm{s}$. This time can also be observed in the system states plot in fig. 1 as the derivative of $x_1(t)$ flips its sign. Additionally, the saturation relay function is a lot less noisy than the sign relay function control input. However, in both cases the disturbance acting on the control input leads to a largely fluctuating signal. Finally, the phase plot in fig. 2 shows that the system states converge to zero with different initial states. It can nicely be seen that the convergence takes place in two steps. First, the states converge to the switching surface. After this, the states slide along the switching surface to the origin. In the end, both states have converged to zero.

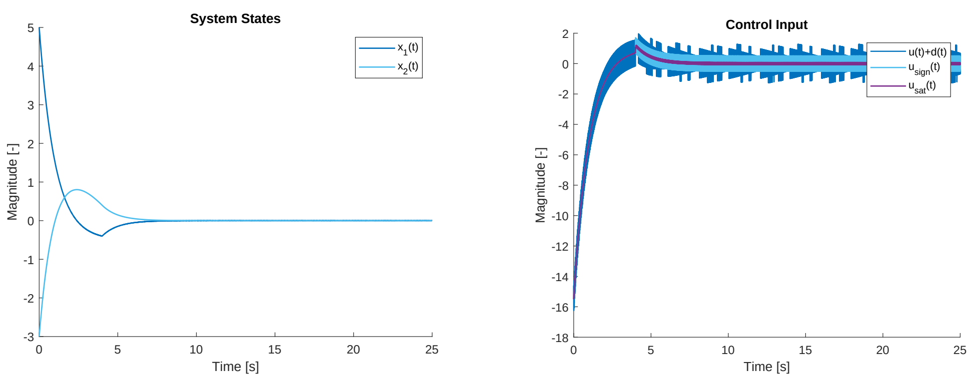


Figure 1: State convergence and control input signal for $\mu = 0.5$ and initial states $x_1^0 = 5$, $x_2^0 = -3$.

The control gain μ can be varied to tweak the system performance. The simulation results for varying μ and for initial states $x_1^0 = 5$ and $x_2^0 = -3$ are shown in fig. 3. It can be observed that both states converge to zero. The convergence is much quicker with a higher control gain. However, a higher control gain also increases the relay chatter as can be seen in the control input plot of fig. 3. Therefore, if a high control gain is used, it is important to design a smooth relay function. Otherwise the chatter will quickly damage the system hardware.

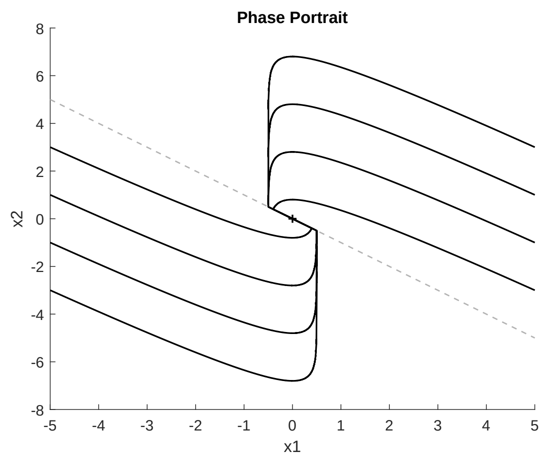


Figure 2: The system states x_1 and x_2 converge to the origin for different initial states.

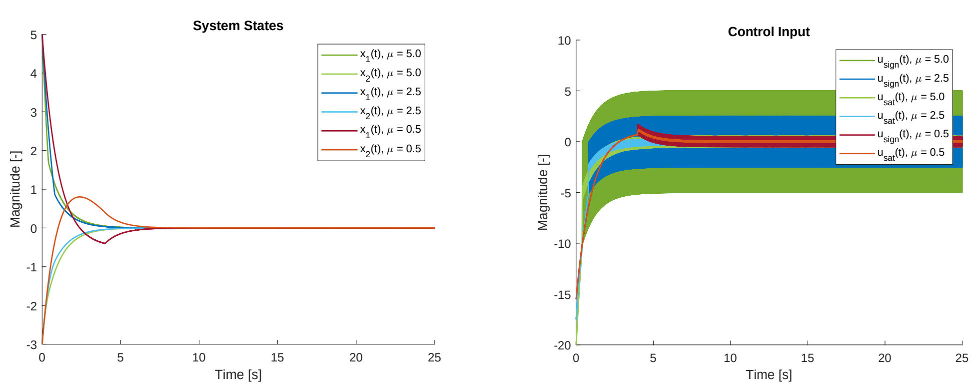


Figure 3: State convergence and control input signal for different control gains μ.

4 Conclusion

The sliding control law proposed in Section 2 makes the states $\boldsymbol{x}$ of the system converge to zero. The sliding controller ensures that $\boldsymbol{x} = 0$ is a global asymptotically stable equilibrium point. Additionally, if appropriate control gains are chosen, the system is insensitive to uncertain system parameters. The simulation results presented in Section 3 show that the states converge to zero in two steps. First, the states converge to the switching surface ($\sigma(\boldsymbol{x}) = 0$). Then, the states slide along the switching surface until they reach the equilibrium point. The phase portrait shows that the initial states do not influence the error convergence. The simulation results also showed that the choice of relay function makes the control input more or less prone to chatter. Finally, the choice of control gain μ determines how quickly the system converges to zero.

References

[1] Khalil, H. (2002). *Nonlinear Systems.* Pearson Education. Prentice Hall.

A Appendix

The simulations were conducted with Simulink and Matlab. The sliding controller was designed with Simulink. The CA2 Matlab live script defines all the simulation parameters. To automate simulation results plotting, two functions were defined in the *functionsContainer* script.

CA2 Sliding Control

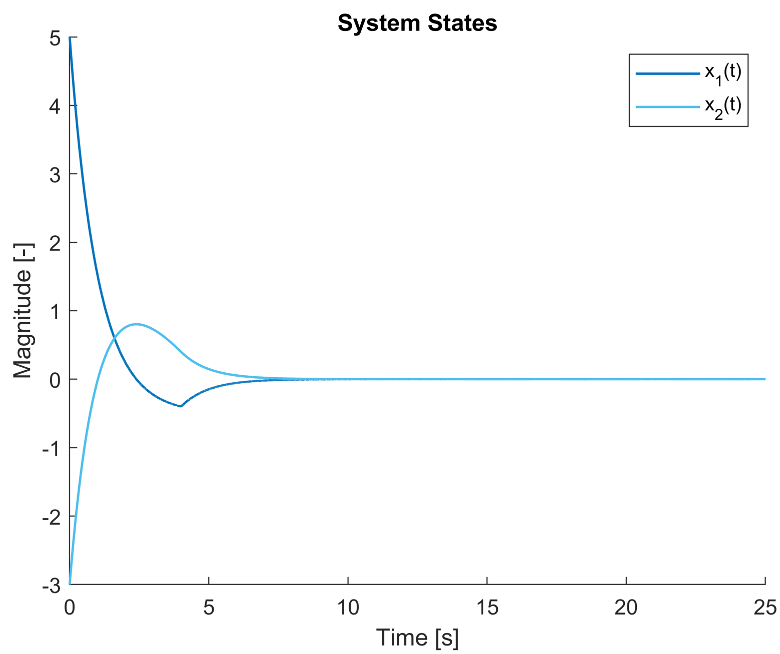
In this CA a sliding controller is designed for a linear system with two states. The controller ensures that x = 0 is an asymptotically stable solution.

```
% Clean up
close all; clear all; clc;

% Plant parameters
a = 2; b = 1;
% State space matrices
A = [2 0; 1 0];
B = [1; 0];
C = [1 0; 0 1];
D = [0; 0];

% Disturbance parameters
d_max = 0.9; d_freq = 628;

% Controller parameters
% Switching surface parameters
c1 = 1; c2 = 1;
p = [c1; c2];
p_T = p';
mu = 0.5; epsilon = 0.01;
```

Run simulations and plot results for default values

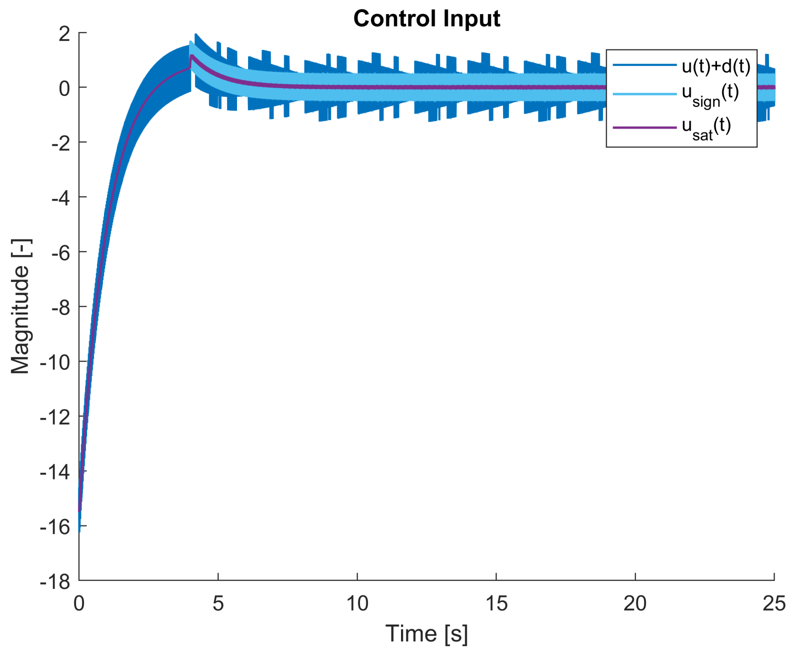
```
close all;
simTime = 25; % [s]
initial_states = [5, -3.0]; % [x1_0, x2_0]
% Plot colors
blue = [0 0.4470 0.7410]; red = [0.6350 0.0780 0.1840]; orange = [0.8500 0.3250 0.0980];
black = [0 0 0]; grey = [0.7 0.7 0.7]; green = [0.4660 0.6740 0.1880]; lgreen = [0.6 0.8, 0.3];
lblue = [0.3010 0.7450 0.9330]; purple = [0.4940 0.1840 0.5560];
out = sim("CA2_Simulink_Model.slx");

myObj = functionsContainer;
fig_num = 1;
t = out.tout;
% Plot states for default parameters
myObj.plotMe(t,[out.yout{1}.Values.Data(:,1),out.yout{1}.Values.Data(:,2)],'default_states.pdf'
    [{'x_1(t)'},{'x_2(t)'}],[blue,lblue],['- ','- '],[1.1 1.1],'Time [s]','Magnitude [-]',...
    'System States',fig_num); fig_num = fig_num+1;
```

1



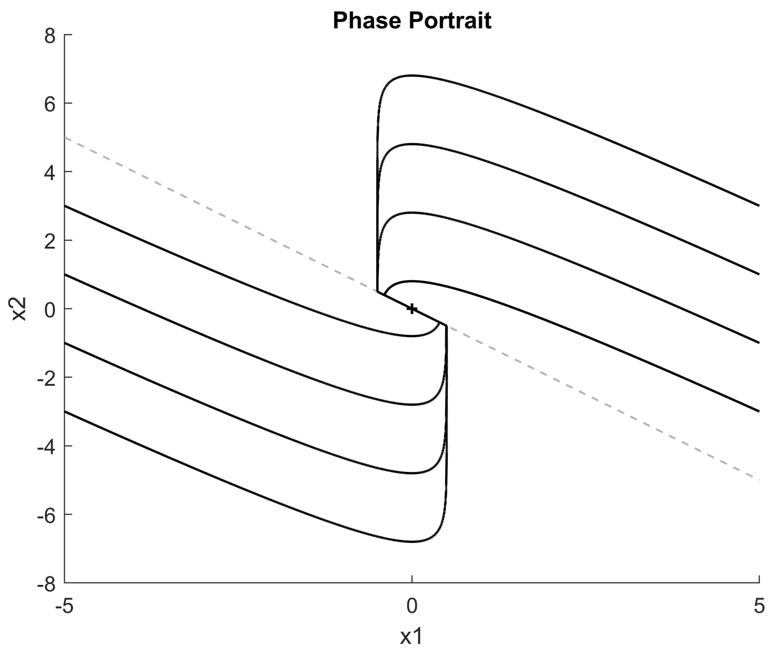

```
% Plot control input for default parameters with sign() and sat()
myObj.plotMe(t,[out.yout{1}.Values.Data(:,4),out.yout{1}.Values.Data(:,3),out.yout{2}.Values.Da
    'default_input.pdf',[{'u(t)+d(t)'},{'u_{sign}(t)'},{'u_{sat}(t)'}],[blue,lblue,purple],['-
    [1 1.1 1.1],'Time [s]','Magnitude [-]','Control Input',fig_num); fig_num = fig_num+1;
```

2



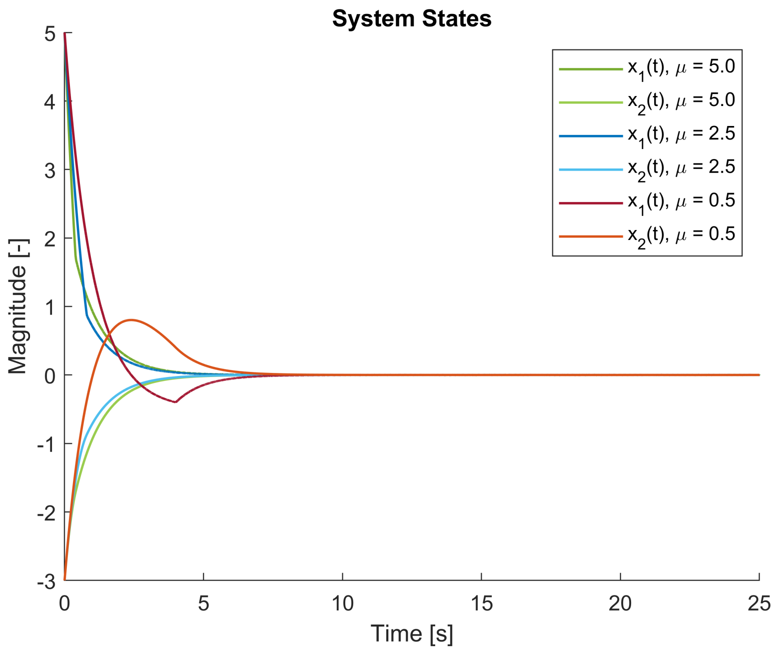

```
% Phase portrait with sign()
% Simulate different initial conditions
initial_states = [5, -1.0]; out2 = sim("CA2_Simulink_Model.slx");
initial_states = [5, +1]; out3 = sim("CA2_Simulink_Model.slx");
initial_states = [5, 3]; out4 = sim("CA2_Simulink_Model.slx");
initial_states = [-5, -3.0]; out5 = sim("CA2_Simulink_Model.slx");
initial_states = [-5, -1.0]; out6 = sim("CA2_Simulink_Model.slx");
initial_states = [-5, +1]; out7 = sim("CA2_Simulink_Model.slx");
initial_states = [-5, 3]; out8 = sim("CA2_Simulink_Model.slx");
myObj.phasePortrait([linspace(-5,5,length(out.yout{1}.Values.Data(:,1)))',out.yout{1}.Values.Da
     out2.yout{1}.Values.Data(:,1),out3.yout{1}.Values.Data(:,1),out4.yout{1}.Values.Data(:,1),
     out5.yout{1}.Values.Data(:,1),out6.yout{1}.Values.Data(:,1),out7.yout{1}.Values.Data(:,1),
     out8.yout{1}.Values.Data(:,1)],[-linspace(-5,5,length(out.yout{1}.Values.Data(:,1)))',...
     out.yout{1}.Values.Data(:,2),out2.yout{1}.Values.Data(:,2),out3.yout{1}.Values.Data(:,2),
     out4.yout{1}.Values.Data(:,2),out5.yout{1}.Values.Data(:,2),out6.yout{1}.Values.Data(:,2),
     out7.yout{1}.Values.Data(:,2),out8.yout{1}.Values.Data(:,2)],'default_phase_portrait.pdf',
     [grey,black,black,black,black,black,black,black,black],['--','- ','- ','- ','- ','- ','- 
     [0.9 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1],'x1','x2','Phase Portrait',fig_num); fig_num = fig_n
```

3

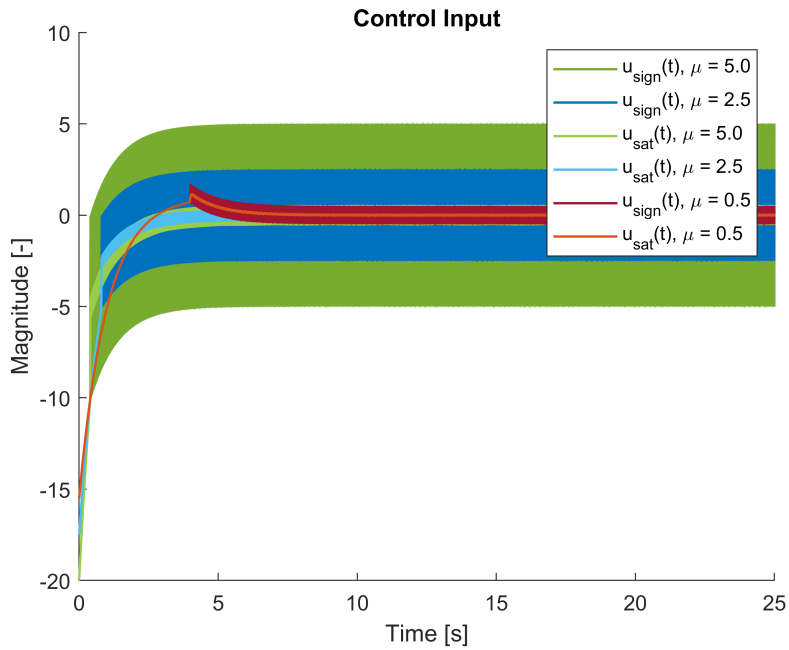


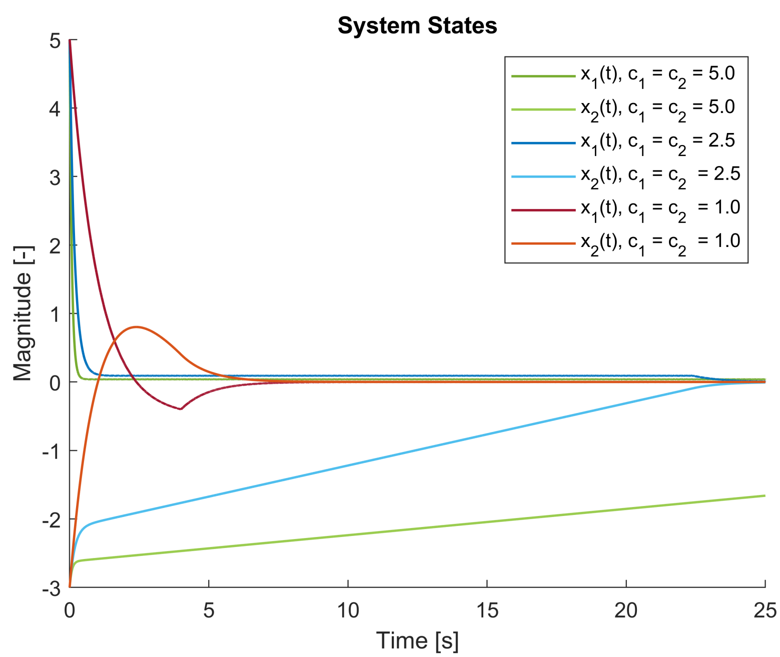
Run simulations and plot results for different gains

```
% Vary mu
initial_states = [5, -3.0];
mu = 2.5; out_mu25 = sim("CA2_Simulink_Model.slx");
mu = 5.0; out_mu50 = sim("CA2_Simulink_Model.slx");

% Plot states for varying mu
myObj.plotMe(t,[out_mu50.yout{1}.Values.Data(:,1),out_mu50.yout{1}.Values.Data(:,2),out_mu25.yo
    out_mu25.yout{1}.Values.Data(:,2),out.yout{1}.Values.Data(:,1),out.yout{1}.Values.Data(:,2
    [{'x_1(t), \mu = 5.0'},{'x_2(t), \mu = 5.0'},{'x_1(t), \mu = 2.5'},{'x_2(t), \mu = 2.5'},..
    {'x_1(t), \mu = 0.5'},{'x_2(t), \mu = 0.5'}],[green,lgreen,blue,lblue,red,orange],['- ','-
    [1.1 1.1 1.1 1.1 1.1 1.1],'Time [s]','Magnitude [-]','System States',fig_num); fig_num = f
```

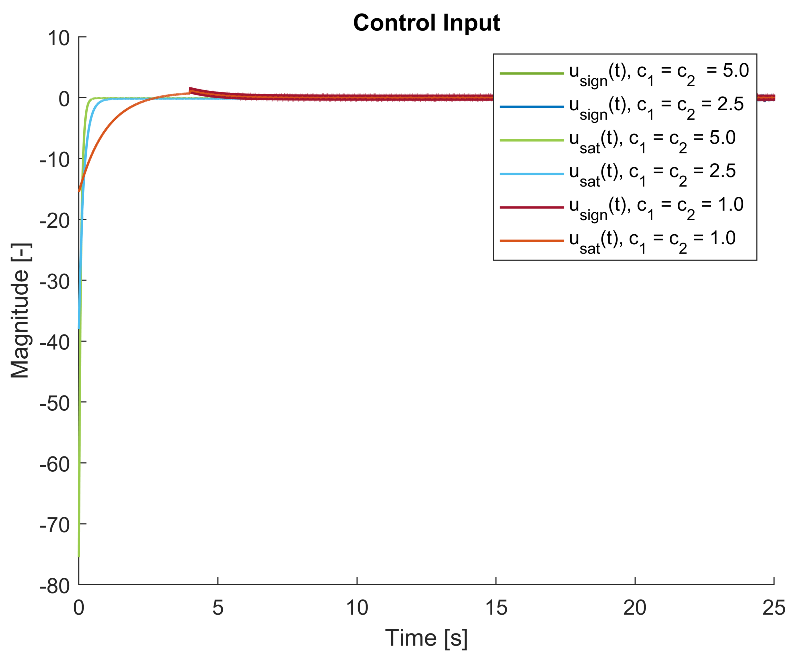



```
% Plot control input for varying mu with sign() and sat()
myObj.plotMe(t,[out_mu50.yout{1}.Values.Data(:,3),out_mu25.yout{1}.Values.Data(:,3), ...
    out_mu50.yout{2}.Values.Data(:,3),out_mu25.yout{2}.Values.Data(:,3), ...
    out.yout{1}.Values.Data(:,3),out.yout{2}.Values.Data(:,3)],...
    'mu_input.pdf',[{'u_{sign}(t), \mu = 5.0'},{'u_{sign}(t), \mu = 2.5'},{'u_{sat}(t), \mu = 
    {'u_{sat}(t), \mu = 2.5'},{'u_{sign}(t), \mu = 0.5'},{'u_{sat}(t), \mu = 0.5'}]...
    ,[green,blue,lgreen,lblue,red,orange],['- ','- ','- ','- ','- ','- '],[1.1 1.1 1.1 1.1 1.1
    'Time [s]','Magnitude [-]','Control Input',fig_num); fig_num = fig_num+1;
```


Control Input
10
5
0
-5
-10
-15
-20
0
5
10
15
20
25
Time [s]
Magnitude [-]
u_sign(t), μ = 5.0
u_sign(t), μ = 2.5
u_sat(t), μ = 5.0
u_sat(t), μ = 2.5
u_sign(t), μ = 0.5
u_sat(t), μ = 0.5


System States
5
4
3
2
1
0
-1
-2
-3
0
5
10
15
20
25
Time [s]
Magnitude [-]
x_1(t), c_1 = c_2 = 5.0
x_2(t), c_1 = c_2 = 5.0
x_1(t), c_1 = c_2 = 2.5
x_2(t), c_1 = c_2 = 2.5
x_1(t), c_1 = c_2 = 1.0
x_2(t), c_1 = c_2 = 1.0

6


Control Input
$u_{sign}(t)$, $c_1 = c_2 = 5.0$
$u_{sign}(t)$, $c_1 = c_2 = 2.5$
$u_{sat}(t)$, $c_1 = c_2 = 5.0$
$u_{sat}(t)$, $c_1 = c_2 = 2.5$
$u_{sign}(t)$, $c_1 = c_2 = 1.0$
$u_{sat}(t)$, $c_1 = c_2 = 1.0$
Magnitude [-]
Time [s]
10
0
-10
-20
-30
-40
-50
-60
-70
-80
0
5
10
15
20
25

7