Robust Control of Nonlinear Systems

TAMZIDUL MINA

SMART LAB TECHNICAL SEMINAR

SPRING 2020

Robust Control of Nonlinear Systems

Weekly Plan

Week 1 (today)

- Nonlinear Systems
- Robust Control
- Sliding Mode Control (Theoretical)

Week 2

- Example problem control formulation with implementation notes (Applied)
- Introduction to and MATLAB/SIMULINK implementation

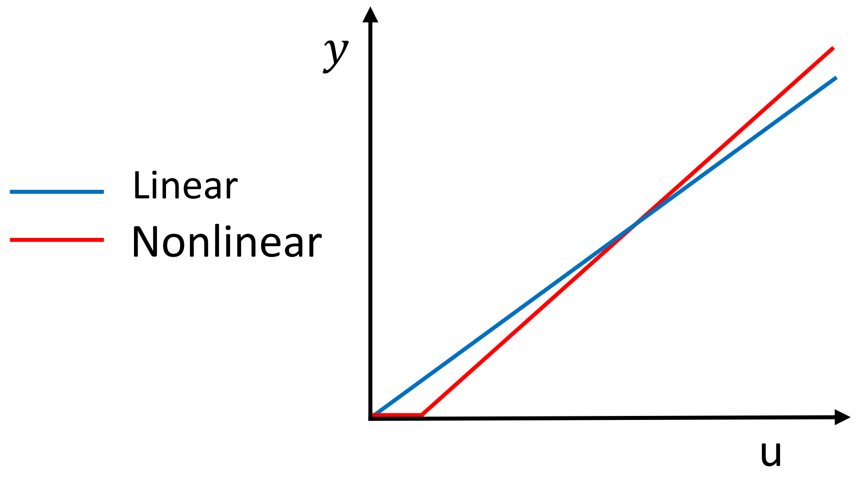
Week 1

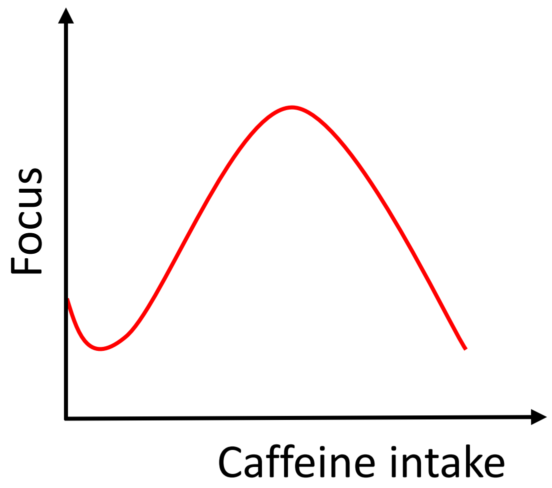
Linear vs. Nonlinear Systems

Linear systems follow the Superposition principle and are Homogeneous.

$$F(u_1) = y_1$$
$$F(u_2) = y_2$$
$$F(u_1 + u_2) = y_1 + y_2$$

$$F(u) = y$$
$$F(au) = ay$$



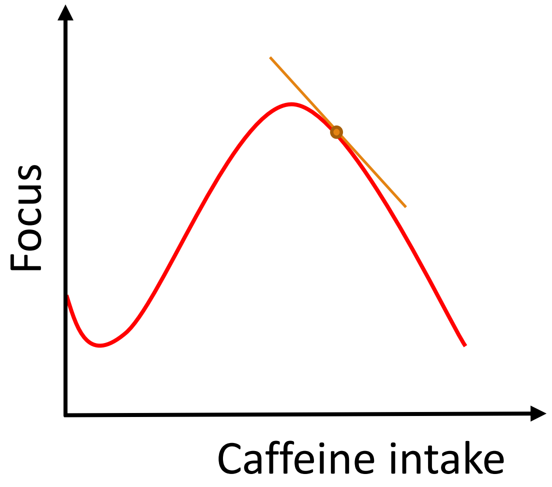


Nonlinear Systems <u>do not</u> follow the Superposition principle nor are homogeneous.

Most systems in real life are nonlinear !

Control Strategies

Linear Control of Linearized Nonlinear Systems



Nonlinear Systems maybe **linearized** around operating points to apply linear control methods.

However,

- Inefficient control input
- Performance not as good as it could be
- Fixed control law cannot accommodate system if system state goes outside the operating region

Nonlinear Control of Nonlinear Systems

Nonlinear Control Methods:

- Feedback Linearization
- Backstepping Control
- Sliding Mode Control

Nonlinear Control Analysis tools:

- Lyapunov methods
- Limit Cycles
- Describing Functions

However, each one has its own advantages and disadvantages.

Nonlinear Control Theory still an active field of research !

Uncertainty and unmodeled dynamics in Nonlinear Systems

Importance of controller robustness

Examples:

UUV underwater pipeline inspection

<u>Imperfect plant data</u> - each plant is slightly different because of the tolerances associated with individual components.

<u>Time varying plants</u> - The dynamics of some plants vary over time. A fixed control model may not accurately depict the plant at all times.

<u>Modelling error</u> - Mechanical and electrical systems are inherently complex to model. Even a simple system requires complex differential equations to describe its behavior.

<u>External disturbance</u> – changing environmental conditions change operating conditions; a fixed controller may not be able to reject disturbance effects

Nonlinear Systems with uncertainty and unmodeled dynamics

Activity

2nd order differential equation model:

$$\underbrace{A\ddot{x}}_{\text{Inertia term}} + \underbrace{B\dot{x}}_{\text{Damping term}} + \underbrace{Cx}_{\text{Restoring term}} = u$$

<u>Activity (4 min)</u>

What nonlinear systems do you deal with in your research?
List some unmodeled dynamics and uncertainties involved with your method

- you do not need equations
- think about what affects as inertia, damping, restoring force in your systems

Robust Control Theory

An approach to controller design that explicitly deals with plant uncertainty.

Designed to function properly provided that uncertain parameters or disturbances are found within some (typically compact) set.

Aim to achieve robust performance and/or stability in the presence of bounded modelling errors.

Example control methods:

- High gain feedback
- H-infinity loop shaping
- Loop transfer recovery (LTR) for Linear quadratic Gaussian (LQG) control
- Sliding Mode Control

Sliding Mode Control

Developed by [Vadim Utkin] (Russian control theorist; currently a professor at Ohio State University)

- Deals explicitly with uncertainty in its approach to controller design
- Tend to be able to cope with small differences between the true system and the nominal model used for design

A nonlinear control method that alters the dynamics of a nonlinear system by application of a discontinuous control signal (or more rigorously, a set-valued control signal) that forces the system to "slide" along a cross-section of the system's normal behavior.

The sliding-mode control scheme involves

1. Selection of a hypersurface or a manifold (i.e., the sliding surface) such that the system trajectory exhibits desirable behavior when confined to this manifold.
2. Finding feedback gains so that the system trajectory intersects and stays on the manifold.

Sliding Mode Control

Consider the n^{th} order scalar nonlinear system:

$$x^n = \underbrace{f(x, \dot{x}, .., x^{n-1}, t)}_{\underline{x}} + \underbrace{b(x, \dot{x}, .., x^{n-1}, t)}_{>0}u$$

Trajectory tracking error:

$$\tilde{x}(t) = x(t) - x_d(t)$$

Goal: $\tilde{x}(t) \to 0$

In real systems, $\boldsymbol{f}$ and $\boldsymbol{b}$ are never exactly known. Modelling errors exist.

Modeling inaccuracies can be classified into two major kinds:

- structured (or parametric) uncertainties
- unstructured uncertainties (or unmodeled dynamics)

But under operating conditions we can estimate upper and lower bounds on these parametric uncertainties.

$$|\hat{f}(\underline{x}, t) - f(\underline{x}, t)| \leq F(\underline{x}, t)$$

(Estimate) (actual) (bound)

$$|\hat{b}(\underline{x}, t) - b(\underline{x}, t)| \leq B(\underline{x}, t)$$

(Estimate) (actual) (bound)

Sliding Mode Control

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

$\dot{s}$ contains u

$s \to 0 \Rightarrow \tilde{x}(t) \to 0$

Step 1: "Replace" the nth order problem with an equivalent 1st order problem.

Create intermediate variable s, such that $s \Rightarrow \begin{cases} \dot{s} \text{ contains } u \\ s \to 0 \Rightarrow \tilde{x} \to 0 \end{cases}$

Consider, for n=2: $\ddot{x} = f + bu$

We could pick $s = \dot{x}$

or, $s = \dot{x} - \dot{x}_d = \dot{\tilde{x}}$

$\dot{s}$ contains u

But $s \nrightarrow 0 \nRightarrow \tilde{x}(t) \nrightarrow 0$

or, $s = \tilde{x}$

But $s \to 0 \Rightarrow \tilde{x}(t) \to 0$

$\dot{s}$ does not contain u

Let us choose, $s = \dot{\tilde{x}} + \lambda\tilde{x}$ (constant > 0)

$s \to 0 \Rightarrow \tilde{x}(t) \to 0$

AND

$\dot{s} = \ddot{x} + \cdots = bu + \cdots$

$\dot{s}$ contain u

Low pass filter

$$s \longrightarrow \boxed{\frac{1}{\frac{d}{dt} + \lambda}} \longrightarrow \tilde{x}$$

Sliding Mode Control

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

$\dot{s}$ contains u

$s \rightarrow 0 \Rightarrow \tilde{x}(t) \rightarrow 0$

For the general case,

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}$$

$\dot{s}$ contain $\frac{d}{dt} x^{(n-1)} = x^{(n)}$,

and therefore, u

(n-1)th order stable linear filter

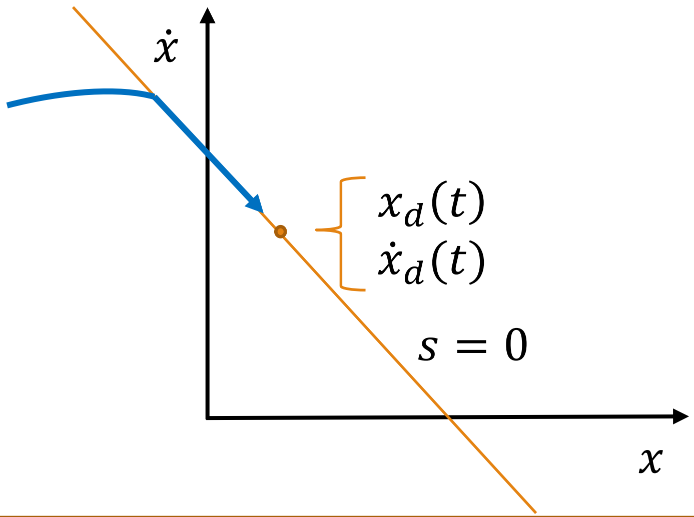
$$s \longrightarrow \boxed{\frac{1}{(\frac{d}{dt} + \lambda)^{n-1}}} \longrightarrow \tilde{x}$$

$$s \rightarrow 0 \Rightarrow \tilde{x}(t) \rightarrow 0$$

Geometrically (visually) what does it look like?

Considering the 2nd order system,

$$s = \dot{\tilde{x}} + \lambda\tilde{x}$$
$$s \rightarrow 0 \Rightarrow \tilde{x}(t) \rightarrow 0$$



For the general n-th order case, $s = 0$ looks like a <u>hyperplane</u>.

Once on this surface in finite time, the state remains there.

It slides along the surface towards $\underline{x}_d(t)$.

Sliding Mode Control

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

$\dot{s}$ contains u

$s \to 0 \Rightarrow \ \tilde{x}(t) \to 0$

What if, as $t \to \infty$, we can only guarantee $|s| \leq \varphi$?

(constant)

For n=2, $s = \dot{\tilde{x}} + \lambda\tilde{x}$

$$s(t) \longrightarrow \boxed{\frac{1}{(\frac{d}{dt}+\lambda)^{n-1}}} \longrightarrow \tilde{x}(t)$$

$|s(t)| \leq \varphi$

After transient

$$\tilde{x}(t) = \underbrace{\text{(effect of I.C.)}}_{\to 0} + \underbrace{\int_0^t e^{-\lambda(t-\tau)} s(r)dr}_{\text{Upper bound}}$$

$$\Rightarrow 1\int_0^t 1 \leq \int_0^1 |\ldots| \leq \varphi \int_0^1 e^{-\lambda(t-\tau)}\, dr \leq \frac{\varphi}{\lambda}$$

For n=n,

$$s \longrightarrow \overbrace{\boxed{\frac{1}{\frac{d}{dt}+\lambda}} \xrightarrow[\leq \frac{\varphi}{\lambda}]{} \cdots\cdots \longrightarrow \boxed{\frac{1}{\frac{d}{dt}+\lambda}}}^{\text{n-1 filters}} \longrightarrow \tilde{x} \leq \frac{\varphi}{\lambda^{n-1}}$$

Activity

Recall, we design hyperplane as, $$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}$$

$$s \longrightarrow \overbrace{\boxed{\frac{1}{\frac{d}{dt} + \lambda}} \longrightarrow \cdots\cdots \longrightarrow \boxed{\frac{1}{\frac{d}{dt} + \lambda}}}^{\text{n-1 filters}} \longrightarrow \tilde{x}$$

For n=2, it was $$s = \dot{\tilde{x}} + \lambda \tilde{x}$$

<u>Activity (3 min)</u>

Derive s for (n=3) 3rd order system in terms of $\tilde{x}$ and λ

<u>Answer</u>

For n=2, it was $$s = \dot{\tilde{x}} + \lambda \tilde{x}$$

For n=3, $$s = \left(\frac{d}{dt} + \lambda\right)\left(\dot{\tilde{x}} + \lambda \tilde{x}\right) = \ddot{\tilde{x}} + 2\lambda\dot{\tilde{x}} + \lambda^2 \tilde{x}$$

Example Problem:

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

$\dot{s}$ contains u

$s \to 0 \Rightarrow \ \tilde{x}(t) \to 0$

Consider a 2nd order scalar system:

$$\ddot{x} + a(t)\ \dot{x}^2 \cos 3x = u \qquad \text{with } 1 \le a(t) \le 2$$

We can choose,

$$\hat{f} = -1.5\ \dot{x}^2 \cos 3x$$

So that,

$$F = 0.5\ \dot{x}^2 |\cos 3x|$$

Design

$$\begin{aligned} s &= \dot{\tilde{x}} + \lambda \tilde{x} \\ \dot{s} &= \ddot{\tilde{x}} + \lambda \dot{\tilde{x}} \\ &= \ddot{x} - \ddot{x}_d + \lambda \dot{\tilde{x}} \\ &= f + bu - \ddot{x}_d + \lambda \dot{\tilde{x}} \end{aligned}$$

We choose a control law that makes $\dot{s} = 0$, if we knew exactly the dynamics of f, i.e. $\hat{f} = f$,

$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda \dot{\tilde{x}}$$

Therefore, plugging in $\mathrm{u} = \hat{u}$,

We get,

$$\dot{s} = f - \hat{f}$$

Sliding Condition

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

$\dot{s}$ contains u

$s \to 0 \Rightarrow \tilde{x}(t) \to 0$

Step 2: Determine rate of convergence based on the Sliding Condition

We know the system converges to $s = 0$,
But how do we ensure that it happens within a finite time?

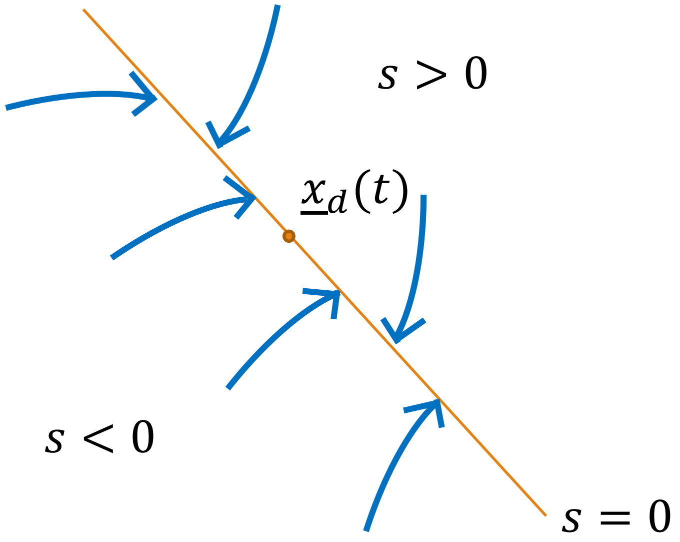
We choose control input u, such that

$$\frac{d}{dt}s^2 \le 0$$

More precisely,

$$\boxed{\frac{1}{2}\frac{d}{dt}s^2 \le -\eta|s|}$$

i.e. we want s to decrease to its absolute value at the rate described by



$\cancel{s}\dot{s} \le -\eta \cancel{s}$

If we solved for time, t

We get, $t \le \frac{|s(t=0)|}{\eta}$

Therefore, $s = 0$ is reached in a finite time!

Continuing the previous example:

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

Step 2: Determine rate of convergence based on the Sliding Condition

Step 3: Determine Control Law

Example problem formulation so far:

$\ddot{x} + a(t)\,\dot{x}^2 \cos 3x = u$
with $1 \le a(t) \le 2$

We had $\hat{f} = -1.5\,\dot{x}^2 \cos 3x$

So that, $F = 0.5\,\dot{x}^2 |\cos 3x|$

Design, $s = \dot{\tilde{x}} + \lambda\tilde{x}$
$\dot{s} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}}$

We chose, $\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\tilde{x}}$

To get, $\dot{s} = f - \hat{f}$

We previously got, $\dot{s} = f - \hat{f}$
For $\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\tilde{x}}$

Following the Sliding Condition,

$$\frac{1}{2}\frac{d}{dt}s^2 = s\dot{s} = s(f - \hat{f})$$

Does not say much about the rate of convergence of s.

We could modify u such that the rate is accounted for:

Trying, $u = \hat{u} - k\,\mathrm{sgn}(s)$
Following the Sliding Condition,

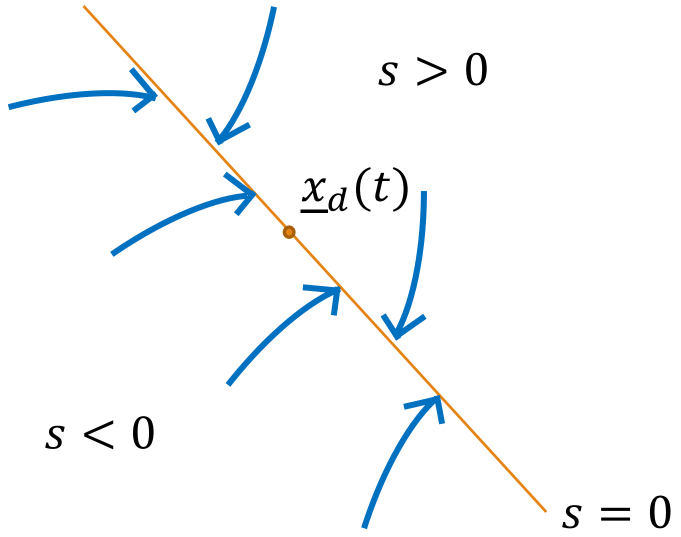
$$\frac{1}{2}\frac{d}{dt}s^2 = s\dot{s} = \underbrace{s(f - \hat{f}) - k\,\mathrm{sgn}(s) \le -\eta|s|}$$

For this condition to hold,
$k = F + \eta$

So the new control law would be,
$$u = \hat{u} - k\,\mathrm{sgn}(s)$$
$$= \left(1.5\,\dot{x}^2 \cos 3x + \ddot{x}_d - \lambda\dot{\tilde{x}}\right) - \left(0.5\,\dot{x}^2|\cos 3x| + \eta\right)\mathrm{sgn}(\dot{\tilde{x}} + \lambda\tilde{x})$$

SMC switching control law:

Switching control law: $u = \hat{u} - (F + \eta)\,\mathrm{sgn}(s)$



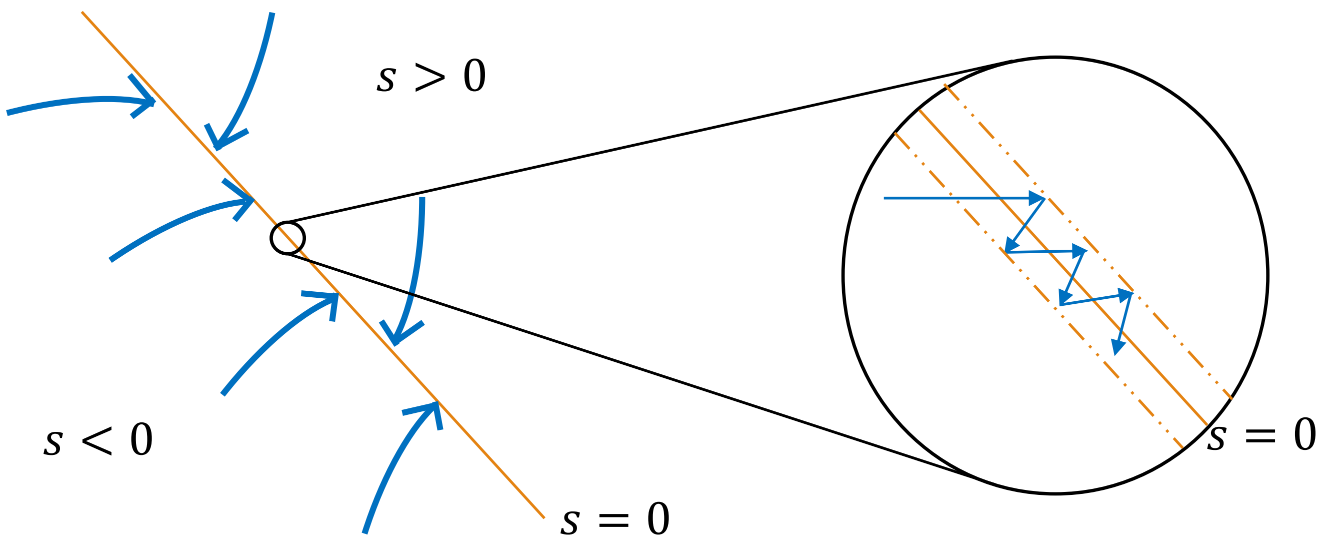
As desired state $\underline{x}_d$ changes , the hyperplane s shifts on the n-dimensional space.

The controller forces the system to go to $s = 0$.

SMC switching control law:

Disadvantage: Control chattering

Switching control law: $u = \hat{u} - (F + \eta)\boxed{\mathrm{sgn}(s)}$



Bigger F (uncertainty bounds) means bigger discontinuity in the control signal due to the *sgn* term.

Chattering occurs at full range $\mathrm{k} = (F + \eta)$.
The overshooting occurs due to system sampling, computation time.

(Maybe okay for certain systems/applications utilizing PWM, but without it, s does not tend to zero, but remains close to it.)

Sliding Mode Control Theory Summary

Consider the n^{th} order scalar nonlinear system:

$$x^n = f(x, \dot{x}, ..., x^{n-1}, t) + b(x, \dot{x}, ..., x^{n-1}, t)u$$

$$|\hat{f}(\underline{x}, t) - f(\underline{x}, t)| \le F(\underline{x}, t) \qquad |\hat{b}(\underline{x}, t) - b(\underline{x}, t)| \le B(\underline{x}, t)$$

Trajectory tracking error:

$$\tilde{x}(t) = x(t) - x_d(t)$$

$$\tilde{x}(t) \to 0$$

<u>Step 1:</u> Create intermediate variable s, such that $s \Rightarrow \begin{cases} \dot{s} \text{ contains } u \\ s \to 0 \Rightarrow \tilde{x} \to 0 \end{cases}$

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}$$

<u>Step 2:</u> Determine rate of convergence based on the Sliding Condition

$$\frac{1}{2}\frac{d}{dt}s^2 \le -\eta|s|$$

<u>Step 3:</u> Determine switching control Law

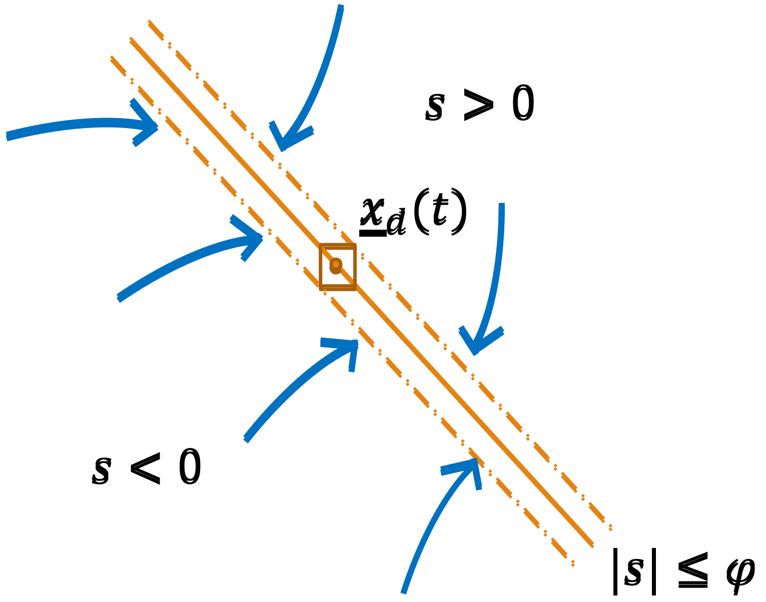
$$u = \hat{u} - k\,\mathrm{sgn}(s)$$

where,

$$\mathrm{k} = (F + \eta)$$

Conclusion

Switching control law: $u = \hat{u} - (F + \eta)\,\boxed{\mathrm{sgn}(s)}$



Chattering occurs within bounds φ.
The control law must be modified to "***smoothen***" the control signal for implementation.

Therefore, Week 2:

- Example problem control formulation with implementation notes (Applied)
- Introduction to and MATLAB/SIMULINK implementation

Thank you

Control design activity:

Consider a 2nd order scalar system:

$$\ddot{x} + \dot{x}\cos 3a(t)x = u \qquad \text{with } 1 \leq a(t) \leq 2$$

Good luck!!

Robust Control of Nonlinear Systems

Tamzidul Mina

Week 2

Introduction to SIMULINK

- Simple example – building blocks on SIMULINK

Example problem controller design

- Example problem SIMULINK implementation

Example problem chatter free controller improvement

- Example problem SIMULINK implementation

Design and implement your own controller!

Brief Introduction to Simulink

Simulink® is a block diagram environment for multi-domain simulation and Model-Based Design.

It supports:

- system-level design
- simulation
- automatic code generation
- continuous test and verification of embedded systems.

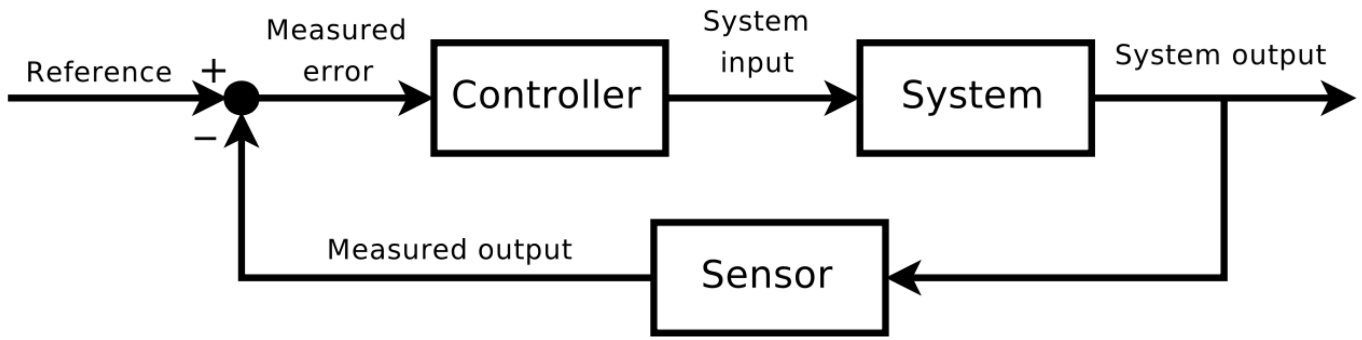
Simulink provides:

- customizable block libraries
- solvers for modeling and simulating dynamic systems.

It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

SIMULINK – Getting Started

Create your first block



Lets start your MATLAB and dive right in!

Open MATLAB, type Simulink in the command window and hit enter

Tasks:

- SIMULINK overview

SIMULINK – Getting Started

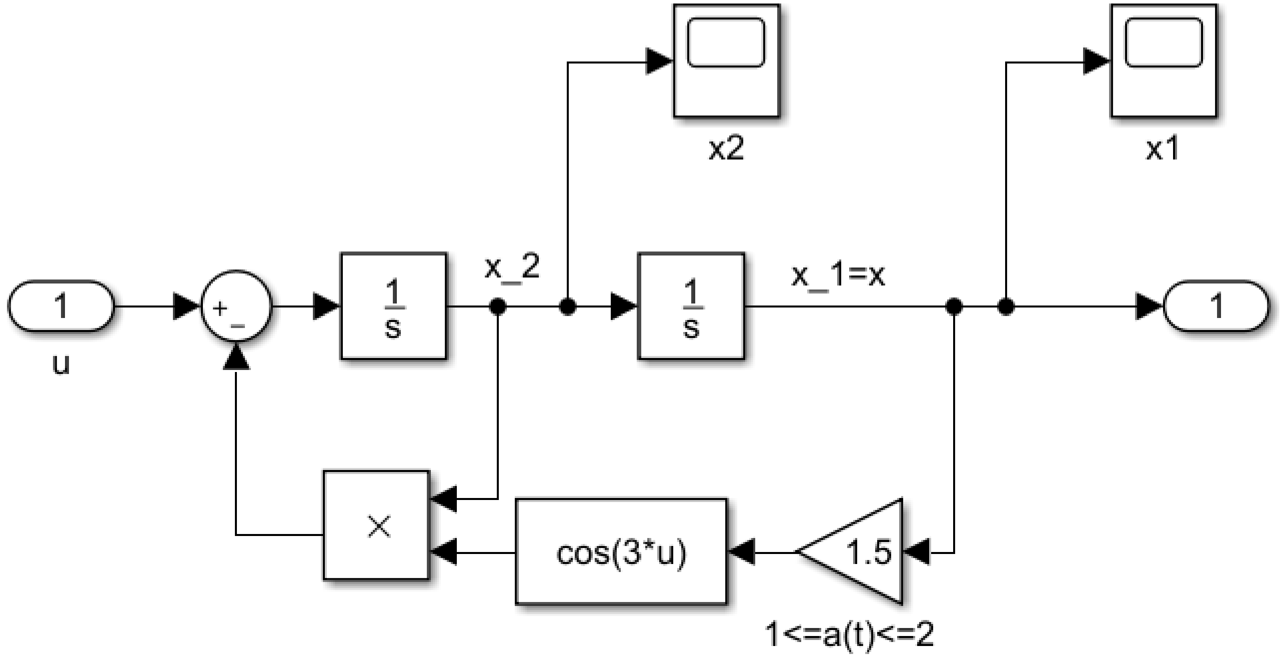
Create your first block

Create the following plant model as a block:

$$\ddot{x} + \dot{x}\cos 3a(t)x = u \qquad \text{with } 1 \leq a(t) \leq 2$$

Trajectory tracking error:

$$\tilde{x}(t) = x(t) - x_d(t) \qquad \text{Goal:} \quad \tilde{x}(t) \rightarrow 0$$



Start with a ‘subsystem’ block to create this.

Example Problem:

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

<u>Step 1</u>: Create intermediate variable s

<u>Step 2</u>: Determine rate of convergence based on the Sliding Condition, η

<u>Step 3</u>: Determine Control Law

Consider a 2nd order scalar system:

$$\ddot{x} + a(t)\ \dot{x}^2 \cos 3x = u \qquad \text{with } 1 \le a(t) \le 2$$

Write this as: $\ddot{x} = -a(t)\ \dot{x}^2 \cos 3x + u$

We can choose, $\hat{f} = -1.5\ \dot{x}^2 \cos 3x$

So that, $F = 0.5\ \dot{x}^2 |\cos 3x|$

Design of sliding manifold:

$$s = \dot{\tilde{x}} + \lambda\tilde{x}$$
$$\dot{s} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}}$$
$$= f + bu - \ddot{x}_d + \lambda\dot{\tilde{x}}$$

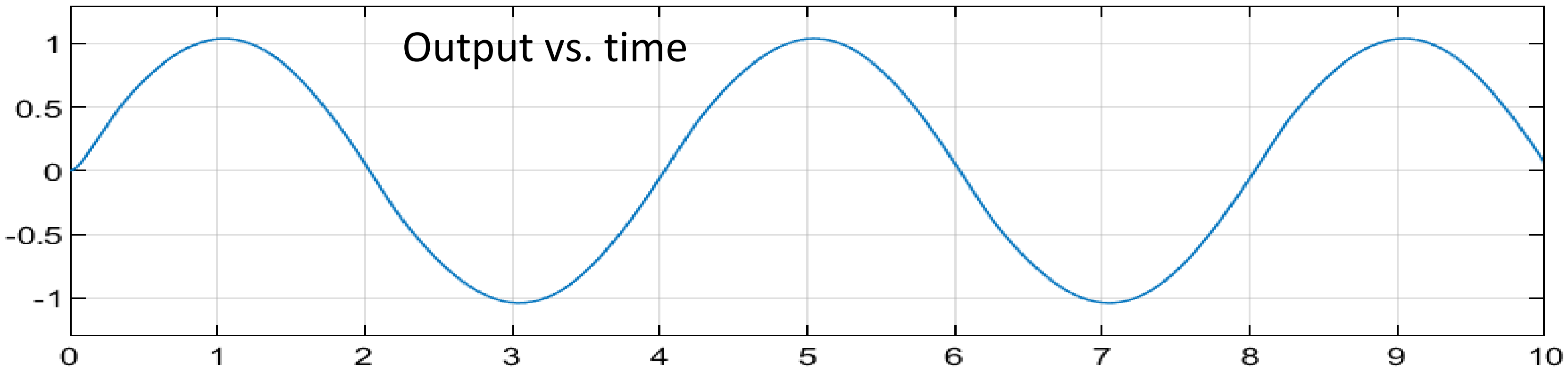
We choose a control law that makes $\dot{s} = 0$, if we knew exactly the dynamics of f, i.e. $\hat{f} = f$,

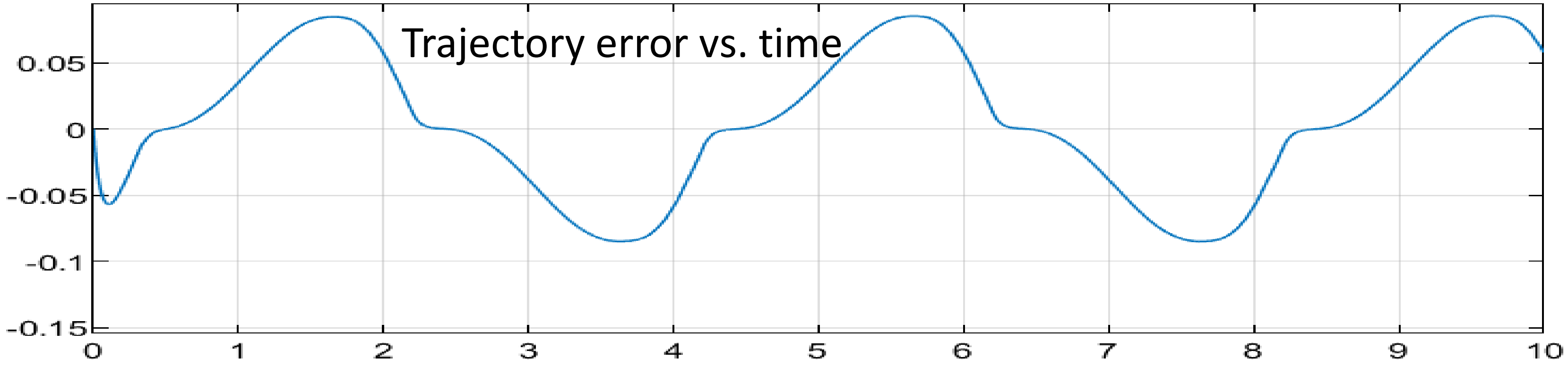
$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\tilde{x}}$$

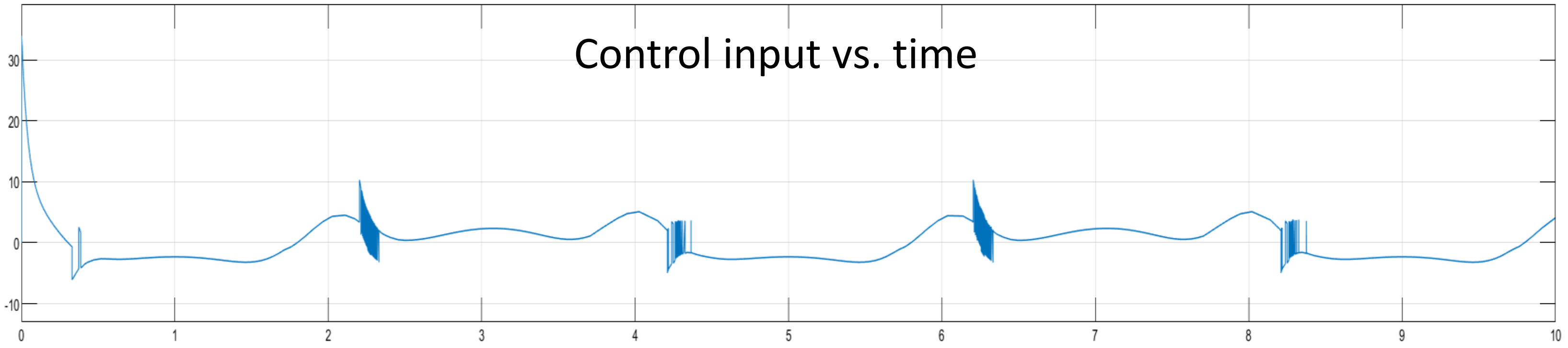
$$u = \hat{u} - k\ \mathrm{sgn}(s)$$
$$= \left(1.5\ \dot{x}^2 \cos 3x + \ddot{x}_d - \lambda\dot{\tilde{x}}\right) - \left(0.5\ \dot{x}^2|\cos 3x| + \eta\right) \mathrm{sgn}(\dot{\tilde{x}} + \lambda\tilde{x})$$

Download file: <u>SlidingMode_sign.mdl</u>
From https://github.com/tmina01/robust-control

Results


Output vs. time
1
0.5
0
-0.5
-1
0 1 2 3 4 5 6 7 8 9 10

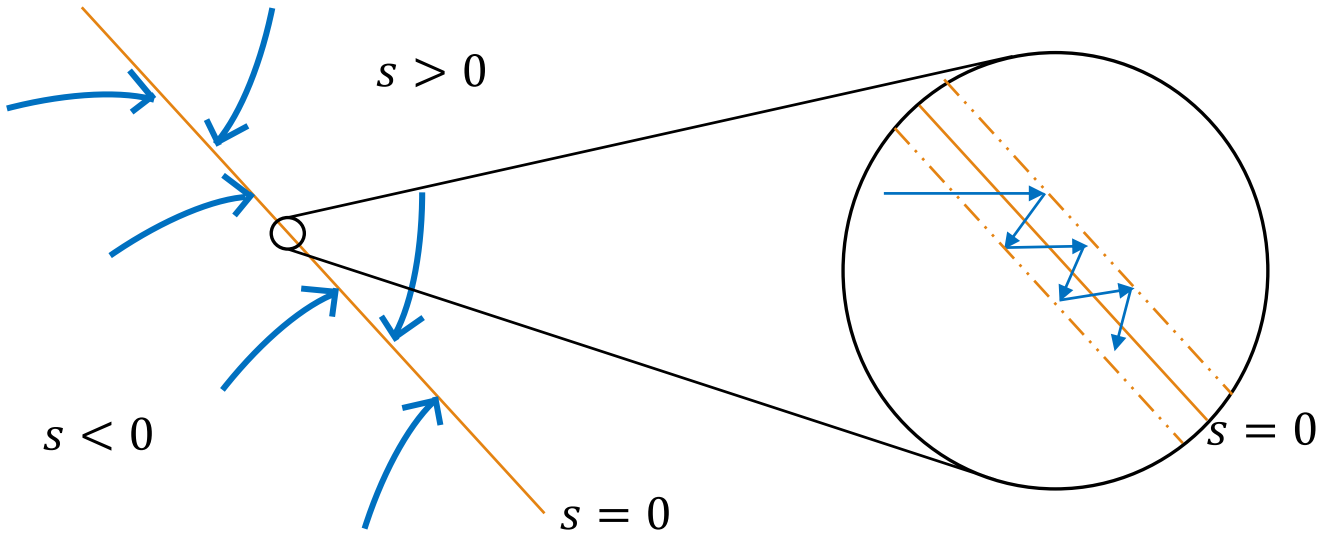

Trajectory error vs. time
0.05
0
-0.05
-0.1
-0.15
0 1 2 3 4 5 6 7 8 9 10

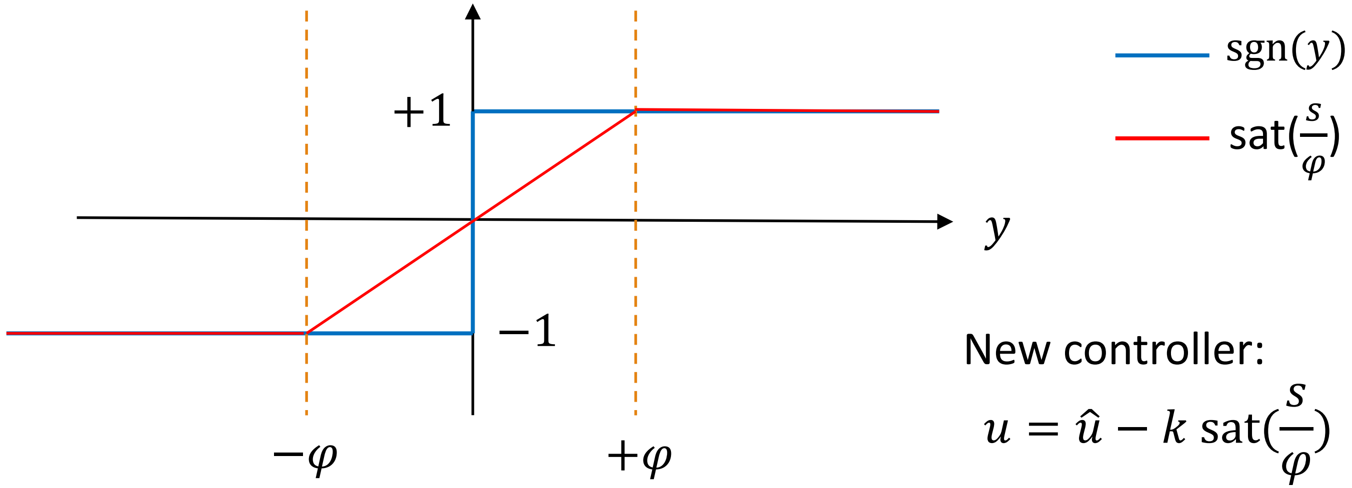

Control input vs. time
30
20
10
0
-10
0 1 2 3 4 5 6 7 8 9 10

Control chatter smoothing

Recall:

Switching control law $u = \hat{u} - (F + \eta)\boxed{\mathrm{sgn}(s)}$





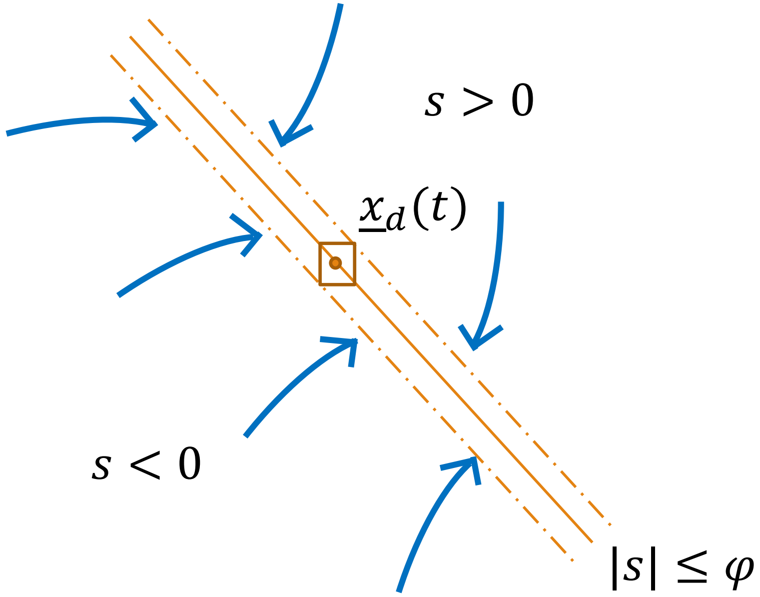
New controller:

$$u = \hat{u} - k\ \mathrm{sat}(\frac{s}{\varphi})$$

Control chatter smoothing

How do we design φ?

Inside the $|s| \leq \varphi$ region:



<u>Step 2:</u> Outside the φ region, i.e. $|s| \geq \varphi$; since k(x, t), the Sliding Condition becomes:

$$\frac{1}{2}\frac{d}{dt}s^2 \leq (\dot{\varphi} - \eta)|s|$$

i.e. if φ is decreasing, the system must converge to within the boundary layer faster than the rate φ is shrinking.

<u>Step 3:</u> Determine switching control Law: $u = \hat{u} - k\, sat(\frac{s}{\varphi})$

where,

$$\mathrm{k} = (F + \eta - \dot{\varphi})$$
$$\dot{\varphi} = k_d - \lambda\varphi$$

Chatter free control of example problem:

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

Step 2: Determine rate of convergence based on the Sliding Condition, η

Step 3: Determine Control Law

Consider a 2nd order scalar system:

$$\ddot{x} + a(t)\ \dot{x}^2 \cos 3x = u \qquad \text{with } 1 \le a(t) \le 2$$

Write this as: $\ddot{x} = -a(t)\ \dot{x}^2 \cos 3x + u$

We can choose, $\hat{f} = -1.5\ \dot{x}^2 \cos 3x$

So that, $F = 0.5\ \dot{x}^2 |\cos 3x|$

Design of sliding manifold:

$$s = \dot{\tilde{x}} + \lambda\tilde{x}$$
$$\dot{s} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}}$$
$$= f + bu - \ddot{x}_d + \lambda\dot{\tilde{x}}$$

We choose a control law that makes $\dot{s} = 0$, if we knew exactly the dynamics of f, i.e. $\hat{f} = f$,

$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\tilde{x}}$$

$$u = \hat{u} - k\ sat\left(\frac{s}{\varphi}\right)$$
$$= \left(1.5\ \dot{x}^2 \cos 3x + \ddot{x}_d - \lambda\dot{\tilde{x}}\right) - \left(0.5\ \dot{x}^2 |\cos 3x| + \eta - \dot{\varphi}\right) sat\left(\frac{s}{\varphi}\right)$$

Download file: SlidingMode_sat.mdl
from https://github.com/tmina01/robust-control

Controller performance observations

Tasks:

- How does the two controller designs compare?
- Change $a(t)$ for both controllers and observe changes
  - In what range for $a(t)$ does the controller prove to be robust?
  - Find the value of $a(t)$ for which the controller starts to fail.
- Change λ and η and observe controller performance.

Design a controller for the following system

Control design activity:

Consider a 2nd order scalar system:

$$\ddot{x} + \dot{x}\cos 3a(t)x = u \qquad \text{with } 1 \leq a(t) \leq 2$$

Good luck!!

Example Problem:

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

Step 2: Determine rate of convergence based on the Sliding Condition, η

Step 3: Determine Control Law

Consider a 2nd order scalar system:

$$\ddot{x} + \dot{x}\cos 3a(t)x = u \qquad \text{with } 1 \le a(t) \le 2$$

Write this as: $\ddot{x} = -\dot{x}\cos 3a(t)x + u$

We can choose, $\hat{f} = -\dot{x}\cos 3(1.5)x$

So that, $F = \dot{x}\cos 3(0.5)x$

Design of sliding manifold:

$$s = \dot{\tilde{x}} + \lambda\tilde{x}$$
$$\dot{s} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}}$$
$$= f + bu - \ddot{x}_d + \lambda\dot{\tilde{x}}$$

We choose a control law that makes $\dot{s} = 0$, if we knew exactly the dynamics of f, i.e. $\hat{f} = f$,

$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\tilde{x}}$$

$$u = \hat{u} - k\ \mathrm{sgn}(s)$$
$$= \left(\dot{x}\cos 3(1.5)x + \ddot{x}_d - \lambda\dot{\tilde{x}}\right) - (\dot{x}\cos 3(0.5)x + \eta)\ \mathrm{sgn}(\dot{\tilde{x}} + \lambda\tilde{x})$$

Chatter free control of example problem:

$$\ddot{x} = f(\underline{x}, t) + b(\underline{x}, t)u$$
$$\tilde{x}(t) = x(t) - x_d(t)$$

Step 1: Create intermediate variable s

Step 2: Determine rate of convergence based on the Sliding Condition, η

Step 3: Determine Control Law

Consider a 2nd order scalar system:

$$\ddot{x} + \dot{x}\cos 3a(t)x = u \qquad \text{with } 1 \le a(t) \le 2$$

Write this as: $\ddot{x} = -\dot{x}\cos 3a(t)x + u$

We can choose, $\hat{f} = -\dot{x}\cos 3(1.5)x$

So that, $F = \dot{x}\cos 3(0.5)x$

Design of sliding manifold:

$$s = \dot{\tilde{x}} + \lambda\tilde{x}$$
$$\dot{s} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}}$$
$$= f + bu - \ddot{x}_d + \lambda\dot{\tilde{x}}$$

We choose a control law that makes $\dot{s} = 0$, if we knew exactly the dynamics of f, i.e. $\hat{f} = f$,

$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda\dot{\tilde{x}}$$

$$u = \hat{u} - k\, sat(\frac{s}{\varphi})$$
$$= \left(\dot{x}\cos 3(1.5)x + \ddot{x}_d - \lambda\dot{\tilde{x}}\right) - (\dot{x}\cos 3(0.5)x + \eta - \dot{\varphi})\, sat(\frac{s}{\varphi})$$