MR8500 - PhD Topics in Marine Control Systems (2020)

Backstepping design on complex nonlinear ODE systems

Lecture 1: Elegant methods

Lecture 2: Transferring a complex system into a familiar form

Zhengru Ren

Nov. 26, 2020

Lecture 2-Application to complex nonlinear systems

0.1 Previously

Lecture 1 - Elegant methods

- The development of backstepping, from simple systems to complex uncertain systems.
- Semi-global stability criteria
- Six modularizable methods
  - Dynamic surface control / commanded filters
  - Finite-time control
  - Neural network and fuzzy logic system
  - Nussbaum function
  - Barrier Lyapunov function
  - Hyperbolic tangent function.

- Robustness-based method: If the control gain (γ) is large enough, the system will stay in a small enough region (δ).
- Approximation-based method: Everything can be estimated. First estimated and then cancel it.

Lemma 1. *A LFC $V(x)$ is bounded if the initial condition $V(0)$ is bounded, $V(x)$ is positive definite and continuous and if a Lyapunov-like inequality holds, i.e.,*

$$\dot{V}(x) \leq -\gamma V(x) + \delta, \tag{1}$$

where $\gamma > 0$ and $\delta > 0$. Define $\rho := \delta/\gamma$,

$$0 \leq V(t) \leq \rho + (V(0) - \rho)\exp(-\gamma t). \tag{2}$$

And it implies that

$$V(t) \leq e^{-\gamma t} V(0) + \int_0^t e^{-\gamma(t-\tau)} \rho(\tau) d\tau, \ \forall t \geq 0, \tag{3}$$

for any finite constant γ.

Some names in titles
Neural adaptive control
Fuzzy adaptive control
Robust adaptive control
Finite-time adaptive control
State feedback
Output feedback

0.2 Outline

Complexities and nonlinearities can be found everywhere.

Lecture 2 - Applications of methods in Lecture 1 to complex nonlinear systems

- A class of systems:
    - State constraints
    - Input nonlinearities (input saturation, deadzone, time-varying control coefficient),
    - Unknown disturbance
    - Time-delay effects
    - Pure-feedback system
    - Event-triggered systems
    - Stochastic systems
- Complex systems:
    - Underactuated system
    - Switched system
    - Multi-agent consensus system.
- Understand the robustness-based method and the approximation-based method

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\begin{aligned}\frac{dx_i}{dt} &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \frac{dx_n}{dt} &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}$$

$$\left\{\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\\ \dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u,\\ y &= x_1,\end{aligned}\right.$$

1.1 State constraints

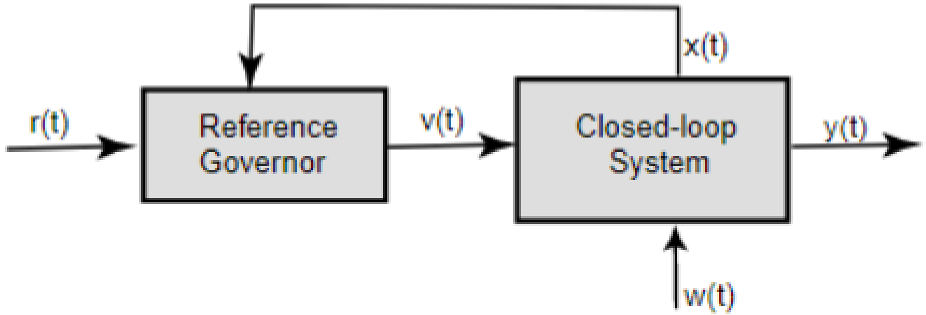
State constraints are due to physical limitations or performance requirements.

Due to the existence of actuator physical limitation and operational limits, there is a clear boundary that the states or the tracking error

- $x(t) \in \mathbb{D}_x$
- $z_1(t) = y(t) - y_d(t) \in \mathbb{D}_{z1}$

- **BLF**
- Commended filter
- Nonlinear mapping

- reference governor



1.2 Unknown disturbance

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1} + d_i, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u + d_n,$$
$$y = x_1,$$

unmatched disturbance

matched disturbance

Disturbances are impossible to be directly measured, resulting in the robustness violation and failure of the direct compensation approaches.

The backstepping problem with unknown disturbance is namely **disturbance-rejection control.**

Unmatched disturbance: when the disturbance d_i enters the system with a different state from the control input u; otherwise it is matched.

Bounded-disturbance assumptions:

- $|d_i| \leq \bar{d}_i$
- $|d_i| \leq \rho_i(\bar{x}_i)\theta_i$
- $|d_i| \leq \big(\rho_{i1}(\bar{x}_i) + \rho_{i2}(\bar{z}_i)\big)\theta_i$

where θ_i unknown bounded constants

$\rho_i(\bar{x}_i) \in \mathbb{R}_+$ known smooth functions for $t > t_0$

1.2 Uncertain disturbance

Robustness-based methods – Projection operator

Lemma 1. The sufficiently smooth projection operator $Proj_d(\cdot)$ is described as [14]

$$Proj_d\left(\mu_i, \hat{\theta}_i\right) = \mu_i - \frac{\pi_1 \pi_2 \nabla p\left(\hat{\theta}_i\right)}{4\left(\varepsilon^2 + 2\varepsilon\theta_0\right)^{n-i} \theta_0^2},$$

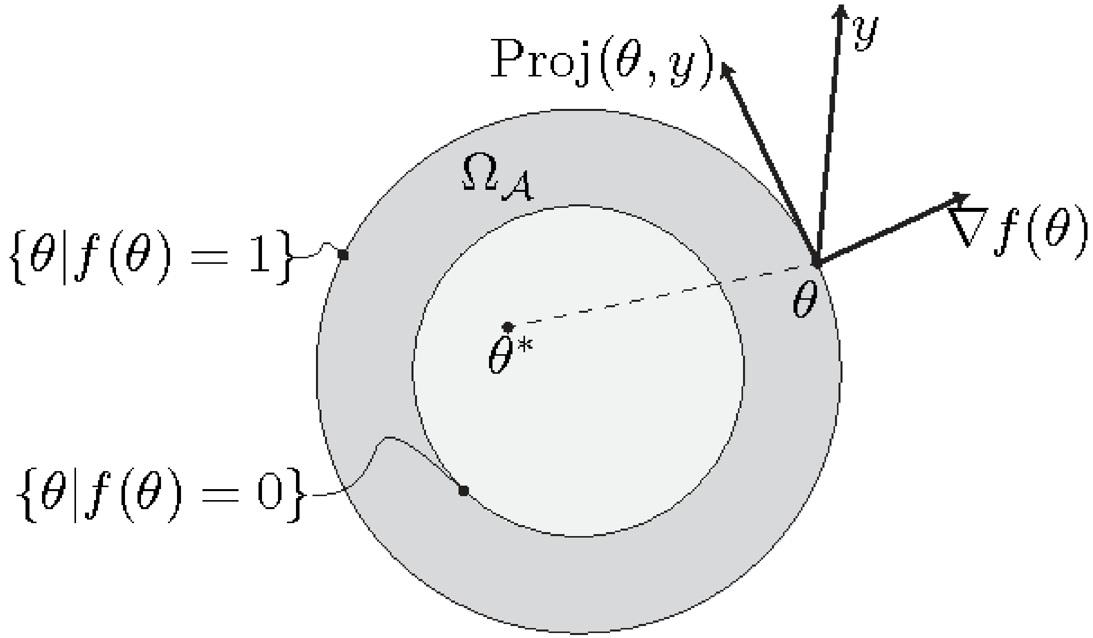
$$i = 1, 2, \cdots, n-1 \quad (9)$$

where,

$$p\left(\hat{\theta}_i\right) = \hat{\theta}_i^T - \theta_0^2, \quad \pi_1 = \begin{cases} p^{n-i}\left(\hat{\theta}_i\right), & p\left(\hat{\theta}_i\right) > 0, \\ 0, & \text{other}, \end{cases}$$

$$\pi_2 = \frac{1}{2}\nabla p\left(\hat{\theta}_i\right)\mu_i + \sqrt{\left(\frac{1}{2}\nabla p\left(\hat{\theta}_i\right)\mu_i\right)^2 + \delta^2},$$

where, ∇ is the gradient operator; ε, δ are any normal constants; θ_0 is the upper bound of θ_i ,that is $\left|\theta_i\right| \le \theta_0$.



Key property: $\tilde{\theta}^\top \left(\mu_i - Proj\left(\mu_i, \hat{\theta}_i\right)\right) \le 0$

1.2 Uncertain disturbance

Robustness-based methods – Projection operator

Example:
$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1} + d_i, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u + d_n,$$
$$y = x_1,$$

Assumptions: $|d_i| \leq \bar{d}_i$

Step 1: (a) Define $\tilde{\theta}_i = \theta - \hat{\theta}_i$ and a Lyapunov function candidate (LFC)

$$V_1(z_1) = V_{QF,1} + \frac{1}{2}\tilde{\theta}_1^\top \Gamma_1^{-1}\tilde{\theta}_1. \quad (1)$$

(b) Because $\dot{\tilde{\theta}}_i = -\dot{\hat{\theta}}_i$, its time derivative becomes

$$\dot{V}_1 = z_1[f_1 + g_1(\alpha_1 + z_2) + \phi_1^\top\theta + d_1 - \dot{x}_{1d}] - \tilde{\theta}_1^\top\Gamma_1^{-1}\dot{\hat{\theta}}_1, \quad (2)$$

(c) The virtual control law and adaptive law are selected as

$$\alpha_1 = g_1^{-1}[-f_1(x_1) - \kappa_1(z_1) - \mathrm{sgn}(z_1)\bar{d}_1 + \dot{x}_{1d} - \phi_1^\top\hat{\theta}_1], \quad (3)$$

$$\dot{\hat{\theta}}_1 = \Gamma_1 \mathrm{Proj}(\phi_1 z_1, \hat{\theta}), \quad (4)$$

where $\kappa_1(z_1)z_1$ is positive definite and $\gamma_1 > 0$. A simple example of $\kappa_1(z_1)$ is $\kappa_1(z_1) = c_1 z_1$ with $c_1 > 0$.

$z_1 d_1 \leq |z_1|\bar{d}_1$

$$\dot{V}_1 = -c_1 z_1^2 z_1 + g_1 z_1 z_2 + z_1 d_1 - \mathrm{sgn}(z_1)z_1\bar{d}_1 + \tilde{\theta}^\top(z_1\phi_1^\top - \mathrm{Proj}(\phi_1 z_1, \hat{\theta}))$$
$$\leq -c_1 z_1^2 z_1 + g_1 z_1 z_2, \quad (5)$$

Key property: $\tilde{\theta}^\top\left(\mu_i - Proj(\mu_i, \hat{\theta}_i)\right) \leq 0$

There are many types of projection operators.

- *The projection operator for two vectors $\theta,\ y \in \mathbb{R}^k$ is now introducted as*

$$Proj(\theta, y, f) = \begin{cases} y - \frac{\nabla f(\theta)\nabla f(\theta)^\top}{\|\nabla f(\theta)\|^2} y f(\theta) & \text{if } f(\theta) > 0 \wedge y^\top \nabla f(\theta) > 0, \\ y & \text{otherwise.} \end{cases} \quad (.4.1)$$

where $f : \mathbb{R}^k \mapsto \mathbb{R}$ is a convex function and $\nabla f(\theta_b) = (\frac{\partial f(\theta)}{\theta_1} \cdots \frac{\partial f(\theta)}{\theta_k})^\top$.

- *The general form of the projection operator is the $n \times m$ matrix extension to the vector definition above.*

$$Proj(\Theta, Y, F) = [Proj(\theta_1, y_1, f_1) \cdots Proj(\theta_m, y_m, f_m)], \quad (.4.2)$$

where $\Theta = [\theta_1 \cdots \theta_m] \in \mathbb{R}^{m\times n}$, $Y = [y_1 \cdots y_m] \in \mathbb{R}^{m\times n}$, and $F = [f_1(\theta_1) \cdots f_m(\theta_m)]^\top \in \mathbb{R}^m$.

- *$\Gamma-$Projection*

$$Proj_\Gamma(\theta, y, f) = \begin{cases} \Gamma y - \Gamma\frac{\nabla f(\theta)\nabla f(\theta)^\top}{\|\nabla f(\theta)\|^2}\Gamma y f(\theta) & \text{if } f(\theta) > 0 \wedge \Gamma y^\top \nabla f(\theta) > 0, \\ \Gamma & \text{otherwise.} \end{cases} \quad (.4.3)$$

- *Parameter projection (Grip et al. 2015)*

$$Proj(\hat{b}, \beta) = \begin{cases} (I_3 - \frac{c(\hat{b})}{\|\hat{b}\|^2}\hat{b}\hat{b}')\beta & \|\hat{b}\| \geq M_b, \hat{b}'\beta > 0 \\ \beta & \text{otherwise.} \end{cases} \quad (.4.4)$$

where $c(\hat{b}) = \min(1, (\|\hat{b}\|2 - M_b^2)/(M_{\hat{b}}2 - M_b^2))$.

- *A projection operator, along the span of the control vector field $g(x)$ onto the tangent space to the constant level sets of the energy function $V(x)$,*

$$M(x) = \left[I - \frac{1}{L_g V(x)} g(x) \frac{\partial V}{denotes\partial x^\top}\right] \quad (.4.5)$$

Projection operator is widely used in earlier works. However, it is no longer popular anymore after the 2000s. A reason is the complex calculation of the terms with Laplace operator ∇f

1.2 Unknown disturbance

Approximation-based methods

Idea: Estimate the unknown and cancel the estimate as much as possible

- Neural network/fuzzy logic system
- High-gain disturbance observer

$$\dot{\xi}_i = -k(\xi_i + kx_i - f_i(x_1, x_2, \ldots, x_i) - b_i x_{i+1})$$
$$\dot{\xi}_n = -k(\xi_n + kx_n - f_n(x_1, x_2, \ldots, x_n) - b_n u)$$
$$\hat{d}_i = \xi_i + kx_i$$
$$\hat{d}_n = \xi_n + kx_n$$
$$\dot{\hat{d}}_i = h(\hat{d}_i, d_i) = -k(\hat{d}_i - d_i)$$

- Generalized disturbance observer

$$\frac{d}{dt}\left[\hat{d}_i^{(j-1)}\right] = l_{ij}\tilde{d}_i + \hat{d}_i^{(j)}, \quad j = 1, 2, \ldots, r-1$$
$$\frac{d}{dt}\left[\hat{d}_i^{(r-1)}\right] = l_{ir}\tilde{d}_i$$

- Exosystem (internal model principle)

$d = \vartheta + \delta_d$, where δ_d is a bounded function, ϑ is the output of a linear exosystem given by

$$\dot{\chi} = \Gamma\chi,$$
$$\vartheta = c^\top \chi,$$

- Specific case (periodic disturbance)

$d(t) = d(t + T)$

$d(t) = B^\top \phi(t) + \delta_d, |\delta_d| \le \bar{\delta}_d$

$$\phi(t) = [\phi_1(t), \ldots, \phi_q(t)]^\mathrm{T}$$

with

$$\phi_1(t) = 1$$
$$\phi_{2j}(t) = \sqrt{2}\sin(2\pi jt/T)$$
$$\phi_{2j+1}(t) = \sqrt{2}\cos(2\pi jt/T), \quad j = 1, \ldots, (q-1)/2$$

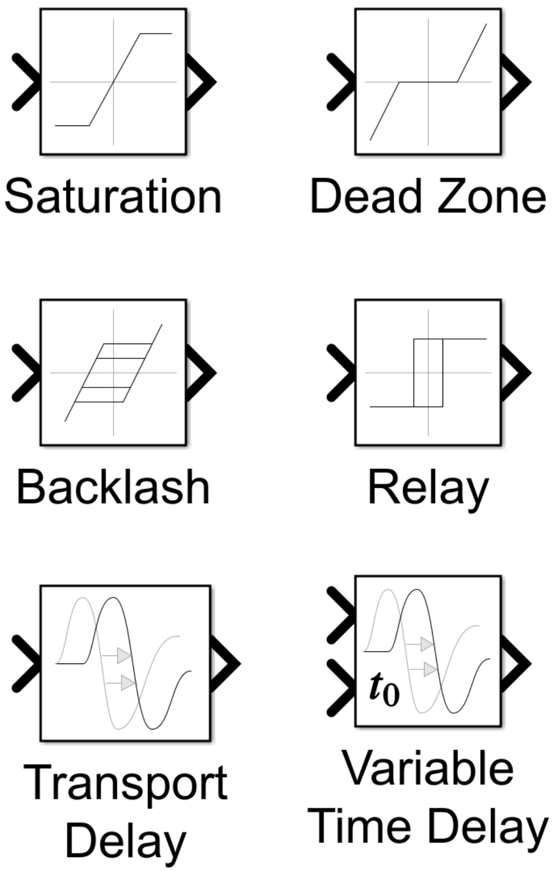
1.3 Input nonlinearities

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u,$$
$$y = x_1,$$

The typical control input $u(t) = v(t)$
A general control input model

$$u\left(v(t - \tau_d(t))\right)$$

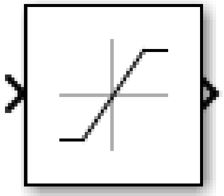
where v control signal
τ_d is the time delay



1.3.1 Input saturation

Definition and approximation

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u,$$
$$y = x_1,$$



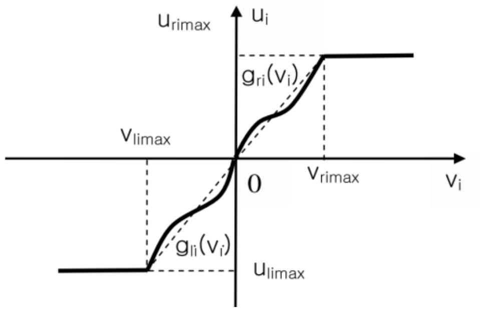
Physical actuators surely have their limits, also called input constraints. When the control input remains within the boundness, the input saturation effects are negligible.

Model of input saturation:

$$u(v) = \mathrm{sat}(v) = \begin{cases} u_{min}, & \text{if } v \le u_{min}, \\ g_s(v), & \text{if } u_{min} < v < u_{max}, \\ u_{max}, & \text{if } v \ge u_{max}, \end{cases}$$

where g_s is a smooth function, and
u_{min} and u_{max} are the minimum and maximum values for u.

Challenges:
1. Nonsmoothness at the breakpoints
2. Limited scope of input



For a simplest linear saturation with symmetric limits, i.e., $g_s(v) = v$ and $-u_{min} = u_{\max} = u_m > 0$, the input saturation is simplified to be

$$u(v(t)) = \mathrm{sat}(v(t)) = \begin{cases} v(t), & \text{if } |v| < u_m, \\ \mathrm{sgn}(v(t))u_m, & \text{if } |v| \ge u_m. \end{cases}$$

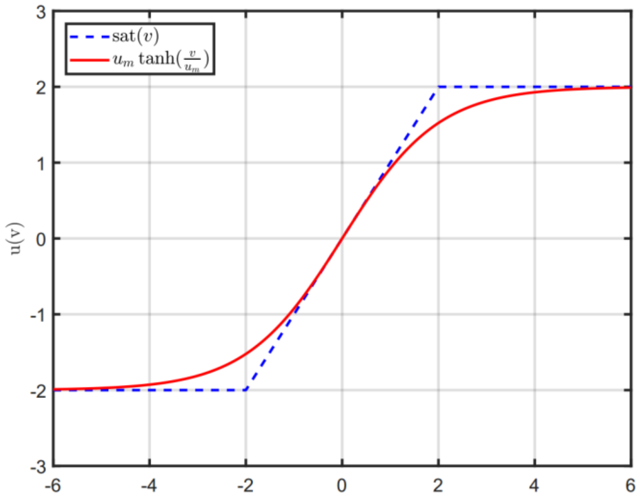
Smooth approximations are adopted to overcome the nonsmoothness.
- The symmetric saturation

$$\mathrm{sat}(v) \approxeq \eta_{sat}(v) = u_m \tanh(\tfrac{v}{u_m}) = u_m \frac{e^{v/u_m} - e^{-v/u_m}}{e^{v/u_m} + e^{-v/u_m}}.$$

- Asymmetric saturation

$$\mathrm{sat}(v) \approxeq \eta_{sat}(v) = \frac{2\bar{u}}{\pi} \arctan\left(\frac{\pi v}{2\bar{v}}\right)$$

where $\bar{u} = u_{max}, \bar{v} = v_{\max}$ if $\frac{u_{\max}}{v_{\max}} \ge \frac{u_{\min}}{v_{\min}}$
$\bar{u} = u_{\min}, \bar{v} = v_{\min}$ if $\frac{u_{\max}}{v_{\max}} \le \frac{u_{\min}}{v_{\min}}$



1.3.1 Input saturation

Solution

The approximation is canceled, and the left approximation difference is defined as $d_{sat}(v) = \text{sat}(v) - \eta_{sat}(v)$.

Consider d_{sat} is a disturbance.

(i) Robustness (disturbance-rejection control)

The approximation difference is bounded with an upper limit

$$\left|d_{sat(v)}\right| \leq u_m\left(1 - \tanh(1)\right) = \overline{d}_{sat}.$$

(ii) Approximation

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n\eta_{sat}(v) + g_n d_{sat}$$
$$\dot{v} = -cv + w,$$

Bounded disturbance

New control input

Unknown control gain

Do not require assumptions on the uncertain parameters within a known compact set and a priori knowledge on the bound of the external disturbance.

3. Auxiliary design system

When the saturation effects are known (u and v are known)

Step n:

$\Delta u = u - v$

$\dot{V}_n = -\sum_{i=1}^{n-1} c_i z_i^2 + z_n(\ldots + f_i + g_n(v + \Delta u) - \dot{\alpha}_{n-1})$

$v = \frac{1}{g_n}(- \cdots - f_i + \dot{\alpha}_{n-1} - c_n(z_n - e))$

Substitute v into $\dot{V}_n$

$\dot{V}_n = -\sum_{i=1}^{n} c_i z_i^2 + c_n z_n e + z_n g_n \Delta u$

Lyapunov function

$V = V_n + \frac{1}{2}e^2$

$\dot{V} \leq -\sum_{i=1}^{n} c_i z_i^2 + c_n z_n e + |z_n g_n \Delta u| + e\dot{e}$

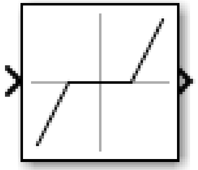
Hence, choose

$$\dot{e} = \begin{cases} -c_e e - c_n z_n - \dfrac{e}{||e||^2}|z_n g_n \Delta u|, & |e| \geq \sigma \\ 0, & |e| < \sigma \end{cases}$$

1.3.2 Input deadzone

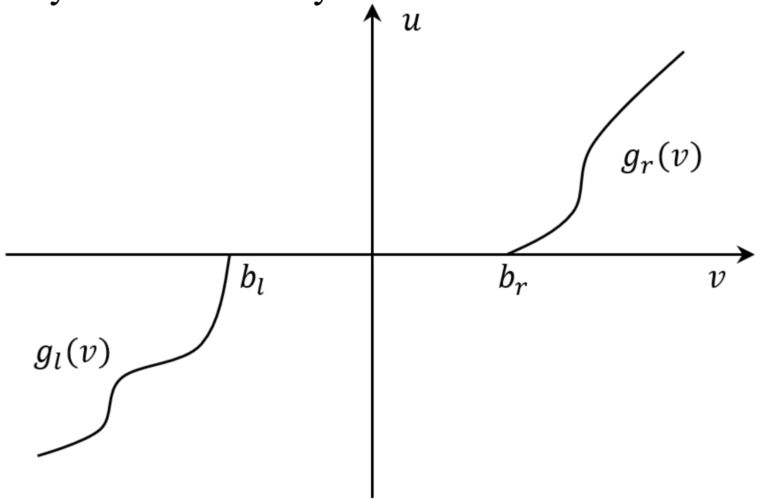
Definition

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u,$$
$$y = x_1,$$



Deadzone occurs frequently in industrial applications, e.g., gear transmission servo system, DC motor, hydraulic aircraft elevator control system, and valve.

Memoryless nonlinearity



Challenges:

- A non-differential function, and insensitive to a small control input
- Undesired chattering, which is a problem in high-precision control
- Deadzone barriers are normally unknown

General model of input deadzone:

$$u = \mathrm{Dead}(v) = \begin{cases} g_r(v), & \text{if } v \geq b_r, \\ 0, & \text{if } b_l < \theta < b_r, \\ g_l(v), & \text{if } v \leq b_l, \end{cases}$$

where g_r and g_l are the functions in the right and left parts, b_l and b_r are the barriers in the right and left parts

Complexity →		
b_l, b_r	Symmetric ($b_l = b_r$)	Asymmetric ($b_l \neq b_r$)
	Constant ($\dot{b}_l = \dot{b}_r = 0$)	Time-varying
$g_l(v)$, $g_r(v)$	Linear ($g_l = k_l v$, $g_r = k_r v$)	Nonlinear
k_l, k_r	Symmetric ($k_l = k_r$)	Asymmetric ($k_l \neq k_r$)
g_l, g_r, b_l and b_r	Known	Unknown

1.3.2 Input deadzone

Linear deadzone and known parameter

1. When g_l and g_r are linear with known k_l, k_r, b_l, and b_r

$$u = \mathrm{Dead}(v) = \begin{cases} k_r(v - b_r), & \text{if } v \geq b_r \\ 0, & \text{if } -b_l < \theta < b_r \\ k_l(v - b_l), & \text{if } v \leq b_l \end{cases}$$

where k_r and k_l are the slopes in the right and left sides.

1.1 When all the parameters are **known**, deadzone inverse is defined as $v = \mathrm{Dead}^{-1}(u)$.

A direct deadzone inverse

$$v = \mathrm{Dead}^{-1}(u) = \begin{cases} u/k_r + b_r, & \text{if } u \geq 0, \\ 0, & if u = 0, \\ u/k_l + b_l, & \text{if } v \leq 0. \end{cases} \qquad \text{(DZI-1)}$$

Remark: The deadzone inverse (DZI-1) is not a smooth function.

Smooth deadzone inverse

$$v \approxeq \eta_{dead}(u) = \frac{u + k_r b_r}{k_r}\phi_r + \frac{u + k_l b_l}{k_l}\phi_l$$

where $\phi_r = \frac{e^{u/e_0}}{e^{u/e_0} + e^{-u/e_0}}$,

$\phi_l = \frac{e^{-u/e_0}}{e^{u/e_0} + e^{-u/e_0}}$,

e_0 is a designed parameter

Other asymmetric deadzone inverse

- $k_r = k_l = k_{dz}$, $b_r = b_l = b_{dz}$

$$\eta_{dead}(v) = k_{dz} v + \frac{k_{dz}}{2\rho_{dz}} \ln \frac{\cosh \rho_{dz}(\theta_{dz} - b_{dz})}{\cosh -\rho_{dz} k_{dz}(v + b_{dz})}$$

- $k_r = k_l = k_{dz}$, $b_r \neq b_l$

$$d_{dead}(v) < k_{dz} \ln \frac{2}{\rho_{dz}}$$

$$\eta_{dead}(v) = k_{dz}\left(v + \frac{b_l - b_r}{2}\right) + \frac{k_{dz}}{2\rho_{dz}} \ln \frac{\cosh \rho_{dz}(v - b_r)}{\cosh -\rho_{dz} k_{dz}(v + b_l)}$$

- $k_r \neq k_l$, $b_r \neq b_l$

$$\eta_{dead}(v) = \frac{1}{\rho_{dz}} \ln \frac{1 + e^{\rho_{dz} k_r (v - b_r)}}{1 + e^{-\rho_{dz} k_l (v + b_l)}}$$

Approximation error $d_{dead}(v) = \mathrm{Dead}(v) - \eta_{dead}(v)$ is bounded and can be made arbitrarily small. $\lim_{\rho_{dz} \to \infty} |\eta_{dead}(v)| = 0$.

Remark: Large ρ_{dz} results in aggressive close-loop response which may degrade the system.

1.3.2 Input deadzone

Linear deadzone and unknown parameter

1.2 In practical applications, the deadzone breakpoints are always unknown.

①**Adaptive deadzone inverse**: adaptive solutions assume that the slopes and breakpoints are unknown parameters $\hat{k}_r$, $\hat{k}_l$, $\widehat{k_r b_r}$, and $\widehat{k_l b_l}$

The deadzone is divided into separate smooth regions

$$V = V_n + \tilde{\theta}_r^\top \Gamma_r^{-1}\tilde{\theta}_r + \tilde{\theta}_l^\top \Gamma_l^{-1}\tilde{\theta}_l$$

$\theta_r = [k_r, k_r b_r]^\top$ and $\theta_l = [k_l, k_l b_l]^\top$, and Γ_r and Γ_l are positive definite matrices.

② Nonlinear deadzone inverse:

Assumptions: k_l, k_r, b_l and b_r are unknown, but stay within known ranges, i.e.,
$k_{dz} \in [k_{\min}, k_{\max}]$, $b_r \in [b_{r\min}, b_{r\max}]$, and $b_l \in [b_{l\min}, b_{l\max}]$

The deadzone function can be separated into a linear term and a bounded disturbance

$$\text{Dead}(v) = k_{dz}v + d_{dead}(v)$$

$$d_{dead}(v) = \begin{cases} -k_r b_r, & \text{if } v \geq b_r, \\ -k_{dz}v, & \text{if } -b_l < v < b_r, \\ -k_l b_l, & \text{if } v \leq b_l. \end{cases} \qquad k_{dz} = \begin{cases} k_r & \text{if } v \geq b_r \\ k_l & \text{if } v \leq b_l \end{cases}$$

- Robustness-based: unknown disturbance
- Approximation-based: NN/FLS

Nonlinear deadzone

2. A more complicated case is the unknown nonlinear g_r and g_l.

Assumptions: b_r and b_l are bounded constants, g_r and g_l are smooth functions with bounded slopes.

$$\text{Dead}(v) = K^\top \Phi v + d_{dead}$$

$$K = \begin{cases} [0, g_l'(\xi(v))]^\top & \text{if } v \geq b_r, \\ [g_r'(\xi(v)), g_l'(\xi(v))]^\top & \text{if } -b_l < v < b_r r, \\ [g_r'(\xi(v)), 1]^\top & \text{if } v \leq b_l, \end{cases}$$

$$\Phi = \begin{cases} [1, 0]^\top & \text{if } v \geq b_r, \\ [1, 1]^\top & \text{if } -b_l < v < b_r r, \\ [0, 1]^\top & \text{if } v \leq b_l, \end{cases}$$

$$d(v) = \begin{cases} -g_r'(\xi(v))b_r & \text{if } v \geq b_r, \\ -g_r'(\xi(v))b_r - g_l'(\xi(v))b_l & \text{if } -b_l < v < b_r r, \\ -g_l'(\xi(v))b_l & \text{if } v \leq b_l, \end{cases}$$

- Robustness-based: Unknown disturbance
- Approximation-based: NN

$$v = \text{Dead}^{-1}(u) = u + u_{NN} \qquad u_{NN} = \begin{cases} g_l^{-1}(u), & \text{if } u < 0 \\ 0, & \text{if } u = 0 \\ g_r^{-1}(u), & \text{if } u > 0 \end{cases}$$

1.3.3 Time-varying control coefficient

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u,$$
$$y = x_1,$$

Control object: $x_1 - x_{1d} \to 0$ for $t \to \infty$;
Additional problem: Unknown control coefficient g_i
Challenges:

- Singularity problem caused by $\frac{1}{g_i}$ when $g_i = 0$

If g_i is a constant with known sign.

$V_1 = \frac{1}{g_1} z_1^2$ (If $g_1 > 0$)

$\dot{V}_1 = z_1\left(\frac{1}{g_1} f_1 + x_2\right)$

$\theta_{g1} := \frac{1}{g_1}$ $\varphi_1 := f_1$

$\dot{V}_1 = z_1\left(\theta_{g1}\varphi_1 + x_2\right)$

Then the problem is transfer to a typical adaptive backstepping.

1.3.3 Time-varying control coefficient

Integral Lyapunov functionals

To avoid the singularity problem, integral Lyapunov functionals

- $V_{1,I} = \int_0^{z_1} \sigma\beta_i(\sigma + x_{1d})d\sigma$
- $V_{i,I} = \int_0^{z_i} \sigma\beta_i(\bar{x}_{i-1}, \sigma+\alpha_{i-1})d\sigma,\ i = 2, \cdots, n$

where $\sigma = \theta z_1$ and $\beta_i = \frac{\overline{g}_i}{g_i}$.

$$\frac{1}{2}z_i^2 \leq V_{i,I} \leq \frac{z_i^2}{\underline{g}_i} \int_0^1 \theta\overline{g}(\theta z_i + \alpha_{i-1})$$

$$\dot{V}_{1,I} = z_1\,\beta_1\,\dot{z}_1 + \int_0^{z_1} \sigma \frac{\partial\beta_1(\sigma + x_{1d})}{\partial x_{1d}}\, d\sigma$$
$$= z_1\,\beta_1(f_1 + g_1x_2 - \dot{x}_{1d})$$
$$+ \dot{x}_{1d}\left[\sigma\beta_1(\sigma + x_{1d})\Big|_0^{z_1} - \int_0^{z_1} \beta_1(\sigma + x_{1d})d\sigma\right]$$
$$= z_1\,\beta_1(f_1 + g_1x_2) - \dot{x}_{1d}\int_0^{z_1} \beta_1(\sigma + x_{1d})d\sigma$$
$$= z_1\left(\beta_1 f_1 - \dot{x}_{1d}\int_0^1 \beta_1(\theta z_1 + x_{1d})d\theta + g_1\beta_1 x_2\right)$$

Choose $z_1\alpha_1 = \mathcal{N}(\chi)\dot{\chi}$

$$\dot{\chi} = c_1z_1^2 + z_1\left(\beta_1 f_1 - \dot{x}_{1d}\int_0^1 \beta_1(\theta z_1 + x_{1d})d\theta\right)$$

$$\dot{V}_{1,i}$$
$$= g_1\beta_1\mathcal{N}(\chi)\dot{\chi} + \overbrace{z_1\left(\beta_1 f_1 - \dot{x}_{1d}\int_0^1 \beta_1(\theta z_1 + x_{1d})d\theta\right) + c_1z_1^2}^{\dot{\chi}} - c_1z_1^2$$
$$+ z_1g_1\beta_1z_2$$

1.3.4 Actuator failure

Introduction

Actuator failures change the output and parameters, introduce additional system uncertainties and disturbances, and result in performance deterioration and even accidents

Challenge: A failure is normally uncertain in time and often unrecoverable

(i) Model-based redundancy approach for fault-tolerant control based on a bank of residual signals generated by multiple online monitoring modules running in parallel with specific possible failures. If the failures are not contained in the bank, the performance is unreliable.

(ii) Adaptive failure compensation design without explicit failure detection, remains the same structure through the running. (We talk here)

Objective: Compensate for the effects of reasoning from the actuator failures, and meanwhile, to ensure the asymptotic tracking performance with a bounded error.

Assumptions (redundant actuators):
The remaining actuators are fully actuated, and the desired control objective is still achievable for up to $m-1$ actuator faults for an SISO system

$$u = [u_1, \cdots, u_m]^\top$$

This assumption ensures the controllability of the plant with the remaining actuation power and the existence of a normal solution for the actuator failure compensation problem.

Failures:
- Total loss of effectiveness (TLOE)
- Partial loss of effectiveness (PLOE)

1.3.4 Actuator failure

Failure models

Static actuator failure models

1. If failure for the j^{th} actuator occurs at t_j

$$u_{j(t)} = \bar{u}_j,\ \forall t \geq t_j, j = 1, \cdots, m,$$

where $\bar{u}_j$ and t_j are unknown

$$u = \sigma\bar{u} + (I_m - \sigma)v$$

where $\sigma = \text{diag}\sigma_1, \cdots, \sigma_m$

failure patterns $\sigma_j = \begin{cases} 1, \text{if the j}^{\text{th}} \text{ actuator fails} \\ 0, \text{otherwise} \end{cases}$

2. Actuator model with both gain fault and bias fault

$$u_j = \rho_j v_j + b_{uj}$$

Failure-free: $\rho_j = 1$ and $b_{uj} = 0$
PLOE: $\rho_j \in (0,1)$ and $b_{uj} = 0$
TLOE: $\rho_j = 0$ and $b_{uj} = 0$
Bias fault: $\rho_j = 0$ and $b_{uj} \neq 0$

Dynamic actuator failure models

3. First-order dynamic actuator failure model

$$\dot{u}_j = -(1 - \sigma_j)\lambda_j(u_j - k_j v_j)$$

second- order dynamic actuator failure model

$$\dot{u}_{1j} = u_{2j}$$

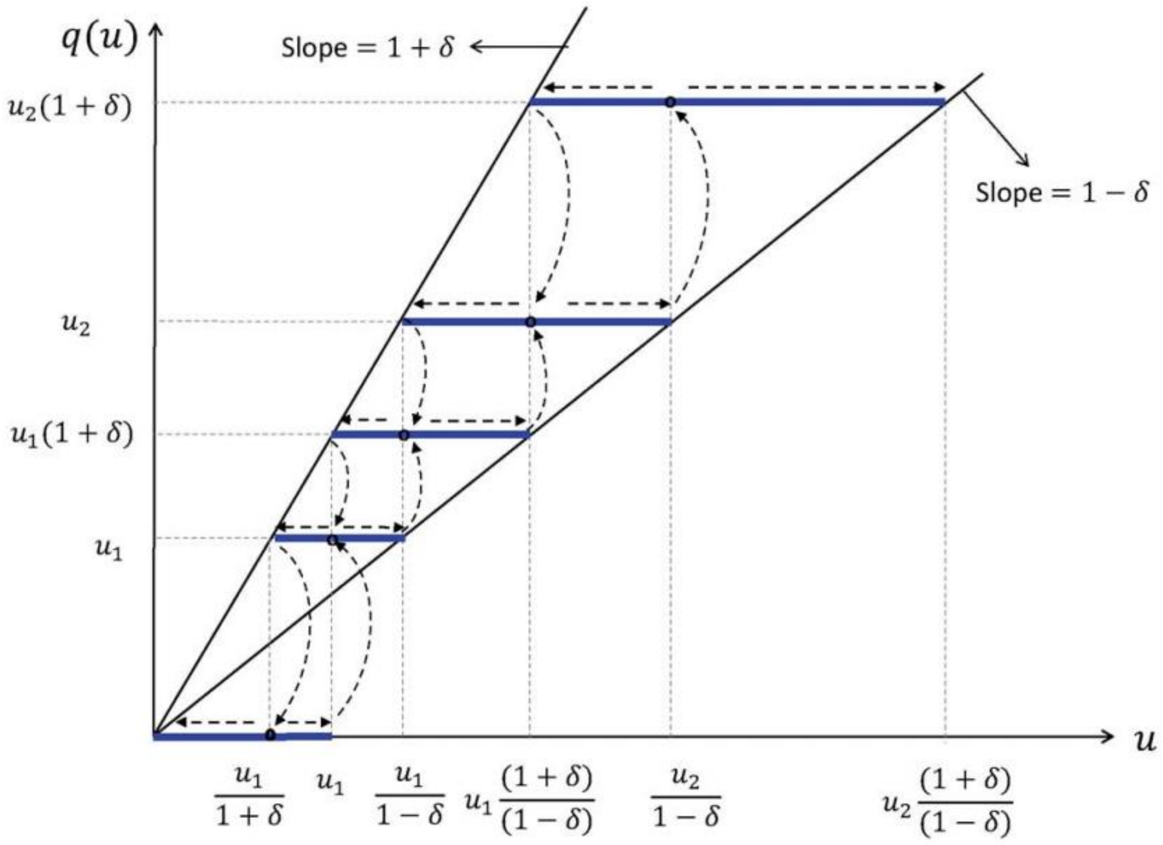
$$\dot{u}_{2j} = -\left(\lambda_{2j} + \sigma_{j\beta_j}\right)u_{2j} + (1 - \sigma_j)\lambda_{1j}\,(k_j v_j - u_{1j})$$

$\lambda_j \gg 1$, $\lambda_{1j} \gg 1$, $\lambda_{1j} \gg \lambda_{2j}$, and $\lambda_{2j} + \beta_j \gg 1$

Failure-free: $\sigma_j = 0$ and $k_j = 1$
PLOE: $\sigma_j = 0$ and $k_j \in (0,1)$
TLOE: $\sigma_j = 1$

1.3.5 Another example

Quantization



$$q(u(t)) = \begin{cases} u_i \operatorname{sgn}(u), & \frac{u_i}{1+\delta} < |u| \le u_i,\ \dot{u} < 0, \text{ or} \\ & u_i < |u| \le \frac{u_i}{1-\delta},\ \dot{u} > 0 \\ u_i(1+\delta)\operatorname{sgn}(u), & u_i < |u| \le \frac{u_i}{1-\delta},\ \dot{u} < 0, \text{ or} \\ & \frac{u_i}{1-\delta} < |u| \le \frac{u_i(1+\delta)}{(1-\delta)},\ \dot{u} > 0 \\ 0, & 0 \le |u| < \frac{u_{\min}}{1+\delta},\ \dot{u} < 0 \text{ or} \\ & \frac{u_{\min}}{1+\delta} \le u \le u_{\min},\ \dot{u} > 0, \\ q\left(u(t^-)\right) & \dot{u} = 0 \end{cases}$$

$$q(u(t)) = u(t) + d(t) \tag{5}$$

where $d(t) = q(u(t)) - u(t) \in \Re^1$. Regarding the nonlinearity $d(t)$, we have the following lemma.

Lemma 1: The nonlinearity $d(t)$ satisfies the following inequality:

$$d^2(t) \le \delta^2 u^2,\ \forall |u| \ge u_{\min}. \tag{6}$$

$$d^2(t) \le u_{\min}^2,\ \forall |u| \le u_{\min}. \tag{7}$$

=>Bounded d

Top journal is easy if you find a undone nonlinearity ☺:
Zhou, J., Wen, C. and Yang, G., 2013. Adaptive backstepping stabilization of nonlinear uncertain systems with quantized input signal. IEEE Transactions on Automatic Control, 59(2), pp.460-464.

1.4 Time-delay effects

Widely existing in chemical systems, biological systems, economic systems, and hydraulic/pneumatic systems

delayed time are often unknown, which can be a constant value for all parameters, constant different values for various parameters, and time-varying.

- Lyapunov-Razumikhin
- Lyapunov-Krasovski (more common)

Lyapunov-Krasovski approach is predictor-like technique according to **Lyapunov-Krasovski Theorem**

Theorem 1 (Lyapunov-Krasovski Theorem). *Let* $f : \mathbb{R} \times \mathcal{C}_n([-r,0]) \to \mathbb{R}$ *map* $\mathbb{R}\times$*(bounded subsets of* $\mathcal{C}_n([-r,0])$*) into bounded subsets of* $\mathbb{R}^n$*. Let* u, v, $w : [0,\infty) \to [0,\infty)$ *be continuous non-decreasing functions for which* u *and* v *are positive definite and* v *is increasing. Assume the following:*

1. *There exists a continuously differentiable function* $V : \mathbb{R} \times \mathcal{C}_n([-r,0]) \to \mathbb{R}$ *such that*

$$u(|\phi(0)|) \leq V(t,\phi) \leq v(|\phi|[-r,0]), \qquad (1)$$

and $\dot{V}(t,\phi) \leq -w(|\phi(0)|)$*, for all* $\phi \in \mathcal{C}_n([-r,0])$ *and* $t \in \mathbb{R}$*. Then the trivial solution is uniformly stable. If, in addition,*

2. $w(s) > 0$ *for all* $s > 0$*, then the system is uniformly asymptotically stable. Finally, if 1. and 2. hold and if we also have* $\lim_{s\to+\infty} u(s) = +\infty$*, then the system is UGAS.*

Just forget this.

1.4.1 State delay

Introduction

$$\dot{x}_i(t) = f_i(\bar{x}_i(t)) + f_{di}(\bar{x}_i(t-\tau_{di})) + g_i(\bar{x}_i(t))x_{i+1}(t),$$
$$\dot{x}_n(t) = f_n(\bar{x}_n(t)) + f_{dn}(\bar{x}_n(t-\tau_{dn})) + g_n(\bar{x}_n(t))u(t),$$

where τ_{di}>0 denotes the delayed time,
$f_{di}(\bar{x}_i(t-\tau_{di}))$are the time-delay terms, and
$\bar{x}_i(t-\tau_{di}) = [x_1(t-\tau_{d1}), x_2(t-\tau_{d2}), \cdots, x_i(t-\tau_{di})]^\top$.

From simple to complex:

- $\tau_{d1} = \cdots = \tau_{dn} = \tau_d$
- $\tau_{d1} \neq \cdots \neq \tau_{dn} \neq \tau_d$

- Known delay
- Unknown delay

Complexity

Assumptions 1(bounded delay):
Unknown time delays are bounded by a known constant, i.e., $\tau_{di} \leq \tau_{d,\max}$

Parameter separation

Assumptions 2(bounded parametric time-delayed terms):
The absolute value of the time-delay term is bounded by known smooth functions $\rho_i(\bar{x}_i)$in several parameter-separation forms

- $|f_{di}(\bar{x}_i(t-\tau_{di}))| \leq \sum_{j=1}^{n} \rho_j(\bar{x}_j)$
- $|f_{di}(\bar{x}_i(t-\tau_{di}))| \leq \rho_i(\bar{x}_i(t-\tau_{di}))$
- $|f_{di}(\bar{z}_i(t-\tau_{di}))| \leq \sum_{j=1}^{i} |z_j(t-\tau_{dj})| \rho_{ij}(\bar{z}_j(t-\tau_{dj}))$
- $|f_{di}(\bar{x}_i(t-\tau_{di}))| = \theta_{di}^\top \phi_{di}(\bar{x}_i(t-\tau_{di})) + \delta_{di}(\bar{x}_i(t-\tau_{di}))$

$\rho_{ij}(\cdot)$is a known continuous and smooth function, $\theta_{di} \in \mathbb{R}^{n_i}$ $\phi_{di}: \mathbb{R}^i \to \mathbb{R}^{n_i}$is a known smooth function vector, δ_{di} is a bounded unknown smooth function, i.e., $|\delta_{di}(\bar{x}_i(t-\tau_{di}))| \leq c_{di}\rho_i(\bar{x}_i(t-\tau_{di}))$ where c_{di} is an unknown constant

A basic integrate-type Lyapunov-Krasovskii functional and its derivative are given by

$$V_{i,LK} = \int_{t-\tau_d}^{t} S_i(\bar{z}_i(\sigma))d\sigma, \tag{1a}$$
$$\dot{V}_{i,LK} = S_i(t) - S_i(t-\tau_d), \tag{1b}$$

where $S_i(\bar{z}_i(t))$ is a positive definite function, e.g., $S_i(\bar{z}_i(t)) = \rho_i^2(\bar{z}_i(t))$.

1.4.1 State delay

Example

$$\dot{x}_i(t) = f_i(\bar{x}_i(t)) + f_{di}(\bar{x}_i(t-\tau_{di})) + g_i(\bar{x}_i(t))x_{i+1}(t),$$
$$\dot{x}_n(t) = f_n(\bar{x}_n(t)) + f_{dn}(\bar{x}_n(t-\tau_{dn})) + g_n(\bar{x}_n(t))u(t),$$

Assumption: $|f_{di}(\bar{z}_i(t-\tau_{di}))| \le \sum_{j=1}^{i} |z_j(t-\tau_{dj})| \rho_{ij}(\bar{z}_j(t-\tau_{dj}))$

$$V_{i,LK} = \int_{t-\tau_d}^{t} S_i(\bar{z}_i(\sigma))d\sigma$$
$$\dot{V}_{i,LK} = S_i(t) - S_i(t-\tau_d)$$

Time-delay effects

LFC

$$V_1 = V_{1,QF}(z_1(t)) + V_{1,KF}$$

Time derivative:

$$\begin{aligned}\dot{V}_1 =& z_1(t)[f_1(\bar{x}_1(t)) + f_{d1}(\bar{x}_1(t-\tau_d)) + g_1(\bar{x}_1(t))x_2] \\ &+ S_1(\bar{z}_i(t)) - S_1(\bar{z}_i(t-\tau_d)) \\ \le& z_1(f_1 + g_1x_2) + z_1(t)|z_1(t-\tau_{d1})|\rho_{11}(\bar{z}_1(t-\tau_{d1})) \\ &+ S_1(\bar{z}_i(t)) - S_1(\bar{z}_i(t-\tau_d)) \\ \le& z_1(f_1+g_1x_2) + \frac{1}{2}z_1^2(t) + \frac{1}{2}z_1^2(t-\tau_{d1})\rho_{11}^2(\bar{z}_1(t-\tau_{d1})) \\ &+ S_1(\bar{z}_i(t)) - S_1(\bar{z}_i(t-\tau_d))\end{aligned}$$

To compensate the effects of the time-delay term with $t-\tau_d$, $S_1(\cdot)$ is designed as follows

$$S_1(z_1(\sigma)) = \frac{1}{2}z_1^2(\sigma)(1+\lambda_{11})\rho_{11}^2(z_1(\sigma))$$

Then, the virtual control is

$$\alpha_1 = -\frac{1}{g_1}\left[\kappa(z_1(t)) + f_1 + \frac{1}{2} + \lambda_{12}z_1(t)\right]$$

Substituting $S_1(\sigma)$ and α_1 into the $\dot{V}_1$ yields

$$\begin{aligned}\dot{V}_1 \le& -\kappa(z_1)z_1 + z_1\bar{g}_1z_2 - \lambda_{12}z_1^2(t) + \frac{1}{2}z_1^2(t)(1+\lambda_{11})\rho_{11}^2(z_1(t)) \\ &- \frac{\lambda_{11}}{2}z_1^2(t-\tau_d)\rho_{11}^2(z_1(t-\tau_d))\end{aligned}$$

where λ_{11}, λ_{12} - coefficients to be designed to compensate the delayed terms in steps 2-n. Since

$z_i f_{di}(\bar{z}_i(t-\tau_{di})) \le \frac{1}{2}\sum_{j=1}^{i} z_j^2(t-\tau_{dj})\rho_{ij}(\bar{z}_j(t-\tau_{dj}))$,

S_i should contain $x_1, \dots, x_i$.

Remark 1: The control law does not depend on the delayed time and is similar to a classic backstepping design. The control gain is higher.
Remark 2: In another words, if the gain is large enough, the effects of the time delay is limited.

Nguang, S. K. (2000). Robust stabilization of a class of time-delay nonlinear systems. *IEEE Transactions on Automatic Control*, *45*(4), 756-762.

1.4.2 Input delay

Introduction and short delay: Emulation

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u(t - \tau_d).$$

- Short delay
- Long delay
- Arbitrarily long delay

The controller is first designed without delay, then the upper bound of the delay is found such that the closed-loop system is still asymptotically stable.

Design controller for a complex System A → Design controller fora similar but simpler System B → Implement controller to a system A

- Continuous-time system ⇐Discrete-time system
- Real-time system ⇐ System with input delay
- …

An example:

Lemma 2.2: Consider the system

$$\dot{Z}(t) = -\varepsilon Z(t - \tau) \qquad (21)$$

where $Z \in R$ and τ and ε are positive real numbers such that $\varepsilon \in (0, 1/2\tau]$. The origin of this system is globally uniformly asymptoti-

Mazenc, F. and Bliman, P.A., 2006. Backstepping design for time-delay nonlinear systems. IEEE Transactions on Automatic Control, 51(1), pp.149-154.

1.4.2 Input delay

Short delay: Pade approximation approach

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u(t - \tau_d).$$

$$\mathcal{L}(u(t-\tau_d)) = \exp(-\tau_d s)\,\mathcal{L}(u(t))$$
$$= \frac{\exp\left(-\frac{1}{2}\tau_d s\right)}{\exp\left(\frac{1}{2}\tau_d s\right)}\mathcal{L}(u(t))$$

(1st-order Taylor polynomial) $\approx \frac{1-\frac{1}{2}\tau_d s}{1+\frac{1}{2}\tau_d s}\mathcal{L}(u(t)) \;\; = \mathcal{L}(x_{n+1} - \mathrm{u}(t))$

$$\Rightarrow \left(1-\tfrac{1}{2}\tau_d s\right)\mathcal{L}(u(t)) = \left(1+\tfrac{1}{2}\tau_d s\right)\mathcal{L}(x_{n+1}) - \left(1+\tfrac{1}{2}\tau_d s\right)\mathcal{L}(u(t))$$
$$\Rightarrow 2\mathcal{L}(u(t)) = \left(1+\tfrac{1}{2}\tau_d s\right)\mathcal{L}(x_{n+1})$$
$$\Rightarrow \tfrac{1}{2}\tau_d s\mathcal{L}(x_{n+1}) = -\mathcal{L}(x_{n+1}) + 2\mathcal{L}(u(t))$$
$$\Rightarrow s\mathcal{L}(x_{n+1}) = -\gamma_d\mathcal{L}(x_{n+1}) + 2\gamma_d\mathcal{L}(u(t)), \text{ where } \gamma_d = \frac{2}{\tau_d}$$
$$\Rightarrow \dot{x}_{n+1} = -\gamma_d x_{n+1} + 2\gamma_d u(t)$$

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)(x_{n+1} - u),$$
$$\dot{x}_{n+1} = -\gamma_d x_{n+1} + 2\gamma u,$$

- Works for unknown delay
- Only works for short delay due to the 1st-order Taylor expression can be less accurate when $\frac{1}{2}\tau_d s$ increases.

Laplace transform from Wikipedia

delayed impulse	$\delta(t-\tau)$	$e^{-\tau s}$		time shift of unit impulse

Exponential function [edit]

The exponential function e^x (with base e) has Maclaurin series

$$e^x = \sum_{n=0}^{\infty}\frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots.$$

It converges for all x.

Deduction tips:

$$z_1 = x_1 - x_{1d}$$
$$z_i = x_i - \alpha_{i-1}$$
$$z_n = x_n - \alpha_{n-1} + \frac{g_n}{\gamma_d}x_{n+1}$$

Step n:

$$\dot{z}_n = z_1\left(f_n + g_n(x_{n+1} - u) - \dot{\alpha}_{n-1} + \frac{g_n}{\gamma_d}(-\gamma_d x_{n+1} + 2\gamma_d u(t))\right)$$
$$\dot{z}_n = z_1(f_n + g_n u(t) - \dot{\alpha}_{n-1})$$

Khanesar, M.A., Kaynak, O., Yin, S. and Gao, H., 2014. Adaptive indirect fuzzy sliding mode controller for networked control systems subject to time-varying network-induced time delay. IEEE Transactions on Fuzzy Systems, 23(1), pp.205-214.

1.4.2 Input delay

Arbitrarily long input delay

$$\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1},$$
$$\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u(t - \tau_d).$$

Arbitrarily long delay:

- Prediction-based boundary control
- Predictor feedback

used on linear time-invariant, finite-dimensional, and completely controllable system

The effects of the delay is compensated with an integration over the delay period. The delay time is estimated by a time-delay identifier when using the prediction-based boundary control.

The main idea of the predictor feedback approach is to modeled the actuator time delay effects as a transport partial differential equation.

However, the distributed terms may not always be easy to compute. Furthermore, this approach is not applicable to nonlinear system due to the inconvenience integration over the delay interval.

A time-delay system is transformed into another dynamics with a delayed system state as its input

1.5 Pure feedback system

Taylor series expansion

$$\dot{x}_i = f_i(\bar{x}_i, x_{i+1}), i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n, u),$$
$$y = x_1,$$

Control object: $x_1 - x_{1d} \to 0$ for $t \to \infty$;
Additional problem: System is not in a strict-feedback form
Challenges:

- No affine appearance of the variables to be used as virtual control (no g_i)

Idea: Transfer into a form with explicit g_i

1. Taylor series expansion

When the system has a strong relative degree and can be transformed into an integrator chain.

$$\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}, u) \quad \Rightarrow \quad \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}q(\mathbf{x}, u)$$
$$y = h(\mathbf{z}) \qquad\qquad y = \mathbf{C}^T\mathbf{x}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\dot{x}_n = q(\mathbf{x}, \bar{u}) + g(\bar{\mathbf{e}})(u - \bar{u}) + d_h$$

where $g(\bar{\mathbf{e}}) = \partial q(\mathbf{x}, u)/\partial u|_{u=\bar{u}(\bar{\mathbf{e}})}$, and d_h stands for higher order term. Suppose a control input u is

$$u = u_f + u_s$$

$$\Rightarrow \quad \dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{B}\mathbf{K}_c^T\hat{\mathbf{e}} + \mathbf{B}[g(\bar{\mathbf{e}})\bar{u} - g(\bar{\mathbf{e}})u_f - g(\bar{\mathbf{e}})u_s + \hat{\zeta} - d]$$
$$e_1 = \mathbf{C}^T\mathbf{e}$$

The high-order terms are modeled as disturbances.

1.5 Pure feedback system

Mean value theorem

$$\dot{x}_i = f_i(\bar{x}_i, x_{i+1}), i \in \mathcal{I}$$
$$\dot{x}_n = f_n(\bar{x}_n, u),$$
$$y = x_1,$$

Control object: $x_1 - x_{1d} \rightarrow 0$ for $t \rightarrow \infty$;
Additional problem: System is not in a strict-feedback form
Challenges:
- No affine appearance of the variables to be used as virtual control (no g_i)

Idea: Transfer into a form with explicit g_i

Mean value theorem
Let $f: [a, b] \rightarrow \mathbb{R}$ be a continuous function on the closed interval $[a, b]$, and differentiable on the open interval (a, b), where $a < b$. Then there exists some $c \in (a, b)$ such that

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$

$$\Rightarrow f(b) = f(a) + f'(c)(b - a)$$

Assumptions:$0 < \underline{g}_i \leq |g_i(\bar{x}_i, \boldsymbol{x}_{i+1})| \leq \bar{g}_i$

Error state:
$$z_1 = x_1 - x_{1d}$$
Error dynamics:
$$\dot{z}_1 = f_i(\bar{x}_1, x_2) - \dot{x}_{1d}$$
There exists $x_2 = \alpha_1^*$, s.t.,
$$0 = f_i(\bar{x}_1, \alpha_1^*) - \dot{x}_{1d}$$
Mean value theorem: there exists
$$g_{1\mu_1} = g_1\,(\bar{x}_1, \mu_1 x_2 + (1 - \mu_1)\alpha_1^*),$$
with $\mu_1 \in (0,1)$, s.t.,
$$f_1(\bar{x}_1, x_2) = f_i(\bar{x}_1, \alpha_1^*) + g_{1\mu_1}(x_1 - \alpha_1^*)$$

Then the error dynamics is
$$\dot{z}_1 = f_1(\bar{x}_1, x_2) - \dot{x}_{1d} = \cancel{f_1(\bar{x}_1, \alpha_1^*)} + g_{1\mu_1}(x_2 - \alpha_1^*) \cancel{- \dot{x}_{1d}}$$
$$= g_{1\mu_1}(\alpha_1 + z_2 - \alpha_i^*)$$
where α_1^* is unknown.

- According to the assumption, $g_{i\mu_i}$ is bounded.
- Backstepping design based on unknown $\alpha_1^* \Rightarrow$ approximation methods.

1.6 Event-triggered systems

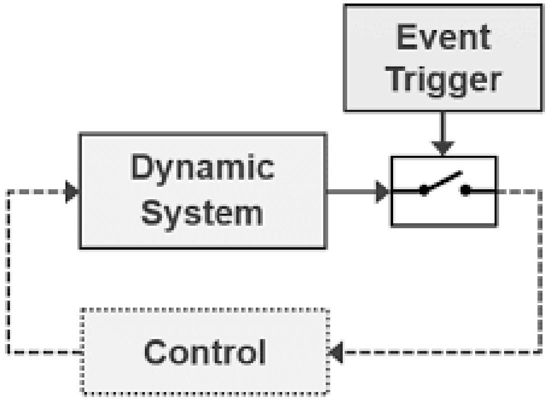
Introduction

The event-triggered control, or event-based system, denotes a system updates aperiodically with the pre-designed event-triggered condition.

Main feature
The execution of control tasks update after the occurrence of an event, and the control input (or measurements) is held between two consecutive updates (zero-order hold).

Advantages

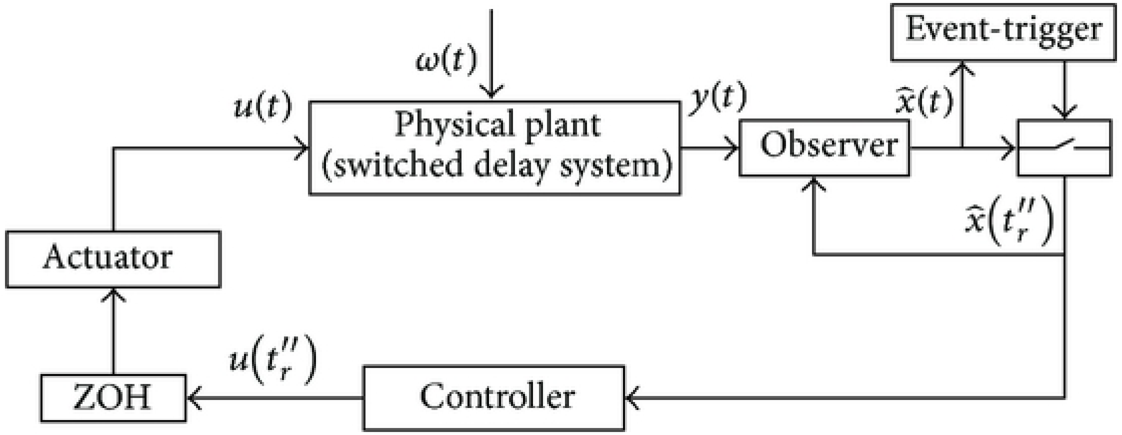
- A more natural sampling way, similar to a human controller.
- Reducing network traffic loads
- Improving resource utilization with minor control performance degradation



Categorization (according to which part is event-triggered)

- Event-triggered control input: Constant control input between two triggered instants, i.e., $u(t) = v(t_k)$ for all $t \in [t_k, t_{k+1})$.

- Event-sampled: State-measurements are event-triggered. the measurements are considered as a jump, i.e., $\hat{x}_{i(t)} = x_{i(t_k)}$, for all $t \in [t_k, t_{k+1})$.



1.6 Event-triggered systems

Event-triggered control input

Event-triggered condition

The event-triggered conditions are designed as the specific error $e_{et} := u(t) - v(t_k)$ is larger than a preset threshold function. The thresholds can be fixed and state-dependent.

- Fixed threshold strategy:

$$t_{k+1} = \inf\{t > t_k \,|\, |e_{et}(t)| > \delta_{et}\} \qquad (1)$$

- State-dependent threshold 1:

$$t_{k+1} = \inf\{t > t_k \,|\, |e_{et}(t)| > \kappa_{et}|u(t)|\} \qquad (2)$$

- State-dependent threshold 2:

$$t_{k+1} = \inf\{t > t_k \,|\, |e_{et}(t)| > \kappa_{et}|u(t)| + \delta_{et}\} \qquad (3)$$

where $\kappa_{et} \in (0,1)$ and $\delta_{et} > 0$.

An example of event-triggered condition (3)

$|u(t) - v(t)| \leq \kappa_{et}|u(t)| + \delta_{et}$

There must $\exists\, \bar{\lambda}_1, \bar{\lambda}_2 \in [0,1]$, s.t.,

Case 1: If $u(t) - v(t) \geq 0$

$$u(t) - v(t) = \bar{\lambda}_1 \kappa_{et}\,\mathrm{sgn}(u(t))\,u(t) + \bar{\lambda}_2 \delta_{et}$$

$$\Rightarrow v(t) = \left(1 - \bar{\lambda}_1 \kappa_{et}\,\mathrm{sgn}(u(t))\right)u(t) - \bar{\lambda}_2 \delta_{et}$$

Case 2: If $u(t) - v(t) < 0$

$$v(t) - u(t) = \bar{\lambda}_1 \kappa_{et}\,\mathrm{sgn}(u(t))\,u(t) + \bar{\lambda}_2 \delta_{et}$$

$$\Rightarrow v(t) = \left(1 + \bar{\lambda}_1 \kappa_{et}\,\mathrm{sgn}(u(t))\right)u(t) + \bar{\lambda}_2 \delta_{et}$$

$$\exists \lambda_1, \lambda_2 \in [0,1], s.t., v(t) = (1 + \kappa_{et}\lambda_1)u(t) + \lambda_2 \delta_{et}$$

Therefore, $u(t) = \frac{v(t)}{1+\kappa_{et}\lambda_1} - \frac{\lambda_1 \delta_{et}}{1+\kappa_{et}\lambda_1}$

$\left|\frac{\lambda_1 \delta_{et}}{1+\kappa_{et}\lambda_1}\right| \leq \frac{\delta_{et}}{1+\kappa_{et}}$ - bounded unknown disturbance

Backstepping design is solved.☺

1.7 Stochastic system

Introduction and stability

Stochastic system:

$$dx = f(x,u)dt + h_w(x,u)dw(t)$$

where $w \in \mathbb{R}^r$ is r-dimensional independent standard Wiener process, $f(x,u)\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h_{w(x,u)}\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{\mathrm{n\times r}}$ are locally Lipschitz for all $t \geq 0$ satisfying $f(0,0) = 0$ and $h_{w(0,0)} = 0$.

- In deterministic models, the output of the model is fully determined by the parameter values and the initial conditions.
- Stochastic models possess some inherent randomness. The same set of parameter values and initial conditions will lead to an ensemble of different outputs.

Different from the disturbance d_i whose amplitude is assumed to be bounded, the magnitude of the disturbance in a stochastic system can be arbitrarily large in sufficiently long period. Hence, stabilities and properties are defined in probability.

The equilibrium point $x(0) = 0$ is said to be

- **Stable in probability** if, for every $\varepsilon > 0$ and $\delta > 0$, there exists an r s.t. if $t > t_0$, $|x_0| < r$ and $i_0 \in S$, then $P\{|x(t)| > \varepsilon < \delta$
- **Asymptotically stable in probability** if it is stable in probability and, for each $\varepsilon > 0$, $x \in \mathbb{R}^n$ and $i_0 \in S$, there is $\lim_{t\to\infty} P\{|x(t)| > \varepsilon = 0$
- **Bounded in probability** if the random variable $|x(t)|$ are bounded in probability uniformly in t, i.e.,

$$\lim_{t\to\infty} \sup_{t>t_0} P\{|x(t)| > R = 0$$

1.7 Stochastic system

LFC and deduction

Stochastic system:

$$dx = f(x,u)dt + h_w(x,u)dw(t)$$

Instead of $\dot{V}$, a new operator is defined as

$$\mathcal{L}V(x) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f + \frac{1}{2}\mathrm{Tr}[h_w^\top \frac{\partial^2 V}{\partial x^2} h_w]$$

where higher-order Hessian term $\frac{1}{2}\mathrm{Tr}\left(h_w^\top \frac{\partial^2 V}{\partial x^2} h_w\right)$ is due to the stochastic noises and $\mathrm{Tr}(\cdot)$ is the trace operator.

Lyapunov-based stability criteria:

- $\mathcal{L}V \leq 0$ Globally asymptotically stable in probability
- $\mathcal{L}V \leq -\gamma V(x,t) + \delta$ Bounded in probability
- $\mathcal{L}V + cV^\alpha \leq 0$ Finite-time stability
- $\mathcal{L}V \leq -\gamma V(x,t) + \frac{1}{c}(g_n\mu_1\mathcal{N}(\chi) + 1)\dot{\chi}\delta$ Nussbaum-type function

=> $E[V(x,t)] \leq V(x_0)e^{-\gamma t} + \frac{\delta}{\gamma} + \frac{\sigma}{c}$

where $\sigma = \sup \int_0^t |E(g_n\mu_1\mathcal{N}(\chi) + 1)\dot{\chi}e^{\gamma\tau}|\, d\tau$

Key steps in the deduction:

- Error states:

$$dz_1 = dx_1 - dx_{1d} = dx_1 - \dot{x}_{1d}dt$$

- To handle the newly involved term $\frac{\partial^2 V}{\partial x^2}$, quartic Lyapunov functions are used:

$$\frac{1}{4}z_i^4 \text{ and } \frac{1}{4}\log\left(\frac{k_{bi}^4}{k_{bi}^4 - z_i^4}\right) \text{ (BLF)}$$

- $\frac{\partial^2 V}{\partial x^2} \Rightarrow x^2$ in $\mathcal{L}V$
- Young's inequality: $x^2 \rightarrow x^4$ in $\mathcal{L}V$, compensate by α_i
- $\mathcal{L}V \leq -\gamma V(x,t) + \delta$

1.8 Topics not included in this lecture

Fractional-order system

$$
\begin{aligned}
&\mathcal{D}^{\alpha_1} x_1 = d_1 x_2 + \psi_1(x_1) + \varphi_1^{\mathrm{T}}(x_1)\theta, \\
&\mathcal{D}^{\alpha_2} x_2 = d_2 x_3 + \psi_2(x_1, x_2) + \varphi_2^{\mathrm{T}}(x_1, x_2)\theta, \\
&\vdots \\
&\mathcal{D}^{\alpha_{n-1}} x_{n-1} = d_{n-1} x_n + \psi_{n-1}(x_1, \ldots x_{n-1}) + \varphi_{n-1}^{\mathrm{T}}(x_1, \ldots x_{n-1})\theta, \\
&\mathcal{D}^{\alpha_n} x_n = bu + \psi_n(x) + \varphi_n^{\mathrm{T}}(x)\theta, \\
&y = x_1,
\end{aligned}
$$

PDE system

$$\partial_t x(z,t) = b(z)\partial_z^2 x(z,t) + c(z)\partial_z x(z,t) + d(z,t)x(z,t)$$

The word "order" may be misleading. It could denote n, $\frac{d^r x}{dt^r}$, and x_{i+1}^r

2.1 Underactuated system

Underactuated: The number of the control inputs u is less than that of state x_i

Idea: An algebraic transformation is utilized to convert the system into a cascade system or a reduced-order strict feedback form with a sliding surface as the error state z_1

Only specific classes of systems can be solved. For example:

A second-order Lagrangian system

$$m_{11}\ddot{q}_1 + m_{12}\ddot{q}_2 + h_1(q,p) = 0, \quad (1)$$
$$m_{21}\ddot{q}_1 + m_{22}\ddot{q}_2 + h_2(q,p) = B(q)\tau, \quad (2)$$

where $\dot{q}_1 = p_1 \in \mathbb{R}^m$, $\dot{q}_2 = p_2 \in \mathbb{R}^n$, $h_1(q,p)$=0 when $q = p = 0$.

Key deduction process (block backstepping)

(1) => $\ddot{q}_1 = -m_{11}^{-1}m_{12}\ddot{q}_2 - m_{11}^{-1}h_1$

$\ddot{q}_1 \rightarrow$(2) => $m_{21}(-m_{11}^{-1}m_{12}\ddot{q}_2 - m_{11}^{-1}h_1) + m_{22}\ddot{q}_2 + h_2 = B\tau$

$\ddot{q}_2 = (m_{22} - m_{21}m_{11}^{-1}m_{12})^{-1}(B\tau + m_{21}m_{11}^{-1}h_1 - h_2) = u$

$\ddot{q}_2 \rightarrow$(1) => $\ddot{q}_1 = -m_{11}^{-1}m_{12}u - m_{11}^{-1}h_1$

New control input

Resulting system:

$$\dot{q}_1 = p_1,$$
$$\dot{q}_2 = p_2,$$
$$\dot{p}_1 = f(q,p) + g(q)u,$$
$$\dot{p}_2 = u,$$

where $f(q,p) = -m_{11}^{-1}(q)\, h_1\,(q,p)$ and $g(q) = -m_{11}^{-1}(q)m_{12}(q)$.

Define the error state: $z_1 = q_2 - K(q_1 + p_1 - gp_2)$

The error dynamics:

$$\dot{z}_1 = p_2 - K(p_1 + f \cancel{+ gu - gu}) = p_2 - K(p_1 + f)$$

Virtual control: $\alpha_1 \rightarrow p_2$

Error state: $z_2 = p_2 - \alpha_1$

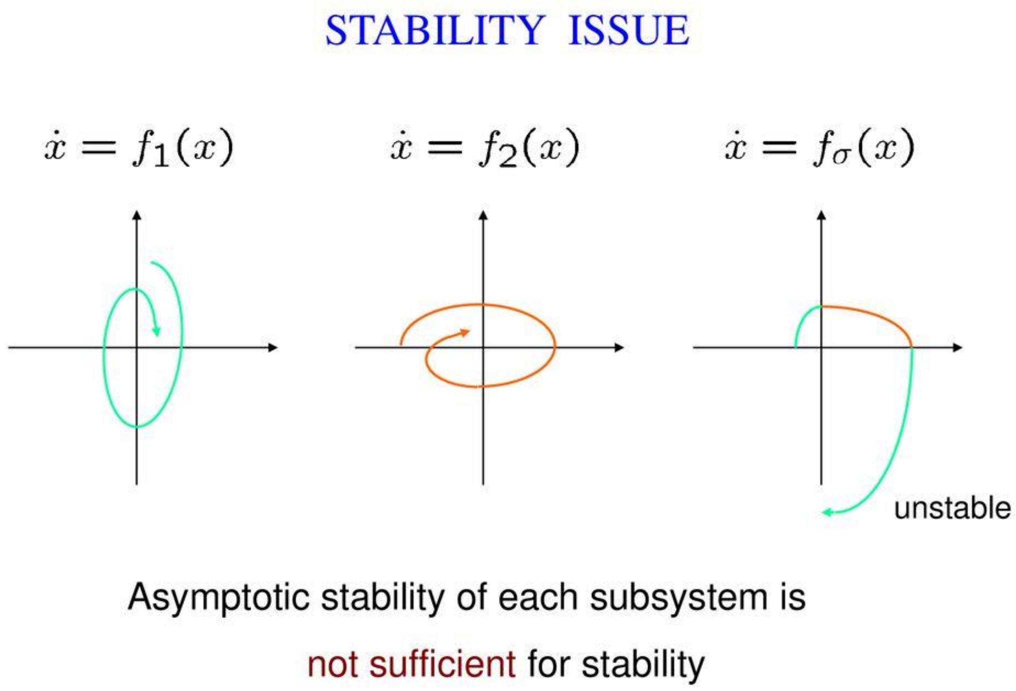
…

A disturbance in presence in z_1 if $h_1(q,p) \neq 0$ when $q = p = 0$, resulting in a bias existed in the results. It is noticed that such design has a poor tracking performance due to time-varying equilibrium.

2.2 Switched system

$$\dot{x}_i = f_{\sigma(t),i}(\bar{x}_i) + g_{\sigma(t),i}(\bar{x}_i)x_{i+1}, i \in \mathcal{I}$$
$$\dot{x}_n = f_{\sigma(t),n}(\bar{x}_n) + g_{\sigma(t),n}(\bar{x}_n)u,$$

where $\sigma(t)\colon \mathbb{R}_+ \rightarrow \Xi = \{1, \cdots, N$ is the piecewise continuous switching signal.



Method 1: Common Lyapunov function

Idea: If there exists a common Lyapunov function to all subsystems, the overall stability of the entire system can be guarantee.

1.1 Simultaneously dominatable assumption: find the most critical condition to design the controller.

1.2 Approximation-based, which considers f_i and g_i are uncertain functions, and the state transformation remains the same as integrator backstepping

1.3 Assume $f_{\sigma,i}\,(\bar{x}_i) \;=\; \sum_{k=1}^{i} x_k\, f_{\sigma,ik}(\bar{x}_i) \;=\; \sum_{k=1}^{i} z_k \varphi_{\sigma,ik}(\bar{x}_i)$

Method 2: Multiple Lyapunov functions

Idea: The Lyapunov function for each subsystem is required to decrease exponentially.

[If there exists a constant $\gamma > 0$ s.t. for any two switching times t_p and t_q with $p < q$, the Lyapunov-like function satisfies $V_{\sigma(t_q)}\left(x(t_{q+1})\right) - V_{\sigma(t_p)}\left(x(t_{p+1})\right) \leq \gamma \left| x(t_{p+1})\right|^2$ then the origin of the system is globally asymptotically stable.]

- Globally asymptotically stable $V_i \leq \gamma V_j$
- Global boundedness $V_i \leq \gamma V_j + \delta$ where δ is positive and bounded.

2.3 Multi-agent consensus problem

The control of the multi-agent system is the most recent research tendency. Composed by a number of intelligent agents, the multi-agent system can accomplish increasingly complex tasks without the intervention of a central controller

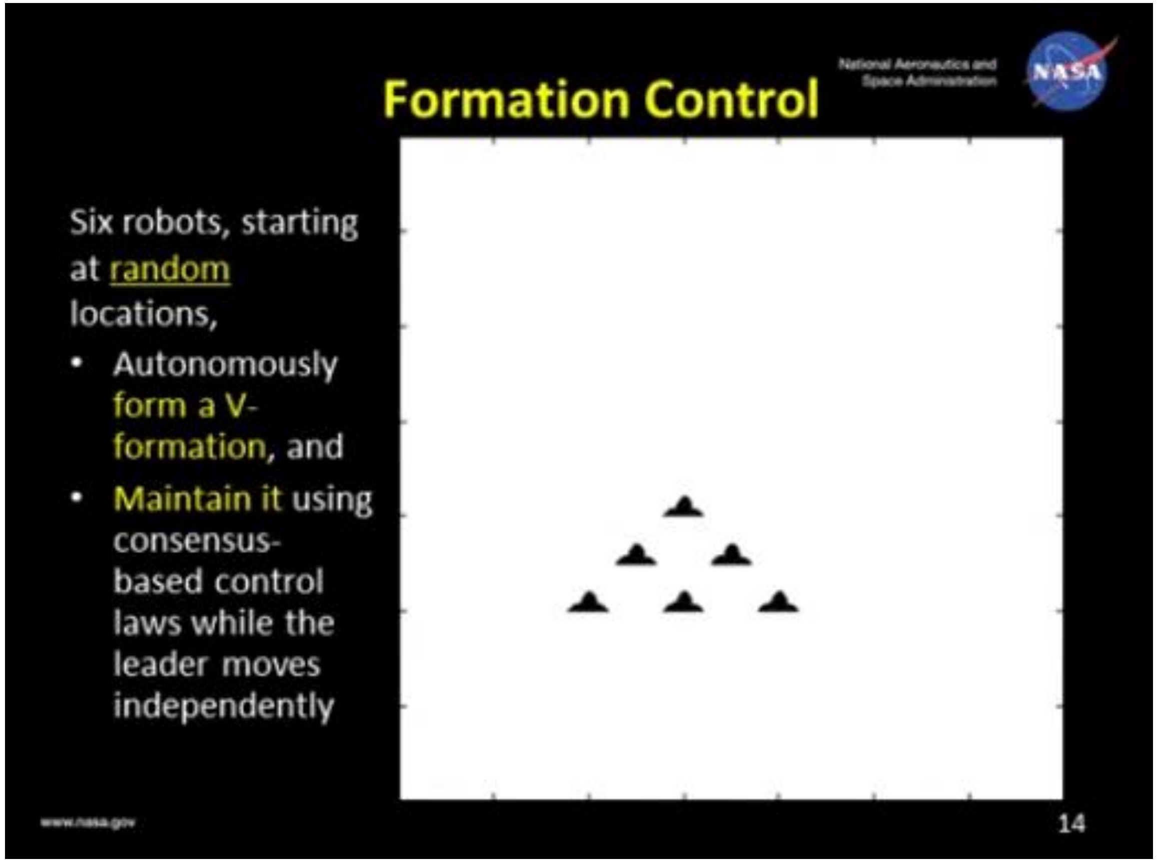
Consensus tracking control of multi-agent systems, also called networked cooperative systems, is suitable to apply the backstepping design.

(i) Cooperative regulation problem (leaderless consensus)
(ii) Cooperative tracking problem (leader-follower consensus)

Suppose that there are N followers in the network, then the system dynamics of a follower is expressed by

$$\begin{cases} \dot{x}_{k,i} &= f_{k,i}(\bar{x}_{k,i}) + g_{k,i}(\bar{x}_{k,i})x_{k,i+1}, \\ \dot{x}_{k,n} &= f_{k,n}(\bar{x}_{k,n}) + g_{k,n}(\bar{x}_{k,n})u_k, \\ y_k &= x_{k,1} \end{cases}$$

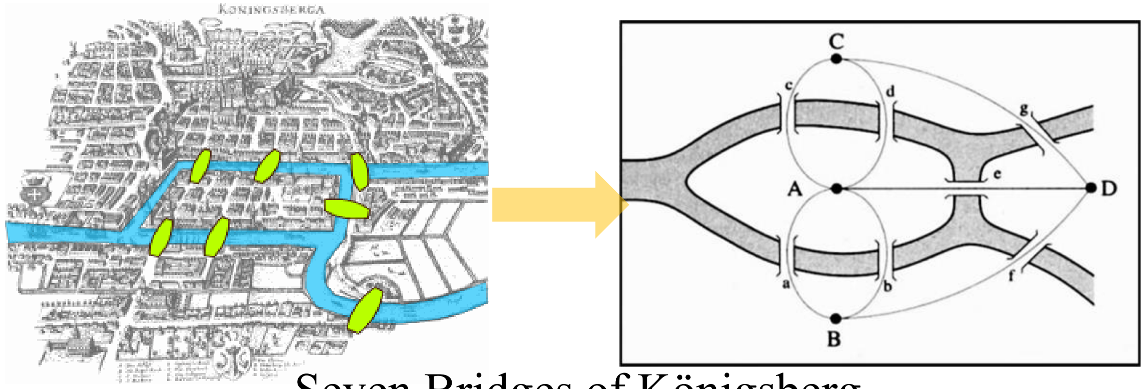
where $k = 1, \cdots, N$ indicates the index of a follower.
Control objective: Stabilize the relative distance or velocity among agents by controlling u_k
Challenge: Only relative measurements are available.



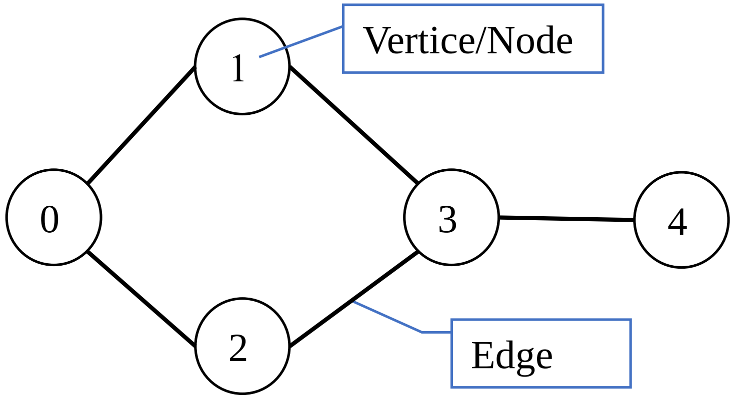
2.3 Multi-agent consensus problem

Graph theory

Graph is everywhere (map navigation, social network, sudoku).



Seven Bridges of Königsberg



Terminology:

Node	vertex	$\mathcal{V} = \{0,1,2,3,4\}$
Edge	connection between nodes	$\mathcal{E} = \{(0,1), (0,2), (1,3), (2,3), (3,4)\}$
Graph	A set of nodes and edges	$\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Neighbors	Nodes connected by edges	neighbors(0)={1,2}
Degree	Number of connected edges	degree(0) = 2, degree (3)=3
Path	Sequence of vertices connected by edges	$0 \rightarrow 1 \rightarrow 3 \rightarrow 2$
Path length	Number of edges in a path	
Cycle	A path with same starting and end vertex	

Connectivity

Two vertices are connected if a path exists between them;

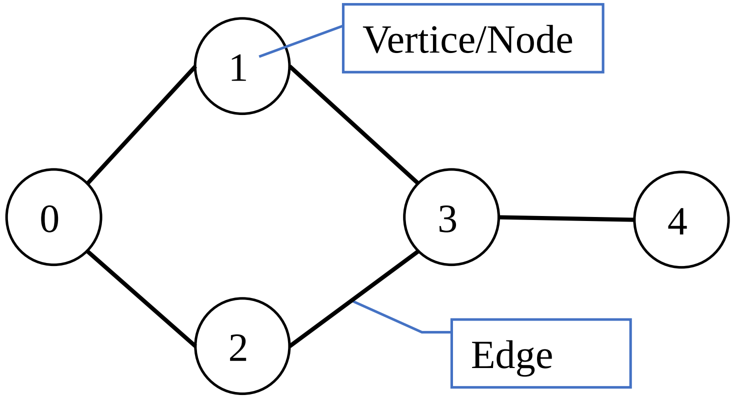
A graph is connected when all vertices are connected;

Undirected graph/directed graph Edges $(u, v) = (v, u)$/ unidirectional edges

Weighted graph Each edge is not treated equally

2.3 Multi-agent consensus problem

Graph theory



Adjacency matrix $\mathcal{A} = [a_{ij}]$

Unweighted: $a_{ij} = \begin{cases} 1, \exists \text{ edge } (i,j) \\ 0, \text{otherwise} \end{cases}$

Weighted: $a_{ij} = \begin{cases} > 0, \exists \text{ edge } (i,j) \\ 0, \quad \text{otherwise} \end{cases}$

	0	1	2	3	4
0		1	1		
1	1			1	
2	1			1	
3		1	1		1
4				1	

Degree matrix $\mathcal{D} = [d_{ij}]$

Unweighted: $d_{ii} = \begin{cases} \text{degree}(i), \\ 0, \quad \text{otherwise} \end{cases}$

Weighted: $d_{ii} = \begin{cases} \sum_{j=1}^{N} a_{ij}, \\ 0, \quad \text{otherwise} \end{cases}$

	0	1	2	3	4
0	2				
1		2			
2			2		
3				3	
4					1

Laplacian matrix $\mathcal{L} = \mathcal{D} - \mathcal{A}$

$$\mathcal{L} = [l_{ij}]$$

Useful properties:

- $\mathcal{L}$ is symmetric and positive semidefinite
- If the graph is connected, $\mathcal{L} + \mathcal{B}$ is positive definite where $\mathcal{B}$ is a diagonal matrix with positive diagonal element.

2.3 Multi-agent consensus problem

Problem formulation

$$\begin{cases} \dot{x}_{k,i} &= f_{k,i}(\bar{x}_{k,i}) + g_{k,i}(\bar{x}_{k,i})x_{k,i+1}, \\ \dot{x}_{k,n} &= f_{k,n}(\bar{x}_{k,n}) + g_{k,n}(\bar{x}_{k,n})u_k, \\ y_k &= x_{k,1} \end{cases}$$

where $k = 1, \cdots, N$ indicates the index of a follower.

Control objective: Stabilize the relative distance or velocity among agents by controlling u_k

The interaction topology among followers is modeled as a **weighted undirected graph** with a fixed topology

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}),$$

where

- $\mathcal{V} = \{1, \cdots, N\}$ is the node set,
- $\mathcal{E} \subset \{(k\_1, k\_2): k_1, k_2 \in \mathcal{V}\}$ is the edge set which means the agent k_1 can obtain information flow from agent k_2, and
- $\mathcal{A} = [a_{k_1 k_2}] \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix.

The **adjacency matrix** is given by

- $a_{k_1 k_1}$=0
- $a_{k_1 k_2} = a_{k_2 k_1}$
- $a_{k_1 k_2} > 0$ if $(k_1, k_2) \in \mathcal{E}$ and $k_1 \neq k_2$.

The Laplacian matrix with graph $\mathcal{G}$ is $\mathcal{L} = [l_{k_1 k_2}] \in \mathbb{R}^{N \times N}$ where

$$\begin{cases} l_{k_1 k_1} = \sum_{k_2=1}^{N} a_{k_1 k_2} \\ l_{k_1 k_2} = -a_{k_1 k_2} \text{ if } k_1 \neq k_2 \end{cases}.$$

The graph could be time-varying and switching.

2.3 Multi-agent consensus problem

Consensus tracking

Follower $\begin{cases} \dot{x}_{k,i} &= f_{k,i}(\bar{x}_{k,i}) + g_{k,i}(\bar{x}_{k,i})x_{k,i+1}, \\ \dot{x}_{k,n} &= f_{k,n}(\bar{x}_{k,n}) + g_{k,n}(\bar{x}_{k,n})u_k, \\ y_k &= x_{k,1} \end{cases}$

Leader $\begin{cases} \dot{x}_{0,i} &= x_{0,i+1}, \\ \dot{x}_{0,n} &= f_0(\bar{x}_{0,n}), \\ y_0 &= x_{0,1} \end{cases}$

where $k = 1, \cdots, N$ indicates the index of a follower.

Control objective: Stabilize the relative distance or velocity among agents by controlling u_k

Challenge: Only relative measurements are available.

Graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$

Define the matrix of the communication weights

$$\mathcal{B} = \text{diag}\{b_1, b_2, \cdots, b_N\}$$

where

$$b_k \begin{cases} > 0, & \text{iff leader and follower } i \text{ communicate} \\ = 0, & \text{otherwise} \end{cases}.$$

There is assumed that at least one agent connects with the leader, i.e., $\sum_{k=1}^{N} b_k > 0$

Assumptions: The augmented graph $\mathcal{G}$ contains a spanning tree with the root node being the leader node 0.
If the assumption holds, matrix $\bar{\mathcal{L}} \coloneqq \mathcal{L} + \mathcal{B}$ is positive definite.

The graph-based consensus error vectors is defined as

- $z_{k,1} := \sum_{k_2=1}^{N} a_{k_1 k_2}(x_{k_1} - x_{k_2}) + b_i (x_{k_1} - x_0)$
- $z_{k,i} = x_{ki} - \alpha_{k(i-1)}$

Control objective: $z_{k,1} \to 0$ as $t \to \infty$.

Define $Z_i = \begin{bmatrix} z_{1,1} \\ z_{2,1} \\ \vdots \\ z_{N,1} \end{bmatrix}$ $X_i = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{N,1} \end{bmatrix}$ $F_i = \begin{bmatrix} f_{1,1} \\ f_{2,1} \\ \vdots \\ f_{N,1} \end{bmatrix}$ $X_i = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{N,1} \end{bmatrix}$ $F_0 = \begin{bmatrix} f_0 \\ f_0 \\ \vdots \\ f_0 \end{bmatrix}$

Error dynamics

$$\begin{cases} z_{1'1} = b_1 x_0 + \sum_{k_2} a_1{}^{k_2} x^{k_2} - \left(\sum_{k_2} a_1{}^{k_2} + b_1\right) x_1 \\ z_{2'1} = b_2 x_0 + \sum_{k_2} a_2{}^{k_2} x^{k_2} - \left(\sum_{k_2} a_2{}^{k_2} + b_2\right) x_2 \\ z_{i'1} = b_i x_0 + \sum_{k_2} a_i{}^{k_2} x^{k_2} - \left(\sum_{k_2} a_i{}^{k_2} + b_i\right) x_i \end{cases} \Rightarrow Z_i = \mathcal{B} x_0 - \underbrace{\begin{bmatrix} \left(\sum_j a_{1j} + b_1\right) & & \\ & \left(\sum_j a_{2j} + b_2\right) & \\ & & \end{bmatrix}}_{\mathcal{D}+\mathcal{B}} X + \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\mathcal{A}} X$$

$$\Rightarrow \dot{Z}_i = -(\mathcal{L} + \mathcal{B})(\dot{X} - \dot{X}_0) \quad (\text{since } \mathcal{L}X_0 = 0)$$

Lyapunov function

- $V_i = V_{i-1} + \sum_{k=1}^{N} V_{i,QF}(z_{k,i})$
- or $V_i = V_{i-1} + \frac{1}{2} Z_i^\top Z_i$

Time derivative becomes: $\dot{V}_i \le -c_i Z_i^\top \bar{\mathcal{L}} Z_i \le -c_i \lambda_{min}(\bar{\mathcal{L}})|Z_i|^2$

Use the inequality: $c_i \lambda_{min}(\bar{\mathcal{L}})|Z_i|^2 \le c_i Z_i^\top \bar{\mathcal{L}} Z_i \le c_i \lambda_{max}(\bar{\mathcal{L}})|Z_i|^2$

Stability criteria: $\dot{V}_n \le -\gamma \sum_{i=1}^{n} |E_i|^2 + \delta$

First things first

- Assumption: boundness; parametric separation
- Inequality: parametric separation
- Cancellation is not perfect unless this is a known system
- Lyapunov-like inequality holds

$$\dot{V}(x) \leq -\gamma V(x) + \delta$$

Backstepping = Cooking

01 — Find what you want to eat

02 — Cut vegetables and meats into correct shapes

03 — Select the sauce

04 — Mix & wok

Define the problem

- Find all nonlinearities
- Separate the problem to several separated subproblems

Transfer each subproblem

- Into a specific form in Lecture 1

Find the correct methods for each subproblem

- Assumption
- Lyapunov function V_i
- Robustness/Approximation

Superposition & recursive design

- Derivative of LFC $\dot{V}_i$
- Virtual control
- Key steps

Keep in mind

- Not all combinations are feasible. Be careful about the assumption

- Tips: To understand a specific method, the original work may be not the best choice. The later works has a better organization.

	Incompatible food	Supprotive combination
Beans	Milk, meat, yogurt, eggs, fish, cheese, fruit	Seeds, bean, grains, vegetables, other nuts
Butter & Ghee	Butter may not combine with other foods as universally as ghee	Grains, vegetables, beans, nuts, seeds, meat, fish, eggs, cooked fruit
Cheese	Hot drinks, eggs, fruit, beans, milk, yogurt	Grains, vegetables
Milk	Any other food (especially BANANAS, eggs, cherries, meat, melons, sour fruits, yeasted breads, yogurt, fish, kitchari, starches)	Milk is best enjoyed alone... Exceptions: rice pudding, oatmeal, dates, almonds
Eggs	Milk, cheese, yogurt, fruit (especially melons), kitchari, potatoes, meat, fish, beans	Grains, non-starchy vegetables
Fruits	Any other food (aside from other fruit) *Exceptions: dates with milk, some cooked combinations	Other fruits with similar qualities (i.e. citrus together, apples with pears, a berry medley, etc.)
Lemons	Cucumbers, tomatoes, milk, yogurt Note: lime can be substituted for use with cucumbers and tomatoes	Usually ok with other foods, if used in small amounts as a garnish or flavoring.
Melons	EVERYTHING (especially dairy, fried food, grains, starches, eggs) *More than most fruit, melons should be eaten alone or not at all.	Other melons (in a pinch)... But it's better to have each type of melon on its own.
Grains	Fruit	Beans, other grains, cheese, eggs, meat, fish, nuts, seeds, vegetables, yogurt
Vegetables	Fruit, milk	Grains, other vegetables, yogurt, meat, fish, nuts, beans, seeds, eggs, cheese
Nightshades	Fruit (especially melon), cucumber, milk, cheese, yogurt Note: potatoes, nightshades include peppers, eggplant and tomatoes.	Seeds, other vegetables, grains, beans, meat, fish, nuts

Summary and some reflections

- Backstepping is very useful in Department of Marine Technology when you work on the control of rigid (flexible) bodies.
- Everything uncancelled is put into the garbage bin δ. With more garbage, more aggressive control is required.
  - The controller can compensate almost all nonlinearities if the control gains are large enough (Increase k_p in a PID)
- To estimate the unknown needs training time, and the training occurs meanwhile with the control.
  - The performance of the system is not guaranteed when the estimator is not ready.
  - To identify the unknown, the system has to be excited. Hours of constant input does not provide useful information than a second.
- Complex theories and designs ensure strictly analytical stability proof, but they are not suitable for most practical applications. The theoretical development is limited. Learning can overcome this but without any guarantee to stability.
- Combination-based innovation
  - A tricky but easy way to graduate.
  - A never-appeared-before inequality/nonlinearity/general system gives you a large amount of Automatica/IEEE TAC.

Callback: The aforementioned examples

Fuzzy logic system

Observer error in V

$x_{1d} \neq 0 \; \forall t \geq 0$

Adaptive Fuzzy Output Feedback Tracking Backstepping Control of Strict-Feedback Nonlinear Systems With Unknown Dead Zones

Appr. by NN/Robust

Neural network

Lyapunov-Krasovskii functional

Integral LFC/ Nussbaum function

Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients

Neurocomputing

...ry 2016, Pages 759-767

Parameter separation

Adaptive backstepping-based fuzzy tracking control scheme for output-constrained nonlinear switched lower triangular systems with time-delays ☆

BLF

Lyapunov-Krasovskii functional

Automatica

Volume 64, February 2016, Pages 70-75

State constraint

Brief paper

Barrier Lyapunov Functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints ☆

Mean value theorem + Integral LFC/Nussbaum function

If you have any question on backstepping, I believe that I am helpful.

zhengru.ren@ntnu.no