TTK4150 Nonlinear Control Systems
Lecture 13

Backstepping

NTNU
Innovation and Creativity

Previous lectures on nonlinear control design

Previous lectures on control design:

- Lyapunov based control design
- Cascaded control: Lemma 4.7 allows for modular design
  (And background material, Sontag and Loria)
- Passivity-based control design
- Input-output linearization

Outline I

Last week

Last week:

- The concepts of relative degree, normal form, external dynamics, internal dynamics and zero dynamics.
- Learned how to design a stabilizing control law using the input-output linearization method, including
  1. Finding the relative degree
  2) Writing the system in normal form
  3) Creating a linear input-output relation by feedback control
  4) Analyzing the zero dynamics
  5) Choosing the transformed input variable v to stabilize the origin of the system, locally or globally
- How to design a control law that solves the local tracking control problem, using the I-O linearization method
- Advantages and disadvantages of the input-output linearization method

Today's goals

Backstepping

After today you should...

- Be able to design a stabilizing control law using the integrator backstepping method
- Be able to discuss the advantages and disadvantages of this method

Literature

Today's lecture is based on

Khalil **Chapter 14**
Section 14.3 pages 589-598

The system to be controlled

The system

Consider the system

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_n\end{bmatrix} = f\left(\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\right) + g\left(\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\right)\cdot x_{n+1}$$

$$\dot{x}_{n+1} = u$$

f, g sufficiently smooth (C^k) in a set $\mathbb{D} \subseteq \mathbb{R}^n$ that contains $x = 0$, and $f(0) = 0$

Control task

Find a control law $u = \alpha(x)$ that stabilizes $x = \begin{bmatrix}x_1\\ \vdots\\ x_{n+1}\end{bmatrix} = 0$

The system to be controlled

A cascade structure

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_n\end{bmatrix} = f\left(\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\right) + g\left(\begin{bmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\right)\cdot x_{n+1}$$

$$\dot{x}_{n+1} = u$$

Cascade structure

We can identify a cascade structure:

$$\dot{\eta} = f(\eta) + g(\eta)\xi \quad (1)$$

$$\dot{\xi} = u \quad (2)$$

$$\xrightarrow{u} \boxed{\Sigma_2} \xrightarrow{\xi} \boxed{\Sigma_1} \xrightarrow{\eta}$$

The backstepping method

Step 1 - Find a stabilizing function for Σ_1

Step 1 - Find a stabilizing function for Σ_1 (Equation (1))

Regard ξ as a *virtual control input* to Σ_1

- Find a *stabilizing function*

$$\xi = \varphi(\eta), \quad \varphi(0) = 0$$

  such that $\eta = 0$ is an asymptotically stable equilibrium point of

$$\dot{\eta} = f(\eta) + g(\eta)\varphi(\eta)$$

- and find a corresponding Lyapunov function to prove this

$$V(\eta) > 0, \quad C^1$$

$$\frac{\partial V}{\partial \eta}\left[f(\eta) + g(\eta)\varphi(\eta)\right] < 0, \quad \forall\, \eta \in \mathbb{D}$$

The backstepping method
Step 2 - Design the actual control input u

Step 2 - Design the actual control input u

Design the actual control input u to stabilize the full system:

- Introduce the error variable as a new state (replacing ξ)

$$z = \xi - \varphi(\eta)$$

- Write the system equations in the new coordinates $\begin{bmatrix} \eta \\ z \end{bmatrix}$

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)(z + \varphi(\eta)) \\ \dot{z} &= \dot{\xi} - \dot{\varphi} \\ &\Downarrow \\ \dot{\eta} &= f(\eta) + g(\eta)\varphi(\eta) + g(\eta)z \\ \dot{z} &= u - \dot{\varphi} \end{aligned}$$

The backstepping method
Step 2 - Design the actual control input u

Step 2 - Design the actual control input u

- Choose the Lyapunov function candidate

$$V_c(\eta, z) = V(\eta) + \frac{1}{2}z^2$$

- Find a control law u which asymptotically stabilizes

$$\begin{bmatrix} \eta \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

  based on $V_c = V(\eta) + \frac{1}{2}z^2$

Because $[\eta, \xi]^T \mapsto [\eta, z]^T$ is a diffeomorphism:
$[\eta, z]^T = 0$ asymptotically stable $\Leftrightarrow$ $[\eta, \xi]^T = 0$ asymptotically stable

The backstepping method
Step 2 - Design the actual control input u

Step 2 - Design the actual control input u

- Choose u such that $\dot{V}_c < 0$ (in (η, z)):

$$u = -\frac{\partial V}{\partial \eta}g(\eta) + \dot{\varphi} - kz \qquad k > 0$$

$$\dot{V}_c = \underbrace{\frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\varphi(\eta)]}_{<0 \, \text{in} \, \eta} \underbrace{-kz^2}_{<0 \, \text{in} \, z} < 0$$

Conclusion

$$u = -\frac{\partial V}{\partial \eta}g(\eta) + \frac{\partial \varphi}{\partial \eta}[f(\eta) + g(\eta)\xi] - k[\xi - \varphi(\eta)]$$

$\Rightarrow$ $(\eta, \xi) = (0, 0)$ is asymptotically stable
(*Globally* asymptotically stable if $\mathbb{D} = \mathbb{R}^n$ and V is radially unbounded in η)

Examples

Read examples 14.8 - 14.9

Example

Consider the system

$$\dot{x}_1 = \sin x_1 - x_1^3 + x_2$$
$$\dot{x}_2 = u$$

Use the backstepping method to design a stabilizing control law (rendering the equilibrium point $x = 0$ GAS).

Application: Active suspension

Example: Active suspension

When designing vehicle suspension systems for cars, there is a dual objective:

- Minimize the vertical acceleration of the car body (for passenger comfort)
- Maximize tire contact with the road surface (for handling)

To this end *active* suspension systems with hydraulic actuators are developed.

Active suspensions should be designed to behave differently on smooth and rough roads. Thic can be achieved by introducing nonlinearities in the controller which make the suspension stiffer near its travel limits:

Application: Active suspension cont.

Example: Active suspension cont.

Active suspension design:

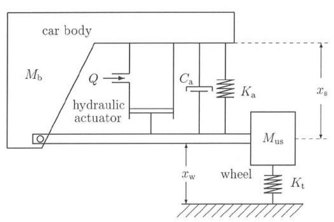

Figure 2.6: Quarter-car model for active suspension design with *parallel* connection of hydraulic actuator with passive spring/damper.

The fluid flow is adjusted by a current controlled valve:

$$\dot{d}_v = -c_v d_v + k_v i_v$$

The resulting flow is (advanced valve, cancels the square-root nonlinearity):

$$\dot{Q} = -c_f Q + k_f i_v$$

Application: Active suspension cont.

Example: Active suspension cont.

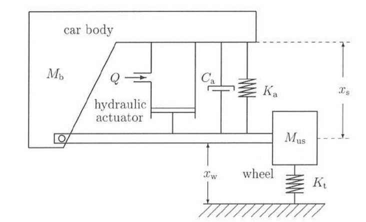

Figure 2.6: Quarter-car model for active suspension design with *parallel* connection of hydraulic actuator with passive spring/damper.

In this parallell configuration, neglecting leakage and compressability, the suspension travel x_s is related to the fluid flow Q through the equation

$$\dot{x}_s = \frac{1}{A} Q$$

Application: Active suspension cont.

Example: Active suspension cont.

The system equations are thus

$$\dot{x}_s = \frac{1}{A} Q$$
$$\dot{Q} = -c_f Q + k_f i_v$$

To apply backstepping, we view the flow Q as a virtual control, and design for it a nonlinear stabilizing function $\varphi(x_s)$ which will stiffen the suspension near its travel limits:

$$Q_{\text{des}} = \varphi(x_s) = -A(c_1 x_s + k_1 x_s^3)$$

Find a stabilizing controller for i_v

Exam

This was the final lecture
First exams then...

Happy holidays