EG4321/EG7040

Nonlinear Control

Dr. Matt Turner

Motivation

Nonlinear control so far

Aims of Lecture

Motivation

Nonlinear control so far

- Quite hard!

Aims of Lecture

Motivation

<u>Nonlinear control so far</u>

- Quite hard!
- Gaps between necessity and sufficiency (conservatism)

<u>Aims of Lecture</u>

Motivation

Nonlinear control so far

- Quite hard!
- Gaps between necessity and sufficiency (conservatism)
- So far, mainly analysis has been treated

Aims of Lecture

Motivation

Nonlinear control so far

- Quite hard!
- Gaps between necessity and sufficiency (conservatism)
- So far, mainly analysis has been treated

Aims of Lecture

1. To introduce the *nonlinear dynamic inversion* (NDI) approach to controller design
2. To examine the mertis/deficiencies of NDI

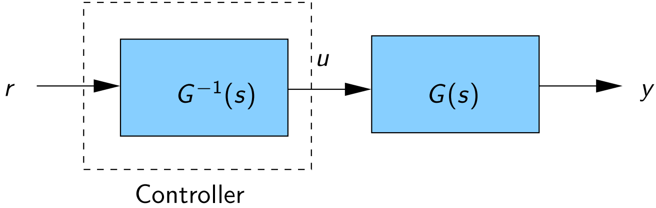
Feedback Linearisation/Nonlinear Dynamic Inversion

Recall ideas from classical control

- Want y to track r

$$\lim_{t\to\infty} y(t) = r(t)$$

<u>Naive approach</u> - invert the plant



- Obviously: $y = r$ *for all time*
- Is this a perfect controller?

Problems with "inversion"

We would not have 70 years of feedback control if inversion was the answer

Many problems:

- Requires invertibility of $G(s)$ - often difficult/impossible
- Requires stability of G - feedforward control
- Requires perfect knowlege of G - very sensitive to perturbations
- Unable to cope with disturbances
- ...

But aspect of inversion are appealing:

- Idea of "cancelling" troublesome (nonlinear?) dynamics attractive

Feedback linearisation - simple case

Assume system is given in form:

$$\begin{aligned} \dot{x} &= Ax + B\Gamma(x)(u - g(x)) \\ y &= Cx \end{aligned}$$

where

- $A \in \mathbb{R}^{n\times n}$; $B \in \mathbb{R}^{n\times m}$
- $\Gamma(.) : \mathbb{R}^n \mapsto \mathbb{R}^{m\times m}$
- $g(.) : \mathbb{R}^n \mapsto \mathbb{R}^m$

i.e a <u>very special</u> structure in which there are clearly visible linear elements

- If $\Gamma(x) \equiv I$ and $g(x) \equiv 0 \Rightarrow$ linearity recovered

Feedback linearisation - simple case II

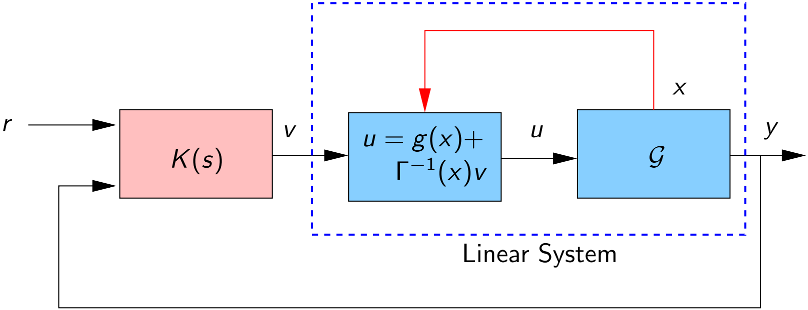
- Choose specially structured control law

$$u = g(x) + \Gamma(x)^{-1} v$$

- Using this in system dynamics

$$\dot{x} = Ax + B\Gamma(x)[g(x) + \Gamma^{-1}(x)v - g(x)]$$
$$\dot{x} = Ax + Bv$$

- Hence we have a linear system driven by the "virtual control" v



Feedback linearisation - simple example

Consider the pendulum dynamics

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l}\sin(x_1) - \frac{b}{m}x_2 + \frac{1}{ml^2}u\end{aligned}$$

This can be re-written as

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 1 \\ 0 & -\frac{b}{m}\end{bmatrix}}_{\mathbf{A}}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \underbrace{\begin{bmatrix}0 \\ \frac{1}{ml^2}\end{bmatrix}}_{\mathbf{B}}\left(u(t) - \underbrace{glm\sin(x_1)}_{\mathbf{g(x)}}\right)$$

Therefore with

$$u(t) = glm\sin(x_1) + v$$

we have

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & -\frac{b}{m}\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}0 \\ \frac{1}{ml^2}\end{bmatrix}v(t)$$

Another simple example [Khalil]

Consider the dynamics

$$\begin{aligned}\dot{x}_1 &= \mathbf{a}\sin(\mathbf{x_2})\\ \dot{x}_2 &= -x_1^2 + u\end{aligned}$$

- Clearly this system does not fit the form introduced earlier
- Introduce *nonlinear change of coordinates*

$$\begin{aligned}z_1 &= x_1\\ z_2 &= \mathbf{a}\sin(\mathbf{x_2})\end{aligned} \quad \Rightarrow \quad \begin{aligned}\dot{z}_1 &= \dot{x}_1\\ \dot{z}_2 &= a\cos(x_2)\dot{x}_2\end{aligned}$$

- Simplifying

$$\begin{aligned}\dot{z}_1 &= z_2\\ \dot{z}_2 &= a\cos\left(\sin^{-1}(z_2/a)\right)\left(-z_1^2 + u\right)\end{aligned}$$

- Thus letting

$$u = z_1^2 + \frac{1}{\cos(\sin^{-1}(z_2/a))}v \quad \Rightarrow \quad \boxed{\begin{aligned}\dot{z}_1 &= z_2\\ \dot{z}_2 &= av\end{aligned}}$$

Extrapolations

If system does not **initially** have a structure suitable for feedback linearisation, it may be possible to change coordinates so that it does

- Coordinate change $T(.) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is typically *nonlinear*:

$$z = T(x) \quad \text{and} \quad \dot{z} = \frac{\partial T(x)}{\partial x}\dot{x}$$

  i.e. $T(x)$ must be differentiable
- Also we need to recover x so mapping $T(.)$ must be invertible:

$$x = T^{-1}(z)$$

  ($T^{-1}(.)$ denotes inverse of mapping not inverse of matrix)
- $T(x)$ must be a **diffeomorphism**
  - Search for $T(x)$ not trivial
  - $T(x)$ not unique