ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

Approximating Nonlinear Models With Linear Models Linear in the State and Input

Problem Statement

- Taylor's linearization yields models linear in $\delta \boldsymbol{x}$ and $\delta \boldsymbol{u}$
- These models, in general, are not linear in $\boldsymbol{x}$ and $\boldsymbol{u}$, but rather affine
- We provide a method of constructing linear models in the state and input for the plants modeled as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$$

- Suppose that we are given an operating state $\boldsymbol{x}_o$ that does not have to be an equilibrium state
- Objective: construct a linear model in $\boldsymbol{x}$ and $\boldsymbol{u}$ that approximates the plant behavior in the vicinity of the operating state $\boldsymbol{x}_o$

What do we wish to do?

- We wish to find constant matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ such that in a neighborhood of $\boldsymbol{x}_o$,

$$\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \approx \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u},$$

  and

$$\boldsymbol{f}(\boldsymbol{x}_o) + \boldsymbol{G}(\boldsymbol{x}_o)\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{B}\boldsymbol{u} \quad \text{for all} \quad \boldsymbol{u}$$

- Since $\boldsymbol{u}$ is arbitrary, we have to have

$$\boldsymbol{G}(\boldsymbol{x}_o) = \boldsymbol{B}.$$

- Thus, we are left with finding a constant matrix $\boldsymbol{A}$ such that in a neighborhood of $\boldsymbol{x}_o$,

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{A}\boldsymbol{x},$$

  and

$$\boldsymbol{f}(\boldsymbol{x}_o) = \boldsymbol{A}\boldsymbol{x}_o$$

Finding $\boldsymbol{A}$ as a constrained optimization

- Let $\boldsymbol{a}_i^\top$ denote the i-th row of the matrix $\boldsymbol{A}$
- Then,

$$f_i(\boldsymbol{x}) \approx \boldsymbol{a}_i^\top \boldsymbol{x}, \quad i = 1, 2, \ldots, n,$$

  and we have

$$f_i(\boldsymbol{x}_o) = \boldsymbol{a}_i^\top \boldsymbol{x}_o, \quad i = 1, 2, \ldots, n,$$

  where the i-th component of $\boldsymbol{f}$ is $f_i : \mathbb{R}^n \to \mathbb{R}$
- Expanding the left hand side about $\boldsymbol{x}_o$ and neglecting second and higher order terms yields

$$f_i(\boldsymbol{x}_o) + \nabla f_i(\boldsymbol{x}_o)^\top (\boldsymbol{x} - \boldsymbol{x}_o) \approx \boldsymbol{a}_i^\top \boldsymbol{x},$$

  where $\nabla f_i(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^n$ is the gradient (a column vector) of f_i at $\boldsymbol{x}$
- Hence,

$$\nabla f_i(\boldsymbol{x}_o)^\top (\boldsymbol{x} - \boldsymbol{x}_o) \approx \boldsymbol{a}_i^\top \left(\boldsymbol{x} - \boldsymbol{x}_o\right),$$

  where $\boldsymbol{x}$ is arbitrary but “close” to $\boldsymbol{x}_o$

Finding $\boldsymbol{A}$ as a constrained optimization—Objective function to be minimized

- We have
$$\nabla f_i(\boldsymbol{x}_o)^\top (\boldsymbol{x} - \boldsymbol{x}_o) \approx \boldsymbol{a}_i^\top (\boldsymbol{x} - \boldsymbol{x}_o),$$
where $\boldsymbol{x}$ is arbitrary but "close" to $\boldsymbol{x}_o$
- Our task: determine a constant vector $\boldsymbol{a}_i$ that is as "close as possible" to $\nabla f_i(\boldsymbol{x}_o)$ and satisfies the constraint $\boldsymbol{a}_i^\top \boldsymbol{x}_o = f_i(\boldsymbol{x}_o)$
- Let
$$E = \frac{1}{2} \left\| \nabla f_i(\boldsymbol{x}_o) - \boldsymbol{a}_i \right\|_2^2$$

Optimization problem formulation

- Find $\boldsymbol{a}_i$ such that

$$\nabla f_i(\boldsymbol{x}_o)^\top (\boldsymbol{x} - \boldsymbol{x}_o) \approx \boldsymbol{a}_i^\top (\boldsymbol{x} - \boldsymbol{x}_o)\,,$$

  where $\boldsymbol{x}$ is arbitrary but "close" to $\boldsymbol{x}_o$
- The constant vector $\boldsymbol{a}_i$ being "close as possible" to $\nabla f_i(\boldsymbol{x}_o)$ must satisfy the constraint, $\boldsymbol{a}_i^\top \boldsymbol{x}_o = f_i(\boldsymbol{x}_o)$
- Recall the objective function,

$$E = \frac{1}{2} \left\| \nabla f_i(\boldsymbol{x}_o) - \boldsymbol{a}_i \right\|_2^2$$

- Constrained optimization problem

$$\left.\begin{array}{l} \underset{\boldsymbol{a}_i}{\text{minimize}}\ E \\ \text{subject to}\ \boldsymbol{a}_i^\top \boldsymbol{x}_o = f_i(\boldsymbol{x}_o) \end{array}\right\}$$

- This is a convex constrained optimization problem
- This means that the first-order necessary condition for a minimum of E is also sufficient!

Solving the convex optimization problem

- The first-order conditions for the optimization problem

$$\nabla_{\boldsymbol{a}_i} E + \lambda \nabla_{\boldsymbol{a}_i} \left(\boldsymbol{a}_i^\top \boldsymbol{x}_o - f_i(\boldsymbol{x}_o) \right) = \boldsymbol{0},$$
$$\boldsymbol{a}_i^\top \boldsymbol{x}_o = f_i(\boldsymbol{x}_o),$$

  where λ is the Lagrange multiplier and the subscript $\boldsymbol{a}_i$ in $\nabla_{\boldsymbol{a}_i}$ indicates that the gradient, ∇ is computed with respect to $\boldsymbol{a}_i$
- Performing the required differentiation yields

$$\boldsymbol{a}_i - \nabla f_i(\boldsymbol{x}_o) + \lambda \boldsymbol{x}_o = \boldsymbol{0},$$
$$\boldsymbol{a}_i^\top \boldsymbol{x}_o = f_i(\boldsymbol{x}_o)$$

- Recall that we consider the case when $\boldsymbol{x}_0 \neq \boldsymbol{0}$

Solution to the convex optimization problem

- Performing manipulations gives

$$\lambda = \frac{\boldsymbol{x}_o^\top \nabla f_i(\boldsymbol{x}_o) - f_i(\boldsymbol{x}_o)}{\|\boldsymbol{x}_o\|^2}$$

- Substituting λ

$$\boxed{\boldsymbol{a}_i = \nabla f_i(\boldsymbol{x}_o) + \frac{f_i(\boldsymbol{x}_o) - \boldsymbol{x}_o^\top \nabla f_i(\boldsymbol{x}_o)}{\|\boldsymbol{x}_o\|^2}\boldsymbol{x}_o, \quad \boldsymbol{x}_o \neq \boldsymbol{0}}$$

- Let

$$D\boldsymbol{f}(\boldsymbol{x}_o) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_o}$$

  be the Jacobian matrix of $\boldsymbol{f}$ evaluated at the operating state $\boldsymbol{x} = \boldsymbol{x}_o$

Linear in $\boldsymbol{x}$ and $\boldsymbol{u}$ approximating model

- Linear model in $\boldsymbol{x}$ and $\boldsymbol{u}$ that approximates the behavior of the nonlinear models in the vicinity of the operating state $\boldsymbol{x}_o \neq \mathbf{0}$,

$$\boxed{\dot{\boldsymbol{x}} = \left[D\boldsymbol{f}(\boldsymbol{x}_o) + \frac{\left(\boldsymbol{f}(\boldsymbol{x}_o) - D\boldsymbol{f}(\boldsymbol{x}_o)\boldsymbol{x}_o\right)\boldsymbol{x}_o^\top}{\|\boldsymbol{x}_o\|^2} \right] \boldsymbol{x} + \boldsymbol{G}(\boldsymbol{x}_o)\boldsymbol{u}}$$

- If the operating pair $(\boldsymbol{x}_o, \boldsymbol{u}_o)$ is an equilibrium pair, then Taylor's linearization yields the linearized model;

$$\frac{d}{dt}\delta\boldsymbol{x} = \left[D\boldsymbol{f}(\boldsymbol{x}_o) + \sum_{k=1}^{m} u_k \frac{\partial \boldsymbol{g}_k}{\partial \boldsymbol{x}} \Bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}_o \\ \boldsymbol{u}=\boldsymbol{u}_o}} \right] \delta\boldsymbol{x} + \boldsymbol{G}(\boldsymbol{x}_o)\delta\boldsymbol{u}$$

where $\delta\boldsymbol{x} - \boldsymbol{x}_0$ and $\delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_0$

Example

- Nonlinear plant model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{g\sin(x_1) - mlax_2^2 \sin(2x_1)/2}{4l/3 - mla\cos^2(x_1)} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{a\cos(x_1)}{4l/3 - mla\cos^2(x_1)} \end{bmatrix} u,$$

  where $g = 9.8\ \mathrm{m/sec}^2$, $m = 2$ kg, $M = 8$ kg, $a = 1/(m+M)$, $l = 0.5$ m.

- Construct a local model that corresponds to $x_1 = 88^\circ \pi/180^\circ$ and $x_2 = 0$
- Let $\beta = \cos(88^\circ)$
- The input matrix

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ -\frac{a\beta}{4l/3 - mla\beta^2} \end{bmatrix}$$

Example—compute the rows of $\boldsymbol{A}$

- First compute the first row $\boldsymbol{A}$
- Note that $f_1(\boldsymbol{x}) = x_2$, and hence

$$\nabla f_1 = \left[\begin{array}{cc} 0 & 1 \end{array}\right]^\top$$

- The operating state is $\boldsymbol{x}_o = \left[\begin{array}{cc} 88^\circ\pi/180^\circ & 0 \end{array}\right]^\top$
- Compute $\boldsymbol{a}_1^\top$, the first row of $\boldsymbol{A}$,

$$\boldsymbol{a}_1^\top = \nabla^\top f_1(\boldsymbol{x}_o) + \left[\begin{array}{cc} 0 & 0 \end{array}\right] = \left[\begin{array}{cc} 0 & 1 \end{array}\right].$$

- Compute $\boldsymbol{a}_2^\top$, the second row of the matrix $\boldsymbol{A}$,

$$\boldsymbol{a}_2^\top = \left[\begin{array}{cc} f_2(\boldsymbol{x}_o)/(88^\circ\pi/180^\circ) & 0 \end{array}\right] \approx \left[\begin{array}{cc} \frac{g}{4l/3 - mla\beta^2}\left(\frac{2}{\pi}\right) & 0 \end{array}\right]$$

Example—local model

- The local model that corresponds to $x_1 = 88^\circ\pi/180^\circ$ and $x_2 = 0$ has the form

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3 - mla\beta^2}\left(\frac{2}{\pi}\right) & 0 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ -\frac{a\beta}{4l/3 - mla\beta^2} \end{bmatrix} u \\ &= \begin{bmatrix} 0 & 1 \\ 9.5669 & 0 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ -0.0052 \end{bmatrix} u\end{aligned}$$

- Note that the local model corresponding to $x_1 = -88^\circ\pi/180^\circ$ and $x_2 = 0$ has the same form as the one above
- This is because the $\cos$ function is an even function

Use MATLAB to compute A and B

```
clear all
clc
x0=[88*pi/180 0]';
syms x1 x2
f=[x2
  (9.8*sin(x1)-0.1*x2^2*sin(2*x1)/2)/(2/3-...
  0.1*cos(x1)^2)];
Df=jacobian(f,[x1 x2]);
Dfx0=eval(subs(Df,[x1 x2],[x0(1) x0(2)]));
fx0=eval(subs(f,[x1 x2],[x0(1) x0(2)]));
A=Dfx0+((fx0-Dfx0*x0)*x0')/norm(x0)^2
g=[0;-0.1*cos(x1)/(2/3-0.1*cos(x1)^2)];
B=eval(subs(g,[x1 x2],[x0(1) x0(2)]))
```