Lecture – 31

Gain Scheduling and Dynamic Inversion

Dr. Radhakant Padhi

Asst. Professor

Dept. of Aerospace Engineering

Indian Institute of Science - Bangalore

Topics

- **A Brief Overview of Gain Scheduling**
  - Philosophy
  - Steps
  - Issues
- **Dynamic Inversion (DI) Design**
  - Philosophy
  - Steps
  - Advantages

Gain Scheduling

An Accepted Practice in Industry

Dr. Radhakant Padhi

Asst. Professor

Dept. of Aerospace Engineering

Indian Institute of Science - Bangalore

Philosophy of Gain Scheduling

- Design local linear controllers, based on linearization of the nonlinear system at several expected operating points (based on the operating values of the scheduling variable(s)).
- Obtain a nonlinear controller for the nonlinear system by interpolating (scheduling) the linear gains, based on the actual value of the scheduling variable(s).

Practice in Industry

- The scheduling variable may or may not be one/more state variable(s). Usual practice is to choose a ‘slowly varying’ meaningful output variable as the scheduling variable.

  In flight control applications, usually dynamic pressure and Mach number are chosen as the scheduling variables

- The robustness, performance, stability etc. are usually justified from extensive simulation studies.
- The design results in a ‘semi-global controller’, depending on the selection of the operating points and validity of linear gains.

Steps of Gain Scheduling

- Linearize the dynamics about an 'operating point'

$$\dot{X} = AX + BU$$

- Design the controller gain K based on it

$$U = -K\,X$$

- Repeat this procedure about several operating points (within the domain of interest) and store the gains

- Interpolate the gains online and obtain a '<u>nonlinear controller</u>'

Issues

- Is the linearized system good enough?
- How many operating points to consider?
- Does the interpolated gain lead to stability?

  There are examples of loss of stability because of scheduling!
- How easy it is to select a set of “scheduling” variables?
- What if the parameters of the model are updated?
- Higher dimensional scheduling vector leads to complexity in interpolations.
- Loss of Minimum-phase property: A serious issue!

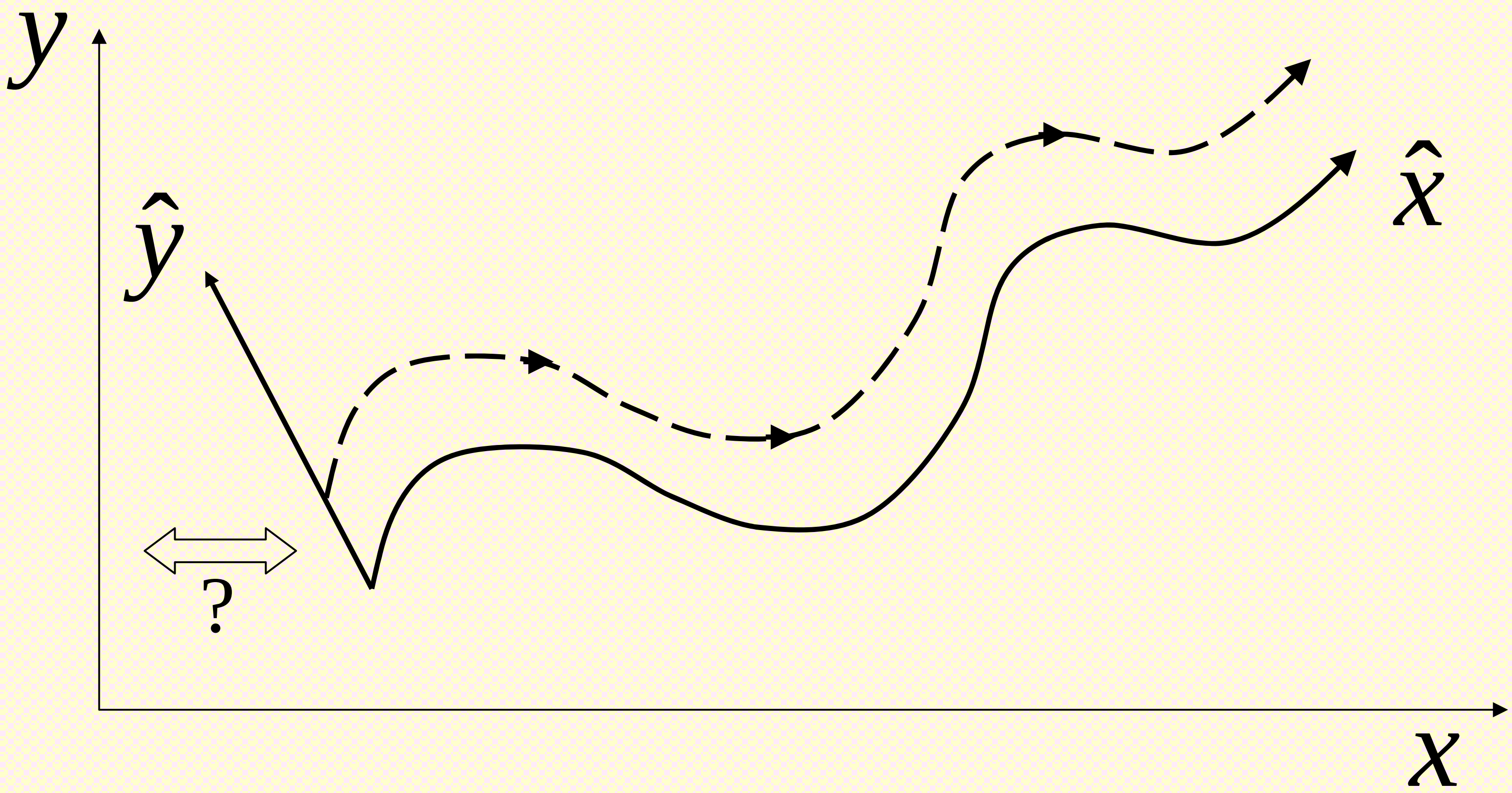
Reference

J. S. Shamma and M. Athans: ***Gain Scheduling: Potential Hazards and Possible Remedies***, IEEE Control System Magazine, June 1992, pp.101-107.

Dynamic Inversion:
A Promising Substitute for Gain Scheduling

Dr. Radhakant Padhi
Asst. Professor
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Philosophy of Dynamic Inversion



Philosophy of Dynamic Inversion

- Carryout a co-ordinate transformation such that the problem “appears to be linear” in the transformed co-ordinates.
- Design the control for the linear-looking system using the linear control design techniques.
- Obtain the controller for the original system using an inverse transformation.
- Intuitively, the control design is carried out by enforcing a stable linear error dynamics.

Problem

- System Dynamics:

$$\dot{X} = f(X, U)$$
$$Y = h(X)$$
$$X \in \mathbb{R}^n, U \in \mathbb{R}^m, Y \in \mathbb{R}^p$$

- Goal (Tracking):

$$Y \to Y^*(t), \quad t \to \infty$$

Assumption: $Y^*(t)$ is smooth

- Special Class:

(control affine & square)

$$\dot{X} = f(X) + [g(X)]U$$
$$p = m, \quad [g_Y(X)] \text{ non-singular } \forall t$$

Dynamic Inversion Design

Known: $\dot{X} = f(X) + [g(X)]U$

$Y = h(X)$

- Derive the output dynamics:

$$\dot{Y} = \left(\frac{\partial h}{\partial X}\right)\dot{X}$$

$$= \left(\frac{\partial h}{\partial X}\right)\left\{f(X) + [g(X)]U\right\}$$

$$= f_Y(X) + [g_Y(X)]U$$

$$\frac{\partial h}{\partial X} \triangleq \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix}$$

- Define Error of Tracking:

$$E(t) \triangleq \left[Y(t) - Y^*(t)\right]$$

Dynamic inversion

- Select a ***fixed*** gain $K > 0$ such that:

$$\dot{E} + K\,E = 0 \qquad \Rightarrow \qquad \boxed{E = e^{-Kt}E_0 \to 0, \quad t \to \infty}$$

Usually $K = diag\left(1/\tau_i\right),\ \tau_i > 0$

- Carry out the algebra:

$$\left(\dot{Y} - \dot{Y}^*\right) + K\left(Y - Y^*\right) = 0$$

$$f_Y\left(X\right) + \left[g_Y\left(X\right)\right]U = \dot{Y}^* - K\left(Y - Y^*\right)$$

- Solve for the controller:

$$\boxed{U = \left[g_Y\left(X\right)\right]^{-1}\left\{\dot{Y}^* - K\left(Y - Y^*\right) - f_Y\left(X\right)\right\}}$$

A Demonstrative Toy Problem

- System Dynamics: $\dot{x} = \left(x + x^2\right) + \left(1 + x^2\right)u$
- Objective: $\left(y = x\right) \to 0 \text{ as } t \to \infty$
- Solution:
  - Desired output: $y^* = 0 \; \forall t, \quad \Rightarrow \dot{y}^* = 0 \; \forall t$
  - Desired error dynamics: $\left(\dot{y} - \dot{y}^*\right) + k\left(y - y^*\right) = 0, \quad k = \left(1/\tau\right) > 0$

$$\left(\dot{x} - 0\right) + k\left(x - 0\right) = 0$$

$$\left\{\left(x + x^2\right) + \left(1 + x^2\right)u\right\} + k\,x = 0$$

  - Control solution:

$$\boxed{u = \frac{1}{1 + x^2}\left[-\left(x + x^2\right) - k\,x\right]}$$

Is a first-order error dynamics always enforced?

Answer: **No!**

The order of the error dynamics is dictated by the "relative degree" of the problem, which is defined as the number of times the output needs to be differentiated so that the control variable appears explicitly.

If a second-order error dynamics needs to be enforced, then the corresponding equation is

$$\ddot{E} + K_V \dot{E} + K_P E = 0$$

Usually $K_V = diag\left(2\xi_i \omega_{n_i}\right),\ K_P = diag\left(\omega_{n_i}^2\right),\ \left(\xi_i, \omega_{n_i} > 0\right)$

Example

- System Dynamics: $$\dot{x}_1 = \left(2x_1x_2 + 3x_1^2 + 2\right) + 5\left(1 + x_1^2 x_2^2\right)u$$ $$\dot{x}_2 = -3x_1$$
- Objective: $$\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \rightarrow 0, \quad as \quad t \rightarrow \infty$$
- Output: $$y = x_2 \quad \text{(selecting a proper } y \text{ is crucial!)}$$ $$\dot{y} = \dot{x}_2 = -3x_1$$ $$\ddot{y} = -3\dot{x}_1 = -3\left[\left(2x_1x_2 + 3x_1^2 + 2\right) + 5\left(1 + x_1^2 x_2^2\right)u\right]$$

Example...contd.

- Solution:
  - Desired output: $y^* = 0\ \forall t \quad \Rightarrow \dot{y}^* = \ddot{y}^* = 0\ \forall t$
  - Desired error dynamics: $e \triangleq \left(y - y^*\right)$

$$\ddot{e} + 2\xi\omega_n\dot{e} + \omega_n^2 e = 0 \qquad 0 < \xi < 1,\ \omega_n > 0$$

$$\ddot{x}_2 + 2\xi\omega_n\dot{x}_2 + \omega_n^2 x_2 = 0$$

  - Control solution:

$$\boxed{u = \frac{1}{15\left(1 + x_1^2 x_2^2\right)}\left[-3\left(2x_1x_2 + 3x_1^2 + 2\right) - 6\xi\omega_n x_1 + \omega_n^2 x_2\right]}$$

When Does the Dynamic Inversion Fail?

Fundamental Principle of Dynamic Inversion:

1. Differentiate y repeatedly until the input u appears.
2. Design u to cancel the nonlinearity.

Q : Is it always possible to design u this way?

Ans: Not necessarily !

It is possible only if the relative degree is "well-defined".

Undefined Relative degree

Undefined relative degree : It may so happen that upon sucessive differentiation of y, u appears . However , the coefficient of u may vanish at X_0, whereas it is non-zero at points arbitrarly close to X_0. In such cases, the relative degree is undefined at X_0.

$$\text{Ex}: \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \rho(x_1, x_2) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = x_1^2, \quad \rho: \text{ Some nonlinear function}$$

Then $\dot{y} = 2x_1\dot{x}_1 = 2x_1x_2$

$$\ddot{y} = 2x_1\dot{x}_2 + 2\dot{x}_1x_2 = 2x_1\left[\rho(x_1, x_2) + u\right] + 2x_2^2$$

Undefined Relative degree

$$\ddot{y} = \underbrace{2x_1\ \rho\left(x_1, x_2\right) + 2x_2^2}_{f_y(X)} + \underbrace{2x_1\ u}_{g_y(X)}$$

As $x_1 = 0$, $g_y(x) = 0.$

Hence, at $x_1 = 0$ the relative degree is **not defined**.

Note: If one choses $y = x_1$, then

$$\dot{y} = \dot{x}_1 = x_2$$

$$\ddot{y} = \dot{x}_2 = \rho\left(x_1, x_2\right) + u$$

and the coefficient of $u = 1 \neq 0$ globally.

In this case the relative degree is **well defined** globally.

Advantages of DI design

- Simple design: No need of tedious gain scheduling (hence, sometimes dynamic inversion is known as a universal gain scheduling design).
- Easy online implementation: It leads to a 'closed form solution' for the controller.
- Asymptotic (rather exponential) stability is guaranteed for the Error dynamics (subjected to the control availability, this is true 'globally').
- No problem if parameters are updated (the updated values can simply be used in the formula).

Thanks for the Attention…!