A

EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2022**

Lecture 3A: Feedback Linearization, Computed Torque Control, Robust Control

Scribes: Jungpyo Lee and Austin Patel

3.1 Model Based Control

The dynamics of a system can be computed to be of the following form:

$m\ddot{\boldsymbol{x}}(t) + b\dot{\boldsymbol{x}}(t) + k\boldsymbol{x}(t) = \boldsymbol{u}(t)$

This leads to the following model-based control law:

$\boldsymbol{u}(t) = m(\ddot{\boldsymbol{x}}^{des}(t) - K_d\dot{\boldsymbol{e}}(t) - K_p\boldsymbol{e}(t)) + b\dot{\boldsymbol{x}}(t) + k\boldsymbol{x}(t)$

This control law contains both a servo-based component used as a PD or PID controller as well as a model-based component used to cancel system dynamics. The model based component is specific to the dynamics of the system.

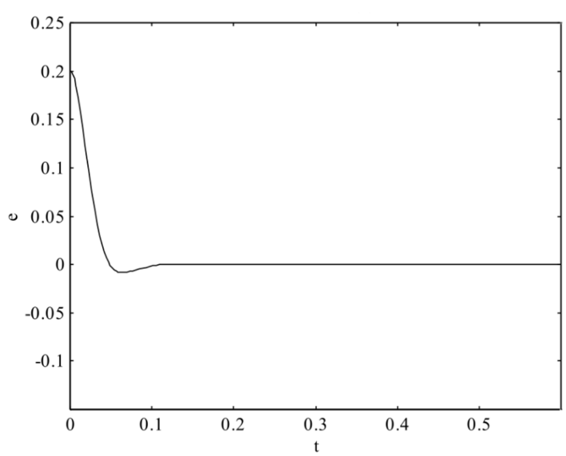
One drawback of this approach is that errors in model parameters will mean that error will converge to 0. In reality, we often can only estimate model parameters to get a control law of the form:

$\boldsymbol{u}(t) = \hat{m}(\ddot{\boldsymbol{x}}^{des}(t) - K_d\dot{\boldsymbol{e}}(t) - K_p\boldsymbol{e}(t)) + \hat{b}\dot{\boldsymbol{x}}(t) + \hat{k}\boldsymbol{x}(t)$

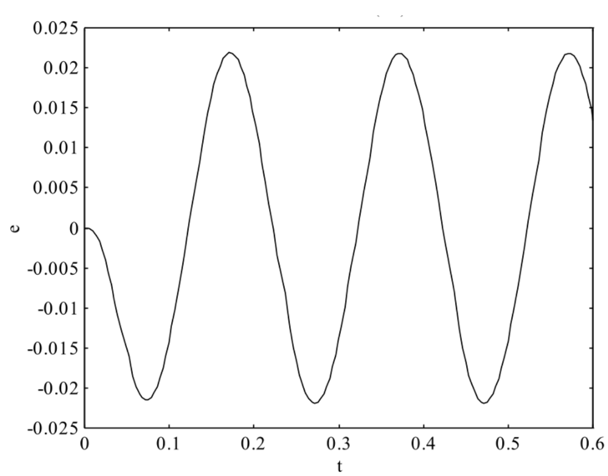
These model errors result in the following error expression:

$\ddot{\boldsymbol{e}} + K_d\dot{\boldsymbol{e}} + K_p\boldsymbol{e} = (1 - \frac{m}{\hat{m}})\ddot{\boldsymbol{x}} + \frac{\hat{b}-b}{\hat{m}}\dot{\boldsymbol{x}} + \frac{\hat{k}-k}{\hat{m}}\boldsymbol{x}$

These model errors will often lead to oscillatory behavior as depicted below.



Perfect model



Imperfect model – 10% errors

3.2 Fully Actuated vs Underactuated

Definition: A control system with coordinates $\boldsymbol{q}$ and inputs $\boldsymbol{u}$ is **fully actuated** if it can achieve any instantaneous acceleration in $\boldsymbol{q}$.

For "control-affine" systems, simple necessary and sufficient conditions for being fully actuated:

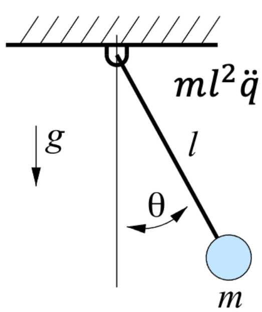
$\ddot{\boldsymbol{q}} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}) + g(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{u}$

Requires rank $g(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dim \boldsymbol{q}$

3.3 Phase Portraits in 2D

Graphical methods can be used to give a qualitative description of the behavior of state space systems (equilibrium, stability, basins of attraction).

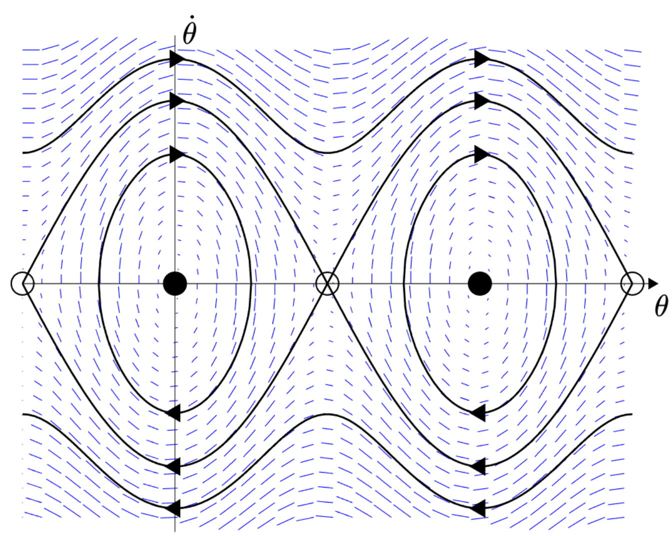
The 2D pendulum system illustrated below with system $\dot{x} = f(x)$ has the following vector field $f(x)$ over the domain of x.



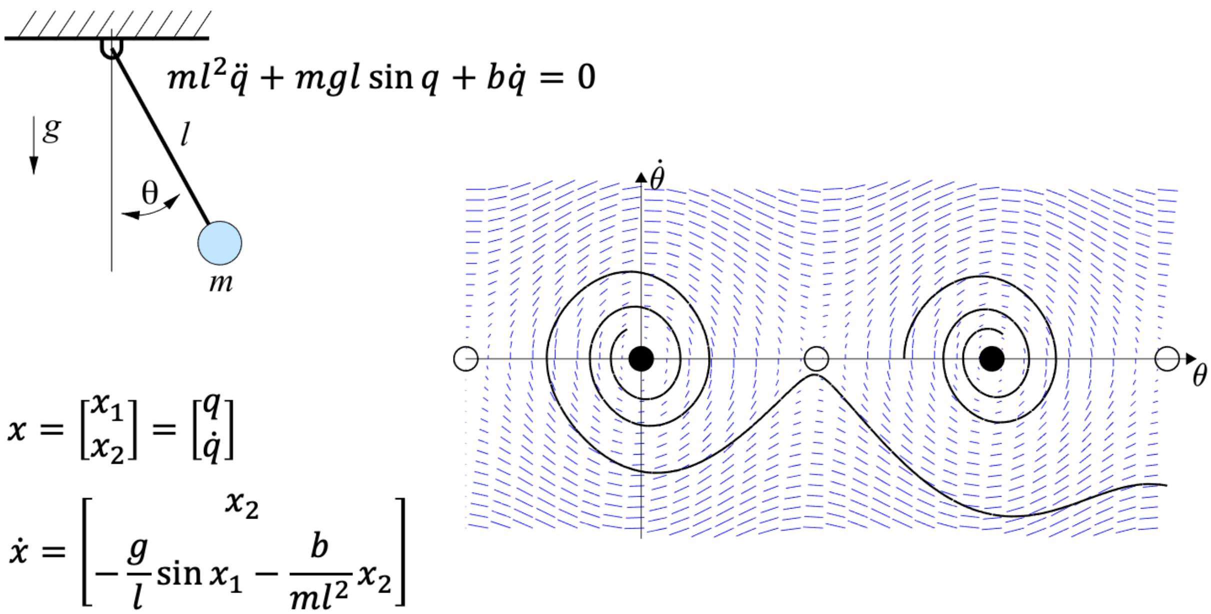
$$ml^2\ddot{q} + mgl\sin q = 0$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 \end{bmatrix}$$



If an additional dampening term is applied, the system will converge around equilibrium points.



3.4 Lyapunov Stability Theorem

For a system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ with $x \in \mathbb{R}, \boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, the equilibrium point $\boldsymbol{x} = 0$ is stable in $D \subset \mathbb{R}^n$ iff there exists a smooth function $V : D \subset \mathbb{R}^n \to \mathbb{R}$ such that:

$V(0) = 0$

$V > 0 \ \forall x \in D = \{0\}$

$\dot{V} \leq 0 \ \forall x \in D$

3.5 Lie Derivatives

The Lie derivative of a function $V(x)$ along a vector field f describes how the function changes along solutions of the differential equation.

$\frac{d}{dt}V(x(t)) = \mathcal{L}_f V(x(t))$

$\mathcal{L}_f V(x) = \frac{dV}{dx}(x) * f(x)$

Using this notation, Lyapunov's stability theorem requires:

$\mathcal{L}_f V(x) < 0$

Dampened Pendulum Lyapunov stability example:

$$V(\boldsymbol{x}) = \frac{1}{2}ml^2x_2^2 - mgl\cos x_1 \qquad \dot{\boldsymbol{x}} = f(\boldsymbol{x}) \qquad f(\boldsymbol{x}) = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 - \frac{b}{ml^2}x_2 \end{bmatrix}$$

$$\mathcal{L}_f V = \frac{dV}{dx} \cdot f(x) = [mgl\sin x_1 \quad ml^2x_2] \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 - \frac{b}{ml^2}x_2 \end{bmatrix}$$

$$\mathcal{L}_f V = (x_2 mgl\sin x_1) + (-x_2 mgl\sin x_1 - bx_2^2)$$

$$\mathcal{L}_f V = -bx_2^2$$

Note that $\mathcal{L}_f V(x)$ is negative for all x_2.

3.6 Input-Output Linearization

Also known as partial feedback linearization. The idea come up with a transformation to turn the nonlinear system into an equivalent linear system.

- State equations: $\dot{x} = f(x) + g(x)u$
- Output: $y = h(x)$
- Goal: $u = \alpha(x) + \beta(x)v$ such that $\dot{y} = v$

Then use the new virtual input v to control y.

Next step: Recipe for constructing $u = \alpha(x) + \beta(x)v$
Taking the derivative of the output we get

$$\dot{y} = \frac{dh}{dx}f + \frac{dh}{dx}gu$$

The Lie derivatives of h with respect to f, g:

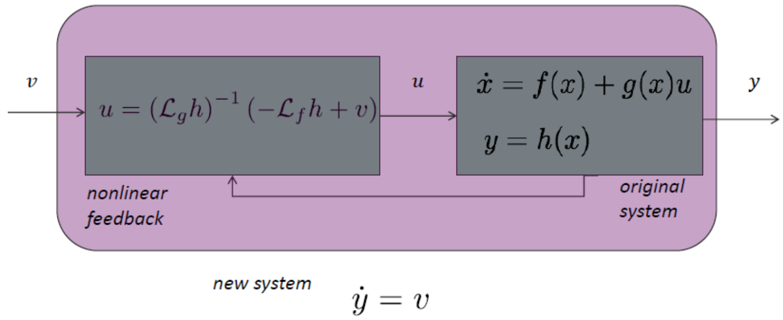
$$\mathcal{L}_f h(x) = \frac{dh}{dx}f(x) \quad \mathcal{L}_g h(x) = \frac{dh}{dx}g(x)$$

If $\mathcal{L}_g h(x) \neq 0$, then

$$u = \frac{1}{\mathcal{L}_g h(x)}(-\mathcal{L}_f h(x) + v)$$

results in

$$\dot{y} = v$$



- Rate of change of output: $\dot{y} = \mathcal{L}_f h + (\mathcal{L}_g h)u$
- Control law

if $\mathcal{L}_g h \neq 0$

$$u = \frac{1}{\mathcal{L}_g h}(-\mathcal{L}_f h + \dot{y}^{des} + k(y^{des} - y))$$

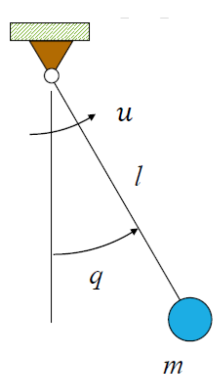
if $\mathcal{L}_g h = 0$

$$\dot{y} = \mathcal{L}_f h$$

3.6.1 SISO Systems

- State: $x \in R^n$
- Input: $u \in R$
- State equations: $\dot{x} = f(x) + g(x)u$
- Output: $y = h(x) \in R$
- Relative degree r: The index of the first nonzero term in the sequence (r is the first nonzero term of $\mathcal{L}_g \mathcal{L}_f^{r-1} h$)
- u = $\frac{1}{\mathcal{L}_g \mathcal{L}_f^{r-1} h}(-\mathcal{L}_f^r h + y^{(r)des} + k_1(y^{(r-1)des} - y^{r-1}) + \cdots + k_r(y^{des} - y))$
- general from of control law: $u = \alpha(x) + \beta(x)v$

Example



- State: $x = \begin{bmatrix} x_1 \\ x2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$
- State equations: $\dot{x} = \begin{bmatrix} x_2 \\ -\frac{g}{l} sin(x_1) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u$
- $h = x_1 \quad \mathcal{L}_g h = 0 \quad \mathcal{L}_f h = x_2 \quad \mathcal{L}_g \mathcal{L}_f h = \frac{1}{ml^2} \quad \mathcal{L}_f^2 h = -\frac{g}{l} sin x_1$
- u $= \frac{1}{\mathcal{L}_g \mathcal{L}_f h}(-\mathcal{L}_f^r h + \ddot{y}^{des} + k_1(\dot{y}^{des} - \dot{y}) + k_2(y^{des} - y))$

3.6.2 MIMO Systems

- State: $x \in R^n$
- Input: $u \in R^m$
- State equations: $x = f(x) + g(x)u$
- Output: $y = h(x) \in R^n$
- Assume each output has relative degree r
- Nonlinear feedback law: $u = (\mathcal{L}_g \mathcal{L}_f^{r-1} h)^{-1}(-\mathcal{L}_f^r + v)$
- Leads to equivalent system : $y^{(r)} = v$

Example: Robot Arm
Fully-actuated robot (n joints, n actuators)

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + N(q) = \tau$$

Dynamic model

- M is the positive definite, $n \times n$ inertia matrix
- C is the $n \times n$ matrix of Coriolis and centripetal forces
- N is the n-dimensional vector of gravitational forces
- τ is the n-dimensional vector of actuator forces and torques

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + N(q) = \tau$$

$$x = \begin{bmatrix} x_1 \\ x2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \qquad u = \tau \in \mathbb{R}^n$$

$$\dot{x} = \begin{bmatrix} x_2 \\ -M(x_1)^{-1}(N(x_1) + C(x_1, x_2)x_2) \end{bmatrix} + \begin{bmatrix} 0 \\ M(x_1)^{-1} \end{bmatrix} u$$

$$h(x) = x_1$$

$$f(x) = \begin{bmatrix} x_2 \\ -M(x_1)^{-1}(N(x_1) + C(x_1, x_2)x_2) \end{bmatrix} \qquad g(x) = \begin{bmatrix} 0 \\ M(x_1)^{-1} \end{bmatrix}$$

$$\mathcal{L}_g h = 0, \mathcal{L}_g \mathcal{L}_f h \neq 0$$

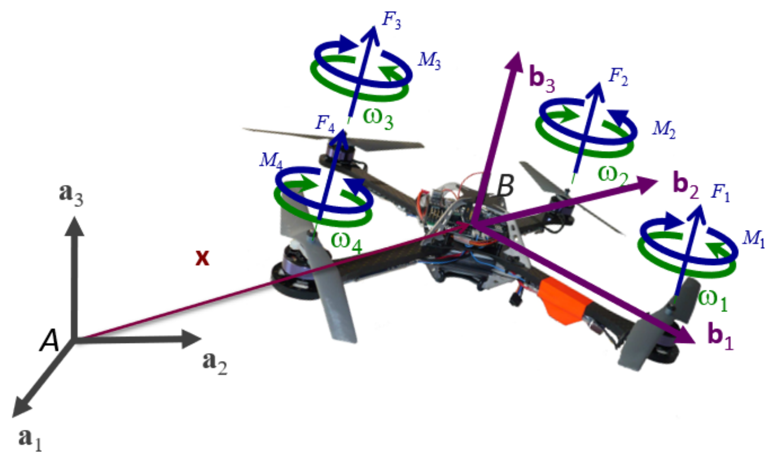
Relative degree is 2

$$u = (\mathcal{L}_g \mathcal{L}_f h)^{-1}(-\mathcal{L}_f \mathcal{L}_f h + \ddot{y}^{des} + k_1(\dot{y}^{des} - \dot{t} + k_2(y^{des} - y))$$

Control law

$$u = M(x_1)(M(x_1)^{-1}(N(x_1) + C(x_1, x_2)x_2) + \ddot{y}^{des} + k_1(\dot{y}^{des} - \dot{t} + k_2(y^{des} - y))$$

3.7 Quadcopter Control



$$F_i = k_F \sigma^2 \quad M_i = k_M \sigma_i^2 \quad \gamma = \frac{k_M}{k_F}$$

Equation of the motion, where R is orientation matrix

$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$$

$$\dot{R} = R\hat{\omega}$$

Euler equation

$$I\dot{\omega} + \omega \times I\omega = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}$$

Input

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} k_p\sigma_1^2 \\ k_p\sigma_2^2 \\ k_p\sigma_3^2 \\ k_p\sigma_4^2 \end{bmatrix}$$

$\sigma_1, \sigma_2, \sigma_3, \sigma_4$ are controlled by motor servos.

$$R = e^{\hat{z}\psi} e^{\hat{y}\theta} e^{\hat{x}\phi} \;\rightarrow\; Yaw,\; Pitch,\; Roll$$

The relationship between $\dot{\psi}$, $\dot{\theta}$, $\dot{\phi}$ and ω is expressed by Jacobian.

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = J(\psi, \theta, \phi)\omega$$

Output (position and yaw)

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \psi \end{bmatrix}$$

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \frac{u_1}{m} \end{bmatrix}$$

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \\ \ddot{y}_4 \end{bmatrix} = \times \times \times + \begin{bmatrix} R_2 & 0 \\ 0 & a_{42} a_{43} a_{44} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

Rank = 2!