EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2020**

Lecture 5: (Non-Linear Control)

Scribes: Mike Huetter,Hamza Kamran Khawaja

5.1 Announcements

- EECS 298 is a companion class that we can audit if people are looking for more advanced lectures:
  - Just sitting in might be beneficial
- The schedule is going to lose a week to account for spring break
- List of papers for lab reports will be up Wednesday or Thursday next week
  - Lab groups will each choose 2 papers for presentations
  - In groups of 2
- HW2 will be available soon

5.2 Model-based control

PD and PID control laws can be applied to real systems to conrol them

- $m\ddot{x}(t) - b\dot{x}(t) + Kx(t) = u(t)$ where $u(t)$ is a force or input to the system
- $u(t) = m(\ddot{x}^{dest}(t) - K_d\dot{e}(t) - K_p e(t)) + b\dot{x}(t) + kx(t)$
  where $(\ddot{x}^{dest}(t) - K_d\dot{e}(t) - K_p e(t))$ is called the feed-forward term, K_d is the derivative coefficient while K_p is the proportional coefficient
  - Servo-Based components
    * the $\ddot{x}^{dest}, K_d, and K_p$ elements are independent of the model
    * Tune PD or PID feedback portions to drive error to 0
  - Model Based components
    * the m,b, and k elements are part of the system
    * These elements cancel system dynamics

5.2.0.1 Model-based Control law using approximations

We can estimate values of m, b, k with $\hat{m}, \hat{b}, \hat{k}$ to obtain:

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

Substituting this into the original equation:

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

Advantages:

- We can decompose the control law into model-dependent and model-independent
- This lets us use the model independent part in any system
- Great for learning algorithms

Disadvantages:

- If model parameters have errors then error will not go to 0
- We estimate m,b, k and have to tune them with trial and error

5.3 Fully Actuated vs Under-actuated

A control system with coordinates q ($q \in \mathbb{R}^6$ for Quadrotors covered) and inputs u is fully actuated if it can achieve any instantaneous acceleration in q.
A necessary condition is for the number of control inputs to be at least as great as the number of degrees of freedom.
Under-actuated Systems are:

- Insufficient number of inputs
- Structure of dynamics
- actuator limits

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

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

Needs rank of $g(q, \dot{q})$= dim(q)

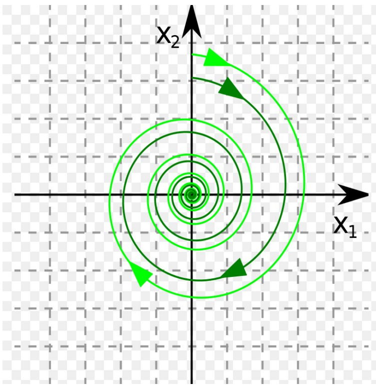
5.4 Holonomic and nonholonomic

Given a dynamical system with coordinates q

- Holonomic constraints are constraints on the configuration q
- Nonholonomic constraints include constraints on the velocities $\dot{q}$ which can not be integrated into holonomic constraints.

Example:
A toy car driving on the ground can achieve any configuration q =(x, y, ϕ) but it cannot drive sideways. This constraint is non-holonoimc because it is on the velocity not the configuration

5.5 Phase Portraits



Specific vector field in 2D
Generated by plotting the f(x) over the domain and placing the resulting vectors

- Circles are for equlibrium points (0 vectors)
- Full circles are stable
- Empty circles are unstable
- Follow vectors to equlibrium points

5.6 Lyapunov Stability theorem

For a system:

$$\dot{x} = f(x) | x \in R^n, f : R^n \to R^n$$

The equilibrium point x=0 is stable in D $\subset R^n$ iff there exists a smooth function V : D $\subset R^n \to R$ such that

$$V(0) = 0$$

$$V > -\forall x \in D - 0$$

$$\dot{V} <= 0 \forall x \in D$$

Lie Derivatives:
For system $\dot{x} = f(x)$
function V(x)
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 $\mathcal{L}_f V(x) < 0$

5.7 Input output linearization

Also known as partial feedback linearization
Used to convert a nonlinear system into an equivalent linear system

- State equations: $\dot{x} = f(x) + g(x)u$
- output: y=h(x)
- goal is to design $u = \alpha(x) + \beta(x)v$
- such that $\dot{y} = v$
- use rate of change of output: $\dot{y} = \mathcal{L}_f h + (\mathcal{L}_g h)u$
- if $\mathcal{L}_g h \neq 0$ then u = $\frac{1}{\mathcal{L}_g h}(-\mathcal{L}_f h + \dot{y}^{des} + k(y^{des} - y))$
- where v is virtual and $v = \dot{y}^{des} + k(y^{des} - y)$
- plugging into our state equations gives $\dot{y} - \dot{y}^{des} + k(y^{des} - y) => \dot{y} = v$
- if $\mathcal{L}_g h = 0$, then the rate of change of output is independent to u and we need to use a higher derivative

Higher Derivatives:

- Relative degree r: index of first nonzero term in the sequence
- r is the first nonzero term of $\mathcal{L}_g \mathcal{L}_f^{r-1} h$
- $u = \frac{1}{\mathcal{L}_g \mathcal{L}_f^{r-1} h}(-\mathcal{L}_f^r h + y_{dest}^{(r)} + k_1(y_{dest}^{(r-1)} - y^{r-1}) + ... + k_r(y_{dest} - ys))$

5.8 Multiple Input Multiple Output (MIMO) Systems

For a MIMO system, we have the state $x \in R^n$ and input $u \in R^m$ The state equation can be written in the form

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

$$y = h(x)$$

For this system, we assume that the output has a relative degree r. With this we can apply the Nonlinear feedback law

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

Which gets us to the equivalent system

$$y^{(r)} = v$$

For a fully actuated robotic arm (n joints, n actuators):

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

where:

- M is the positive definite nxn inertia matrix
- C is the n x 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

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

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \qquad u = \tau \in \mathbb{R}^n$$

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

$$h(x) = x_1$$

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

$$h(x) = x_1$$

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

$$u = (\mathcal{L}_g \mathcal{L}_f h)^{-1}((-\mathcal{L}_f^2 h + \ddot{y}^{des} + k_1(\dot{y}^{des} - \dot{y}) + k_2(y^{des} - y))$$

With the control law

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

5.9 image sources

- Phase Diagram: https://favpng.com/png$_{v}iew/mathematics - equilibrium - point - phase - portrait - mathematics - differential - equation - chaos - theory - png/siNtyytG$