CDS 110b: Lecture 1.1
Course Overview

Doug MacMynowski
4 Jan 2010

- Course administration for CDS 110b
- Introduce modern (optimization-based) control system structure

Course Admin

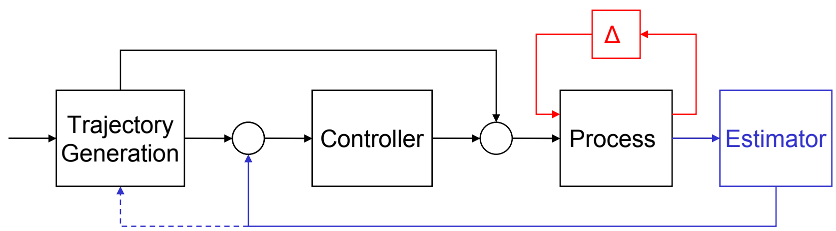
Web page:
- http://www.cds.caltech.edu/~macmardg/wiki/index.php?title=CDS_110b,_Winter_2010
- Or go to www.cds.caltech.edu/~macmardg and click on CDS110b link
- Notes typically posted by noon
- HW and solutions only available on web (not handed out in class)

Lectures:
- M 2-3 and W 1-3?
- We can change this if we can find a better time (but I'm at Stanford on Fridays)

Course texts:
- R. M. Murray, *Optimization-based Control*, 2008 preprint (link on website)
- Doyle, Francis, Tannenbaum, *Feedback Control Theory*, (link on website)
- Useful reference: Friedland, *Control System Design,* (I will put on reserve at SFL)

Grading:
- 50% Homework: Due on **Monday.**
  Three 2-day grace periods can be used any time during the quarter
- 20% Midterm: open-book, covering weeks 1-4
- 30% Final: open-book, covering weeks 1-9

Control System Design



Design controller
- Use process model, design in frequency domain
- Use process model, design in state-space with estimator and state feedback

Robustness to uncertainty
- Intuitive, through stability margins
- Formally, with rigourous analysis

Goals:
- Stabilization of unstable equilibrium
- Disturbance rejection (maintain equilibrium despite disturbances)
- Tracking (follow unknown reference trajectory, possibly with feedforward)
- Trajectory generation: Include design of reference as part of control design to accomplish some higher-level task

Classical vs "Modern" Control Design

Plant model:

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x) \end{aligned}$$

Linearize:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

Transfer function

$$\begin{aligned} y(s) &= C(sI - A)^{-1}Bu(s) \\ &= G(s)u(s) \end{aligned}$$

Design

$$u(s) = -K(s)y(s)$$

Design estimator

$$\dot{\hat{x}} = A\hat{x} + L(y - C\hat{x}) + Bu$$

Design state fdbk

$$u = -K(\hat{x} - x_d) + u_d$$

Implement:

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c(y - r) \\ u &= C_c x_c + k_r r \end{aligned}$$

Classical vs "Modern" Control Design, cont'd

Classical: (1940's to ~1960)
- Frequency domain design
- Graphical tools (Bode, Nyquist, Nichols, Root Locus)
- Intuition about how to tweak design
- MIMO hard!

"Modern": (1960's to…)
- State-space (time-domain)
- MIMO handled automatically
- Extends to nonlinear more easily
- Systematic design procedure
- Optimal design
- Key ideas:
  - Analytical specification of control goals
  - Systematic design for minimum "cost" (some performance metric)

Course Overview

1. Optimization: **Given** $J = J(x,u,t,T)$ **Find** $u = u(x,t,T) = \arg\min_u J$
   - Leads to on-line implementation of optimal control (RHC)
   - or optimal state feedback u = –Kx
   - Trajectory generation

   Weeks 1 – 4

2. Optimal estimation: **Given** $y = Cx + n$ **Compute** $\hat{x}$
   - Optimization: minimize covariance of error…
   - need to model stochastic processes

   Weeks 5 – 7

3. Robustness
   - How to quantify, analyze, and design
   - Formalizes classical control loop-shaping ideas
   - Connects time- and frequency-domain design

   Weeks 8 – 9

4. Brief comments on implementation, discrete-time control