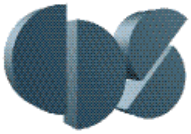
CDS 110b: Lecture 1-1
Modern Control Design



Richard M. Murray
7 January 2008

Goals

- Describe the course administration for CDS 110b
- Introduce modern (optimization-based) control systems structure
- Describe the trajectory tracking problem and gain scheduling

Reading

- Course syllabus (handout + web)
- Course notes: Trajectory Generation and Tracking (web)

Course Administration

Course web page: http://www.cds.caltech.edu/~murray/wiki/cds110b

- Homework and solutions will be posted on web only (not handed out in class)
- Notes for whiteboard lectures will typically be available by noon the day of class

Lectures: MWF 2-3 pm

- Will sometimes only have two lectures per week; schedule posted on course web page

Course Texts

- R. M. Murray, *Optimization-Based Control.* Preprint, 2008.
- K. J. Astrom and R. M. Murray, *Feedback Systems*, Princeton University Press, 2008.

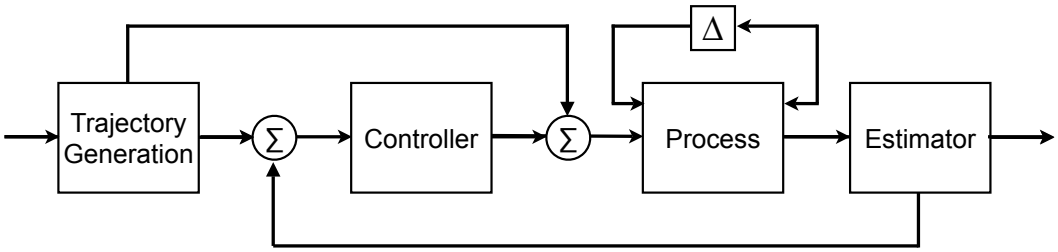
Grading

- Homework: 50% - weekly sets, due on Mondays. Max of 3 two-day grace periods allowed.
- Midterm: 20% - open book, will cover trajectory generation, optimal control, RHC
- Final: 30% - open book, out the last day of class, due the last day of finals
- Optional course project (to be discussed on Friday) - replaces midterm, final & some HW

Collaboration: encouraged!

- Write up your own solutions, including MATLAB scripts and plots
- No collaboration on midterm or final / project report done individually (based on joint work)

Modern Control System Design



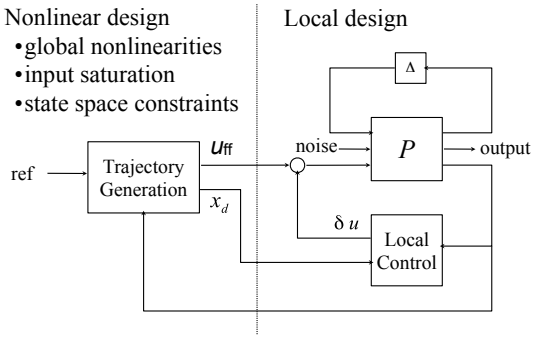
Traditional Control System: controller + process

- Corresponds to "inner loop" of most control system designs

Modern Control System: optimization-based design + robust analysis

- Replace reference with reference trajectory (Weeks 1-4)
- Replace process output with estimated output (Weeks 5-8)
- Replace "inner loop" controller with robust controller (Week 9-10 + CDS 212/213)

Two Degree of Freedom Control Design



Key idea

- Separate problems of generating a *feasible* trajectory from local *tracking*
- Use linear control theory for generating local control laws based on the *error*
- Use nonlinear and optimal control for generating feasible trajectories
- Recompute the trajectory during operation ⇒ *receding horizon control*

Case 1: stabilize equilibrium point

- Given equilibrium point (x_d, u_{ff}), find a controller $u = \alpha(x, x_d, u_{ff})$ that stabilizes x_d
- State space: $u = -K(x - x_d) + u_{ff}$
- Variation: $u = -K(x - x_d) + k_r\, r$

Case 2: track a reference trajectory

- Given a reference trajectory r(t), find a controller $u = \alpha(x, r(\cdot))$ such that

$$\lim_{t\to\infty} (r(t) - y(t)) = 0$$

- Controller can often depend on r at future times (anticipation)

Case 3: optimize performance criterion

- Find controller (trajectory + feedback) that minimizes a cost function:

$$\int_0^T L(x,u)\,dt + V(x(T),u(T))$$