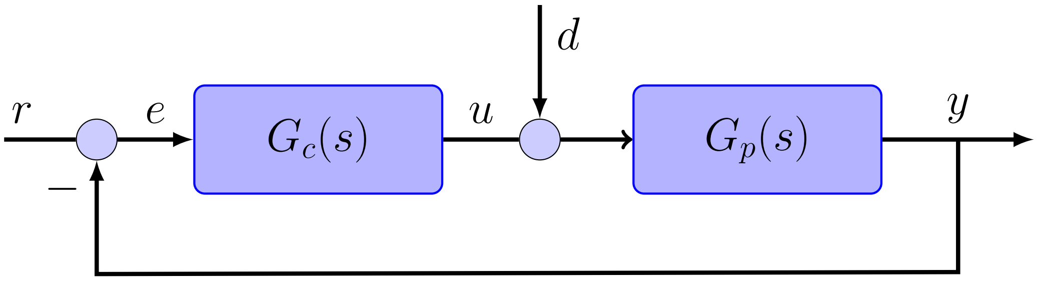
Lecture #1

16.30/31 Feedback Control Systems

- Motivation

16.30/31: Introduction



- **Goal:** Design a controller $G_c(s)$ so that the *system* has some desired characteristics. Typical objectives:
  - **Stabilize** the system (Stabilization)
  - **Regulate** the system about some design point (Regulation)
  - **Follow** a given class of command signals (Tracking)
  - Reduce response to **disturbances.** (Disturbance Rejection)

- Typically think of closed-loop control → so we would analyze the closed-loop dynamics.
  - Open-loop control also possible (called "feedforward") – more prone to modeling errors since inputs not changed as a result of measured error.

- Note that a typical *control system* includes the sensors, actuators, and the control law.
  - The sensors and actuators need not always be physical devices (*e.g.*, economic systems).
  - A good selection of the sensor and actuator can greatly simplify the control design process.
  - Course concentrates on the design of control law given the rest of the system (although we will need to model the system).

Why Use Control

- Typically easy question to answer for aerospace because many vehicles (spacecraft, aircraft, rockets) and aerospace processes (propulsion) need to be controlled just to function
  - Example: aircraft doing aggressive maneuvers difficult to fly by hand - dynamics are unstable and nonlinear (aerodynamic and geometric)

Fig. 1: Vertical hover

Fig. 2: MIT Urban Challenge LR3.

  - Example: The LR3 dynamics of the MIT's DARPA Urban Challenge vehicle are nonlinear, unstable, constrained by limitations.
    - Car will not track desired path without feedback control.

- But there are also many stable systems that simply require **better performance** in some sense (e.g., faster, less oscillatory), and we can use control to **modify/improve** this behavior.

Feedback Control Approach

- Establish control objectives
  - Qualitative – don't use too much fuel
  - Quantitative – settling time of step response $<$3 sec
  - Typically requires that you understand the process (expected commands and disturbances) and the overall goals (bandwidths).
  - Often requires that you have a strong understanding of the physical dynamics of the system so that you do not "fight" them in inappropriate (*i.e.*, inefficient) ways.

- Select sensors & actuators
  - What aspects of the system are to be sensed and controlled?
  - Consider sensor noise and linearity as key discriminators.
  - Cost, reliability, size, . . .

- Obtain model
  - Analytic (FEM) or from measured data (system ID)
  - Evaluation model → reduce size/complexity → Design model
  - Accuracy? Error model?

- Design controller
  - Select technique (SISO, MIMO), (classical, state-space)
  - Choose parameters (ROT, optimization)

- Analyze closed-loop performance. Meet objectives?
  - Analysis, simulation, experimentation, . . .
  - Yes ⇒ done, No ⇒ iterate . . .

Assumptions and Philosophy

- I assume that given a relatively simple (first or second order) system, and a set of stability/performance requirements:
  - Given a controller, you can analyze the stability & expected performance
  - You have a reasonably good idea how to design a controller using classical techniques (Bode and/or root locus)

- Will not focus too much on the classical design in this course, but personally think it is important to know "the design process" for classical controllers, and you should be able to provide a "classical interpretation" of any advanced control technique as a "sanity check"
  - Try to answer "why did it do that"?
  - Not always possible, but always a good goal.
  - Texts provide full design process – we will only review it

- Observation: classical control the process tends to be a bit cryptic, is designer intensive (lots of simple but detailed calculations), and typically very iterative
- So our focus in this class is on state space methods for several reasons:
  - More systematic design tools exist - can be easily codified and implemented numerically – easy integrate optimization
  - Easily handle larger-scale systems (many modes) and many sensors and actuators (MIMO).

State-Space Approach

- Basic questions that we will address about the state-space approach:
  - What are state-space models?
  - Why should we use them?
  - How are they related to the transfer functions used in classical control design?
  - How do we develop a state-space model?
  - How do we design a controller using a state-space model?

- **Bottom line:**

1. **What:** representation of the dynamics of an n^{th}-order system using n first-order differential equations:

$$\begin{aligned} & m\ddot{q} + c\dot{q} + kq = F \\ \Rightarrow \quad & \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u \\ \Rightarrow \quad & \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \end{aligned}$$

2. **Why:**
   - State variable form convenient way to work with complex dynamics. Matrix format easy to use on computers.
   - Transfer functions only deal with input/output behavior, but state-space form provides easy access to the "internal" features/response of the system.
   - Allows us to explore new analysis and synthesis tools.
   - Great for multiple-input multiple-output systems (**MIMO**), which are very hard to work with using transfer functions.

3. **How:** There are a variety of ways to develop these state-space models. We will explore this process in detail.
   - Linearization of nonlinear models
   - Derivation from simple linear dynamics

4. **Control design:** Split into 3 main parts
   - *Full-state feedback* –fictitious since requires more information than typically (ever?) available
   - *Observer/estimator design* –process of "estimating" the system state from the measurements that are available.
   - *Dynamic output feedback* –combines these two parts with provable guarantees on stability (and performance).
   - Fortunately there are **very** simple numerical tools available to perform each of these steps

   - Removes much of the "art" and/or "magic" required in classical control design → **design process more systematic.**

- **Observation:** – Linear systems theory involves extensive use of linear algebra.
  - Will not focus on the theorems/proofs in class – details will be handed out as necessary, but these are in the textbooks.
  - Will focus on using the linear algebra to understand the behavior of the system dynamics so that we can modify them using control. **"Linear algebra in action"**
  - Even so, this will require more algebra that most math courses that you have taken . . . .

Nonlinear and Robustness

- Significant difference from previous years is that we will spend more time on the analysis of the system stability assuming various types of basic nonlinearities
  - All systems are nonlinear to some extent
  - How well will the controller based on linear model assumptions work on the full nonlinear system?
- Goal is to develop tools to work with a nonlinear model, linearize it, design a controller, and then analyze the performance on the original nonlinear system.
  - Also interested in understanding how the addition of certain types of nonlinearities (saturation, rate limits, stiction) might influence stability/performance.
- Will also explicitly consider the effects of uncertainty in the dynamic models
  - Basic issues such as gain and timing errors
  - Parameter errors
  - Unmodeled dynamics

Implementation Issues

- Many Universities used to teach an entire class on digital control analysis and design
  - I did at Stanford for many years and have a good set of notes on the subject if you are interested

- But with the increase in processor speeds and the ability to develop code right from simulink (e.g., RTW), there is not much point discussing all of the intricacies of digital design
  - But sometimes you have to use a small computer and/or you don't get all the cycles

- So we will discuss the process enough to capture the main issues and the overall implementation approach
  - Bottom line is that as long as you "sample" fast enough the effects are not catastrophic - it just adds a little bit more delay to the feedback loop.
  - Easily predicted and relatively easy to account for.
  - Provides feedback on how fast you need to go – go faster costs more money and can create numerical implementation issues

- Much of the implementation done for the labs will be using code automatically developed straight from simulink
  - You are probably familiar with Matlab, but should practice with Simulink as well, as that will be used for the HW and the labs.

Feedback

- Muddy cards and office hours.
  - Help me to know whether my assumptions about your backgrounds is correct and whether there are any questions about the material.

16.30 / 16.31 Feedback Control Systems
Fall 2010