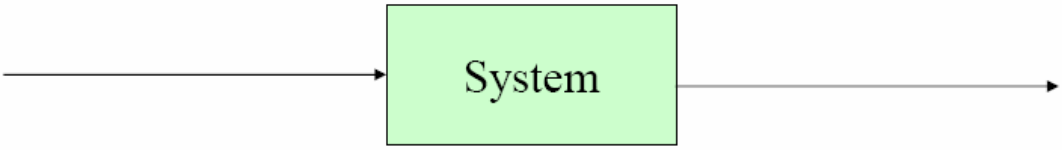
An Introduction to Control and Digital Control Ideas

Jee-Hwan Ryu

School of Mechanical Engineering
Korea University of Technology and Education

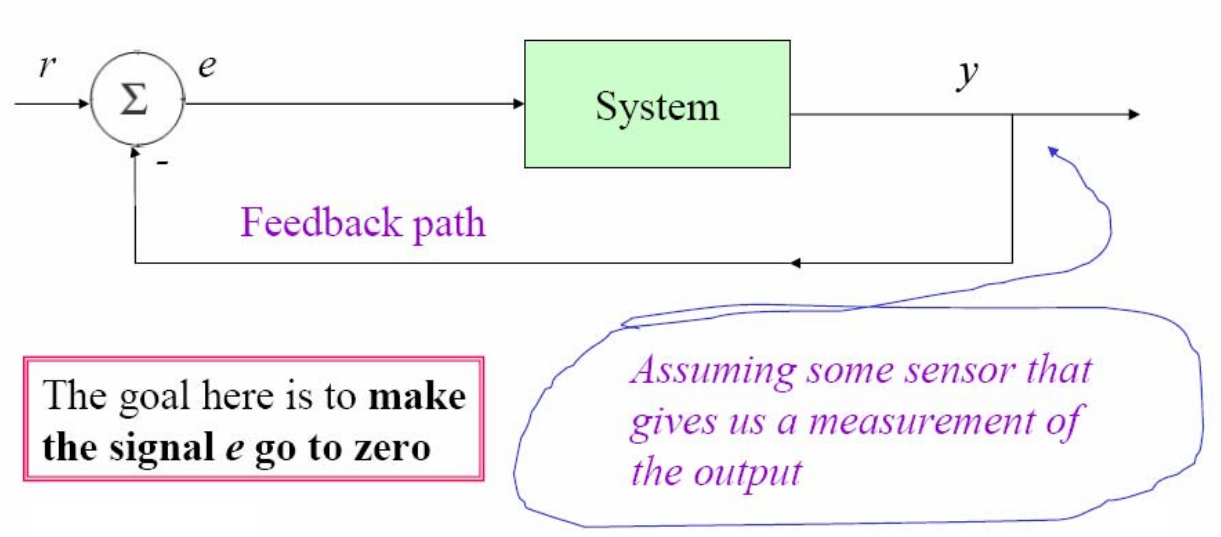
Suppose that we are given a system, and we want to make it behave in some particular, specified way, in response to various inputs



This is the "**Control Problem**" – it can be thought of as **choosing the input** and **changing** the system so that it has a desired input-output behavior

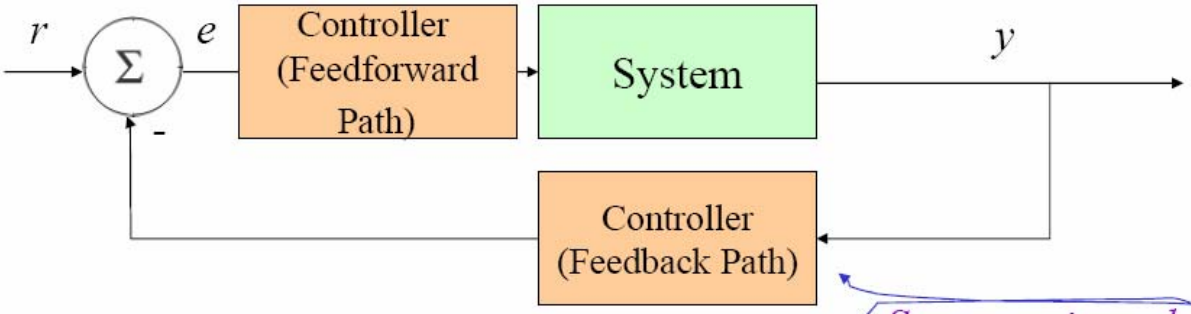
For example, suppose that we want to make the output of the system "**track**" (that is "**match**" or "**follow**") a certain shape

We can think of this as making the system output, y, follow the system **reference input**, *r*. Let e be the error *y = r-e*



The goal here is to somehow **make the signal *e* go to zero** (e is either a discrete time or a continuous time signal)

We do this by incorporating "controllers" into the overall architecture, so that the resulting system has the desired transfer function



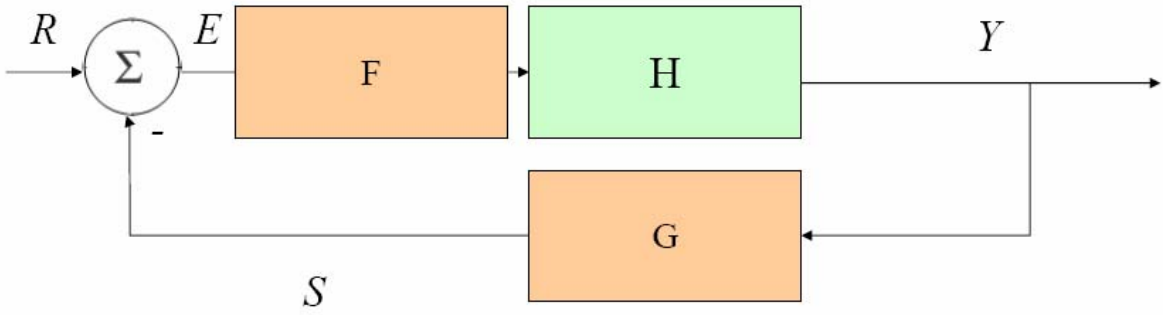
Desired transfer function in this case?

Feedback path

$\frac{Y}{R} = 1$

Sensor gain and dynamics can be included in this

We assume that all 3 subsystems below are linear and time invariant (either all in continuous time or all in discrete time). Names are given below for their transfer functions (in whichever domain is appropriate)



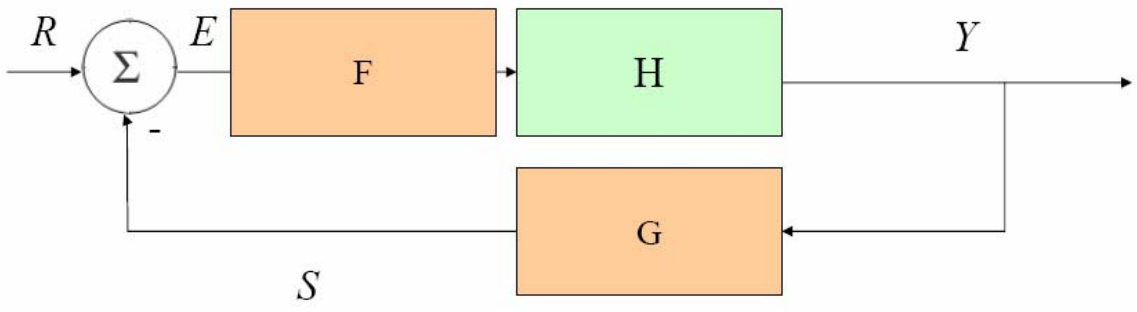
In the frequency domain, we can multiply inputs with transfer functions to get outputs

$E = R - S$
$Y = HFE$
$S = GY$

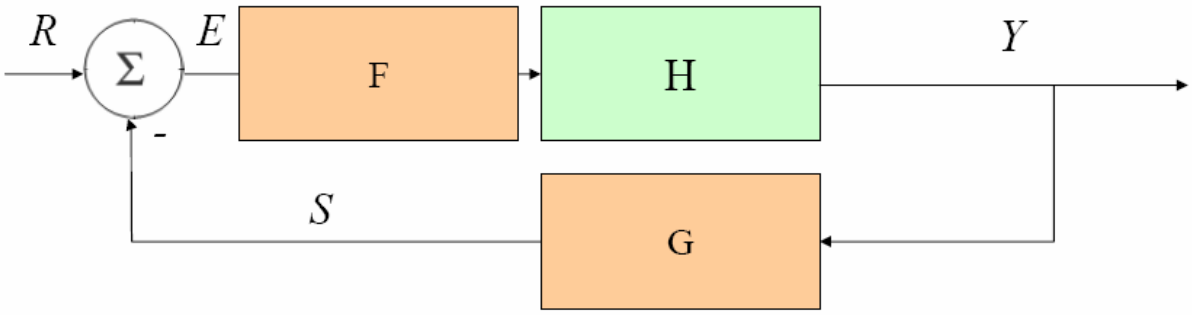
$Y = HF(R-S) = HF(R-GY)$

$[1+HFG]Y=HFR$

$$\frac{Y}{R} = \frac{HF}{1+HFG}$$



We choose F and G (already given H)—that is, we design controllers in either the feedback path, feedforward path, or both – to make Y/R be what we want



$$\frac{Y}{R} = \frac{HF}{1+HFG}$$

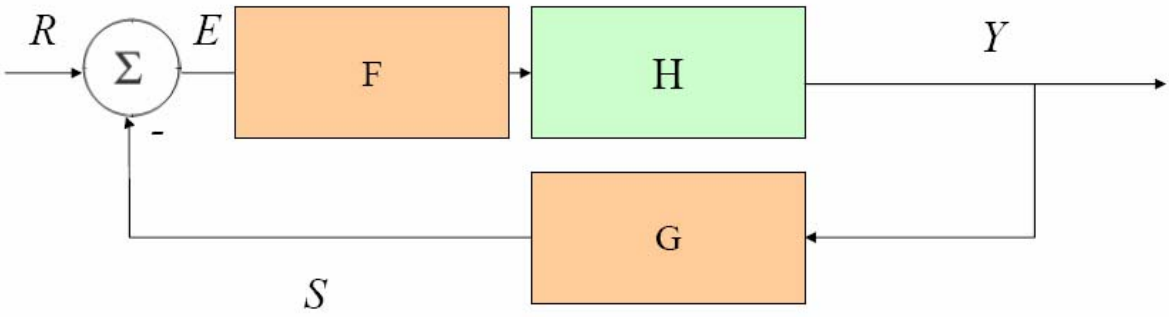
Closed Loop Transfer Function

$$\frac{Y}{R} = \frac{HF}{1+HFG}$$

The poles of this "closed loop transfer function are determined by

$1+HFG = 0$

and the zeros are determined by $HF=0$



So you can think of the controller design process as changing the gain, poles and zeros of the original system H, by selecting these quantities for the controllers *F* and *G*

There are a great many techniques to do this controller design – this is the topic of various courses in control

- **Feedback control systems**
- **Linear control system**
- **Digital control**
- **Adaptive control**
- **Etc.**

Some History of the Control-I

- Water clock (Greeks, 300 BC)
- Steam engine
- Classical Control (1930 – 1960)
  - Feedback amplifiers, servomechanisms
  - Continuous time, time invariant, SISO, mostly linear
  - Root locus, Nyquist methods, Bode plot

Some History of the Control-II

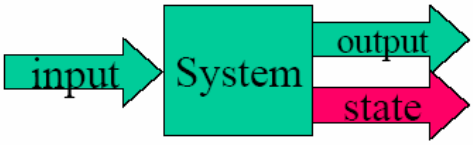
- Modern Control – State space approach
  - Kalman (1960)
  - MIMO, time-varying
  - Use of linear feedback to place all poles of LTI systems
- **With the advent of low cost computation → digital control implementations became preferred**

Digital Control Problem

- Usually trying to control a continuous time system, but using a digital controller to control it
- Need to be able to convert from continuous time signal to discrete time signal ("A/D converter") – **this is just sampling**
- Need to be able to convert from discrete time to continuous time signal ("D/A converter") – **may ways to do this**

Linear Time Invariant System (in continuous time, t)

- input u(t)
- output y(t)
- state x(t)



- system described by **impulse response function h(t)**
- y(t) = h(t)* u(t) where * denotes the convolution operation
- Taking Laplace Transforms of signals to get U(s), Y(s) and H(s), we have Y(s) = H(s)U(s)
- Convolution (in time domain) involves integrals
- **Calculation in Laplace ("s") or Fourier ("frequency") domain, involving multiplication, is much easier**

Linear Time Invariant System (in discrete time, k)

- input u[k]
- output y[k]
- state x[k]

- system described by **impulse response sequence h[k]**
- y[k] = h[k]* u[k] where * denotes the convolution operation in discrete time
- Taking Z Transforms of signals to get U(z), Y(z) and H(z), we have Y(z) = H(z)U(z) *--so the Z Transform has the same role for discrete time systems that the Laplace Transform does for continuous time systems*
- Convolution (in time domain) involves sums
- **Calculation in z (frequency) domain, involving multiplication, is much easier**

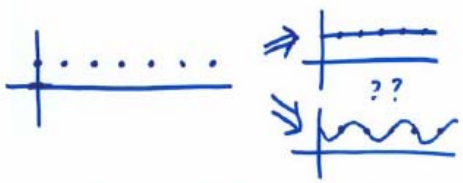
Digital Signal Processing

- Continuous time signal source and continuous time result but use a digital representation to store process
- Need to be able to convert from continuous time signal to discrete time signal ("A/D converter") – this is sampling and often compression
- Often need to be able to convert from discrete time to a continuous time signal ("D/A converter") – many ways to do this

Sampled Signals

- Converting continuous time signal into discrete time system is unambiguous—e.g., sample value every T [note that we are throwing away information]

- Converting discrete time signal into continuous time system is not clear---many ways to do



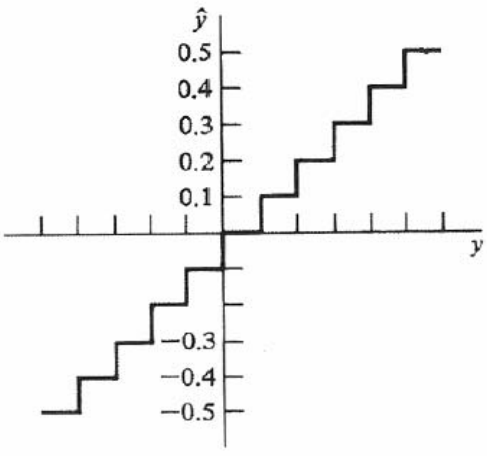
- Need extra information to know how to interpolate

Sampled Signals

- If you want to sample a continuous time signal, and then be able to reconstruct it perfectly—you need additional information such as:
  - How to interpolate points (eg., hold constant or linear interpolation)
  - The highest frequency content of the signal and the knowledge that you sample faster than twice that frequency (the Nyquist Sampling Theorem)
- Usually signals don't have sharp cutoff frequencies—but you can add filters into the design that make this so

Quantization

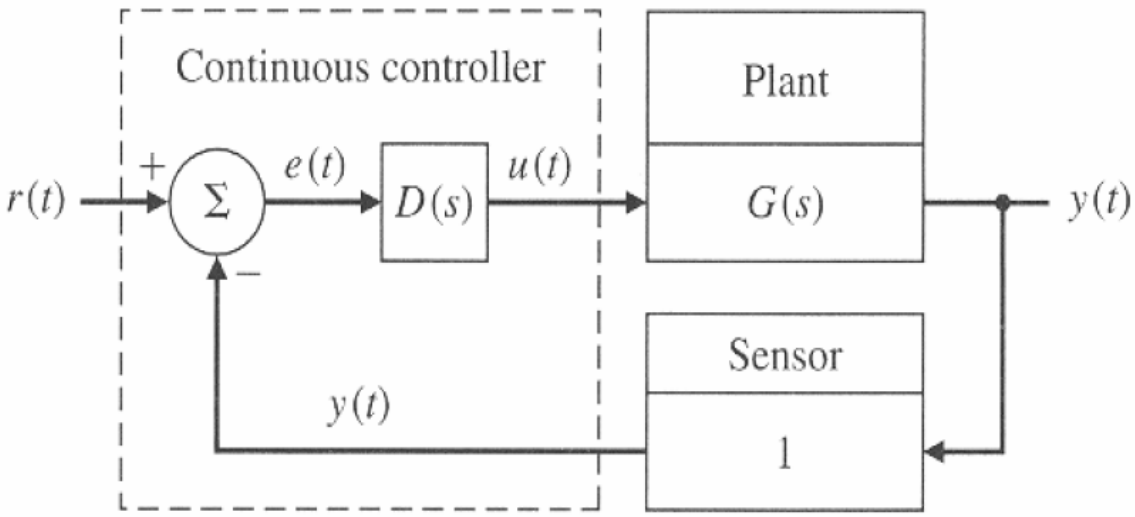
- Digital signals are discrete time, and the signal magnitude can only take discrete values [this is a source of error, if the quantization steps are too large]



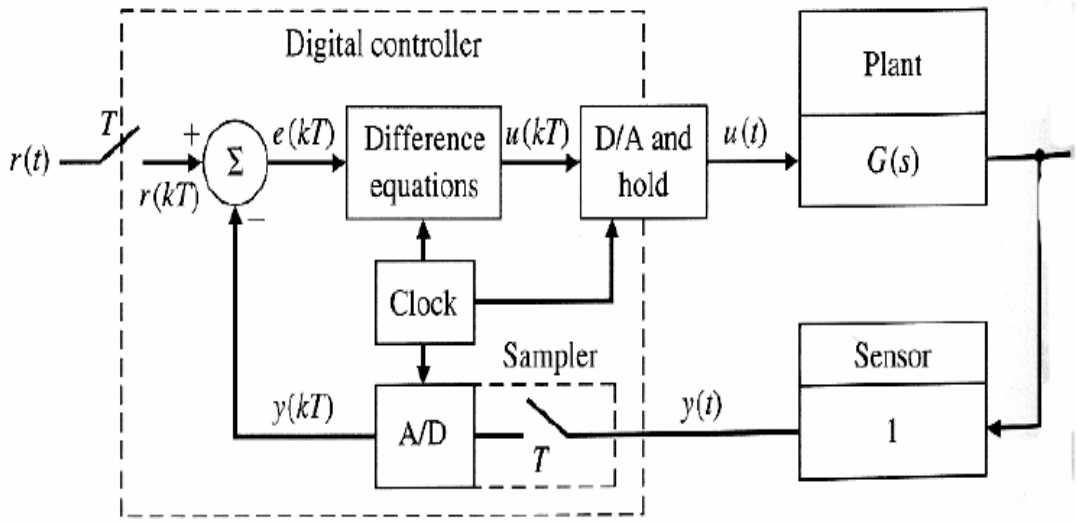
Why Use Digital Control ?

- Often Cheaper
- Usually Smaller/lighter
- Usually needs Less power
- Often More precise
- Can Re-program
- Can use same component (with different programming) ➔ Generallity

Continuous Control (SISO case)



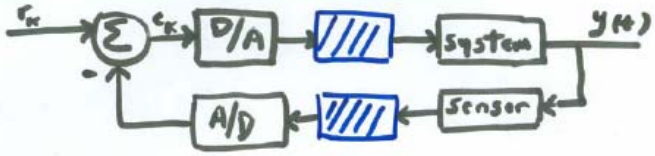
Digital Control (SISO case)



Two Approaches to Digital Control

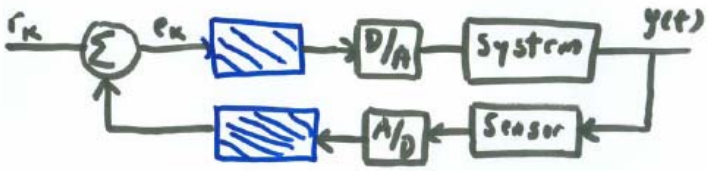
1. Do controller design in continuous time, then implement digitally

"emulation"



2. Convert system from CT to DT, then do controller design in discrete time

"direct digital design"



System Design Specifications-I

- In time domain
  - Stability
  - Step response (rise time, settling time, overshoot, steady-state error)
- In frequency domain
  - Filtering properties (high pass, low pass, band pass)

System Design Specifications-II

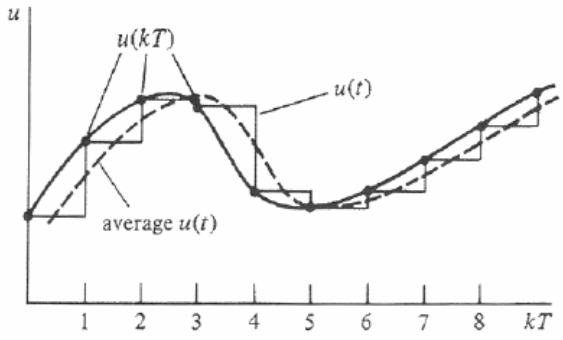
- Robustness
  - Resistance to modeling errors, noise, small time variations
  - Gain margin, phase margin, sensitivity
- Optimal Control
  - Maximize or minimize a specified objective function
  - Usually a cost on trajectory tracking error and on energy/control input; also may have costs based on trajectory endpoints

System Design Specifications-III

- Usually the designer must extract this information from the customer
- Translate the design needs into “intermediate specifications” – things like z plane or s plane pole locations
- For digital control, selection of the sampling time T is an additional consideration (and perhaps associated filters for anti-aliasing)

D/A and A/D = T/2 Delay

- Most common D/A is “zero order hold”
  - Between sampling times, set the analog signal to the same value (“staircase approximation”)
  - Considering ZOH as a system, it is causal, linear
- Sampling with period T, and then using ZOH is like including a T/2 delay in the system



Impact of this in emulation…

- If you sample fast enough (>30 times system bandwidth) ➔ not a big source of error
- If sampling less than 30 times bandwidth, need to include the T/2 delay in the continuous time system model – so that the continuous time controller is designed to compensate for it

Emulation Design vs Direct Digital Control

- Emulation
  - Can use continuous time methods (well developed)
  - Few new tools needed
  - Works well if sampling fast
  - Mapping of control law from continuous time to discrete time is not exact
  - Ignore continuous system response between sampling times

Emulation Design vs Direct Digital Control

- Direct digital control
  - Design of discrete time control law (and thus digital closed loop system) is exact for any sampling rate
  - Ignore continuous system response between sampling times