An Introduction to Control Theory

Hongwei Zhang

http://www.cs.wayne.edu/~hzhang

Acknowledgment: Joseph Hellerstein, Sujay Parekh, Chenyang Lu

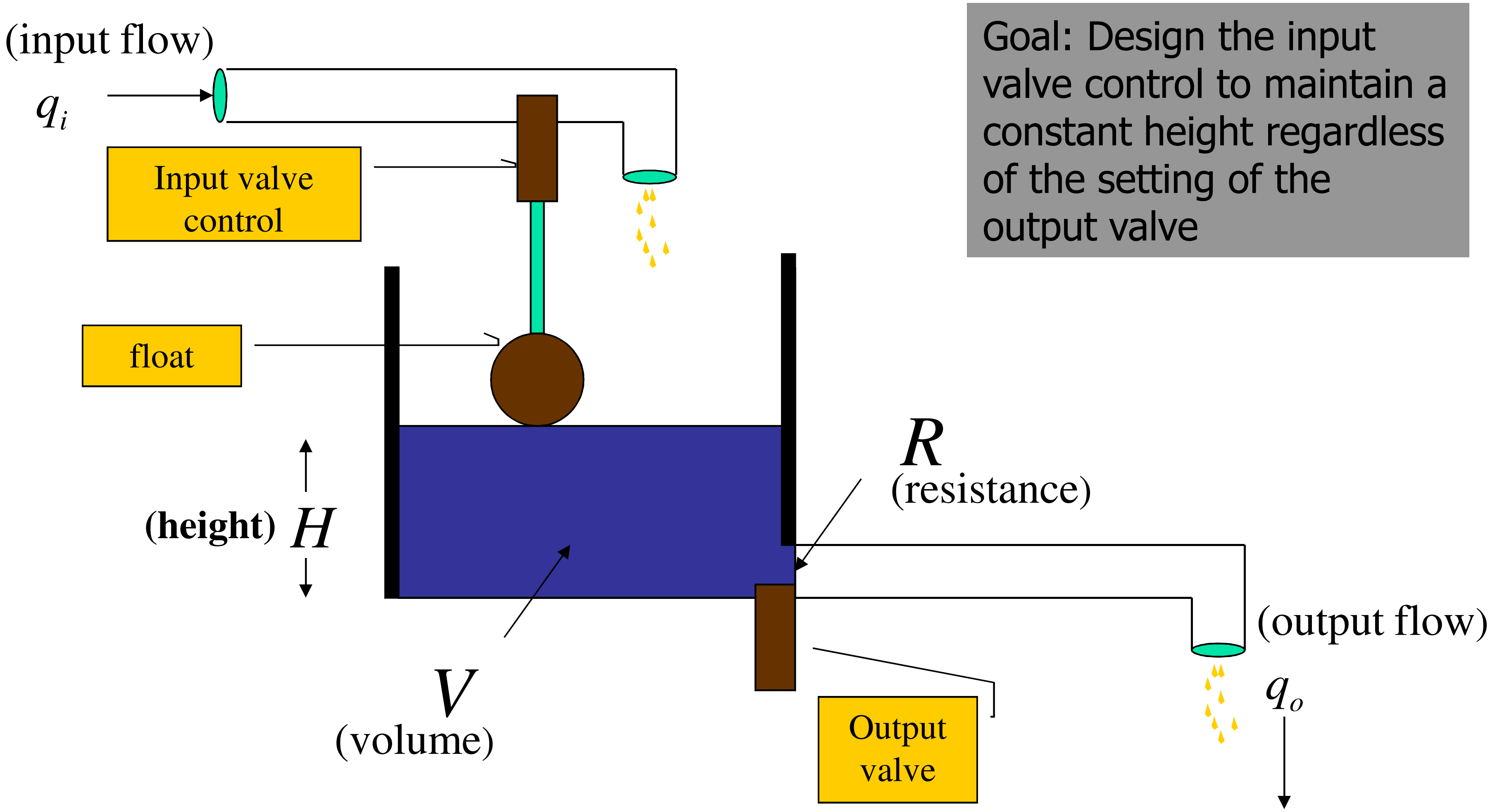
Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems

Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems

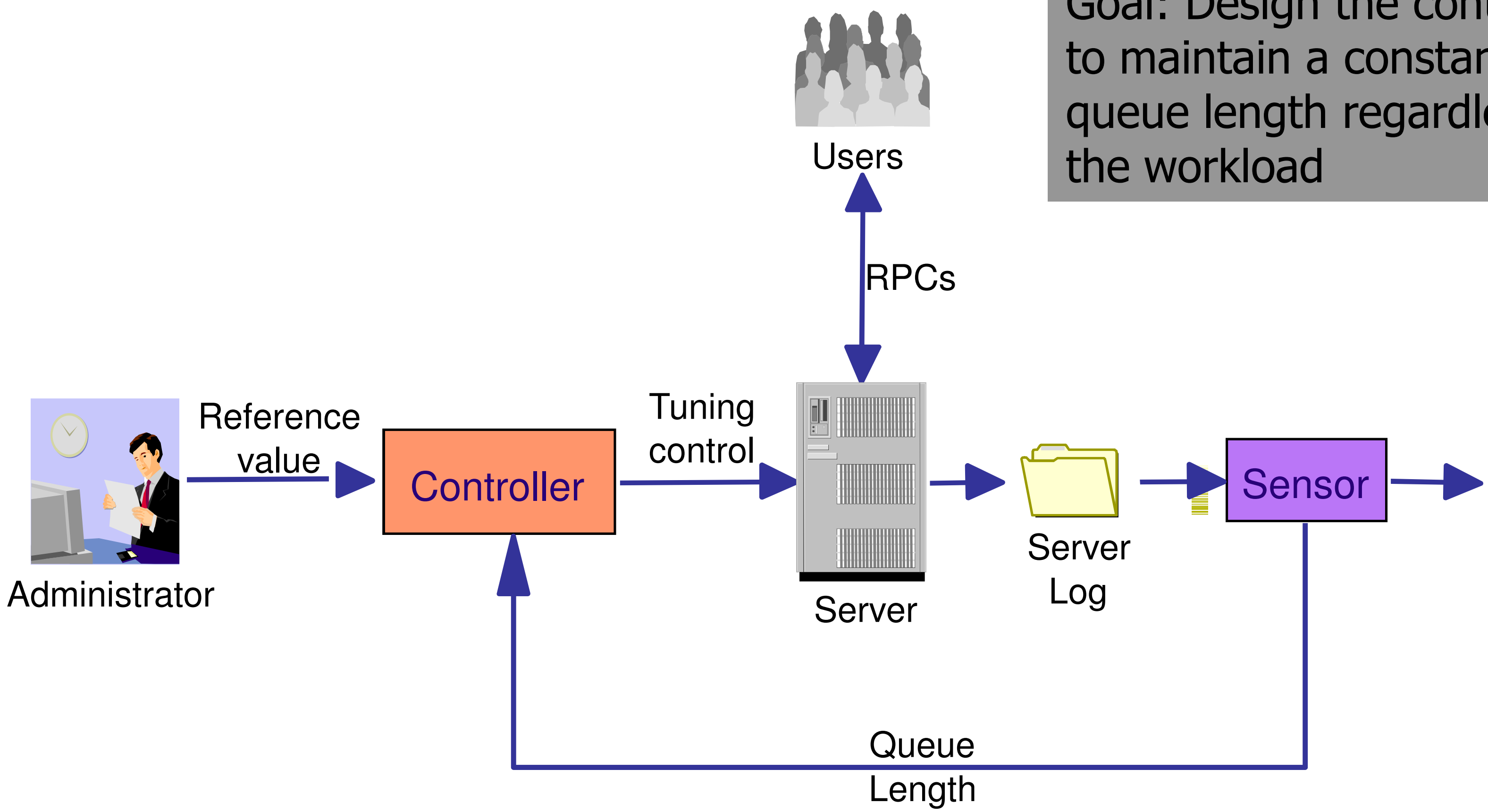
Example 1: Liquid Level System



Goal: Design the input valve control to maintain a constant height regardless of the setting of the output valve

Example 2: Admission Control

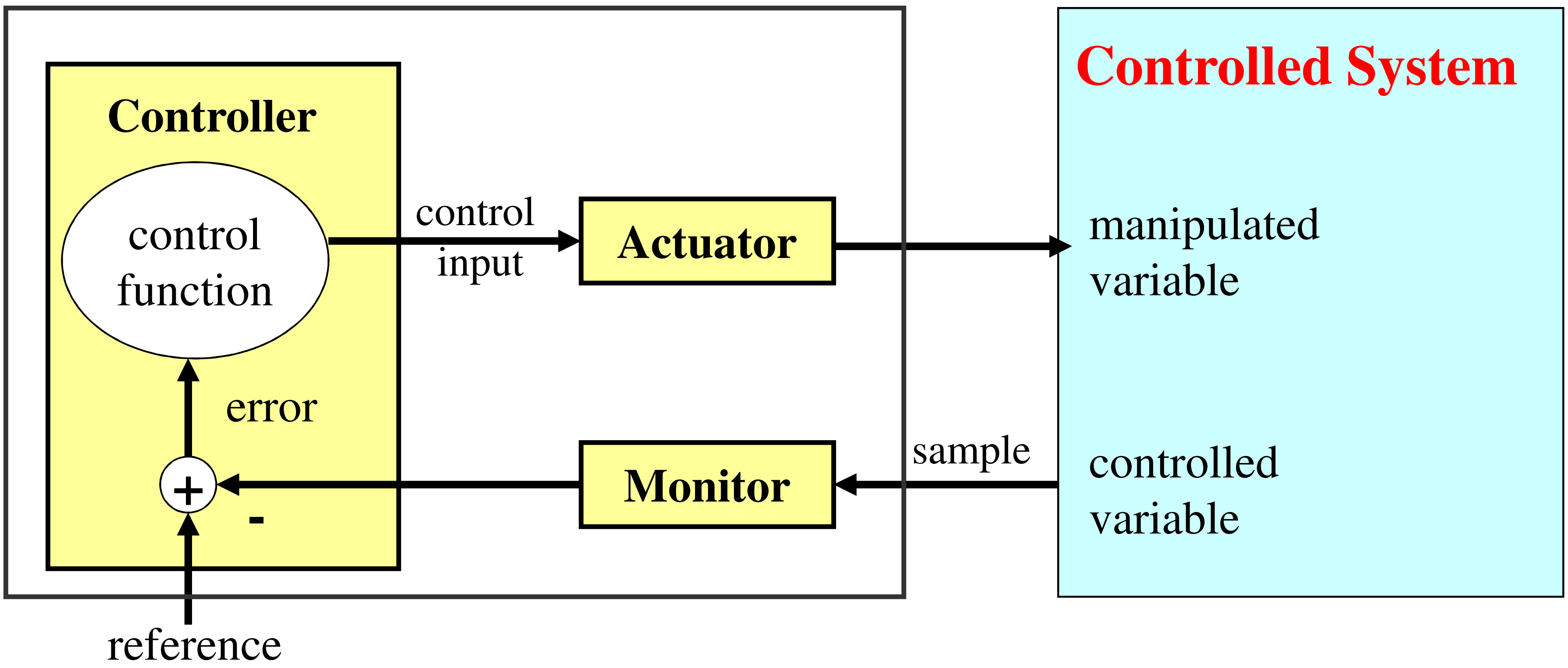
Goal: Design the controller to maintain a constant queue length regardless of the workload



Open-loop control

- Compute control input without continuous variable measurement
  - Simple
  - Need to know **EVERYTHING ACCURATELY** to work right
    - Cruise-control car: friction(t), ramp_angle(t)
    - E-commerce server: Workload (request arrival rate? resource consumption?); system (service time? failures?)
- Open-loop control fails when
  - We don't know everything
  - We make errors in estimation/modeling
  - Things change

Feedback (close-loop) Control



- Measure variables and use it to compute control input
  - More complicated (so we need control theory)
  - Continuously measure & correct
    - Cruise-control car: measure speed & change engine force
    - E-commerce server: measure response time & admission control
    - Embedded network: measure collision & change backoff window
- Feedback control theory makes it possible to control well even if
  - We don't know everything
  - We make errors in estimation/modeling
  - Things change

Why feedback control?
Open, unpredictable environments

- Deeply embedded networks: interaction with physical environments
  - Number of working nodes
  - Number of interesting events
  - Number of hops
  - Connectivity
  - Available bandwidth
  - Congested area
- Internet: E-business, on-line stock broker
- Unpredictable off-the-shelf hardware

Why feedback control?
We want QoS guarantees

- Deeply embedded networks
  - Update intruder position every 30 sec
  - Report fire <= 1 min
- E-business server
  - Purchase completion time <= 5 sec
  - Throughput >= 1000 transaction/sec
- <u>The problem:</u>

provide QoS guarantees in open, unpredictable environments

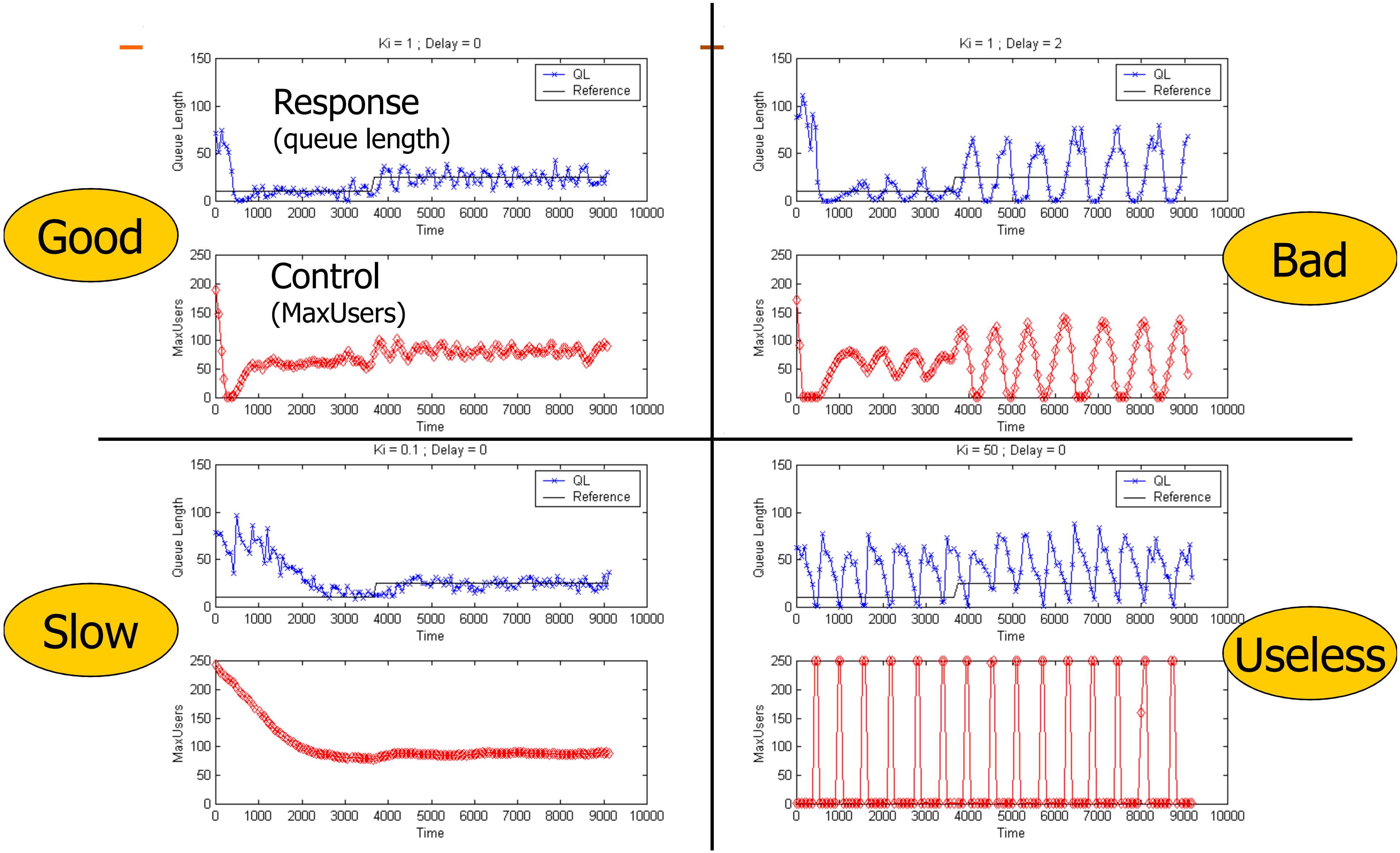
Advantages of feedback control theory

- Adaptive resource management *heuristics*
  - Laborious design/tuning/testing iterations
  - Not enough confidence in face of untested workload
- *Queuing theory*
  - Doesn't handle feedbacks
  - Not good at characterizing transient behavior in overload

- *Feedback control theory*
  - Systematic approach to analysis and design
    - Transient response
    - Consider sampling times, control frequency
    - Taxonomy of basic controls;

      Select controller based on desired characteristics
  - Predict system response to some input
    - Speed of response (e.g., adjust to workload changes)
    - Oscillations (variability)
  - Approaches to assessing stability and limit cycles

Example: Control & Response in an Email Server



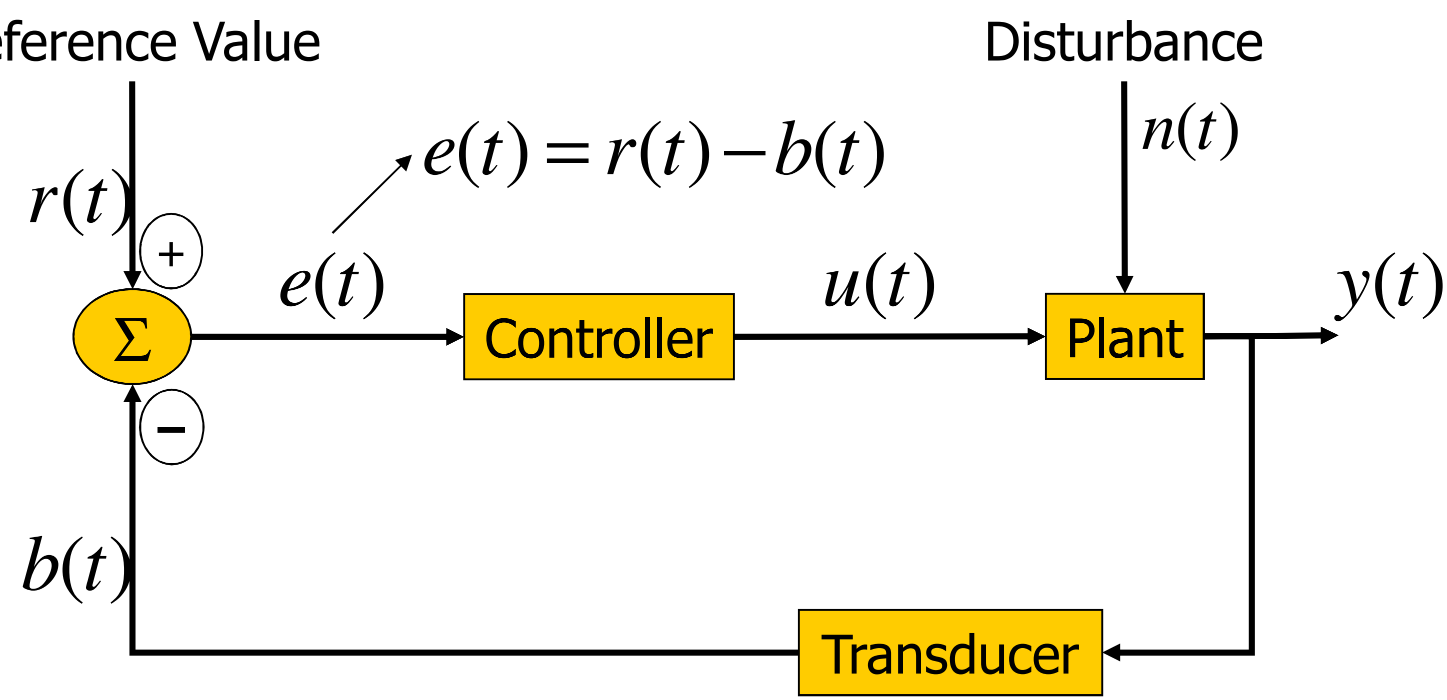
Examples of CT in CS

- Network flow controllers (TCP/IP – RED)
  - C. Hollot et al. (U.Mass)
- Lotus Notes admission control
  - S. Parekh et al. (IBM)
- QoS in Caching
  - C. Lu et al. (U.Va)
- Apache QoS differentiation
  - C. Lu et al. (U.Va)

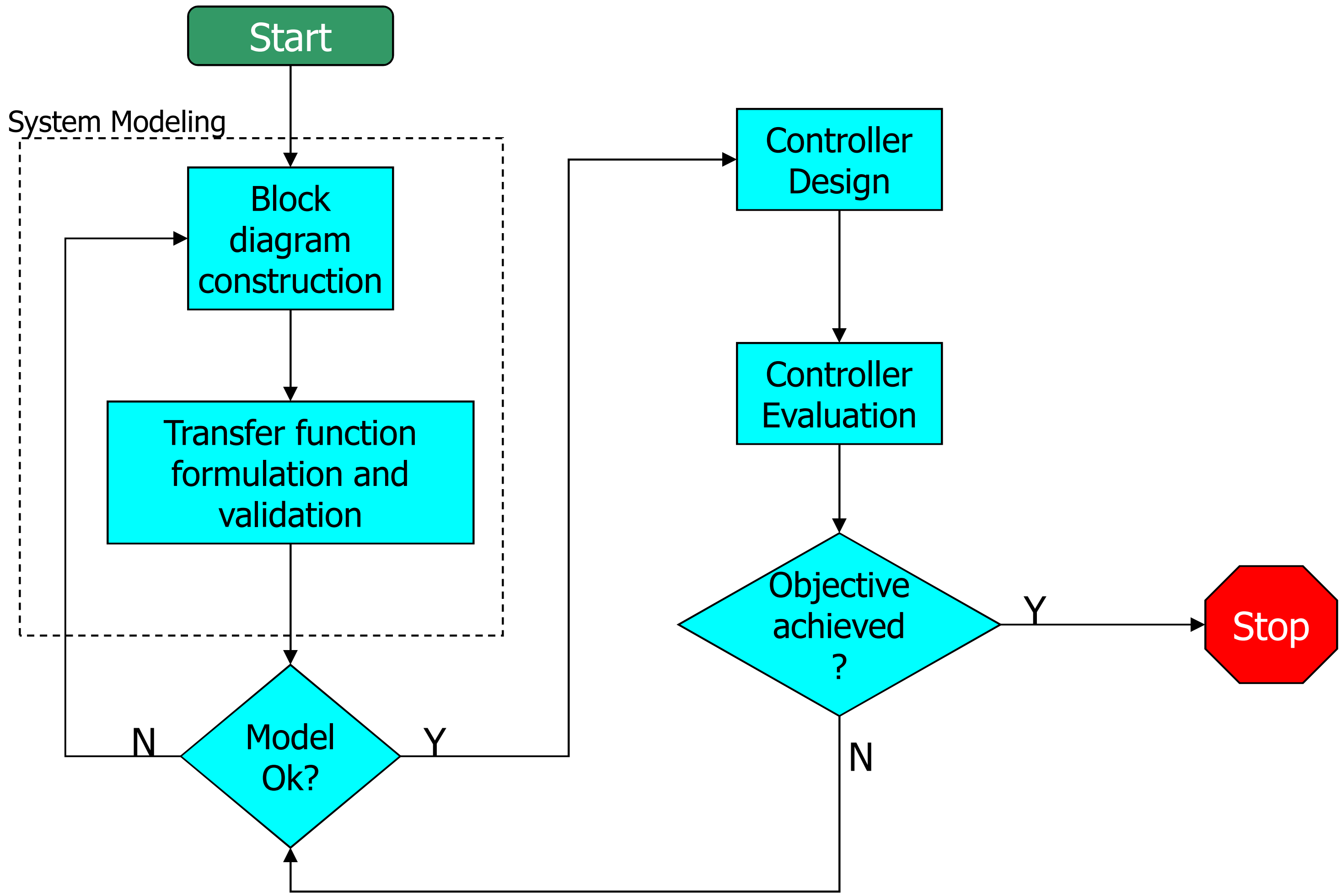
Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems

Feedback Control System



Controller Design Methodology



Control System Goals

- *Regulation*
  - thermostat, target service levels
- Tracking
  - robot movement, adjust TCP window to network bandwidth
- Optimization
  - best mix of chemicals, minimize response times

Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems

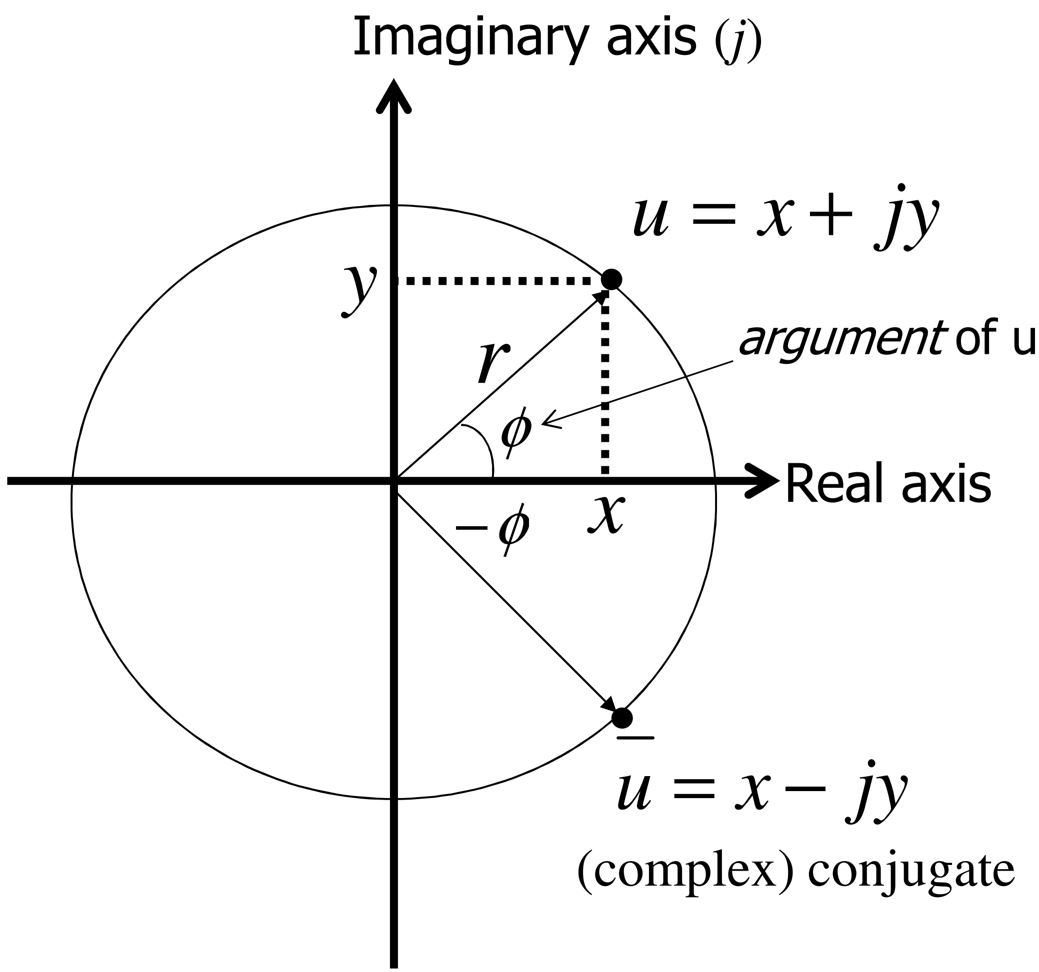
System Models

- **Linear** vs. non-linear
  - *Principle of superposition*: for linear systems, the net response at a given place and time caused by two or more stimuli is the sum of the responses which would have been caused by each stimulus individually.
- **Deterministic** vs. Stochastic
- **Time-invariant** vs. Time-varying
  - Are coefficients functions of time?
- **Continuous-time** vs. Discrete-time
  - $t \in R$ vs. $k \in Z$

Approaches to System Modeling

- First Principles
  - Based on known laws
    - Physics, Queueing theory
  - Difficult to do for complex systems
- Experimental (System ID)
  - Statistical/data-driven models
  - Requires data
  - Is there a good "training set"?

The Complex Plane (review)



$$\angle u \equiv \phi = \tan^{-1}\frac{y}{x}$$

$$|u| \equiv r \equiv |\bar{u}| = \sqrt{x^2 + y^2}$$

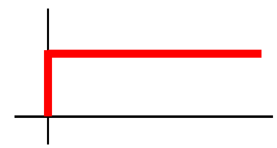
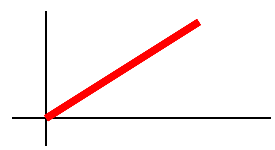
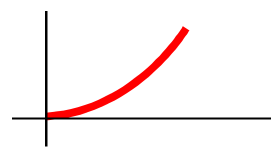
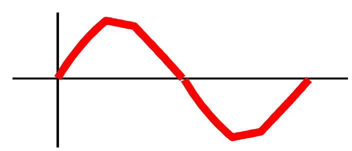
modulus or *absolute value* of u

Basic Tool For Continuous Time: Laplace Transform

$$\mathbf{L}[f(t)] = F(s) = \int_0^\infty f(t)e^{-st}dt$$

- Convert time-domain functions and operations into frequency-domain
  - $f(t) \rightarrow F(s)$ $(t \in$ ☑, $s \in$ ⊞$)$
  - Linear differential equations (LDE) $\rightarrow$ algebraic expression in Complex plane
- Graphical solution for key LDE characteristics
- Discrete systems use the analogous z-transform

Laplace Transforms of Common Functions

Name	$f(t)$		$F(s)$
Impulse	$f(t) = \begin{cases} 1 & t = 0 \\ 0 & t > 0 \end{cases}$		1
Step	$f(t) = 1$		$\frac{1}{s}$
Ramp	$f(t) = t$		$\frac{1}{s^2}$
Exponential	$f(t) = e^{at}$		$\frac{1}{s-a}$
Sine	$f(t) = \sin(\omega t)$		$\frac{1}{\omega^2 + s^2}$

Properties of Laplace Transform

Addition/Scaling	$L[af_1(t) \pm bf_2(t)] = aF_1(s) \pm bF_2(s)$
Differentiation	$L\left[\frac{d}{dt}f(t)\right] = sF(s) - f(0\pm)$
Integration	$L[\int f(t)dt] = \frac{F(s)}{s} + \frac{1}{s}[\int f(t)dt]_{t=0\pm}$
Convolution	$L\left[\int_0^t f_1(t-\tau)f_2(\tau)d\tau\right] = F_1(s)F_2(s)$

Initial-value theorem	$f(0+) = \lim_{s\to\infty} sF(s)$
Final-value theorem	$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$

Insights from Laplace Transforms

- What does the Laplace Transform say about f(t)?
  - Value of f(0)
    - Initial value theorem
  - Value of f(t) at steady state (if it converges)
    - Final-value theorem
  - Does f(t) converge to a finite value?
    - Poles of F(s): whether within unit circle
  - Does f(t) oscillate?
    - Poles of F(s) : whether the imaginary part is 0

Digital/Discrete Control

- More useful for computer systems
- Time is discrete
  - denoted k instead of t
- Main tool is z-transform

$$\mathbf{Z}[f(k)] = F(z) = \sum_{k=0}^{\infty} f(k) z^{-k}$$

  - $f(k) \rightarrow F(z)$, where z is complex
  - Analogous to Laplace transform for s-domain
- Root-locus analysis has similar flavor (discussed later)

z-Transforms of Common Functions

Name	$f(t)$	$F(s)$	$F(z)$
Impulse	$f(t)=\begin{cases}1 & t=0\\0 & t>0\end{cases}$	1	1
Step	$f(t)=1$	$\frac{1}{s}$	$\frac{z}{z-1}$
Ramp	$f(t)=t$	$\frac{1}{s^2}$	$\frac{z}{(z-1)^2}$
Exponential	$f(t)=e^{at}$	$\frac{1}{s-a}$	$\frac{z}{z-e^a}$
Sine	$f(t)=\sin(\omega t)$	$\frac{1}{\omega^2+s^2}$	$\frac{z\,\mathrm{Sin}\,\omega}{z^2-2(\mathrm{Cos}\,\omega)z+1}$

Properties of z-Transforms

Property	Time Domain	Z - Transform
Scaling	$y(k) = au(k)$	$Y(z) = aU(z)$
Addition	$y(k) = u(k) + v(k)$	$Y(z) = U(z) + V(z)$
Unit delay	$y(k) = u(k\text{-}1)$	$Y(z) = z^{-1}U(z)$
n - delay	$y(k) = u(k\text{-}n)$	$Y(z) = z^{-n}U(z)$
Unit shift	$y(k) = u(k+1)$	$Y(z) = zU(z)\text{-}zu(0)$
n - shift	$y(k) = u(k+n)$	$Y(z) = z^{n}U(z) - z^{n}u(0) - \ldots - zu(n-1)$

Z-Transform: Final Value Theorem

- Z-Transform provides a convenient way to determine the steady state value of a signal, if one exists

- Allows us to determine if steady state error exists (among other things):

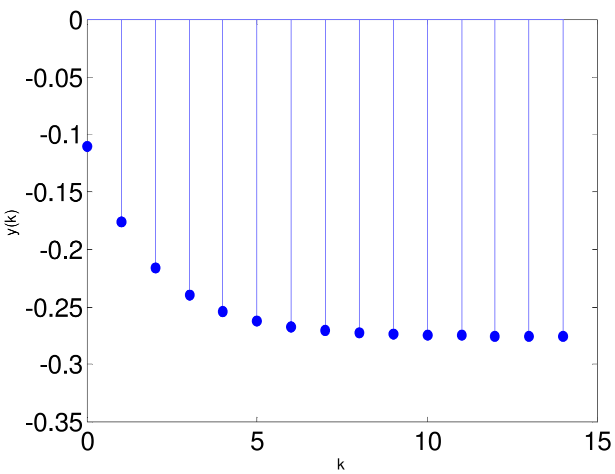
  Theorem: If all of the poles of $(1-z)Y(z)$ lie within the unit circle, then

  $$\lim_{k\to\infty} y(k) = \lim_{z\to 1}(z-1)Y(z)$$

Example

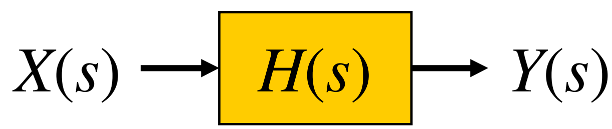
$$Y(z) = \frac{-0.11z}{z^2 - 1.6z + 0.6} = \frac{-0.11z}{(z-1)(z-0.6)}$$

$$(z-1)Y(z)\,|_{z=1} = \frac{-0.11z}{z-0.6}\,|_{z=1} = -0.275$$



Transfer Function

- Definition
  - H(s) = Y(s) / X(s)

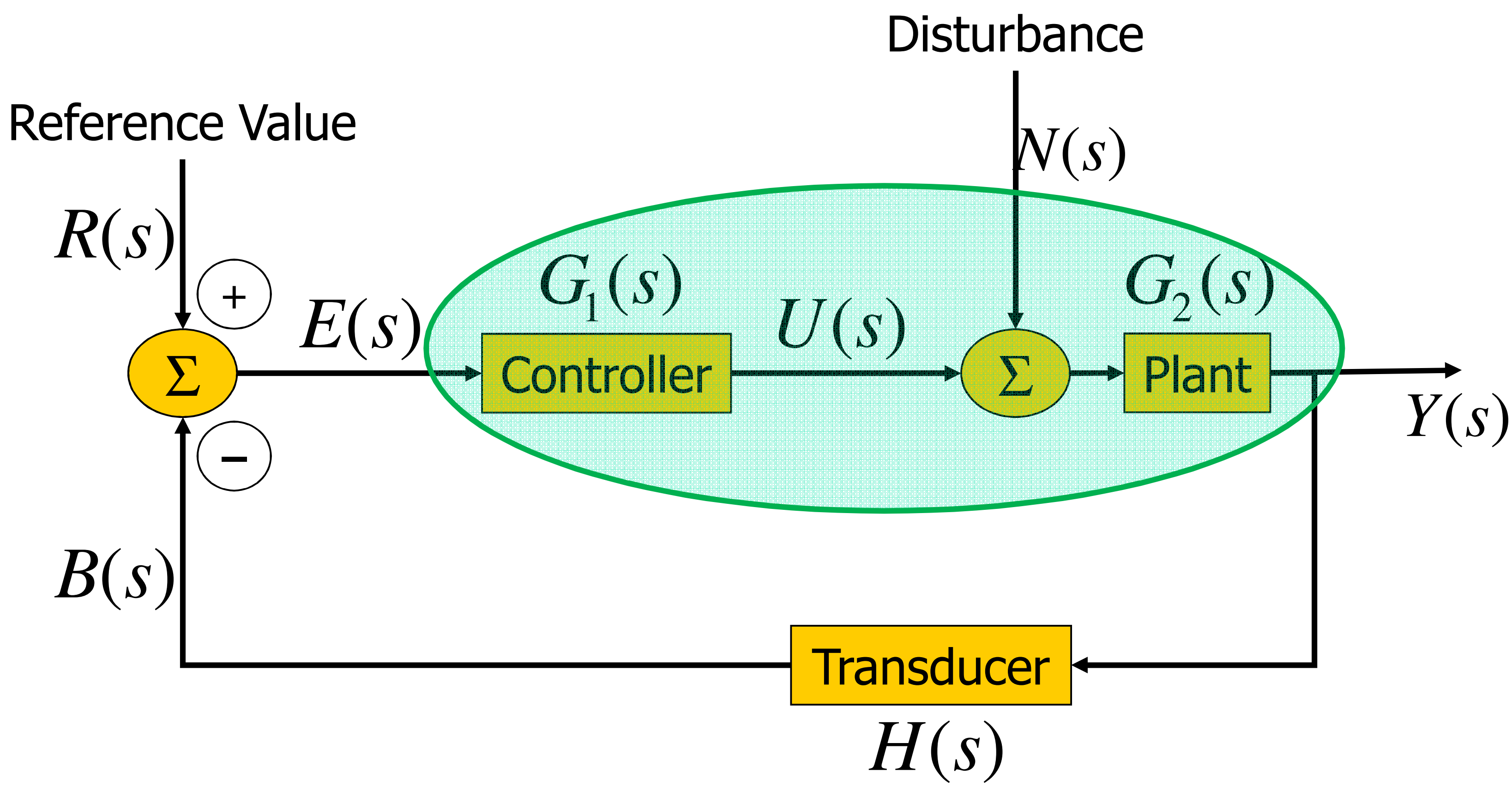


- Relates the output of a linear system (or component) to its input
- Describes how a linear system responds to an impulse (where X(s)=1)
- All linear operations allowed
  - Scaling, addition, multiplication

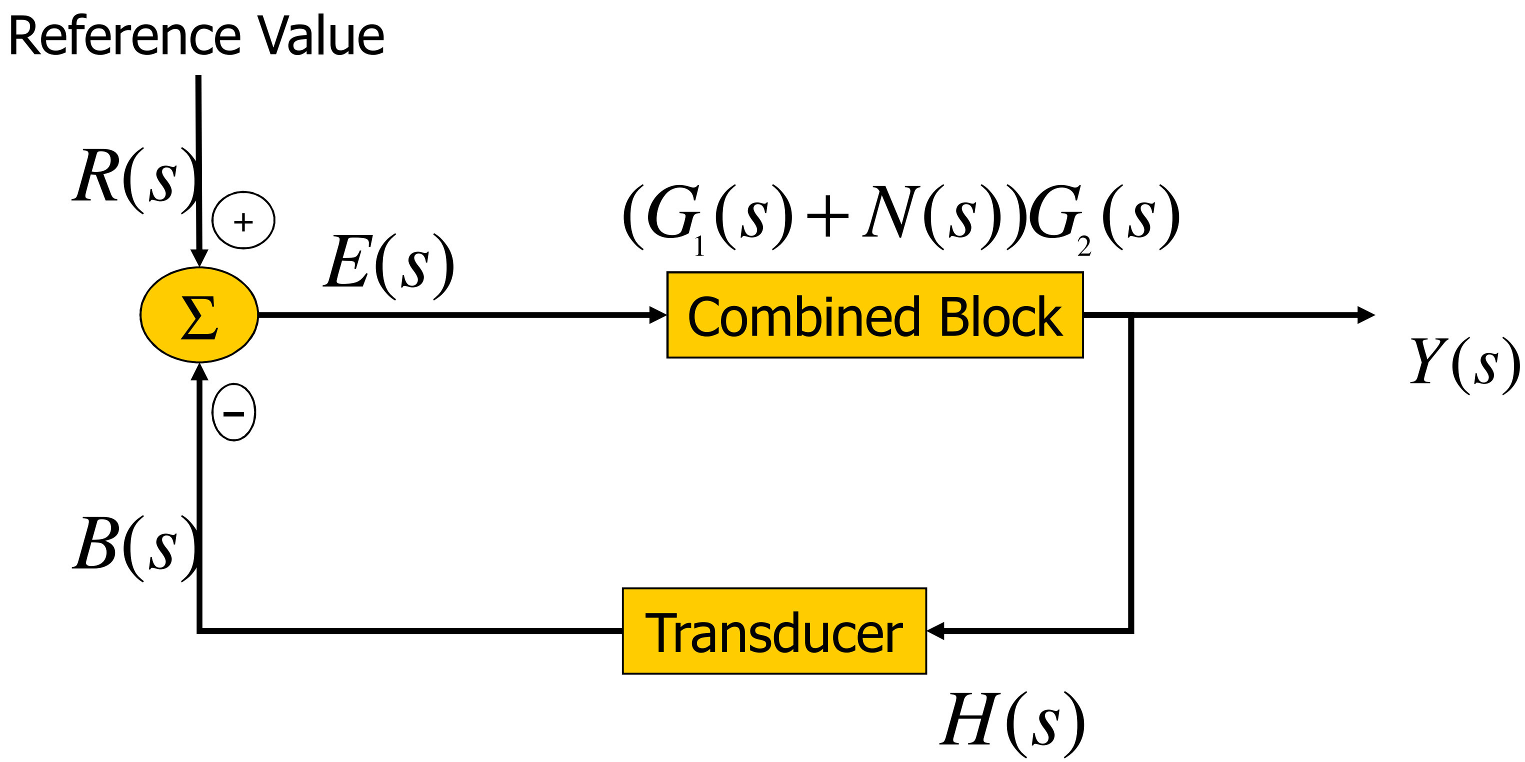
Block Diagrams

- Pictorially expresses flows and relationships between elements in system
- Blocks may recursively be systems
- Rules
  - Cascaded elements: convolution
  - Summation and difference elements
- Can simplify

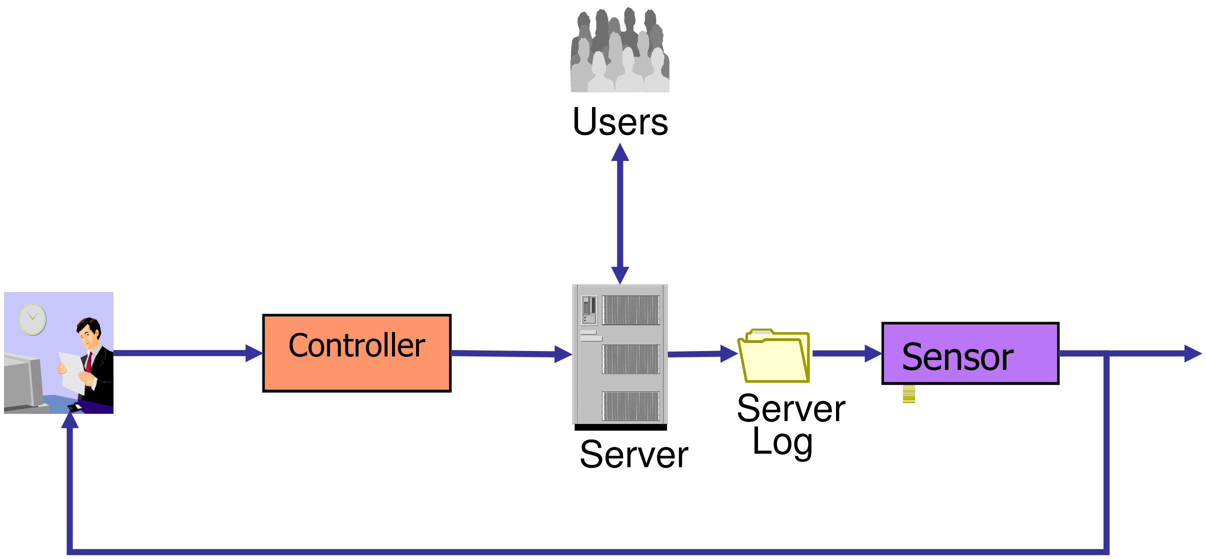
Block Diagram of System

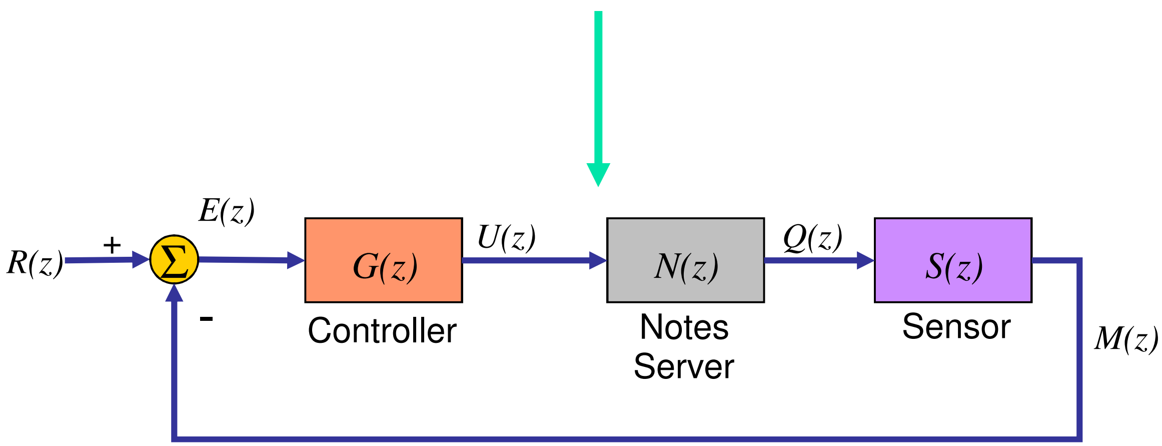


Combining Blocks

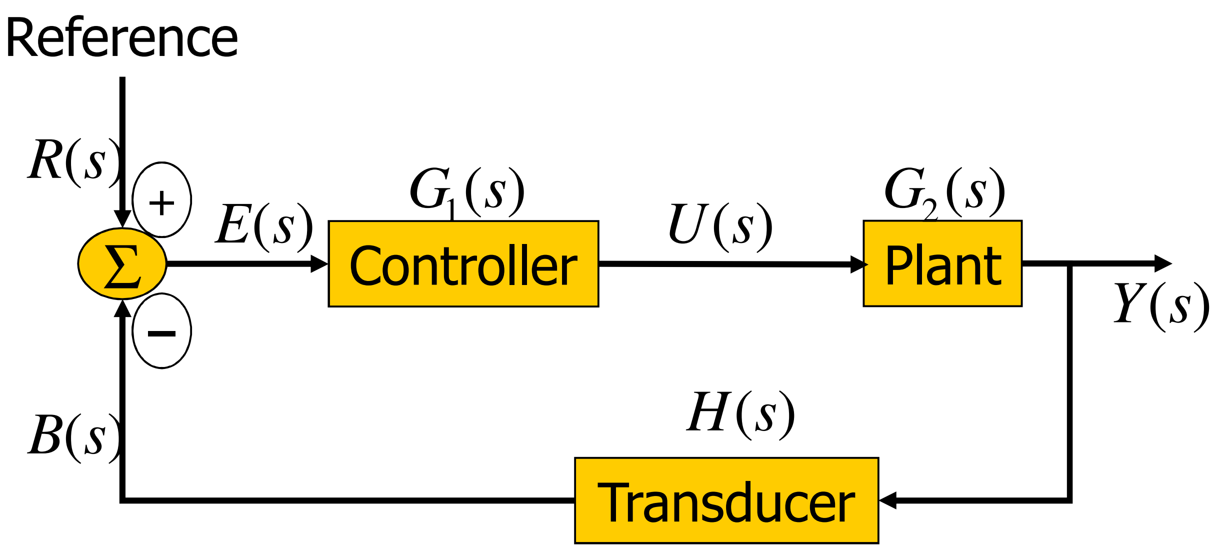


Block Diagram of Access Control





Key Transfer Functions



Feedforward: $\dfrac{Y(s)}{E(s)} = \dfrac{Y(s)}{U(s)}\dfrac{U(s)}{E(s)} = G_1(s)G_2(s)$

Feedback : $\dfrac{Y(s)}{R(s)} = \dfrac{G_1(s)G_2(s)}{1+G_1(s)G_2(s)H(s)}$

Open-Loop: $\dfrac{B(s)}{E(s)} = G_1(s)G_2(s)H(s)$

What can we do with Transfer Functions?

- Predict output for a given input
- Predict Stability of system
  - Unstable systems are highly undesirable
- Calculate steady-state gain
  - Compute operating ranges, state space reachability
- Use it as a basis for lowering system order
  - Lower-order systems are easier to work with
- Compose TFs to build/study complex systems (block diagram)
- Easily simulate the system behavior

Rational Laplace Transforms

$$F(s) = \frac{A(s)}{B(s)}$$

$$A(s) = a_n s^n + \ldots + a_1 s + a_0$$

$$B(s) = b_m s^m + \ldots + b_1 s + b_0$$

$$\text{Poles}: s^* \ni B(s^*) = 0 \text{ (So, } F(s^*) = \infty)$$

$$\text{Zeroes}: s^* \ni A(s^*) = 0 \text{ (So, } F(s^*) = 0)$$

Poles and zeroes are complex; system oscillate if poles have non - zero imaginary parts.

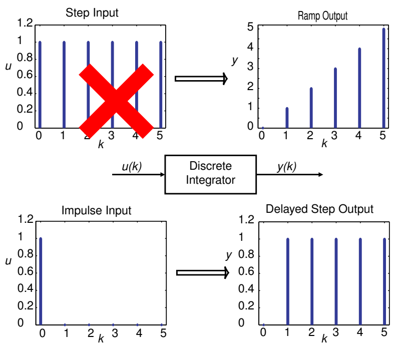
Order of system = # poles = m

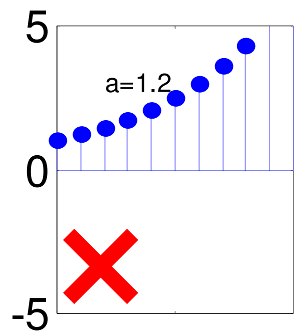
Bounded Signals

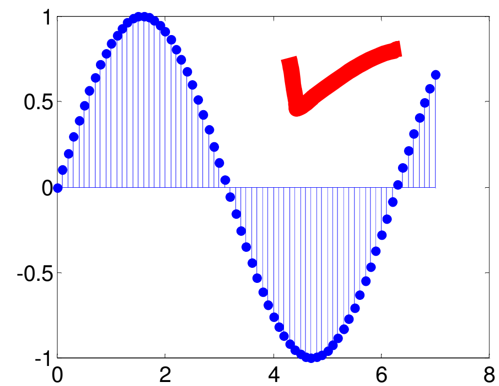
W

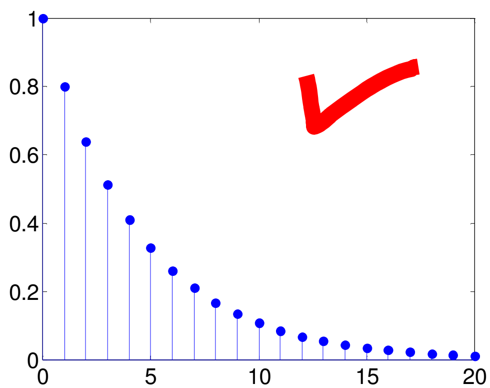
Definition:

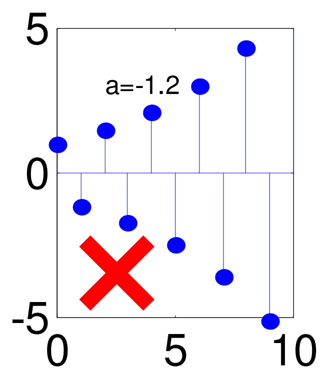
$\{u(k)\}$ is <u>bounded</u> if $\exists$ constant M such that $|u(k)| \leq M$ for all k.

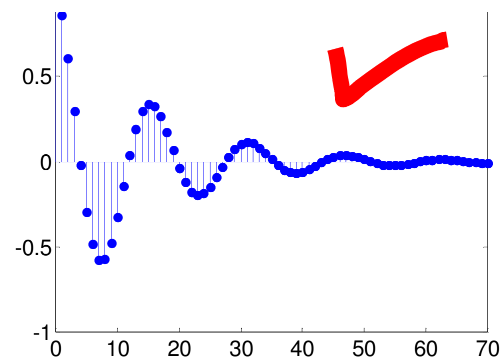












Lets Formalize this…

BIBO Stability

Definition:

A system is BIBO stable if for *any* bounded input {*u(k)*}, its output {*y(k)*} is bounded.

- NOTE: Bad reaction to unbounded inputs is OK?!
- Are these systems BIBO stable?

Unity	$y(k+1) = 1$
P Controller	$y(k+1) = K_P\, u(k)$
Integrator	$y(k+1) = y(k) + u(k)$
I Controller	$y(k+1) = y(k) + K_I\, u(k)$
M/M/1/K	$y(k+1) = 0.49y(k) + 0.033u(k)$
Mystery	$y(k+1) = -1.3y(k) + 2.3u(k)$

There must be a better way…

BIBO Stability Theorem

<u>Theorem</u>:
A system *G(z)* is BIBO stable iff all the poles of *G(z)* are inside the unit circle.

System	Time domain Eq	Transfer Function	Poles	
Unity	y(k+1) = 1	G(z) = 1	N/A	✓
P Controller	$y(k+1) = K_P u(k)$	$G(z) = K_P$	N/A	✓
Integrator	y(k+1) = y(k) + u(k)	G(z) = 1/(z-1)	z=1	✗
I Controller	$y(k+1) = y(k) + K_I u(k)$	$G(z) = K_I/(z-1)$	z=1	✗
M/M/1/K	y(k+1) = 0.49y(k) + 0.033u(k)	G(z) = 0.033/(z-0.49)	z = 0.49	✓
Mystery	y(k+1) = -1.3y(k) + 2.3u(k)	G(z) = 2.3/(z+1.3)	z = -1.3	✗

Steady-state Gain

- *G(z)* can be used to characterize steady-state reaction of system
  - Reaction to a constant input
- Given
  - BIBO stable system *G(z)*
  - Apply step input *U(z)*

NOTE

  - Output MUST converge

- Definition: Steady-state Gain (aka DC Gain)

$$\text{SSGain} = \frac{y_{\text{ss}}}{u_{\text{ss}}}$$

Steady-state Gain Theorem

- Theorem: Steady state gain of a system $G(z)$ is given by $G(1)$

- Proof
  - System response to a unit step (ie u_{ss}=1) is given by

$$Y(z) = G(z)\frac{z}{z-1}$$

  - Applying final-value theorem to get y_{ss}

$$\begin{aligned} y_{ss} &= \lim_{k\to\infty} y(k) = \lim_{z\to 1}(z-1)Y(z) \\ &= \lim_{z\to 1}(z-1)G(z)\frac{z}{z-1} \\ &= \lim_{z\to 1} zG(z) \\ &= G(1) \end{aligned}$$

$$y(k+1) = a_1 y(k) + \ldots + a_n y(k-n+1) + b_1 u(k) + \ldots + b_m u(k-m+1)$$

- For a general ARX transfer function, steady-state gain is

$$G(1) = \frac{b_1 + \cdots + b_m}{1 - a_1 - \cdots - a_n}$$

System Order

- System order = # poles
  - Same as *n* in ARX formula
  - Same as # initial conditions needed to seed the difference equation
- Why does it matter?
  - "Complexity" of system response proportional to system order
  - Higher ($\geq 2^{nd}$) order systems have complex poles
    - Implies oscillatory factors in system response
  - More difficult to design/analyze higher-order systems

Simulating Transfer Functions

- For complex transfer functions, cannot analyze
  - Solution: Simulate

$$G(z) = \frac{b_1 z^{n-1} + \cdots + b_m z^{n-m}}{z^n - a_1 z^{n-1} - \cdots - a_n}$$

is equivalent to:

$$y(k) = a_1 y(k-1) + \ldots + a_n y(k-n)$$
$$+ b_1 u(k-1) + \ldots + b_m u(k-m)$$

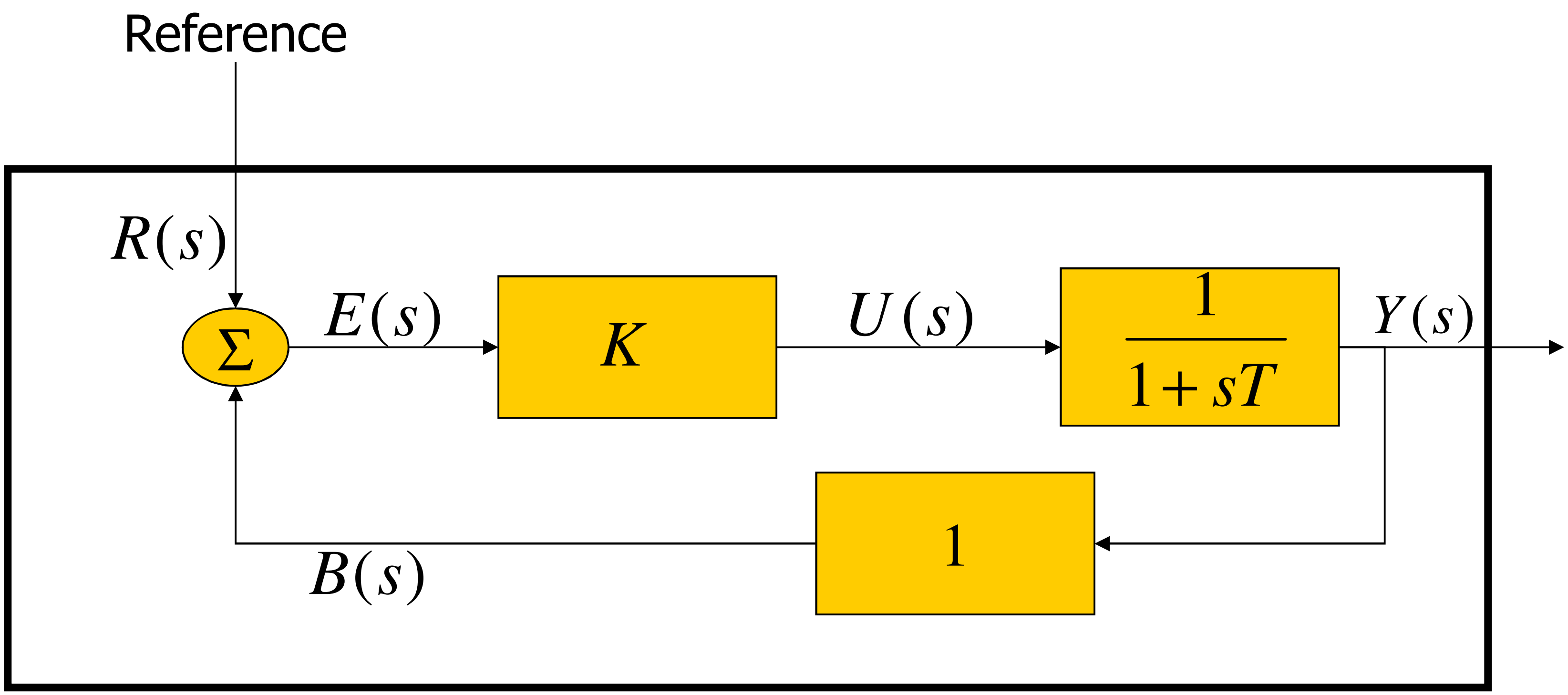
  - Inputs
    - Initial conditions (depends on system order)
    - *u(k)*

Transfer Functions: Summary

- Transfer Functions are also expressed as polynomials in z
- We can predict BIBO stability from the poles of a TF
- We can calculate steady-state gain from a TF
- We can build simplified TF from a high-order TF that has similar steady-state gain and settling time (see Chapter 6 of [1])
- It is usually easy to convert a difference eq to a TF
  - Can usually plug into the general ARX form
- One can simulate a TF by converting back to difference equation

First Order System

$$\frac{Y(s)}{R(s)} = \frac{K}{1+K+sT} \approx \frac{K}{1+sT}$$



First Order System

Impulse response	$\dfrac{K}{1+sT}$	Exponential
Step response	$\dfrac{K}{s} - \dfrac{K}{s+1/T}$	Step, exponential
Ramp response	$\dfrac{K}{s^2} - \dfrac{KT}{s} - \dfrac{KT}{s+1/T}$	Ramp, step, exponential

No oscillations (as seen by poles)

Second Order System

$$\text{Impulse response: } \frac{Y(s)}{R(s)} = \frac{K}{Js^2 + Bs + K} = \frac{\omega_N^2}{s^2 + 2\xi\omega_N s + \omega_N^2}$$

Oscillates if poles have non-zero imaginary part (ie, $B^2 - 4JK < 0$)

Damping ratio: $\zeta = \frac{B}{B_c}$ where $B_c = 2\sqrt{JK}$

Undamped natural frequency: $\omega_N = \sqrt{\frac{K}{J}}$

Second Order System: Parameters

Interpretation of damping ratio:

$\zeta = 0$: undamped oscillation (Re = 0, Im ≠ 0)

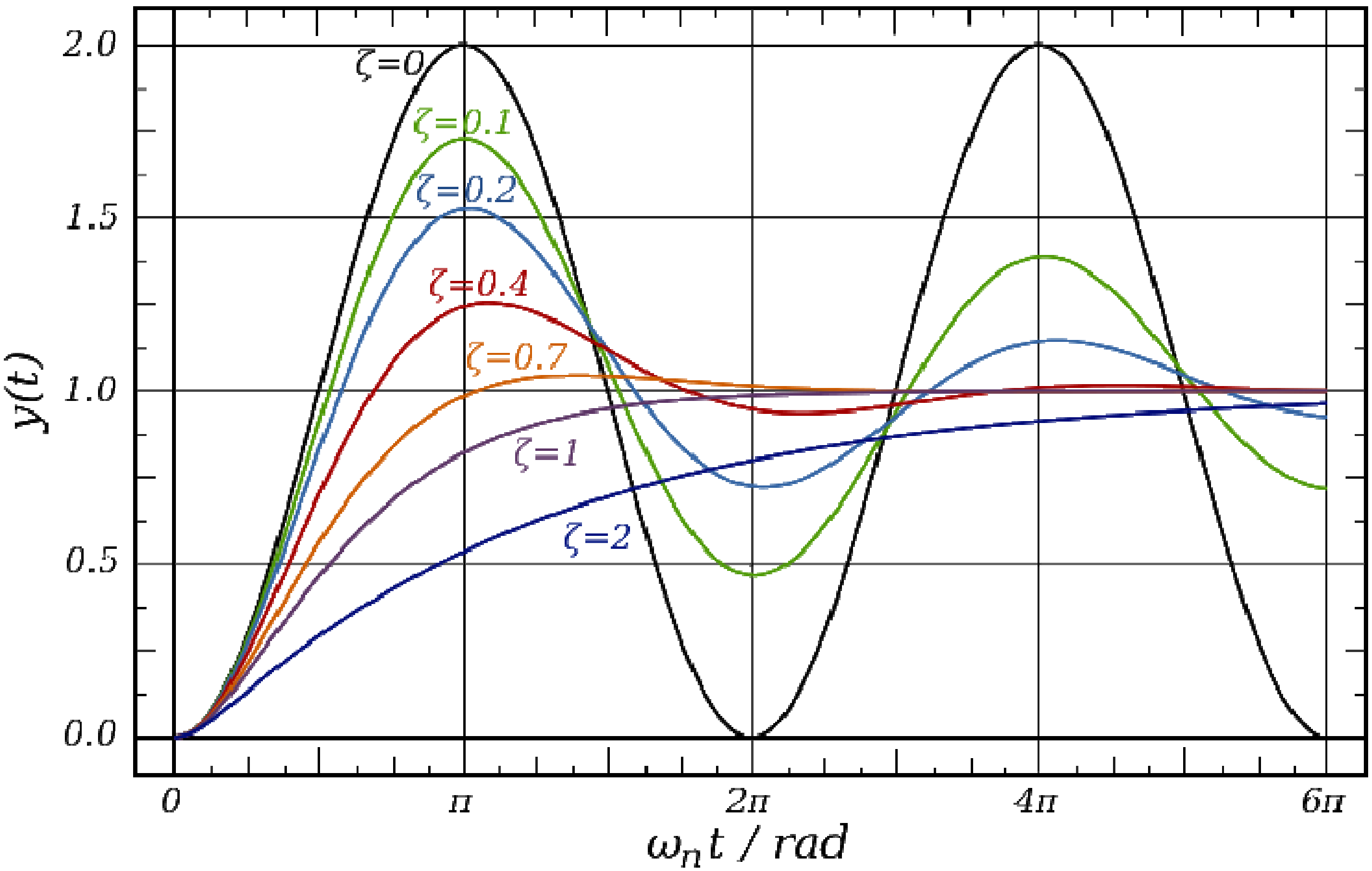
$0 < \zeta < 1$: underdamped (Re ≠ 0 ≠ Im)

$1 \leq \zeta$: overdamped (Re ≠ 0, Im = 0)

Interpretation of undamped natural frequency:

ω_N gives the frequency of the oscillation

(or frequency of free vibration for mechanical systems)



2.0
1.5
1.0
0.5
0.0
y(t)
ζ=0
ζ=0.1
ζ=0.2
ζ=0.4
ζ=0.7
ζ=1
ζ=2
0
π
2π
4π
6π
ωnt / rad

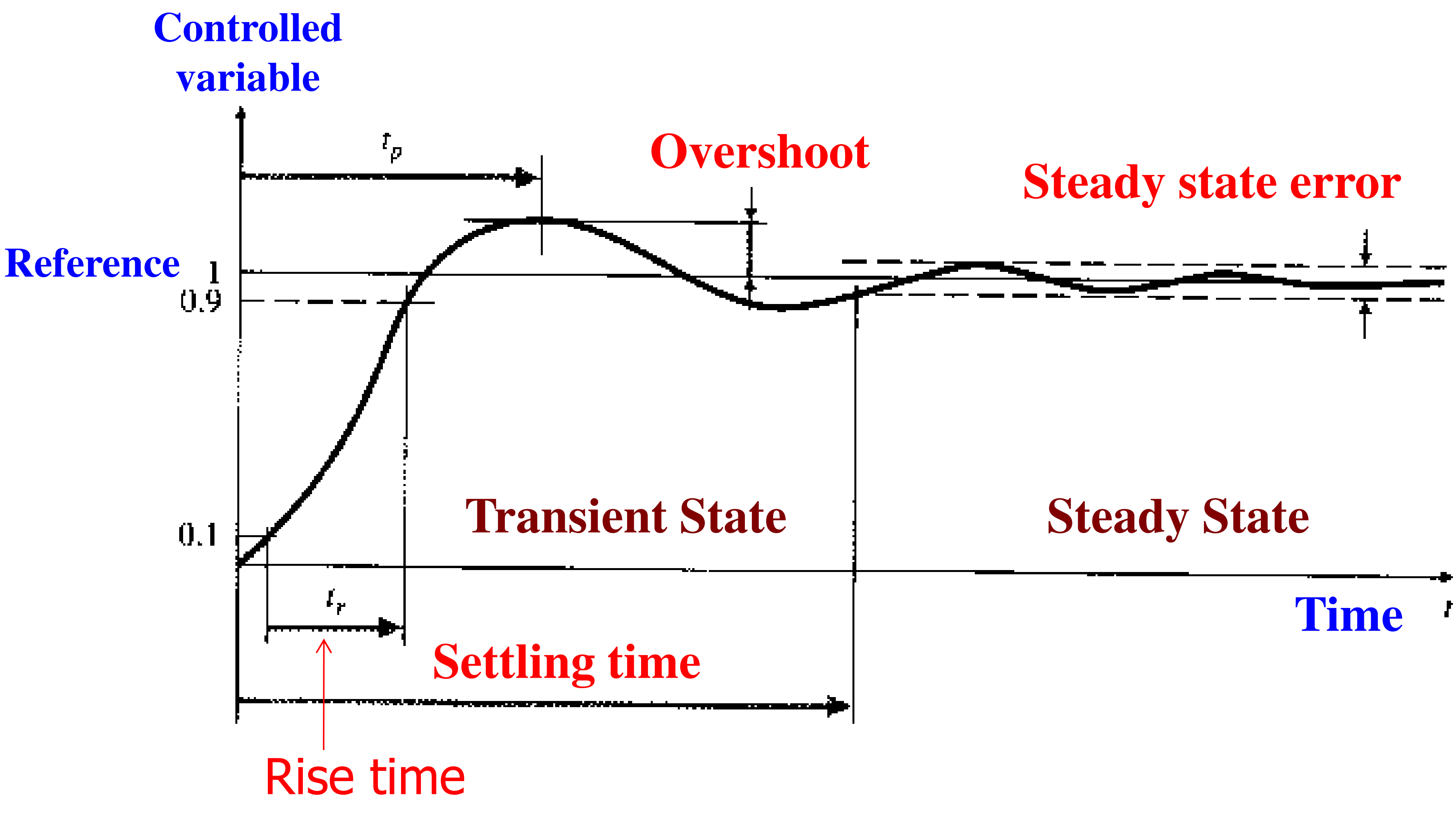
Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems
- References

Objectives of Control for Computing Systems: SASO

- Stability
- Accuracy
- *Settling time*
- *Overshoot*

Characteristics of Transient Response

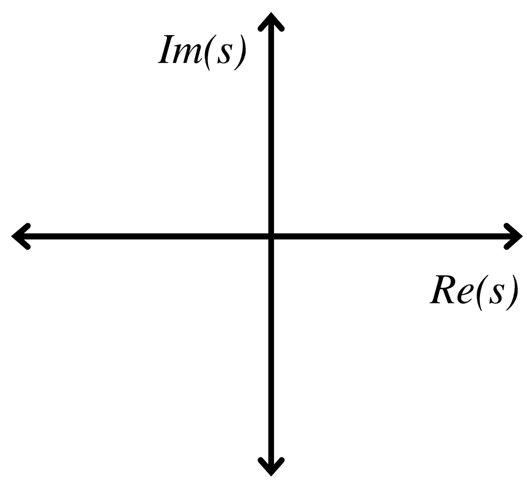


Transient Response

- Estimates the shape of the curve based on the foregoing points on the x and y axis
- Typically applied to the following inputs
  - Impulse
  - Step
  - Ramp
  - Quadratic (Parabola)

Effect of pole locations

Oscillations
(higher-freq)



Faster Decay
(e^{-at})

Faster Blowup
(e^{at})

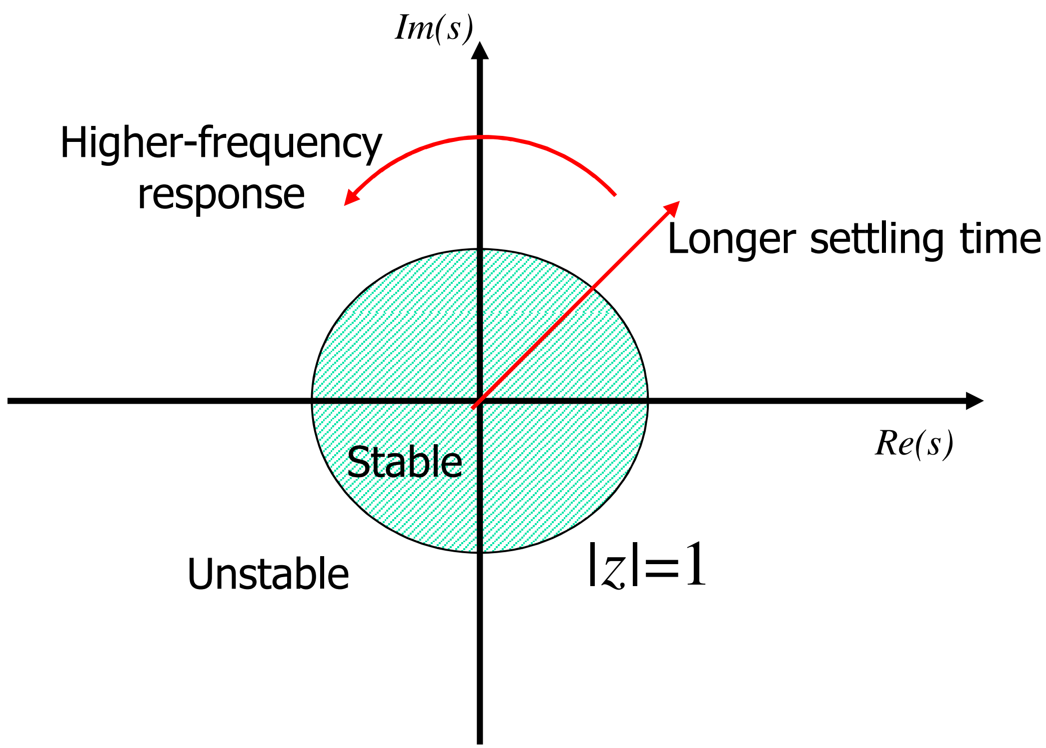
Root-locus Analysis

- Based on characteristic eqn of closed-loop transfer function
- Plot location of **roots** of this eqn
  - Same as **poles** of closed-loop transfer function
  - Parameter (gain) varied from 0 to ∞
- Multiple parameters are ok
  - Vary one-by-one
  - Plot a root "contour" (usually for 2-3 params)
- Quickly get approximate results
  - Range of parameters that gives desired response

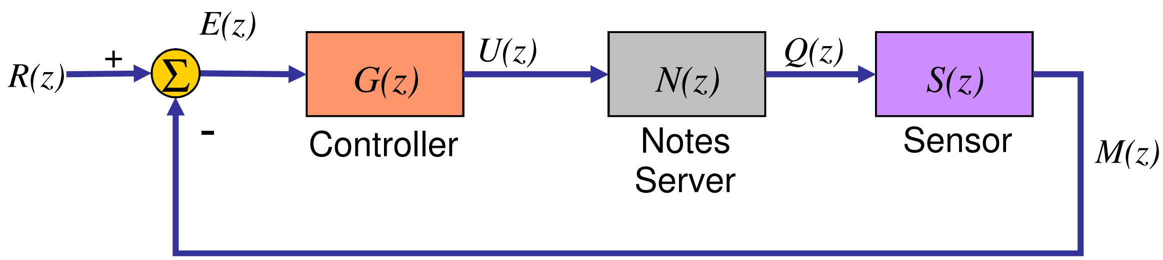
Root Locus analysis of Discrete Systems

- Stability boundary: $|z|=1$ (Unit circle)
- Settling time: distance from Origin
- Oscillation frequency/speed: location relative to Im. axis
  - Right half => slower
  - Left half => faster

Effect of discrete poles



System ID for Admission Control



ARMA Models

$$q(t) = a_1 q(t-1) + b_0 u(t)$$

$$m(t) = c_1 m(t-1) + d_0 q(t) + d_1 q(t-1)$$

Control Law

$$u(t) = u(t-1) + K_i e(t)$$

Transfer Functions

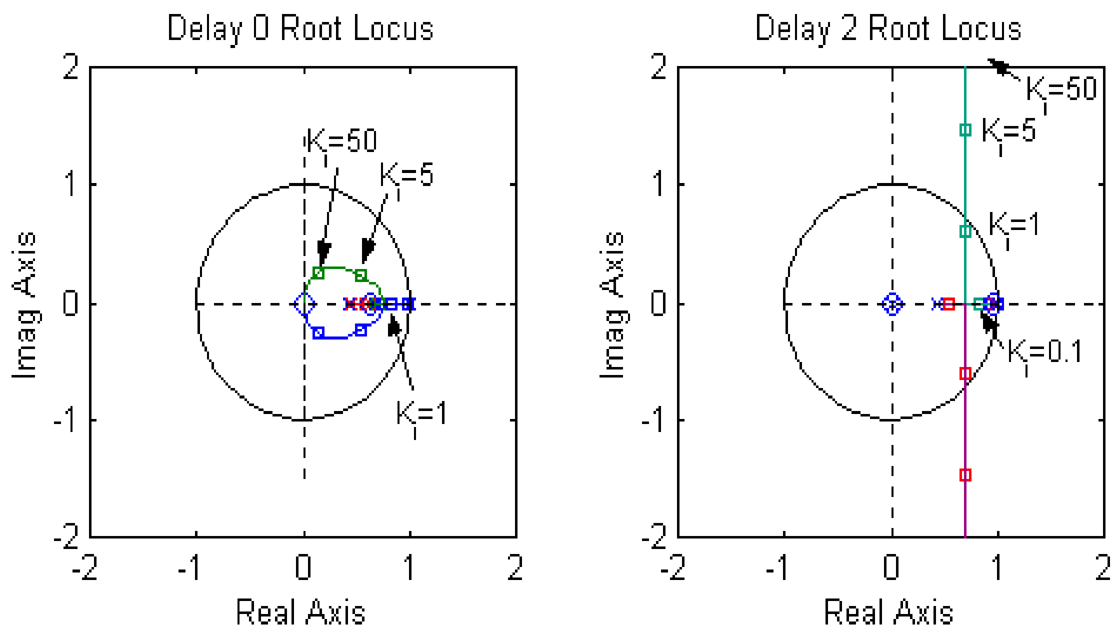
$$N(z) = \frac{b_0 z}{z - a_1}$$

$$S(z) = \frac{d_0 z + d_1}{z - c_1}$$

$$G(z) = \frac{K_i z}{z - 1} \frac{1}{z^\delta}$$

Open-Loop: $$N(z)S(z)G(z) = \frac{b_0 z}{z - a_1} \frac{d_0 z + d_1}{z - c_1} \frac{K_i z}{z - 1} \frac{1}{z^\delta}$$

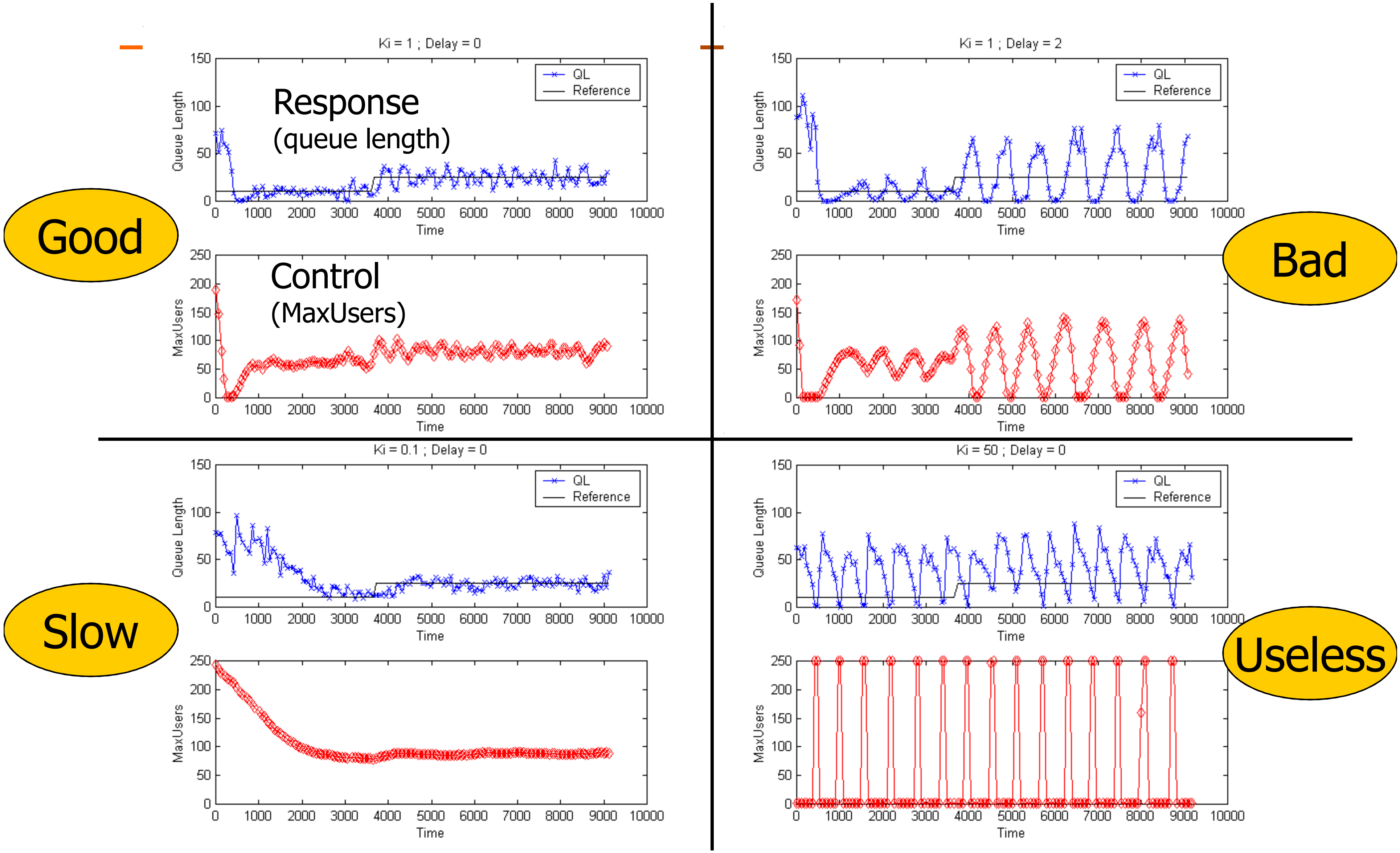
Root Locus Analysis of Admission Control



Predictions:

- K_i small => No controller-induced oscillations
- K_i large => Some oscillations
- K_i v. large => unstable system (delay=2)
- Usable range of K_i for delay=2 is small

Experimental Results



Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems
- References

Basic Control Actions: u(t)

Proportional control: $u(t) = K_p e(t)$ $\quad \dfrac{U(s)}{E(s)} = K_p$

Integral control: $u(t) = K_i \int_0^t e(t)dt$ $\quad \dfrac{U(s)}{E(s)} = \dfrac{K_i}{s}$

Differential control: $u(t) = K_d \dfrac{d}{dt} e(t)$ $\quad \dfrac{U(s)}{E(s)} = K_d s$

Effect of Control Actions

- Proportional Action
  - Adjustable gain (amplifier)
  - May have non-zero steady-state error
- Integral Action
  - Eliminates bias (steady-state error)
  - Can cause oscillations
- Derivative Action (“rate control”)
  - Effective in transient periods
  - Provides faster response (higher sensitivity)
  - Never used alone

Basic Controllers

- Proportional control is often used by itself
- Integral and differential control are typically used in combination with at least proportional control
  - eg, Proportional Integral (PI) controller:

$$G(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_I}{s} = K_p\left(1 + \frac{1}{T_i s}\right)$$

Summary of Basic Control

- Proportional control
  - Multiply e(t) by a constant
- PI control
  - Multiply e(t) and its integral by separate constants
  - Avoids bias for step
- PD control
  - Multiply e(t) and its derivative by separate constants
  - Adjust more rapidly to changes
- PID control
  - Multiply e(t), its derivative and its integral by separate constants
  - Reduce bias and react quickly

Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems
- References

Advanced Control Topics

- Robust Control
  - Can the system tolerate noise?
- Adaptive Control
  - Controller changes over time (adapts)
- MIMO Control
  - Multiple inputs and/or outputs
- Stochastic Control
  - Controller minimizes variance
- Optimal Control
  - Controller minimizes a cost function (e.g., function of error)
- Nonlinear systems
  - Challenging to derive analytic results

Outline

- Examples and Motivation
- Control Theory Vocabulary and Methodology
- Modeling Dynamic Systems
- Transient Behavior Analysis
- Standard Control Actions
- Advanced Topics
- Issues for Computer Systems

Issues for Computer Science

- Most systems are non-linear
  - But linear approximations may do
    - E.g., fluid approximations
- First-principles modeling is difficult
  - Use empirical techniques
- Control objectives are different
  - Optimization rather than regulation
- Multiple Controls
  - State-space techniques
  - Advanced non-linear techniques (eg, NNs)

References

- Control Theory Basics
  - [1] Joseph L. Hellerstein, Yixin Diao, Sujay Parekh, Dawn M. Tilbury, *Feedback Control of Computing Systems*, Wiley-IEEE Press, 2004. (ISBN: 978-0-471-26637-2)
  - [2] G. Franklin, J. Powell and A. Emami-Naeini. "*Feedback Control of Dynamic Systems, 3rd ed*". Addison-Wesley, 1994.
  - [3] K. Ogata. "*Modern Control Engineering, 3rd ed*". Prentice-Hall, 1997.
  - [4] K. Ogata. "*Discrete-Time Control Systems, 2nd ed*". Prentice-Hall, 1995.

- Applications in Computer Science
  - C. Hollot et al. "*Control-Theoretic Analysis of RED*". IEEE Infocom 2001
  - C. Lu, et al. "*A Feedback Control Approach for Guaranteeing Relative Delays in Web Servers*". IEEE Real-Time Technology and Applications Symposium, June 2001.
  - S. Parekh et al. "*Using Control Theory to Achieve Service-level Objectives in Performance Management*". Int'l Symposium on Integrated Network Management, May 2001
  - Y. Lu et al. "*Differentiated Caching Services: A Control-Theoretic Approach*". Int'l Conf on Distributed Computing Systems, Apr 2001
  - S. Mascolo. "*Classical Control Theory for Congestion Avoidance in High-speed Internet*". Proc. 38th Conference on Decision & Control, Dec 1999
  - S. Keshav. "*A Control-Theoretic Approach to Flow Control*". Proc. ACM SIGCOMM, Sep 1991
  - D. Chiu and R. Jain. "*Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks*". Computer Networks and ISDN Systems, 17(1), Jun 1989