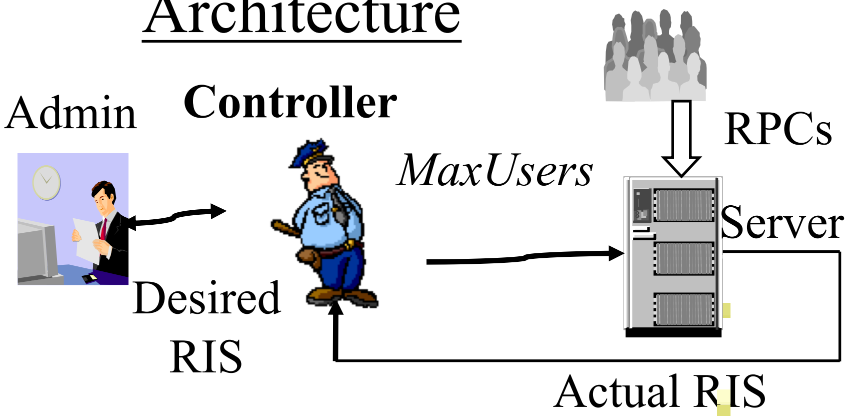
Using Control Theory in Performance Management

Joseph L. Hellerstein
eScience Institute
Computer Science & Engineering

November 14, 2018

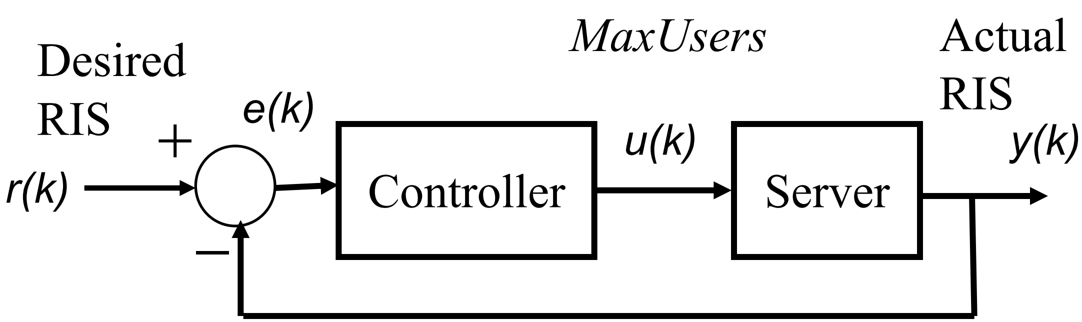
Example: Control of the IBM Lotus Domino Server

Architecture



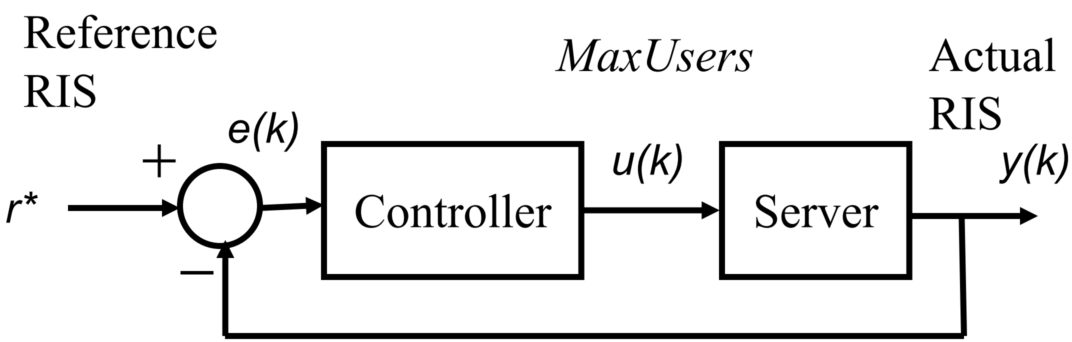
RIS = RPCs in System
(users in active state)

Block Diagram



Lab 1

Block Diagram



ARX* Models

Control error: $e(k)=r^*-y(k)$

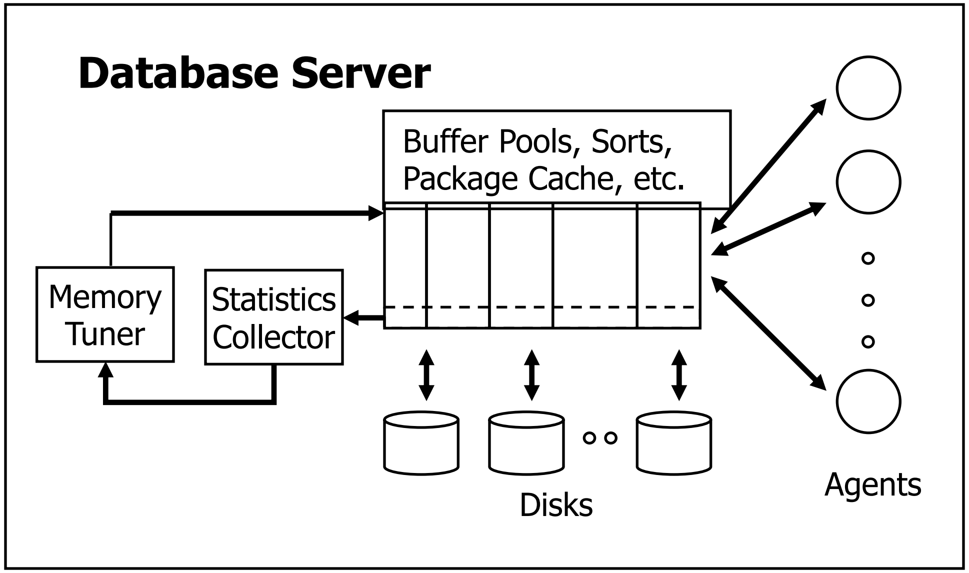
Normalized *MaxUsers:* $u(k)=K_P^*e(k)$

System model: $y(k)=(0.43)y(k-1)$
$+(0.47)u(k-1)$

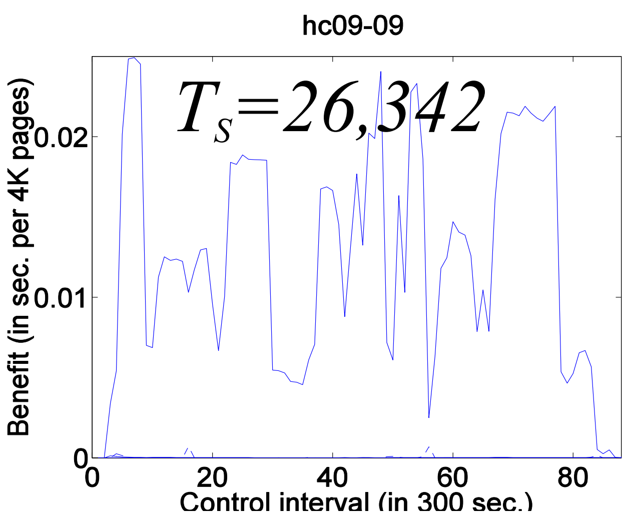
- Spreadsheet file CTShortClass, tab 1 (P Control).
  - Proportional controller: *MaxUsers(k+1)* = $K_P^*(r^*-y(k))$
  - What is the effect of *K* on
    - Accuracy: (want $r^*=y(k)$=200)
    - Stability
    - Convergence rate (settling time)
    - Overshoot

*ARX is autoregressive with an external input

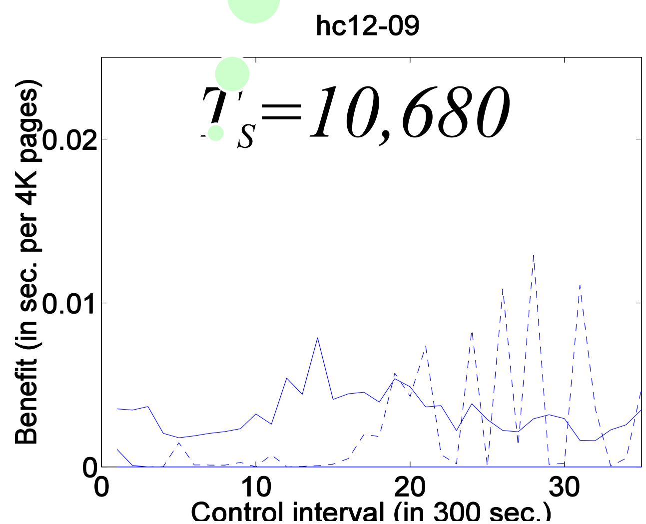
Application of CT to a DBMS



59% Reduction in Total RT



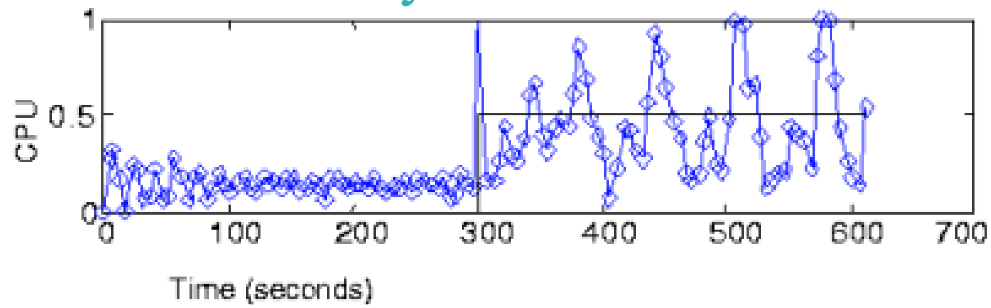
Without Controller

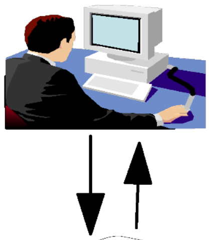


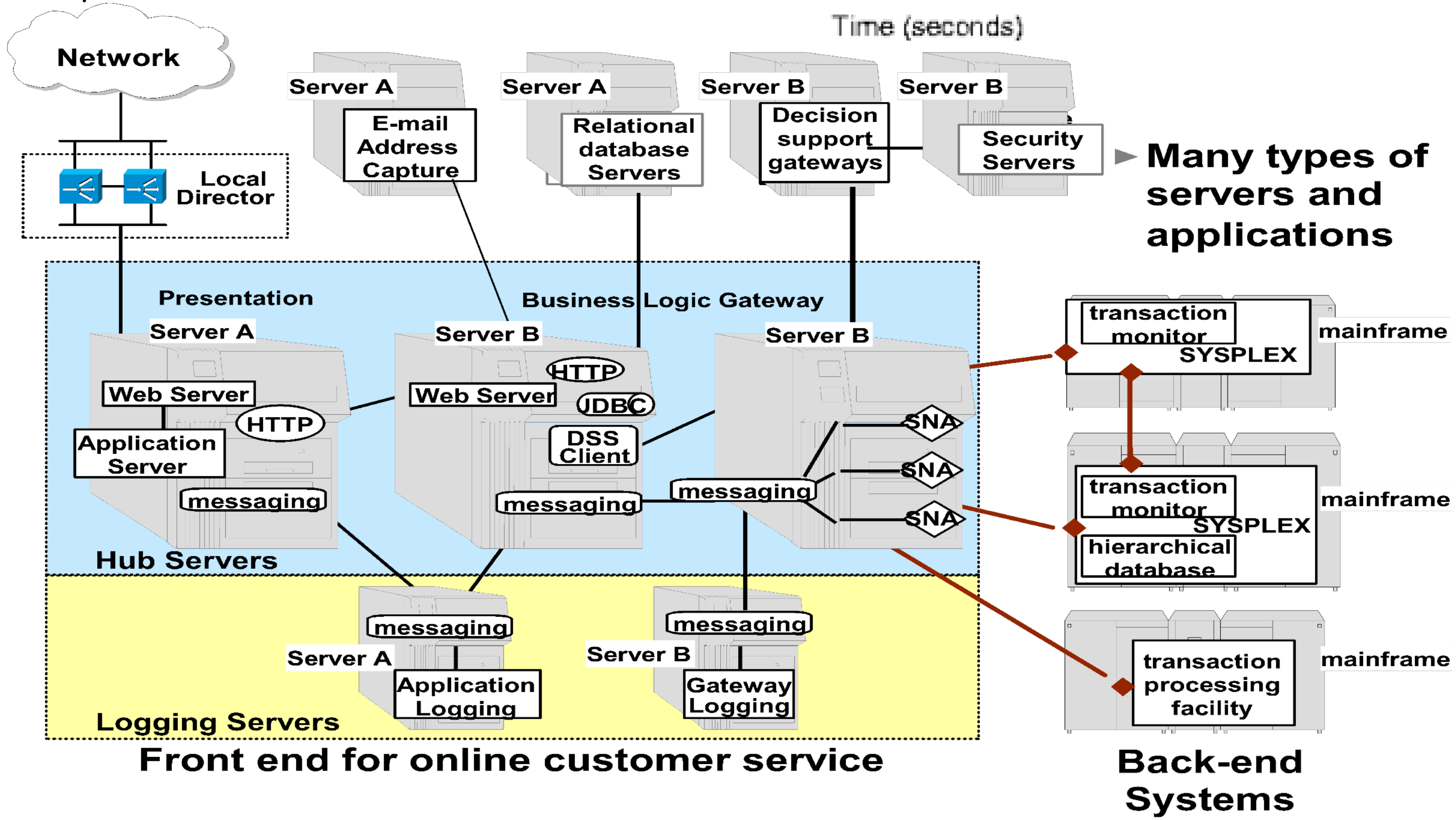
With Controller

Why Control Theory?

Unstable System

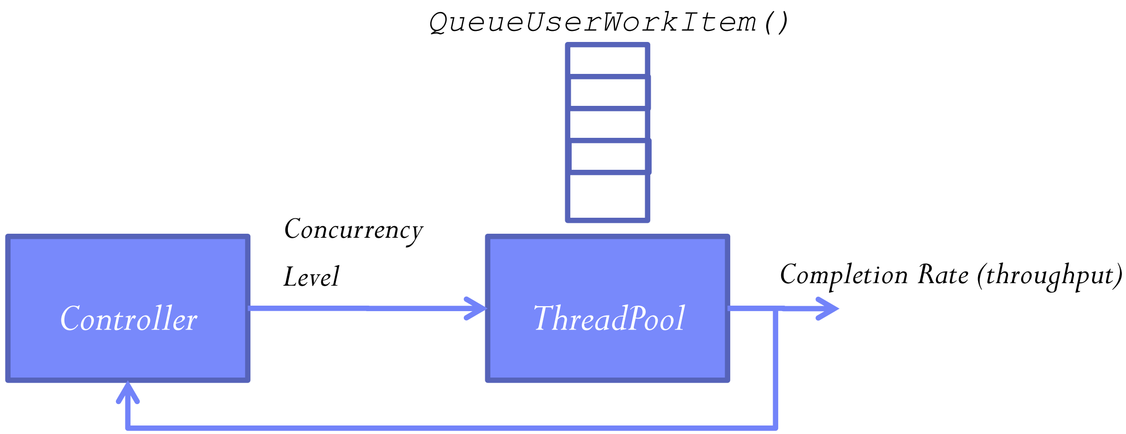






- Stabililty
- Accuracy
- Settling time
- Overshoot

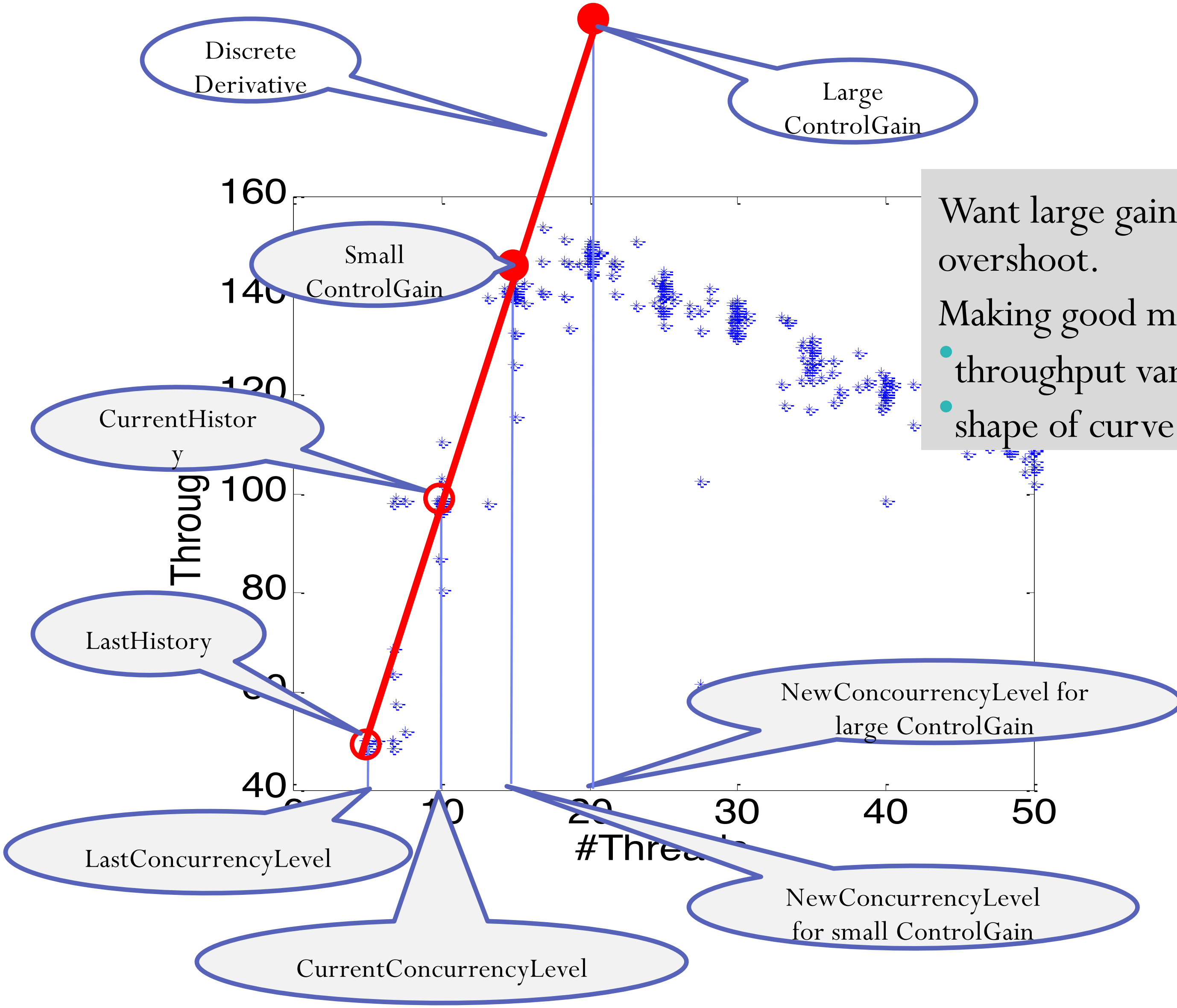
Optimizing Throughput in the Microsoft .NET ThreadPool



- Current ThreadPool
  - Objective: Maximize CPU utilization and thread completion rates
  - Inputs: ThreadPool events, CPU utilization
  - Techniques
    - Thresholds on inter-dequeue times, rate of increasing workers, change in rate of increasing workers
    - States: Starvation, RateIncrease, RateDecrease, LowCPU, PauseInjection
- New approach
  - Objective: Maximize thread completion rate
  - Inputs: ThreadPool events
  - Technique: Hill climbing

SIGMETRICS 2008: Introduction to Control Theory. Abdelzaher, Diao, Hellerstein, Yu, and Zhu.

Hill Climbing Controller



Want large gain so move quickly, but not overshoot.

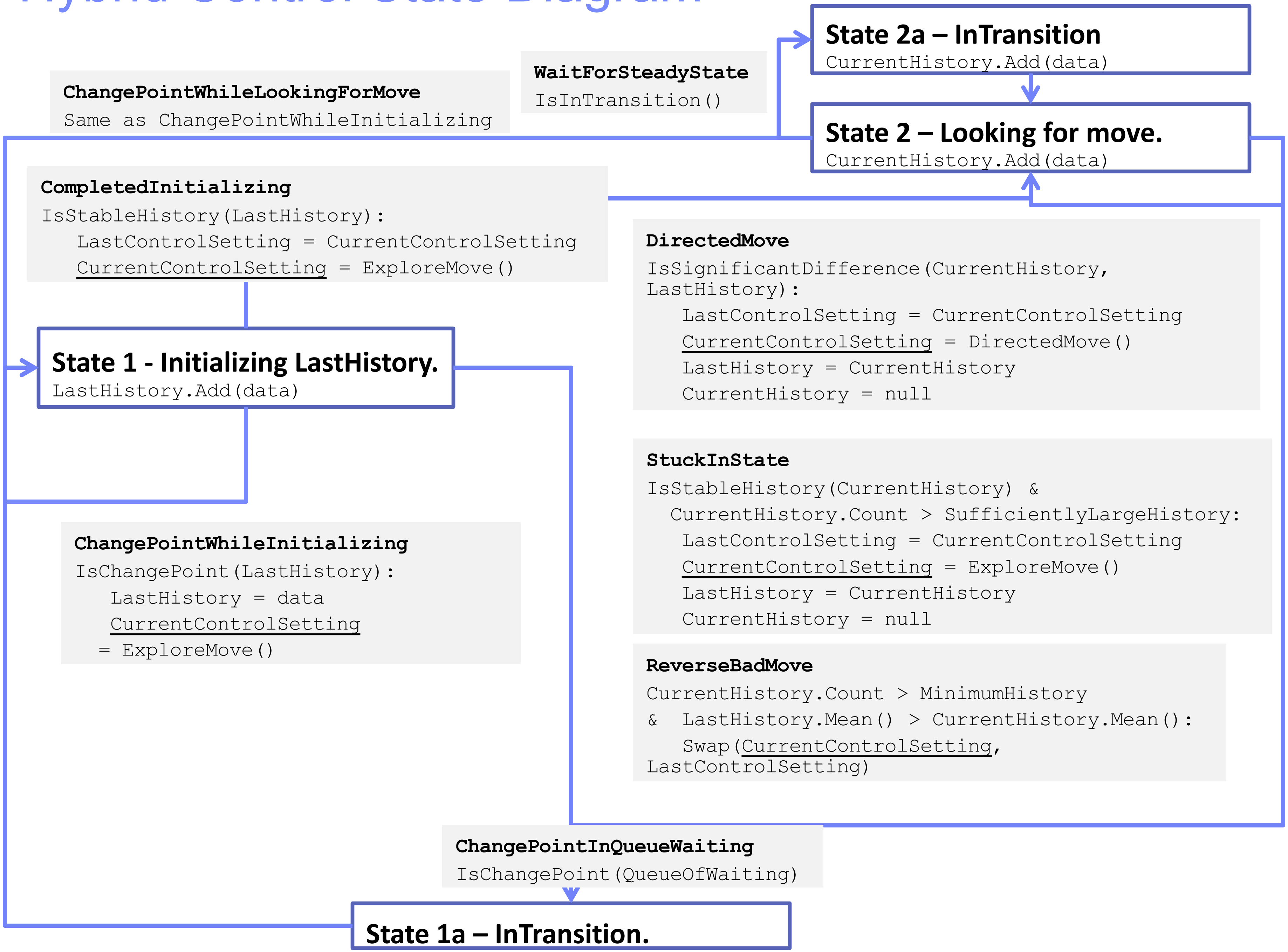
Making good moves depends on

- throughput variance
- shape of curve

○ = History mean

(50 work items:100ms with 10%CPU, 90% wait. 2.2GHz dual core X86.)

Hybrid Control State Diagram



Goals

- Control theory "boot camp" for software designers with no background in control theory or linear systems theory
  - Be able to formulate and solve basic control problems
  - Know references so can solve more complex problems
- Covers about 50% of the material presented in a semester class at Columbia University
- Excludes
  - Modeling: System identification, multiple input multiple output (MIMO) models, non-linear models
  - Control: control design, MIMO control, empirical tuning, adaptive control, stochastic control
  - Tools: MATLAB
  - Running examples: Apache HTTP server, *M/M/1/K* queueing system, streaming, load balancing
- Reference
  - "Feedback Control of Computing Systems", Hellerstein, Diao, Parekh, Tilbury. Wiley, 2004

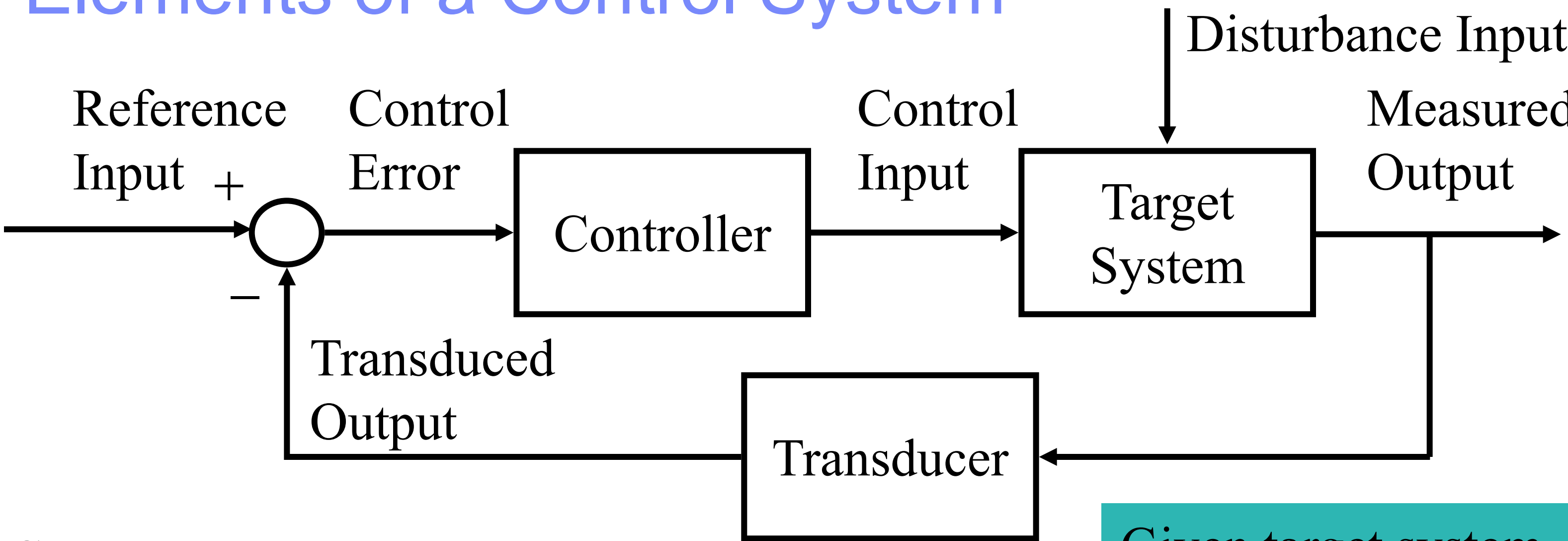
Agenda

- Introduction:
  - Control system architecture, goals, and metrics.
- Theory: Part 1
  - Signals, Z-Transforms
- Theory: Part 2
  - Transfer functions
  - Analyzing composed systems
  - Q&A / Buffer
- Control Analysis
  - Basic controllers, precompensation, filters
  - Structured as a design exercise
- Real world applications (Various publications)
  - DB2 Utilities throttling and self-tuning memory management

M1 - Introduction

Reference: “Feedback Control of Computer Systems”, Chapter 1.

Elements of a Control System



Given target system, transducer
Control theory finds controller
that adjusts control input
to achieve measured
output in the presence of
disturbances.

Components

- Target system: what is controlled
- Controller: exercises control
- Transducer: translates measured outputs

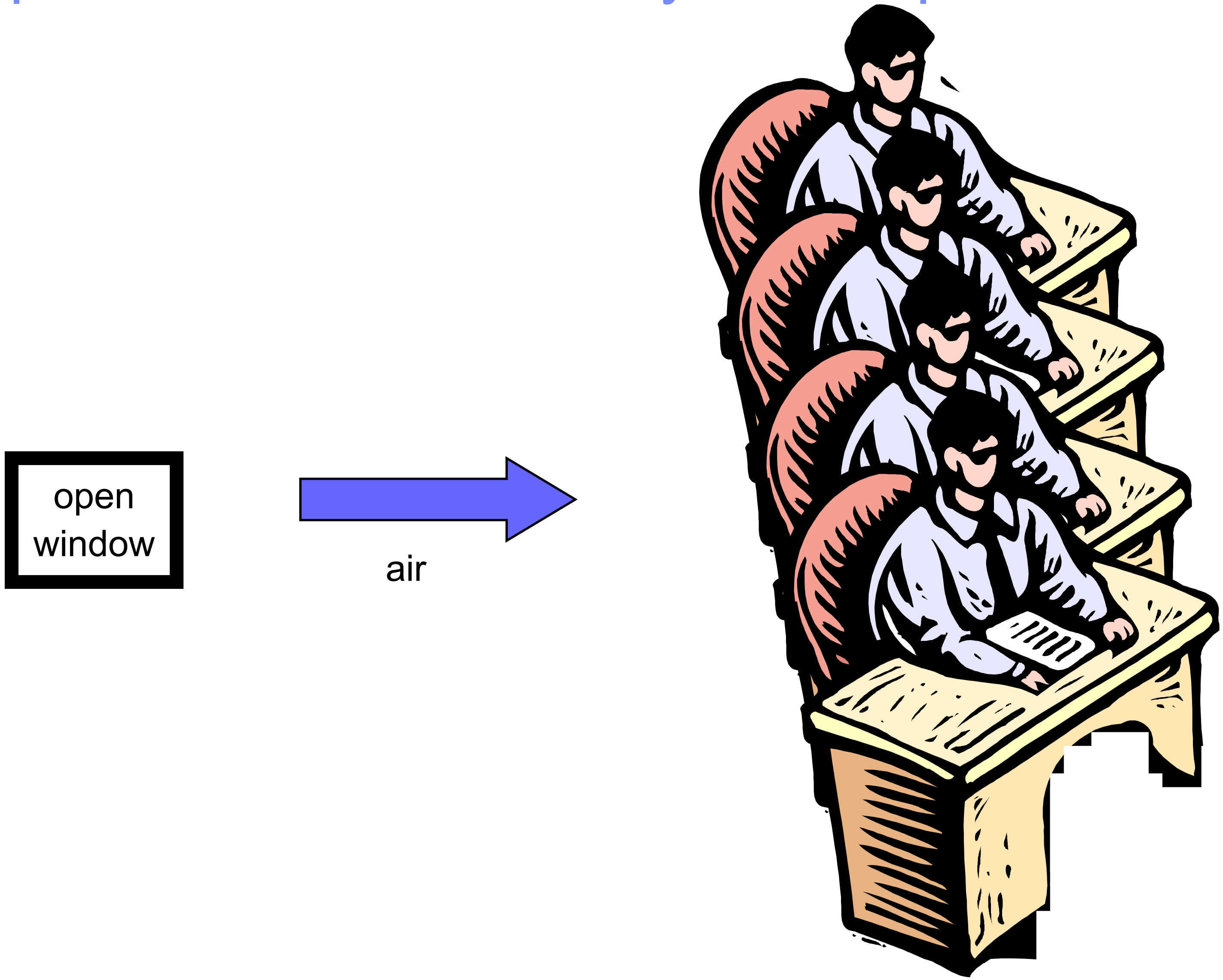
Data

- Reference input: objective
- Control error: reference input minus measured output
- Control input: manipulated to affect output
- Disturbance input: other factors that affect the target system
- Transduced output: result of manipulation

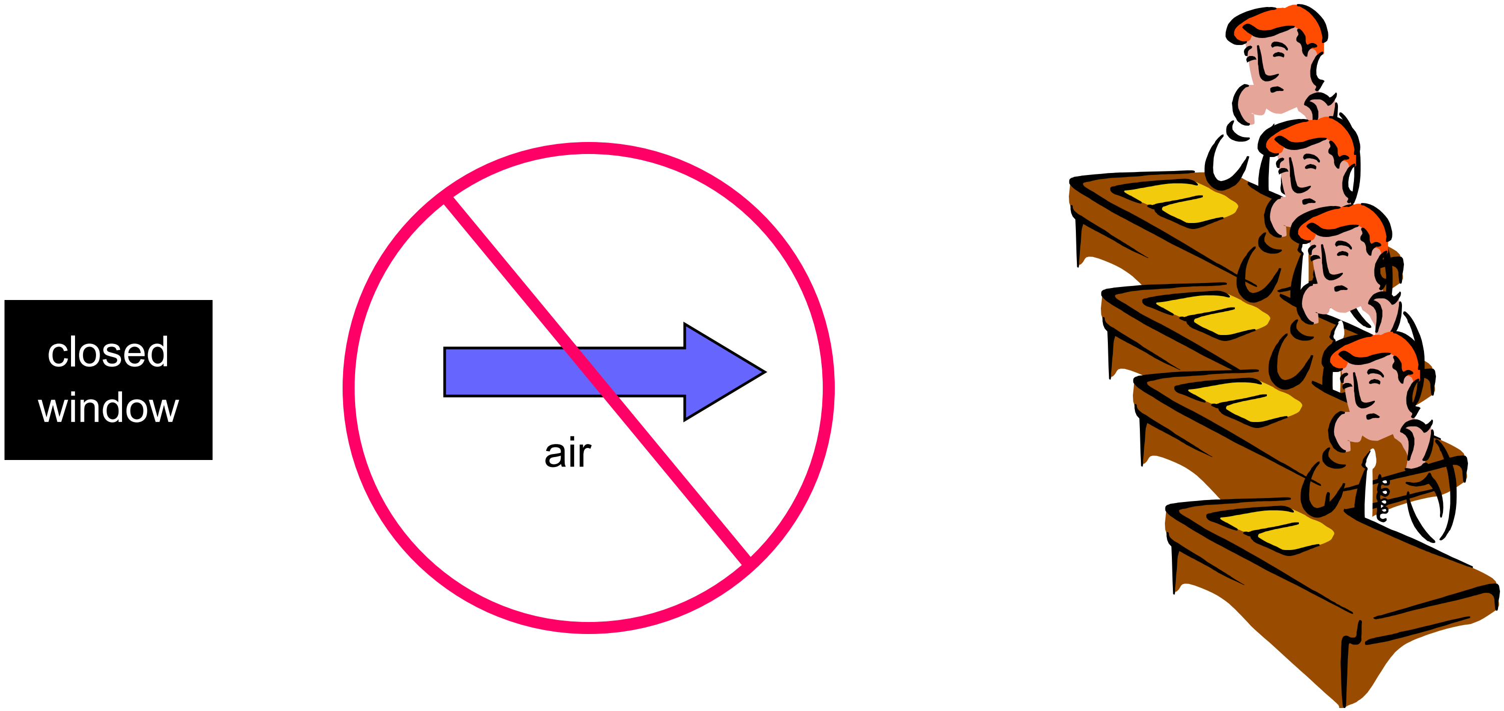
The Yawning Control System

- Description of system
  - Room full of people
  - Assumptions
    - People yawn because they need more oxygen
    - Yawning consumes more oxygen than regular breathing
    - Can open windows to increase oxygen flow, but it’s winter
- Control objective
  - Constrain yawning while maximizing temperature
- Questions
  - What are the main components of the system? A block diagram?
  - What control policies can achieve the objective?
  - What does it mean for this system to be unstable? What would make it unstable?

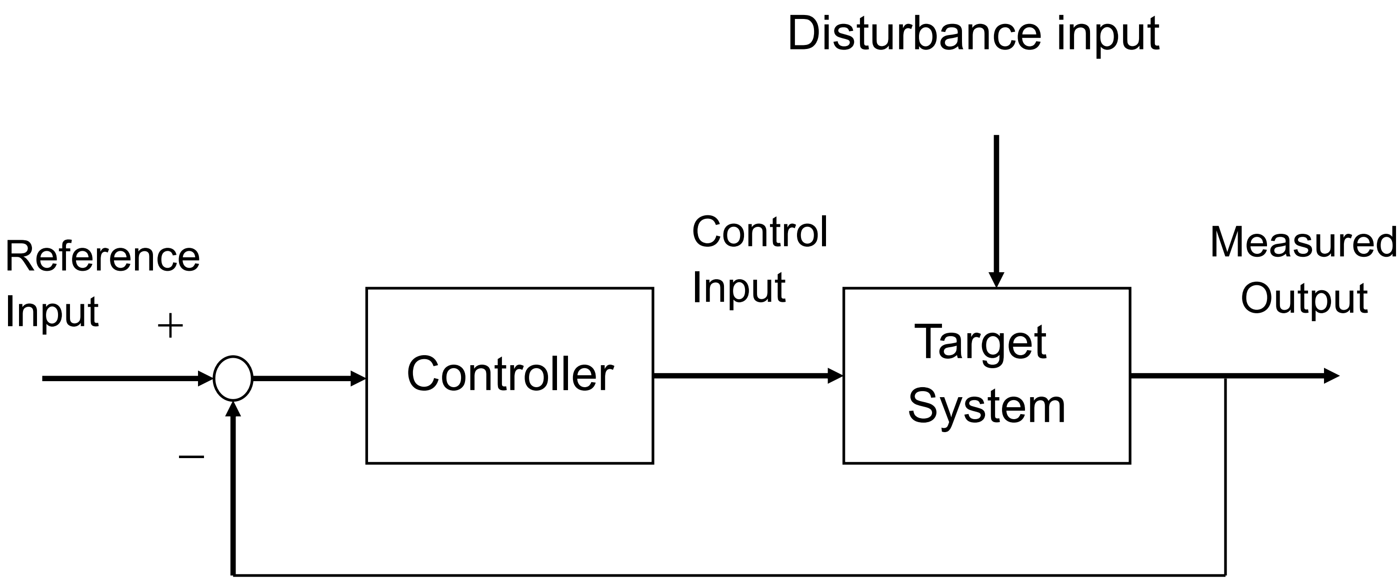
Operation of the Yawn System: Open Window



Operation of the Yawn System: Closed Window

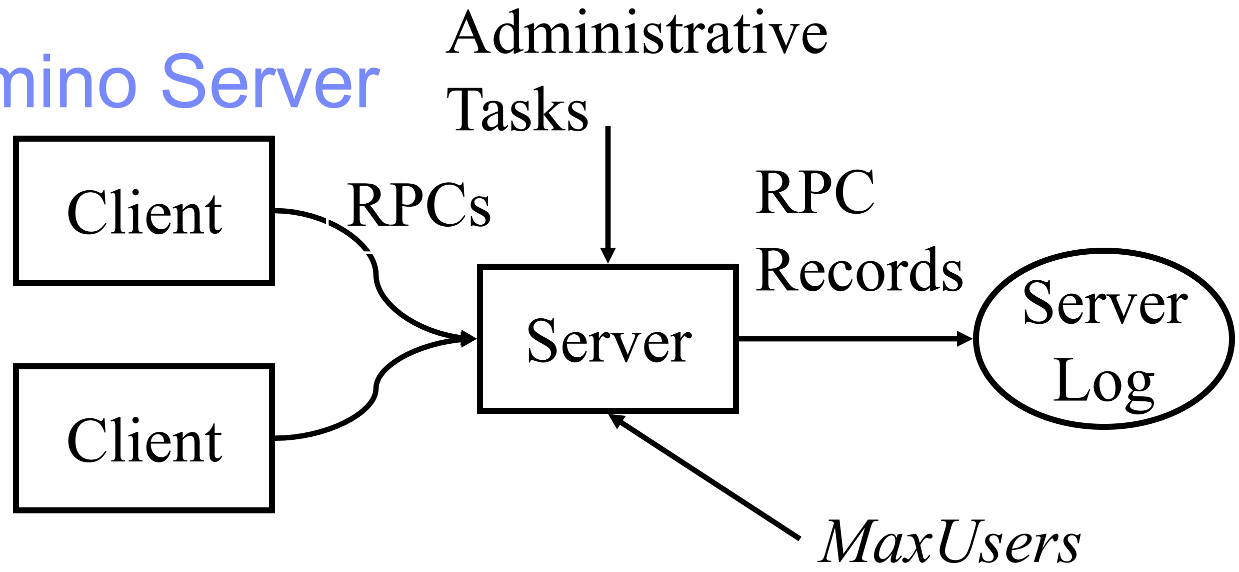


Feedback For Yawning System

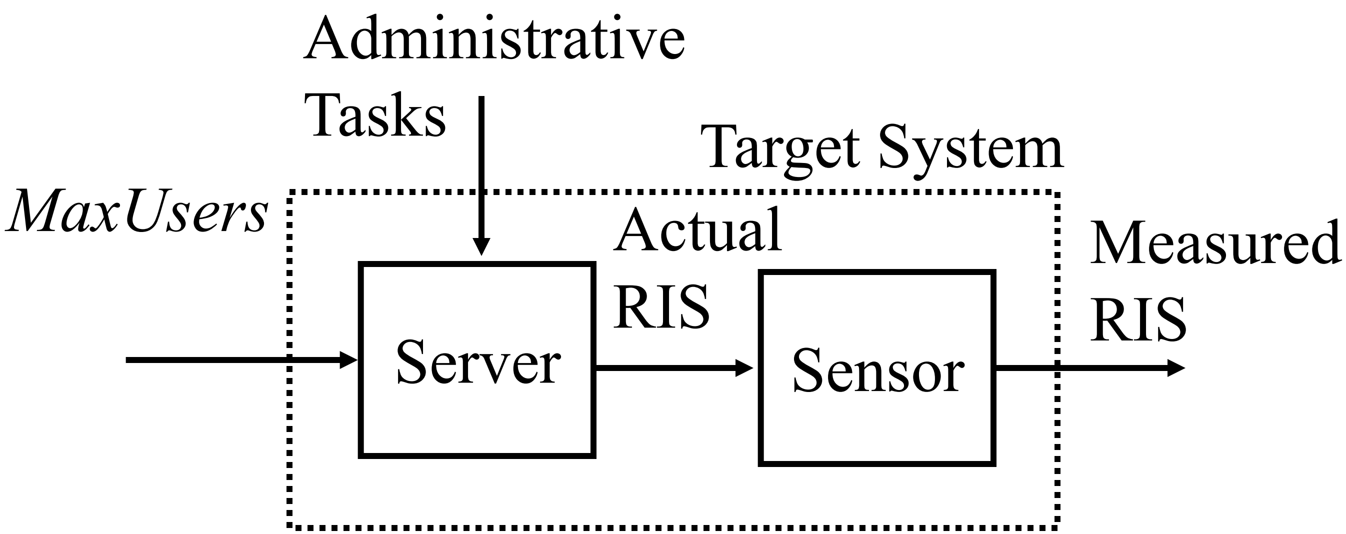


What is the	**Answers**
Target System	Windows + Students
Controller	Who/what determines the height of the windows
Reference input	Maximum tolerable yawn rate
Control input	Height of the window
Disturbance input	Add/remove people, opening door, …
Measured output	Observed yawn rate

IBM Lotus Domino Server



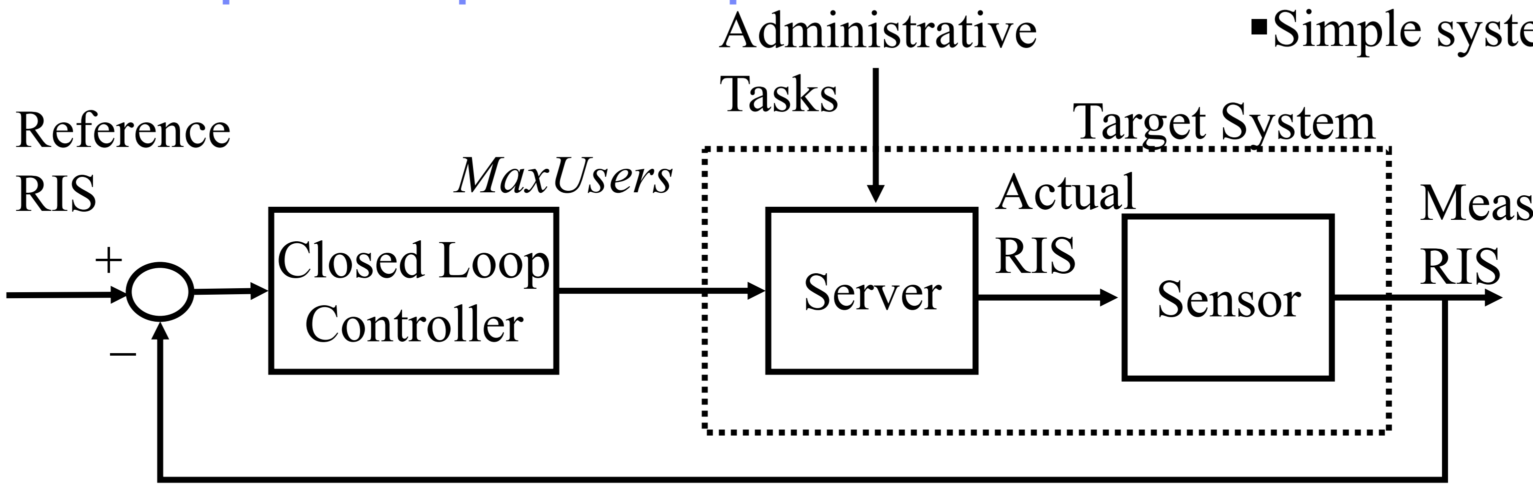
Architecture



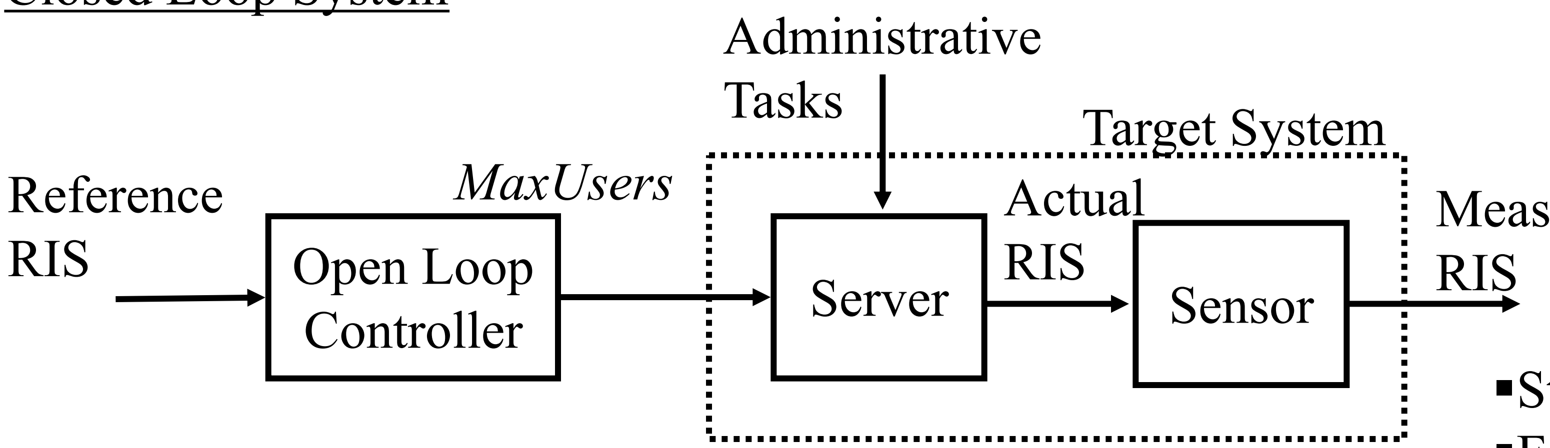
Block Diagram

Closed Loop vs. Open Loop

- Adapts
- Simple system model



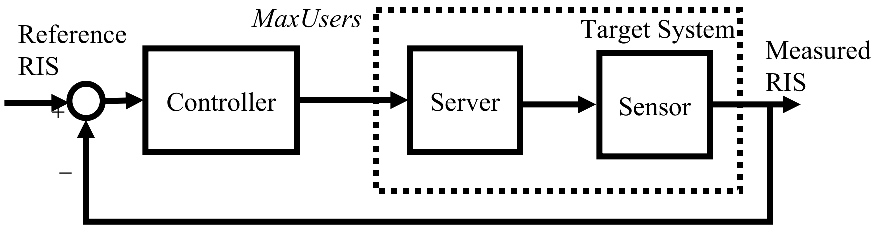
<u>Closed Loop System</u>



- Stable
- Fast settling

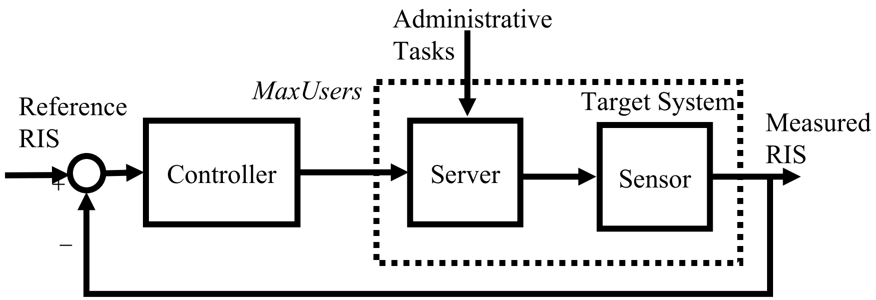
<u>Open Loop System</u>

Types of Control



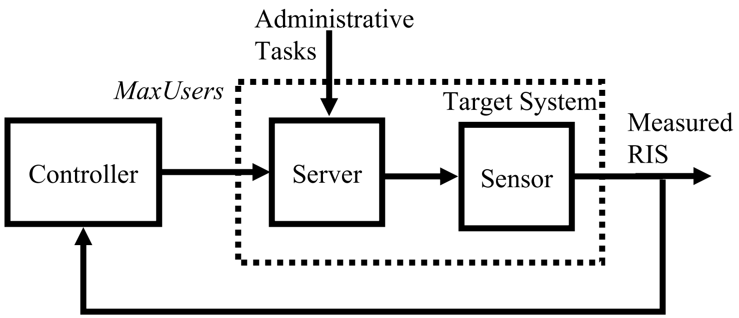
- Manage to a reference value
- Ex: Service differentiation, resource management, constrained optimization

Regulatory Control



- Eliminate effect of a disturbance
- Ex: Service level management, resource management, constrained optimization

Disturbance Rejection



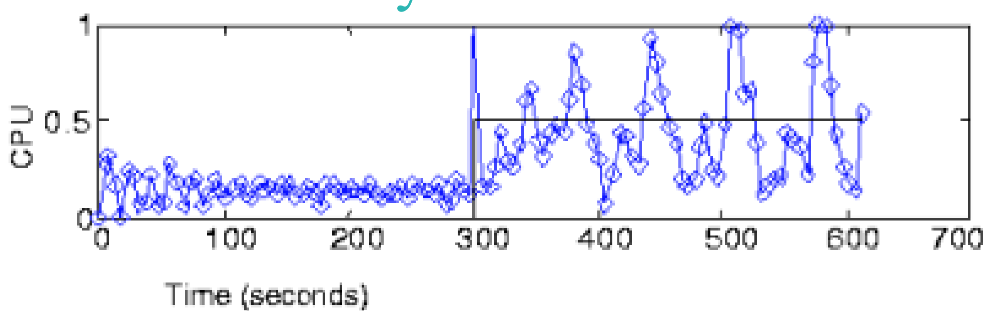
- Achieve the “best” value of outputs
- Ex: Minimize Apache response times

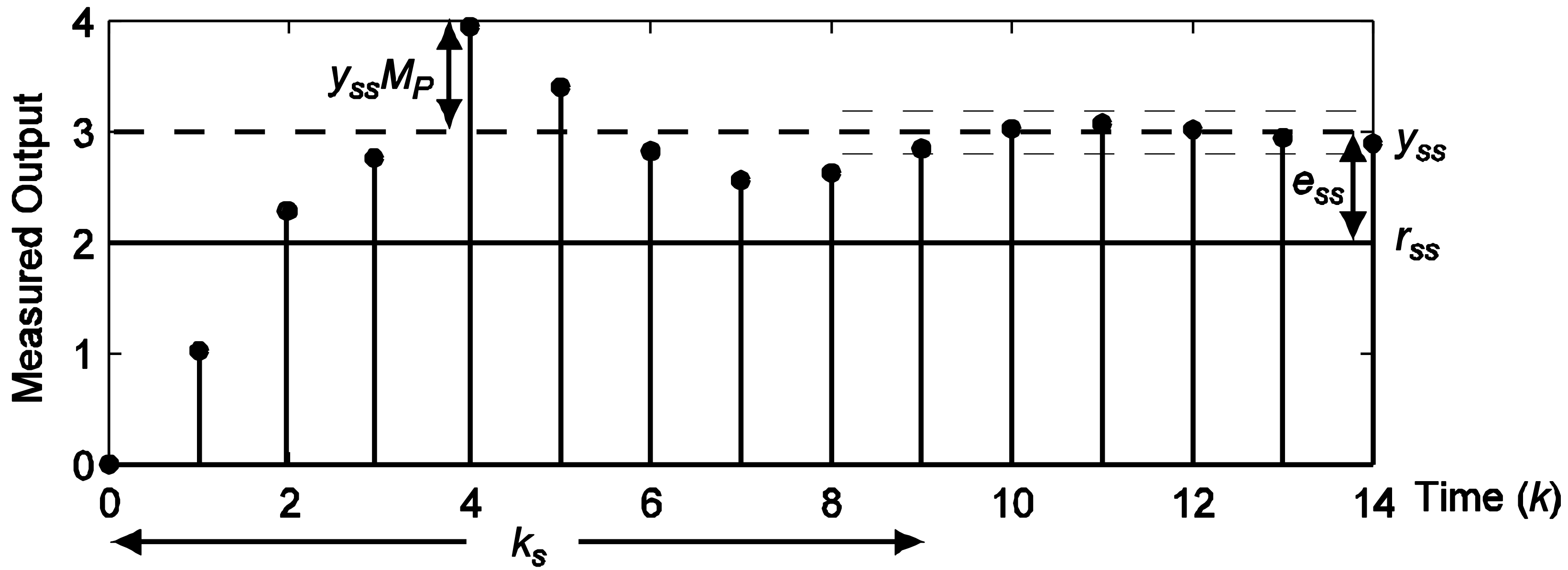
Optimization

The SASO Properties of Control Systems

Stability **A**ccuracy Short **S**ettling Small **O**vershoot

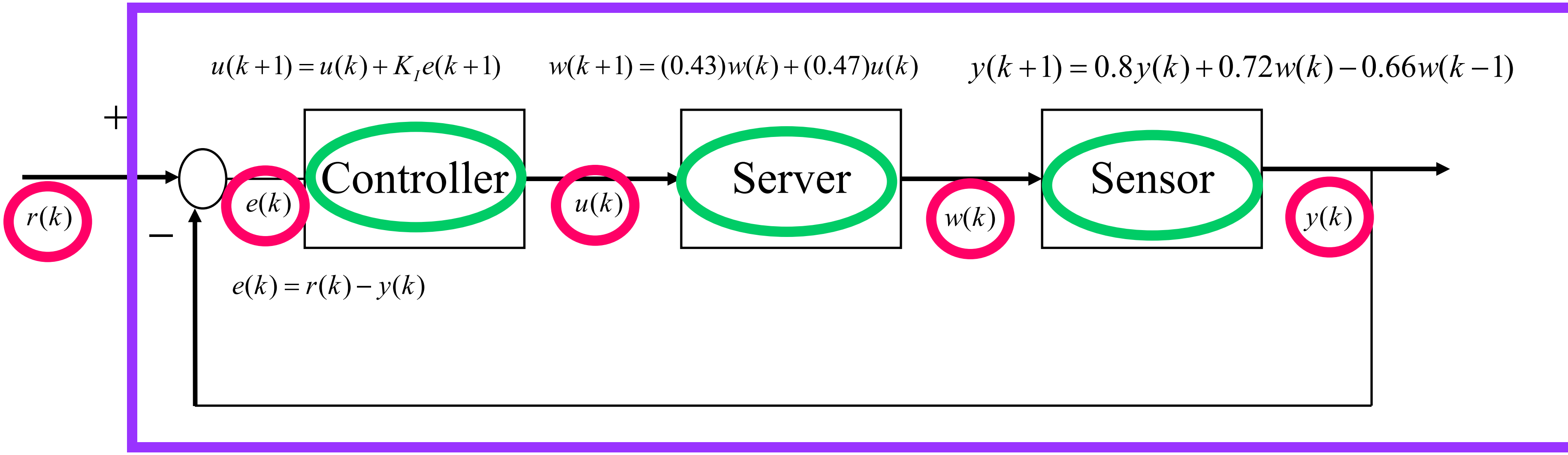
Unstable System





M2 - Theory

Motivating Example



The problem

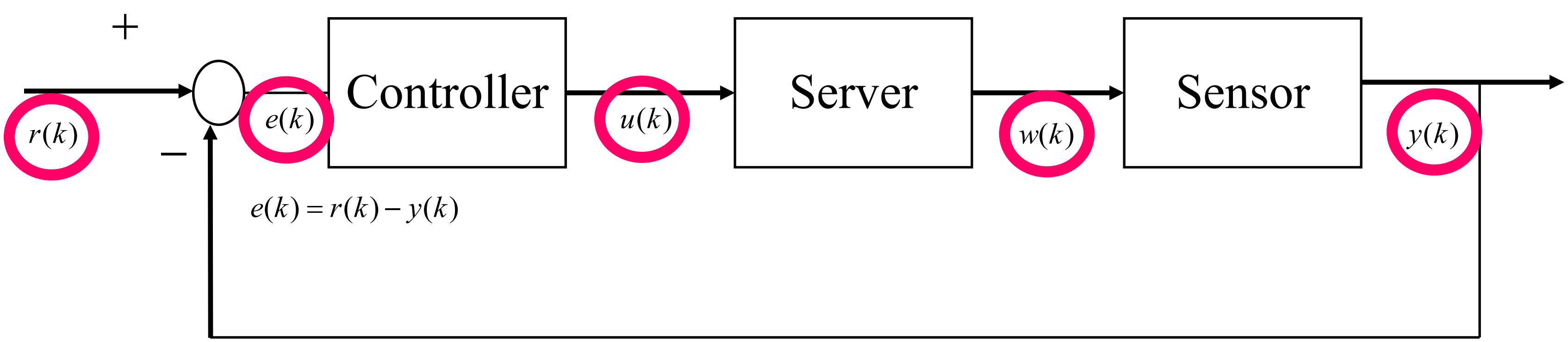
Want to find *y(k)* in terms of K_I so can design control system that is stable, accurate, settles quickly, and has small overshoot.

a. Signals

b. Transfer functions

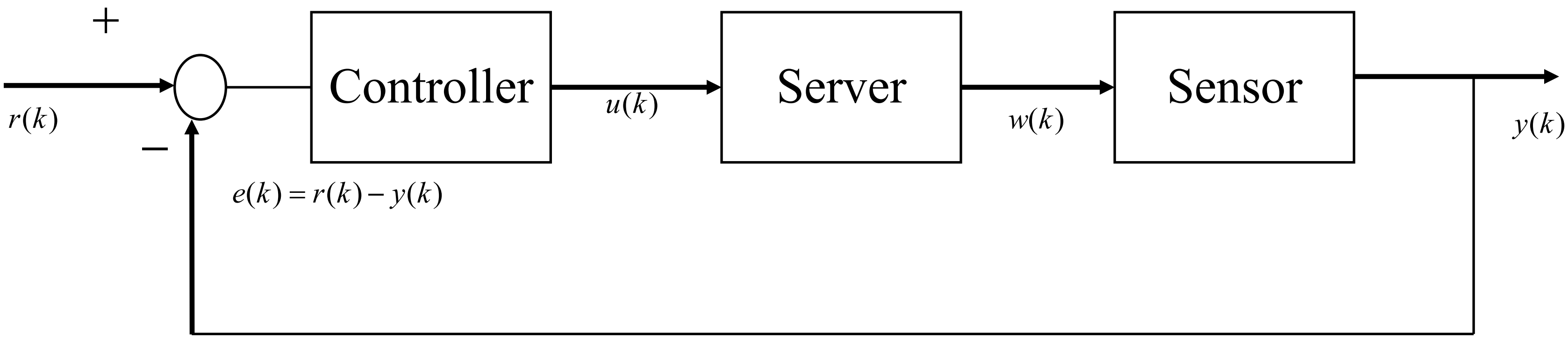
c. Composition of components – end-to-end system

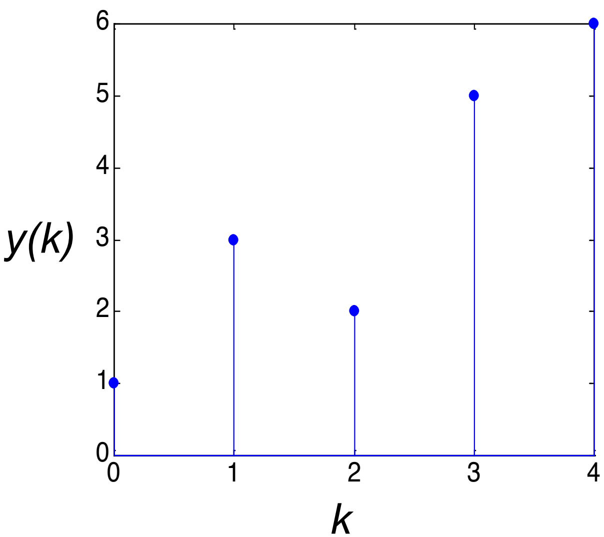
M2a – Theory
Signals



Reference: "Feedback Control of Computer Systems", Chapter 3.

Signals





<u>Time domain representation</u>

y(0)=1

y(1)=3

y(2)=2

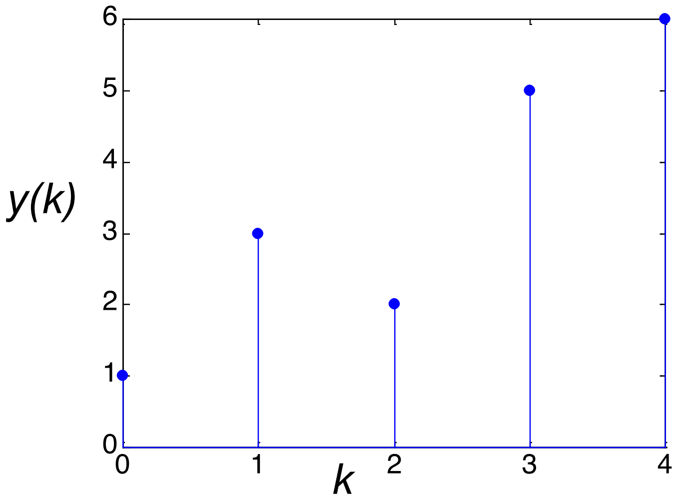
y(3)=5

y(4)=6

A signal is a real-valued function of time.

Issue: Time domain analysis is cumbersome in studying complicated control systems

Z-Transform of a Signal



Time domain representation	**z domain representation**
$y(0)=1$	$1z^{0}$ +
$y(1)=3$	$3z^{-1}$ +
$y(2)=2$	$2z^{-2}$ +
$y(3)=5$	$5z^{-3}$ +
$y(4)=6$	$6z^{-4}$

z is time shift;
z^{-1} is time delay ⇨

$z^{0}=1$: $k=0$ (current time)

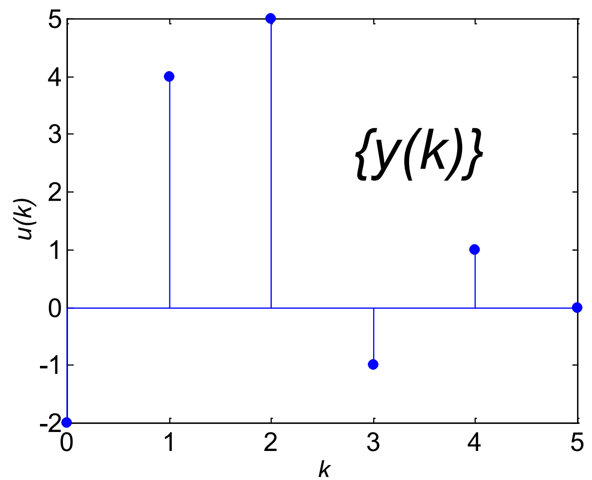
z^{-1}: $k=1$ (one time unit in the future)

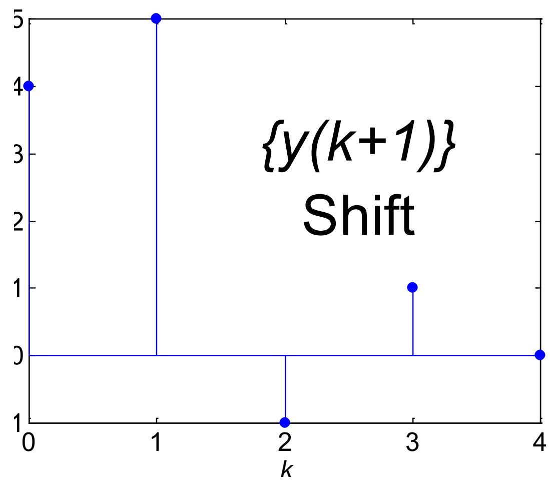
z^{-2}: $k=2$ (two time units in the future)

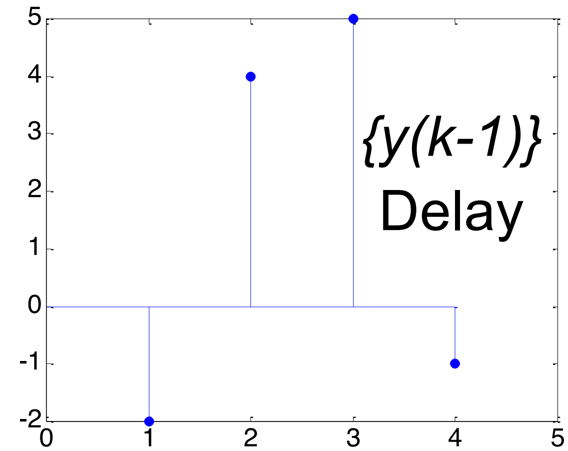
If $\{y(k)\}=y(0), y(1), \ldots$ is a signal, then its z-Transform is $Y(z)=\sum_{k=0}^{\infty} y(k) z^{-k}$

Signal Shifts and Delays

$$V(z) = zU(z) = 4 + 5z^{-1} - z^{-2} + z^{-3}$$







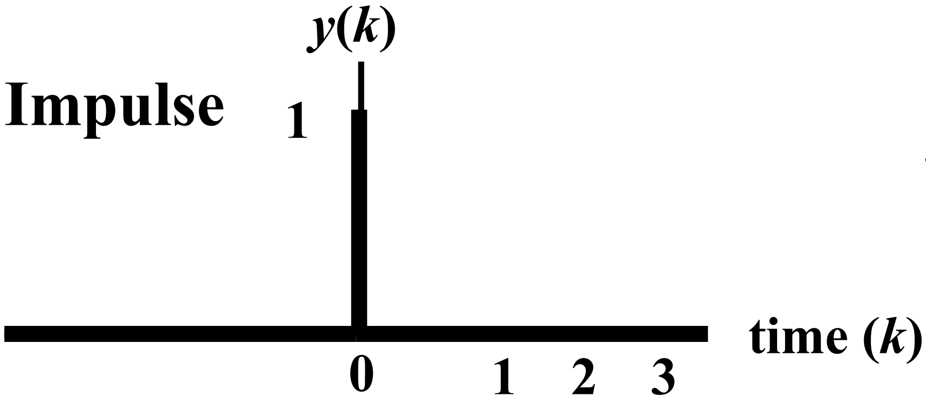
$$U(z) = -2 + 4z^{-1} + 5z^{-2} - z^{-3} + z^{-4}$$

(Drop exponents >0.)

$$V(z) = z^{-1}U(z) = 2z^{-1} + 4z^{-2} + 5z^{-3} - 1z^{-1}$$

Common Signals: Impulse

Impulse



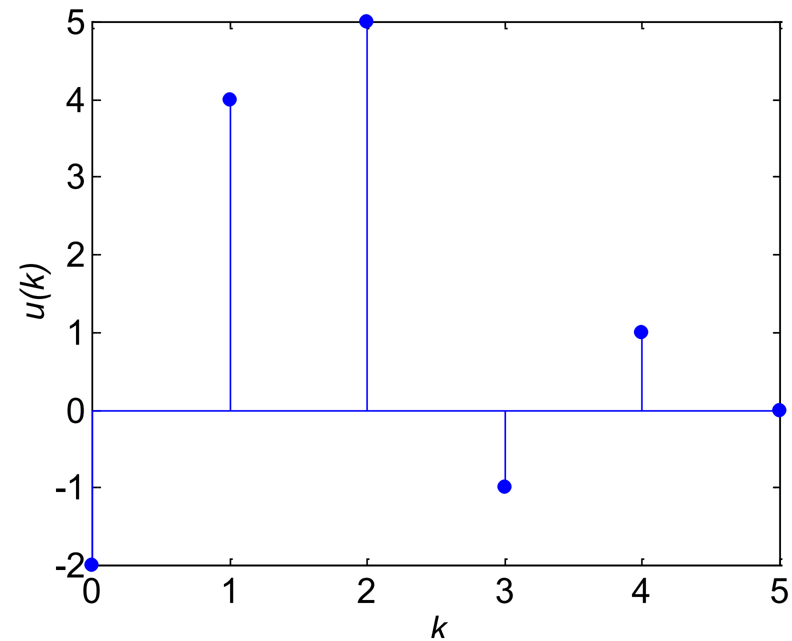
$$y(0) = 1; y(k) = 0, k > 0$$

$$Y(z) = 1z^{0} + 0z^{-1} + 0z^{-2} + \ldots$$

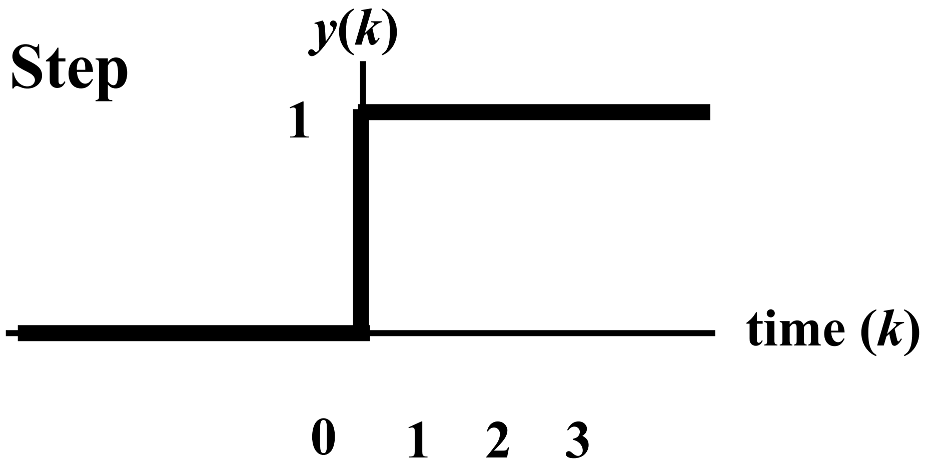
$$= 1$$

$$U(z) = -2 + 4z^{-1} + 5z^{-2} - z^{-3} + z^{-4}$$

This can be viewed as a sum of impulses at time 0, 1, 2, 3, and 4.



Common Signals: Step



$$y(k) = 1, k \geq 0$$

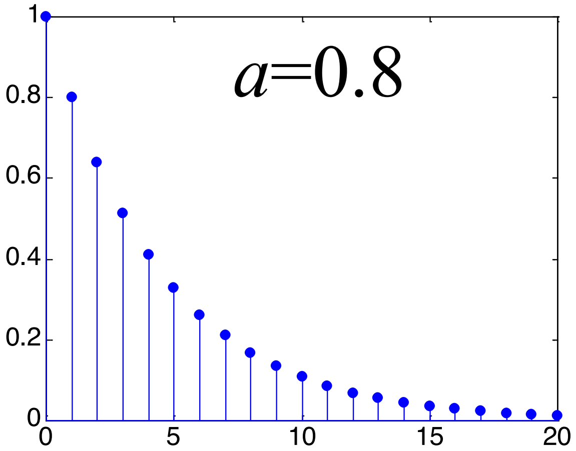
$$Y(z) = 1z^{0} + 1z^{-1} + 1z^{-2} + \ldots$$

$$= \frac{1}{1 - z^{-1}}$$

$$= \frac{z}{z - 1}$$

Common Signals: Geometric

$$\text{Geometric}: y(k) = a^k$$



$$Y(z) = 1 + az^{-1} + a^2 z^{-2} + \ldots$$

$$= \frac{z}{z-a}$$

$$Y(z) = 1 + 0.8z^{-1} + 0.64z^{-2} + \ldots$$

$$= \frac{z}{z-0.8}$$

Properties of Z-Transforms of Signals

Signals: $U(z) = u(0)z^0 + u(1)z^{-1} + u(2)z^{-2} + \ldots$

$$V(z) = v(0)z^0 + v(1)z^{-1} + v(2)z^{-2} + \ldots$$

Shift: $zU(z) = u(0)z^1 + u(1)z^0 + u(2)z^{-1} + \ldots$

$$= u(1)z^0 + u(2)z^{-1} + \ldots$$

Delay: $U(z)/z = u(0)z^{-1} + u(1)z^{-2} + u(2)z^{-3} + \ldots$

Scaling: $aU(z) = au(0)z^0 + au(1)z^{-1} + au(2)z^{-2} + \ldots$

$$= z\text{-Transform of } \{au(k)\}$$

Sum of signals: $u(0)z^0 + u(1)z^{-1} + u(2)z^{-2} + \ldots + v(0)z^0 + v(1)z^{-1} + v(2)z^{-2} + \ldots$

$$= (u(0) + v(0))z^0 + (u(1) + v(1))z^{-1} + (u(2) + v(2))z^{-2} + \ldots$$

$$= U(z) + V(z)$$

Poles of a Z-Transform

Definition: Values of *z* for which the denominator is 0

Easy to find the poles of a geometric: $$V(z) = \frac{z}{z-a}$$ Pole is *a*.

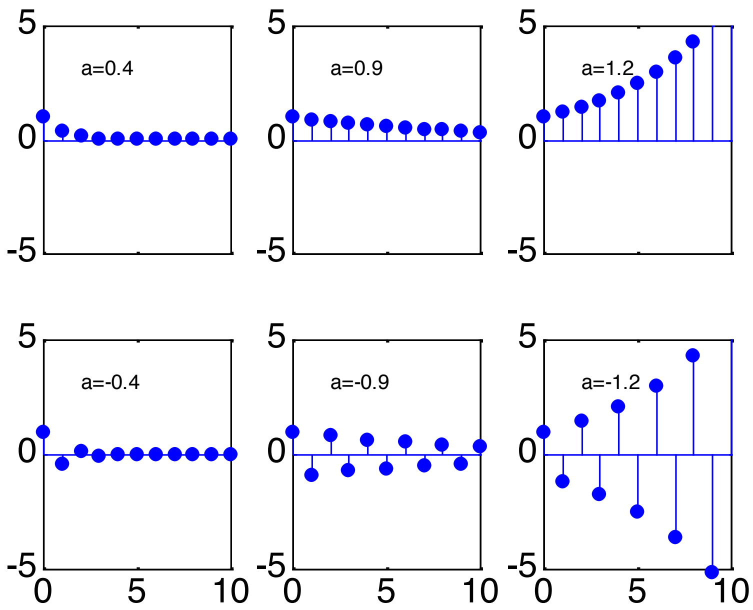
What are the poles of the following Z-Transform? $$Y(z) = \frac{7z^2 - 6z}{z^2 - 1.8z + 0.8}$$

Easy if sum of geometrics $$Y(z) = \frac{5z}{z-1} + \frac{2z}{z-0.8}$$

Poles determine key behaviors of signals

Effect of Pole on the Signal

$$y(k) = a^k \Leftrightarrow \frac{z}{z-a}$$



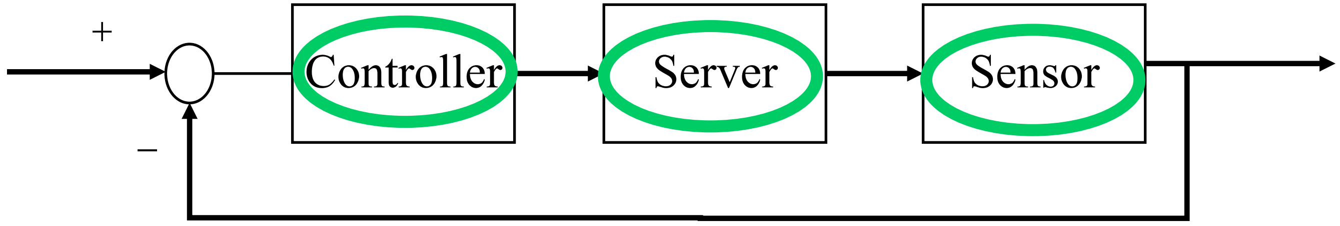
Why?

$$\frac{z}{z-a} = 1 + az^{-1} + a^2 z^{-2} + \ldots \Leftrightarrow (1, a, a^2, \ldots)$$

- What happens when
  - $|a|$ is larger?
  - $|a|>1$?
  - $a<0$?

- $|a|>1$
  - Does not converge
- Larger $|a|$
  - Slower convergence
- $a<0$
  - Oscillates

M2b – Theory
Transfer Functions



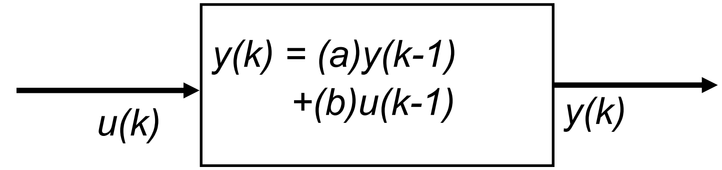
Reference: “Feedback Control of Computer Systems”, Chapter 3.

Motivation and Definition

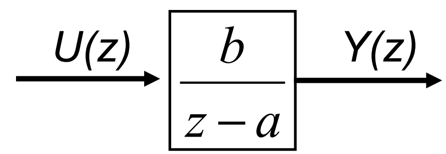
Motivation:

ARX model relates *u(k)* to *y(k)*

(ARX is autoregressive with external input.)



The picture can't be displayed.



Transfer function expresses this relationship in the z domain

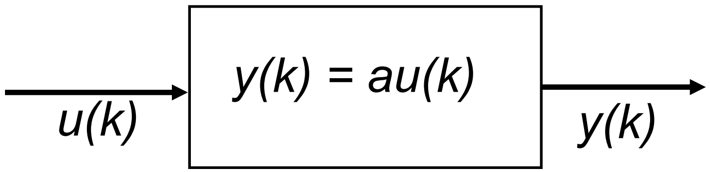
$$G(z) = \frac{\text{Output Signal}}{\text{Input Signal}} = \frac{Y(z)}{U(z)}$$

or $Y(z) = G(z)U(z)$

assuming initial conditions are 0.

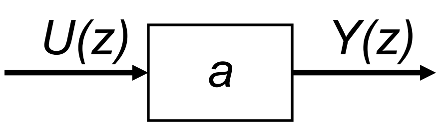
A transfer function is specified in terms of its input and output.

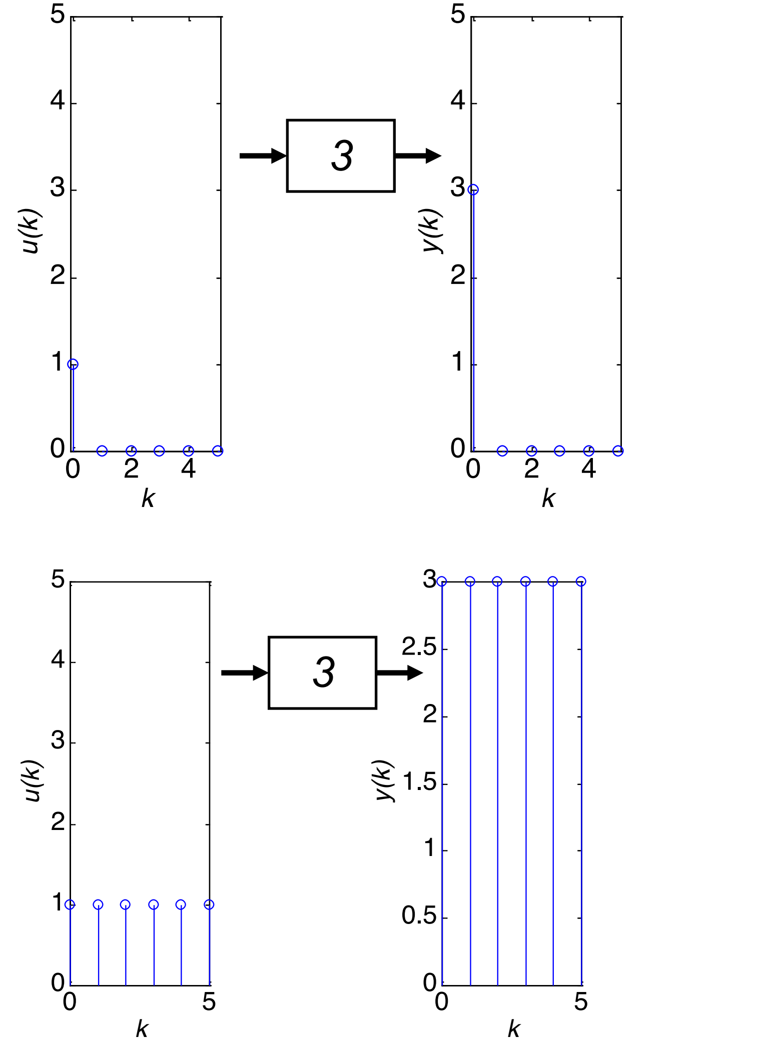
Constant Transfer Function



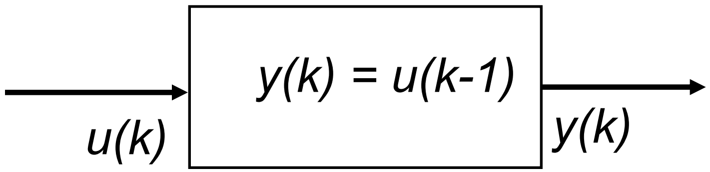
$$Y(z) = aU(z)$$

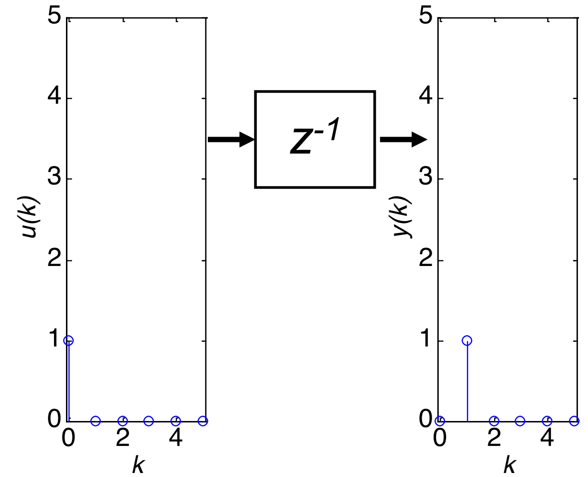
$$G(z) = \frac{Y(z)}{U(z)} = a$$





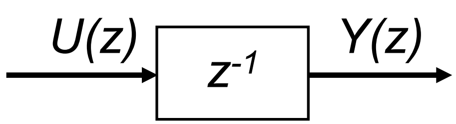
1-Step Time-Delay Transfer Function

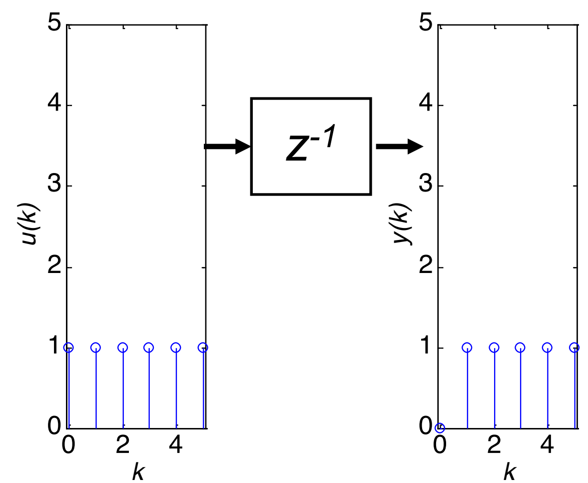




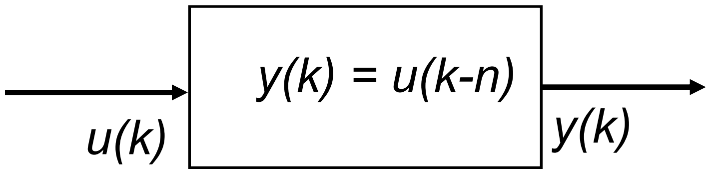
$$Y(z) = z^{-1}U(z)$$

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



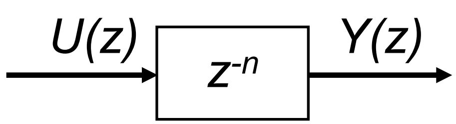


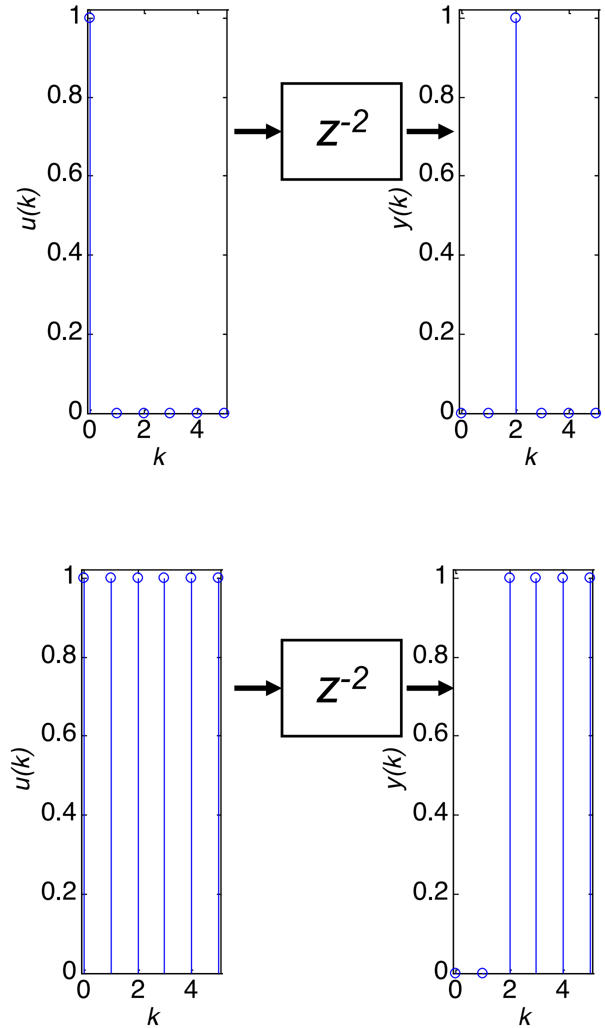
n-Step Time-Delay Transfer Function



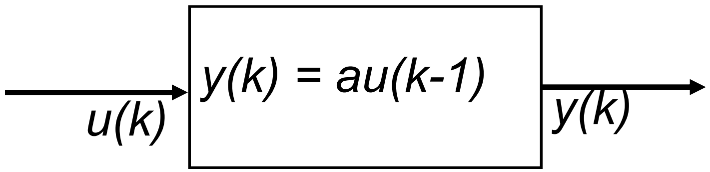
$$Y(z) = z^{-n}U(z)$$

$$G(z) = \frac{Y(z)}{U(z)} = z^{-n}$$



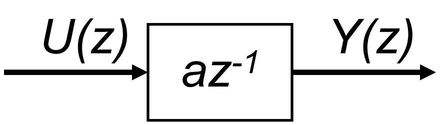


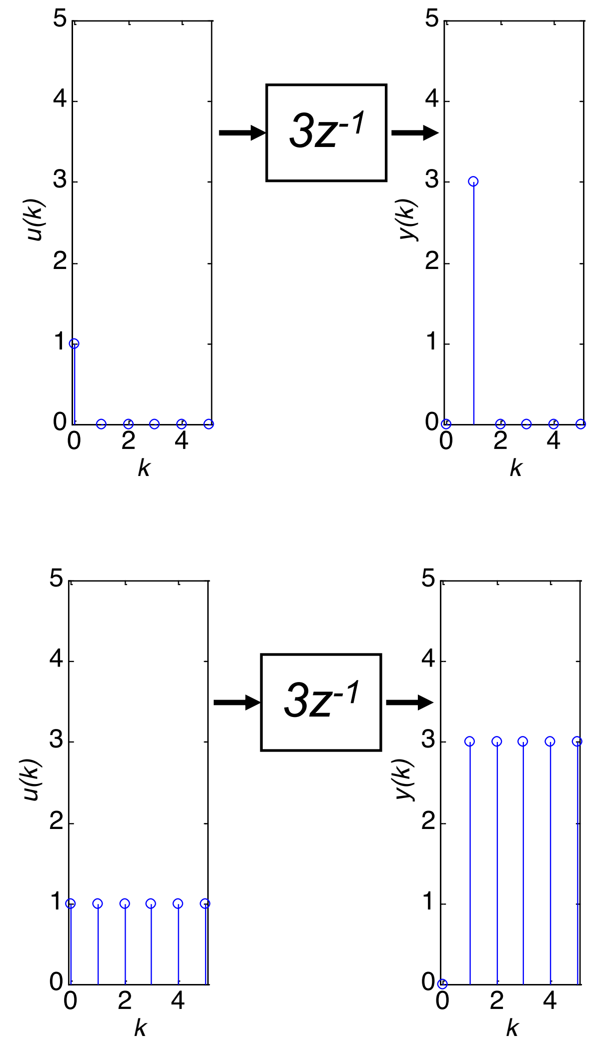
Combining Simple Transfer Functions



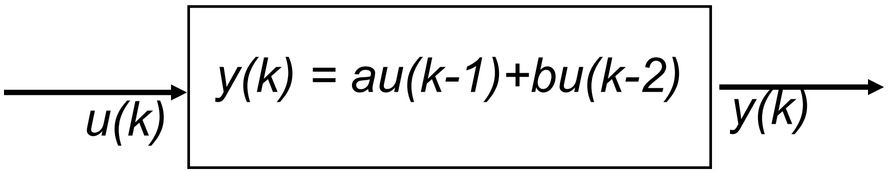
$$Y(z) = az^{-1}U(z)$$

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



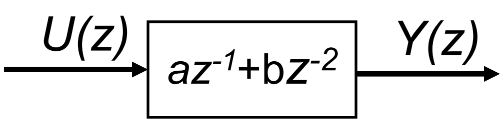


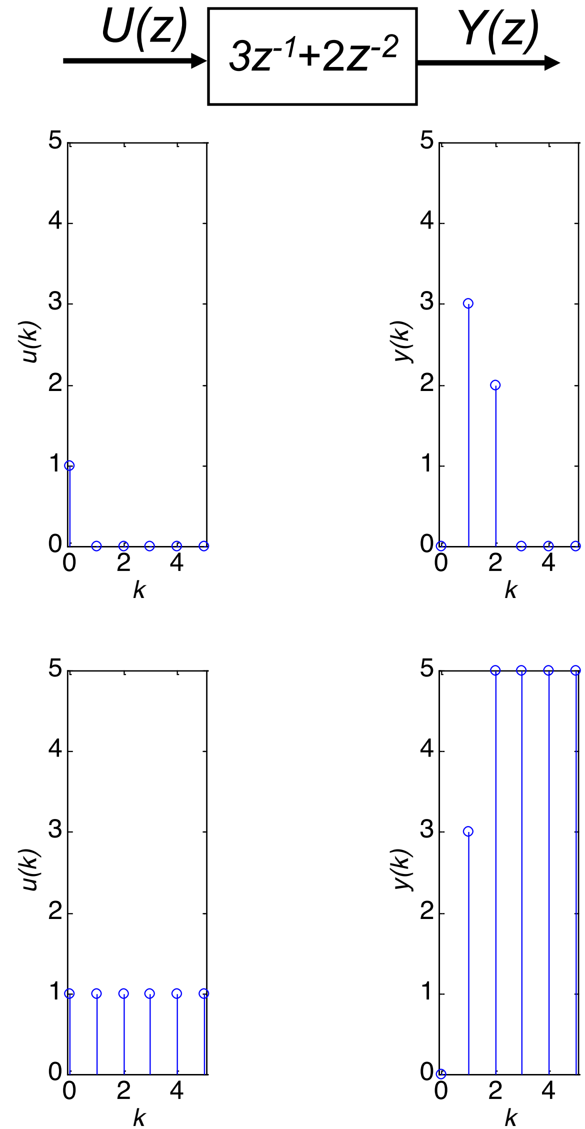
Additional Terms in T.F.



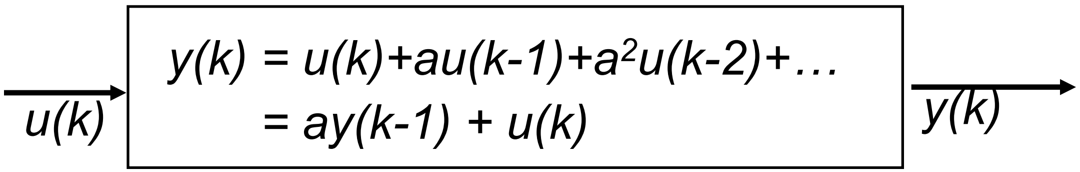
$$Y(z) = (az^{-1} + bz^{-2})U(z)$$

$$G(z) = \frac{Y(z)}{U(z)} = az^{-1} + bz^{-2}$$

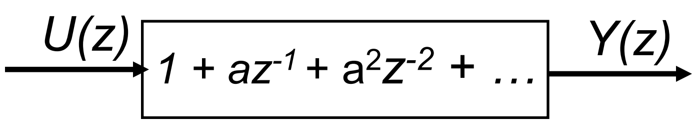




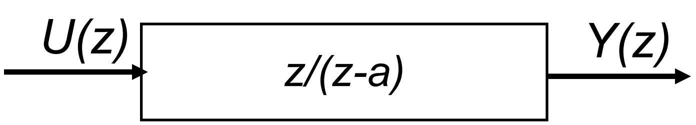
Geometric Sum of T.F.

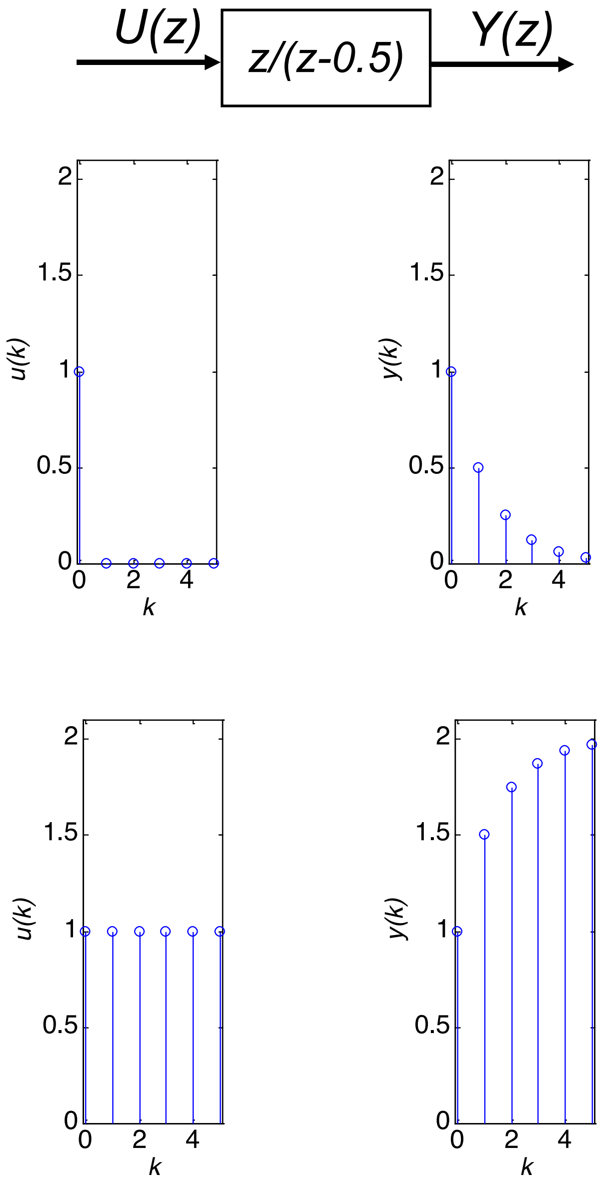


$$G(z) = \frac{Y(z)}{U(z)} = (1 + az^{-1} + a^2 z^{-2} + \ldots)$$



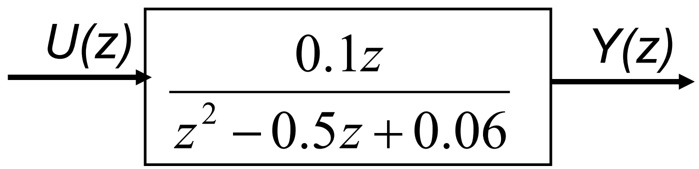
$$G(z) = \frac{Y(z)}{U(z)} = 1 + az^{-1} + a^2 z^{-2} + \ldots = \frac{z}{z-a}$$

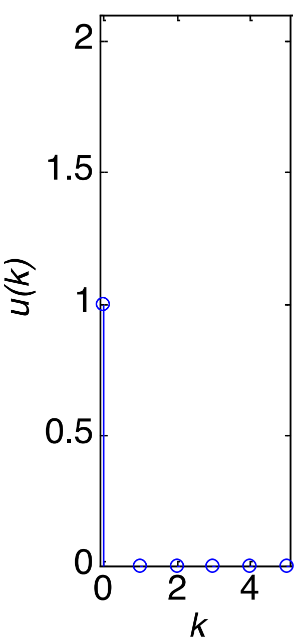


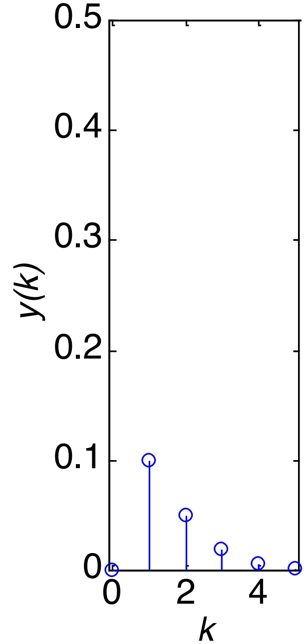


Complicated Transfer Functions

Decompose into a sum of geometrics







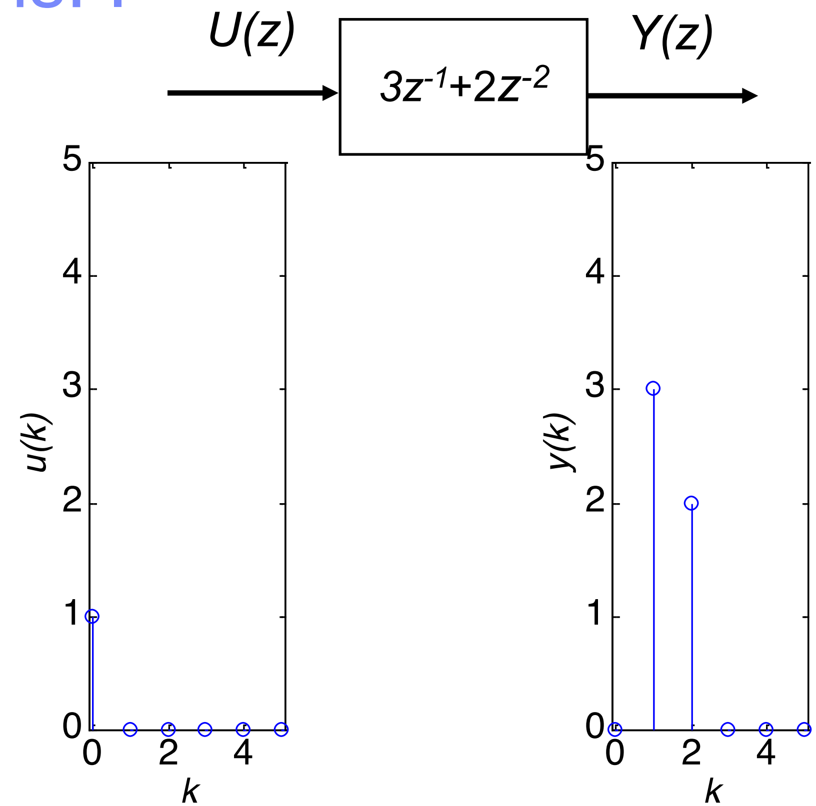
$$G(z)=\frac{0.1z}{z^2-0.5z+0.06}$$
$$=\frac{z}{z-0.3}-\frac{z}{z-0.2}$$
$$=1+0.3z^{-1}+0.09z^{-2}+\ldots-1-0.2z^{-1}-0.04z^{-2}+\ldots$$
$$=0.1z^{-1}+0.05z^{-2}+\ldots$$

- **Partial fraction expansion** allows rational polynomials to be decomposed into a sum of geometrics
- Poles of the original polynomial are the poles of the geometrics

Interpreting Transfer Functions: I

Signal generated by an impulse input

Example:



$$G(z) = \frac{Y(z)}{U(z)}$$

$$Y(z) = G(z)U(z) = G(z)(1) = G(z)$$

Interpreting Transfer Functions: II

Specifies an ARX model

Given a transfer function:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{z}{z - 0.5}$$

So, $Y(z)(z - 0.5) = zU(z)$ or $zY(z) = 0.5Y(z) + zU(z)$

Recall that:

$$aY(z) \Leftrightarrow ay(k)$$

$$zY(z) \Leftrightarrow y(k+1)$$

Which gives us:

$y(k+1) = 0.5y(k) + u(k+1)$ which is equivalent to

$$y(k) = 0.5y(k-1) + u(k)$$

This means that transfer functions are trivial to simulate!

Constructing Transfer Functions

- Given a ARX model, how do we construct its transfer function?
- Method: Term by term conversion from time domain to *z* Domain: (1) substitute for *z* expressions and (2) factor to obtain the ratio of output to *z*-Transforms.
- Example

$$\text{Given } y(k) = (0.43)y(k-1) + (0.47)u(k)$$

$$y(k) \Leftrightarrow Y(z)$$

$$y(k-1) \Leftrightarrow z^{-1}Y(z)$$

$$u(k) \Leftrightarrow U(z)$$

$$\text{Substitute}: Y(\mathrm{z}) = (0.43)z^{-1}Y(z) + (0.47)U(z)$$

$$\text{Factor}: \frac{Y(z)}{U(z)} = \frac{(0.47)z}{z-0.43}$$

A Pop Quiz

What is the transfer function for

$$y(k+1) = (0.8)y(k) + (0.72)w(k) - (0.66)w(k-1)$$

Hint : $y(k+1) \Leftrightarrow zY(z)$

Step 1 : Substitute

$$zY(z) = (0.8)Y(z) + (0.72)W(z) - (0.66)z^{-1}W(z)$$

Step 2 : Factor

$$\frac{Y(z)}{W(z)} = \frac{(0.72)z - 0.66}{z^2 - (0.8)z}$$

Test Your Knowledge Again

Given the transfer function $G(z) = \dfrac{b_m z^m + b_{m-1} z^{m-1} + \ldots + b_0}{a_n z^n + a_{n-1} z^{n-1} + \ldots + a_0}$

Why must it be that $n \geq m$?

Write the ARX model:

$$a_n y(k+n) + a_{n-1} y(k+n-1) + \ldots + a_0 = b_m u(k+m) + b_{m-1} u(k+m-1) + \ldots + b_0$$

Adjust time so that $k+n \to k$

$$a_n y(k) + a_{n-1} y(k-1) + \ldots + a_0 = b_m u(k+m-n) + b_{m-1} u(k+m-n-1) + \ldots + b_0$$

If $m > n$, then $y(k)$ is a function of one or more $u(k+m-n)$ in the future!

Psychic System!

Poles of a Transfer Function

Poles: Values of *z* for which the denominator is 0.

Example:

$$H(z) = \frac{0.1z}{z^2 - 0.5z + 0.06} = \frac{z}{z - 0.3} - \frac{z}{z - 0.2}$$

Poles: 0.3, 0.2

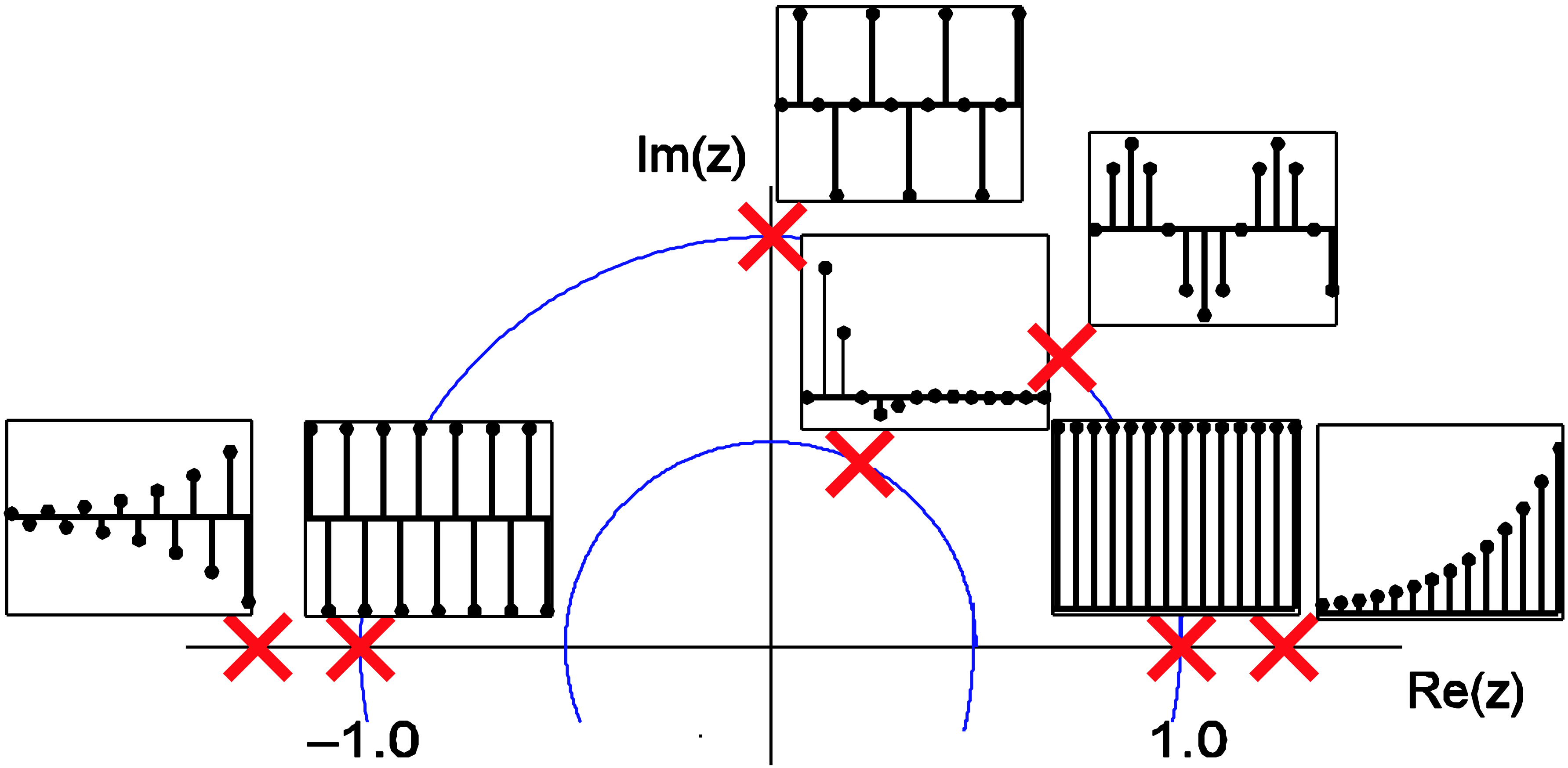
Poles

- Determine stability
- Major effect on settling time, overshoot
- **Dominant pole** – pole that determines the transient response

$$G(z) = \frac{z}{z - a} = 1 + az^{-1} + a^2 z^{-2} + a^3 z^{-3} + \ldots$$

- $|a|>1$
  - Does not converge
- $|a|<1$ but large
  - Slower convergence
- $a<0$
  - Oscillates

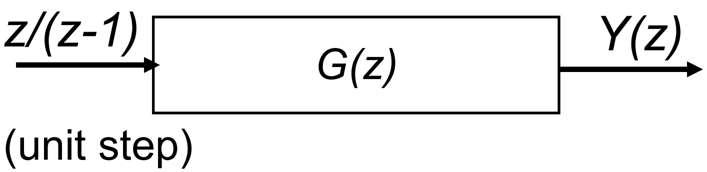
Almost All You'll Ever Need to Know About Poles



$$G(z) = \frac{z}{z-a} = 1 + az^{-1} + a^2 z^{-2} + a^3 z^{-3} + \ldots$$

Settling Time (k_s) of a System

Definition and result: Time until an input signal is within 2% of its steady state value



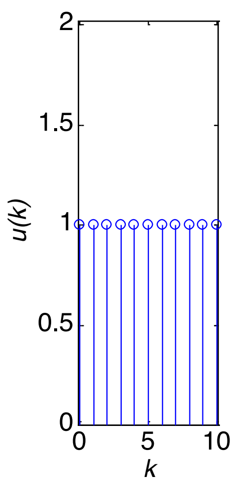
$$k_S \approx \frac{-4}{\ln|a|}, \text{ where } |a| \text{ is the largest pole of } G(z),$$

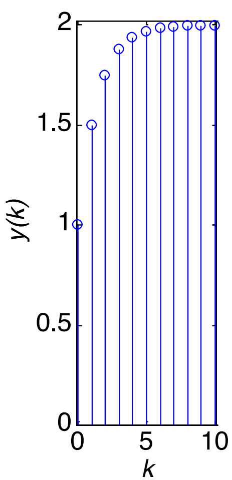
$$\text{with equality for } G(z) = \frac{z}{z-a} = 1 + az^{-1} + a^2z^{-2} + a^3z^{-3} + \ldots$$

Examples:

$$G(z) = \frac{z}{z - 0.5}$$

$$k_S \approx \frac{-4}{\ln 0.5} \approx 6$$

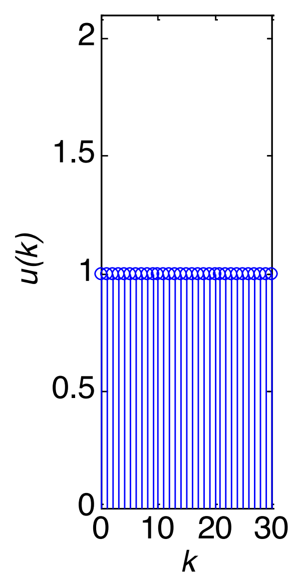


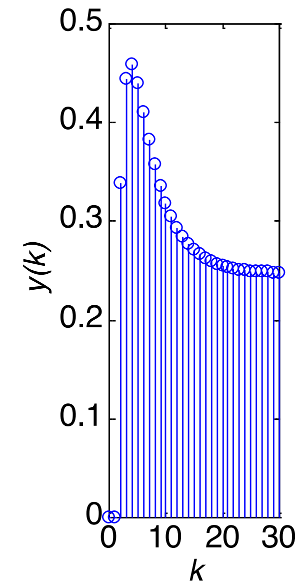


$$G(z) = \frac{0.34z - 0.31}{z^3 - 1.23z^2 + 0.34z},$$

$$\text{poles}: 0, 0.43, 0.8$$

$$k_S \approx \frac{-4}{\ln 0.8} \approx 18$$

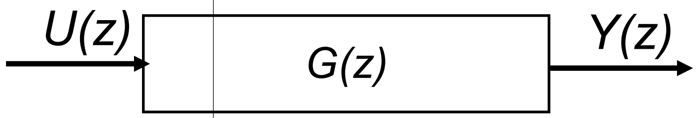




Steady State Gain (ssg) of a Transfer Function

Steady state gain is the steady state output in response to a step input.

The picture can't be displayed.

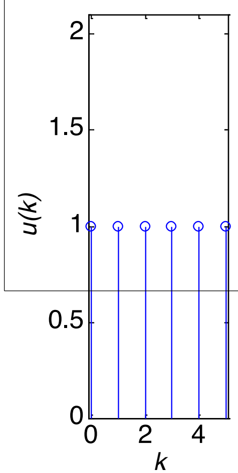


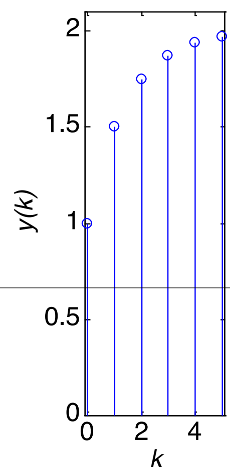
ssg of *G(z)* is $\dfrac{y(\infty)}{u(\infty)} = G(1)$

where *U(z)* is a step input.

<u>Example:</u>

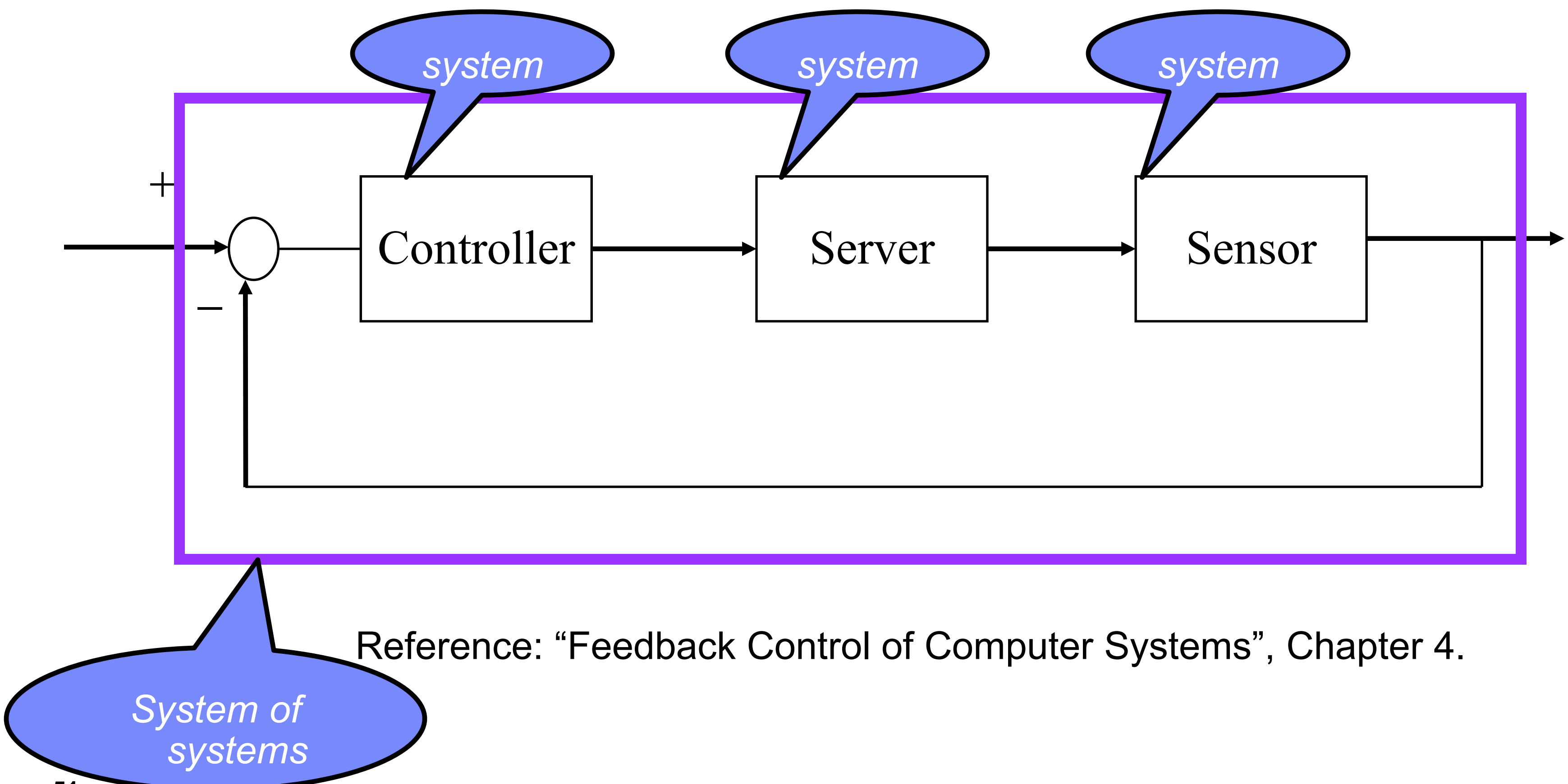
$$u(\infty) = 1 \qquad G(z) = \frac{z}{z - 0.5} \qquad y(\infty) = 2$$





$$\frac{y(\infty)}{u(\infty)} = \frac{2}{1} = 2 = G(1)$$

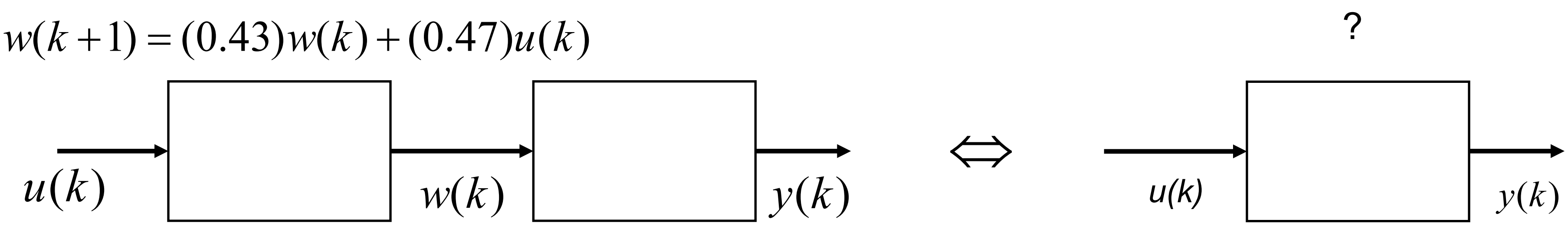
M2c – Theory
Composition of Systems



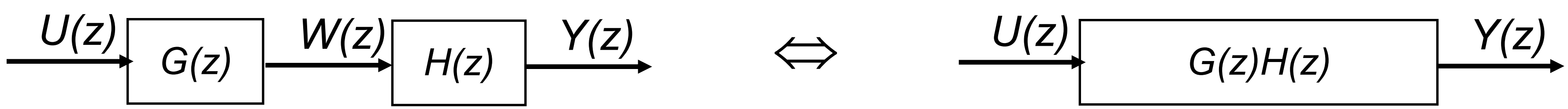
Reference: “Feedback Control of Computer Systems”, Chapter 4.

Transfer Functions In Series

$$w(k+1) = (0.43)w(k) + (0.47)u(k)$$



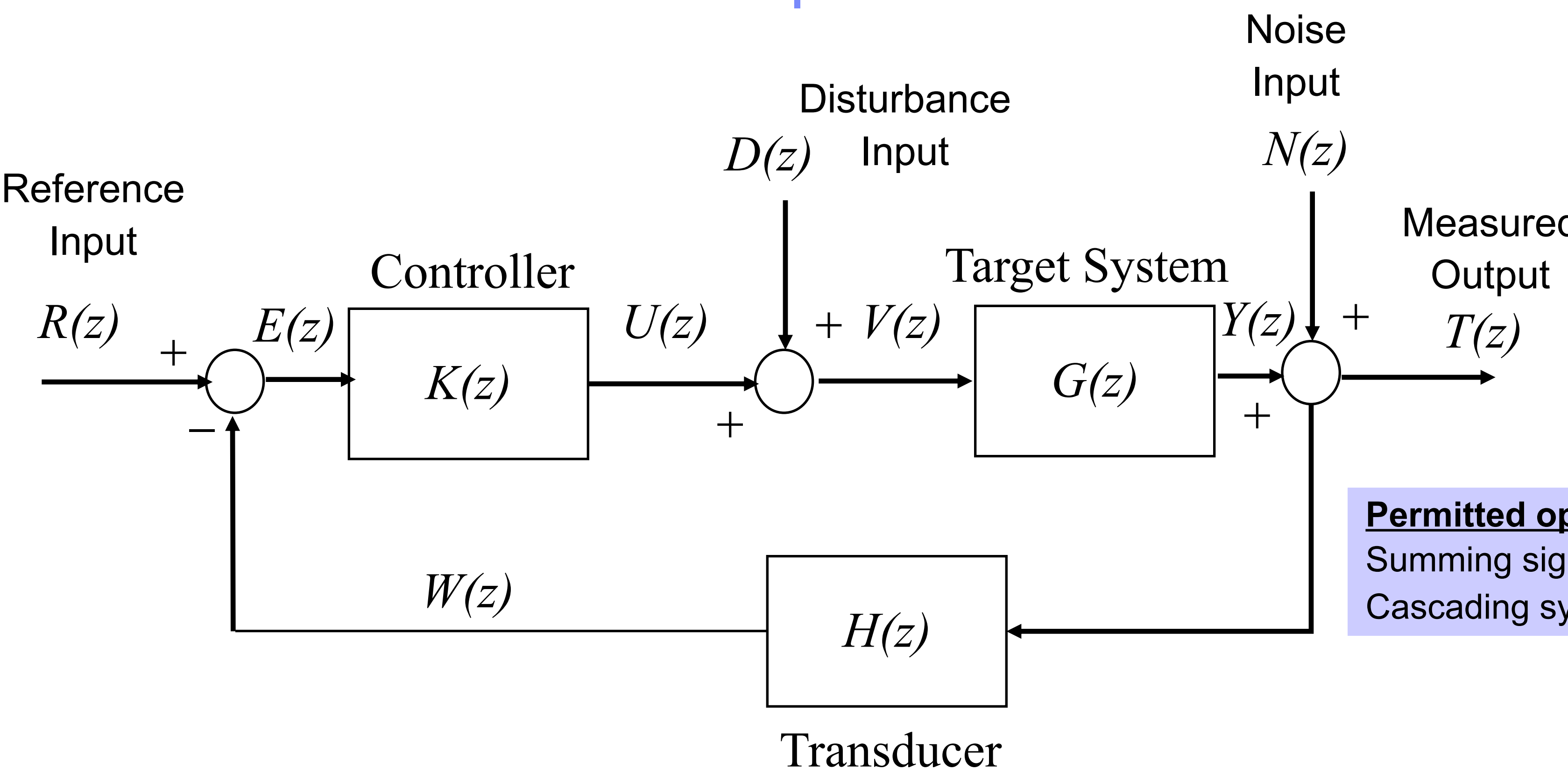
$$y(k+1) = 0.8y(k) + 0.72w(k) - 0.66w(k-1)$$



$$\frac{Y(z)}{U(z)} = \frac{W(z)}{U(z)}\frac{Y(z)}{W(z)} = G(z)H(z)$$

T.F. provide an easy way to analyze the behavior of complex structures.

Canonical Feedback Loop

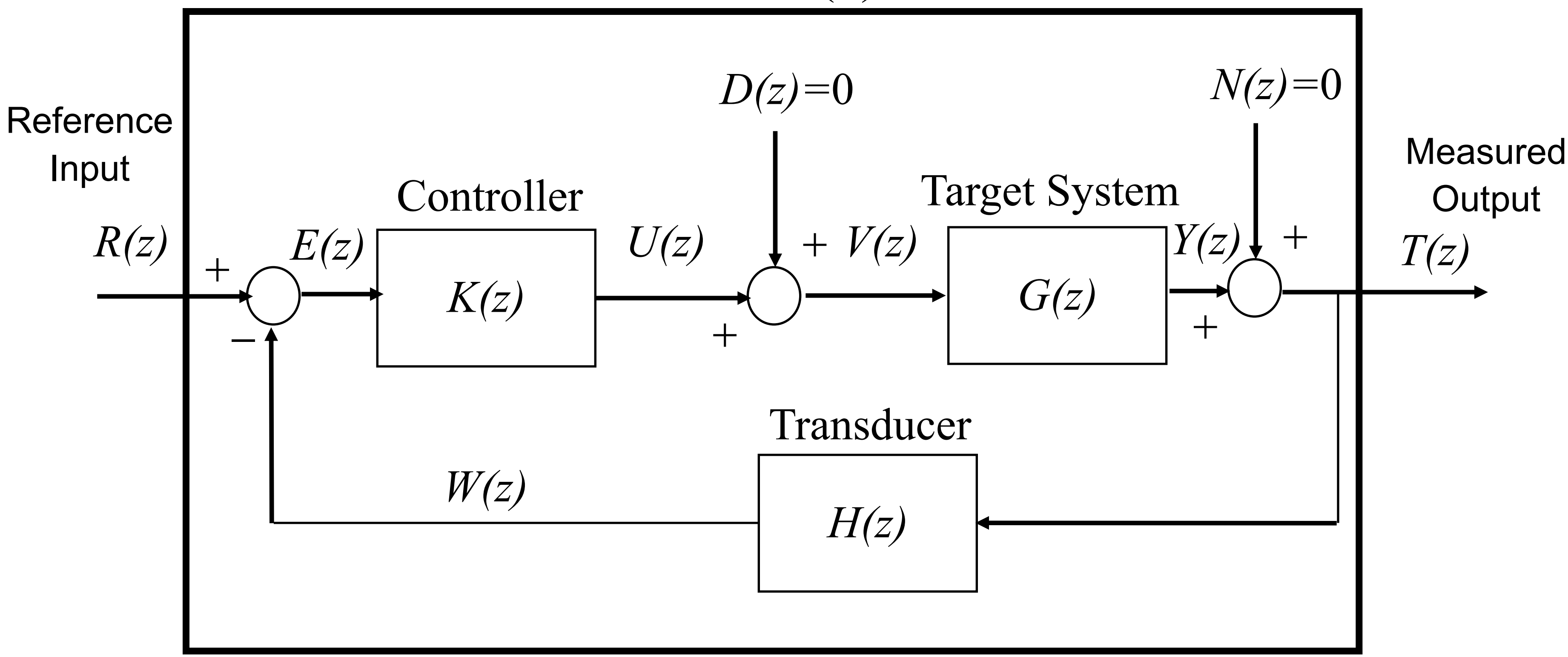


Permitted operations
Summing signals
Cascading systems

Want to analyze characteristics of the entire system: its stability, settling time, and accuracy (ability to achieve the reference input).

It's all done with transfer functions!

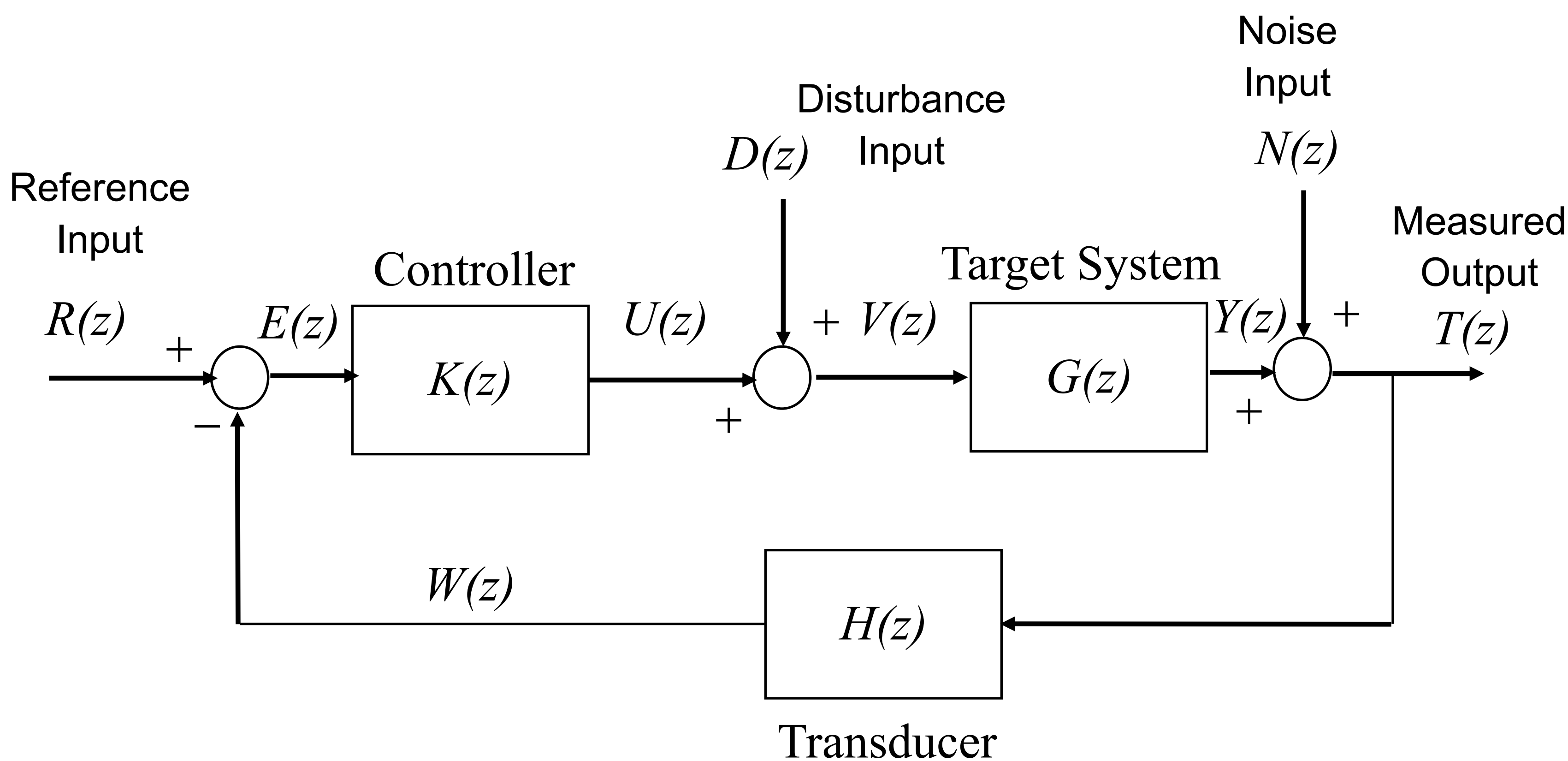
$$F_R(z) = \frac{T(z)}{R(z)}$$



View the dark rectangle as a large transfer function $F_R(z)$ with input $R(z)$ and output $T(z)$.

- System is stable if the largest pole of $F_R(z)$ has an absolute value that is less than 1
- System is accurate if $t(n)=r(n)$ for large n, or $F_R(1)=1$
- System settling time is short if the poles of $F_R(z)$ have a small absolute value
- System has oscillations if there are poles of $F_R(z)$ that are negative or imaginary

Canonical Feedback Loop Has Many T.F.



$F_R(z)$

Transfer function from the reference input to the measured output

$F_D(z)$

Transfer function from the disturbance input to the measured output

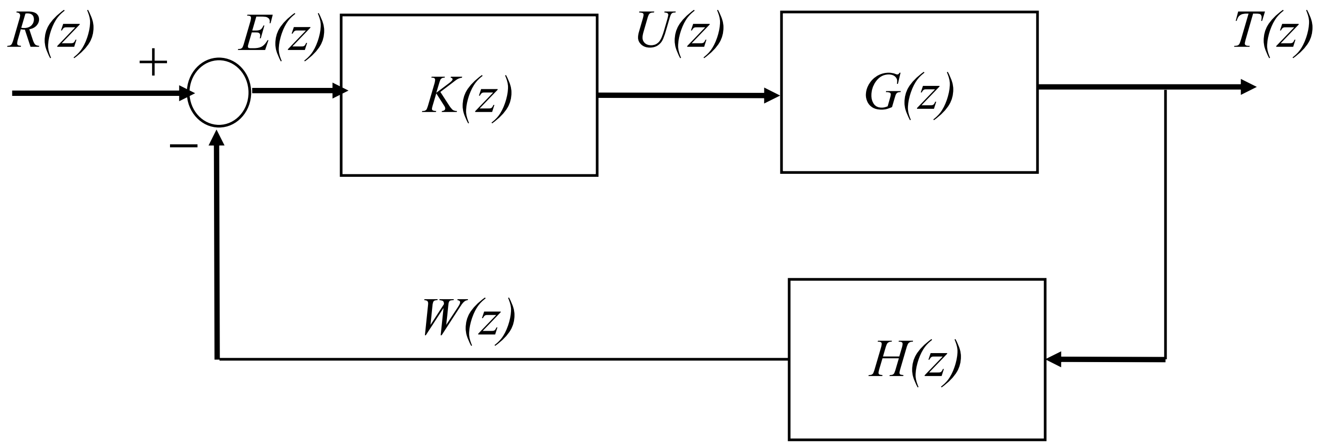
$F_N(z)$

Transfer function from the noise input to the measured output

Computing $F_R(z)$

The only non-zero input is *R(z)*.

Simplified block diagram since *D(z)=0=N(z)*

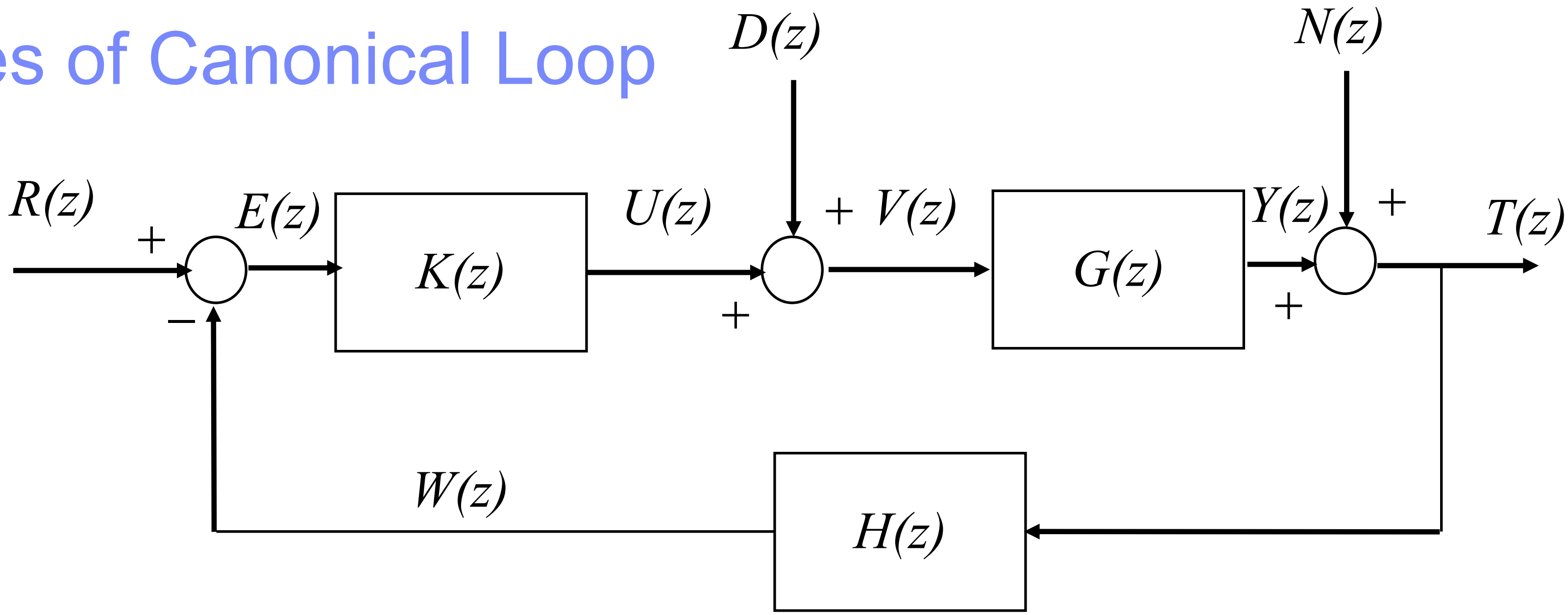


A set of equations relates *R(z)* to *T(z)* based on our previous results

W(z) = *H(z)T(z)* by the definition of a transfer function.
E(z) = *R(z)*-*W(z)* since this is an addition of signals.
T(z) = *E(z)K(z)G(z)* since *K(z)* and *G(z)* are in series.
T(z) = *(R(z)-H(z)T(z))K(z)G(z)* by substitution.

$$F_R(z) = \frac{T(z)}{R(z)} = \frac{K(z)G(z)}{1 + K(z)G(z)H(z)}$$

Properties of Canonical Loop



Reference to Output

$$F_R(z) = \frac{T(z)}{R(z)} = \frac{K(z)G(z)}{1+K(z)G(z)H(z)}$$

Disturbance to Output

$$F_D(z) = \frac{T(z)}{D(z)} = \frac{G(z)}{1+K(z)G(z)H(z)}$$

Noise to Output

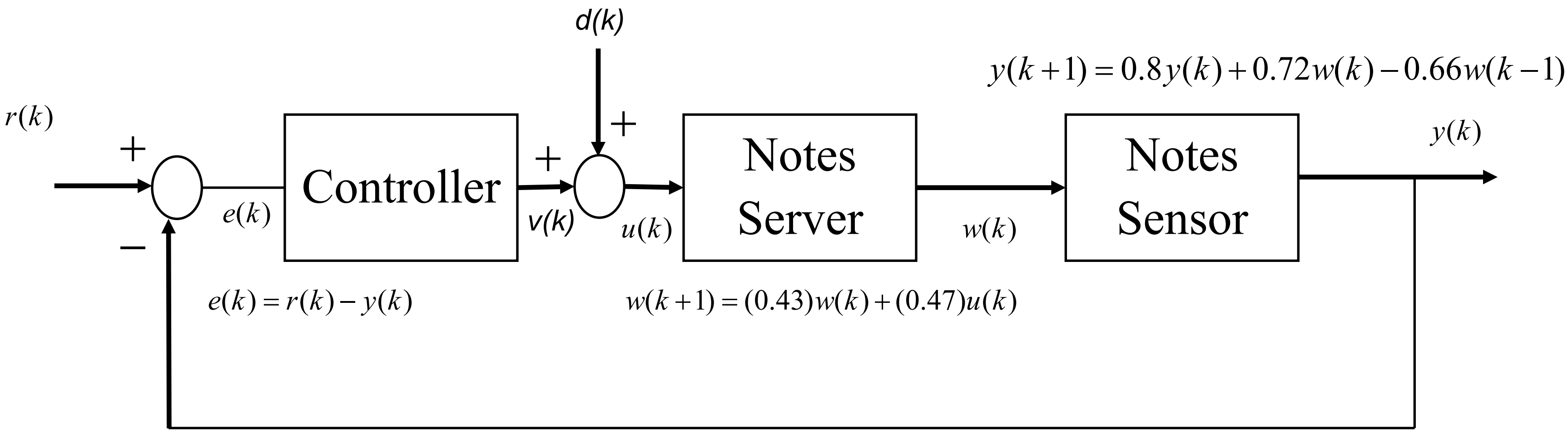
$$F_N(z) = \frac{T(z)}{N(z)} = \frac{1}{1+K(z)G(z)H(z)}$$

What can we say about the stability and settling times of these three transfer functions?
They are the same!

When is the system accurate in the sense that *T(z)=R(z)*? $\boldsymbol{F_R(1)=1}$

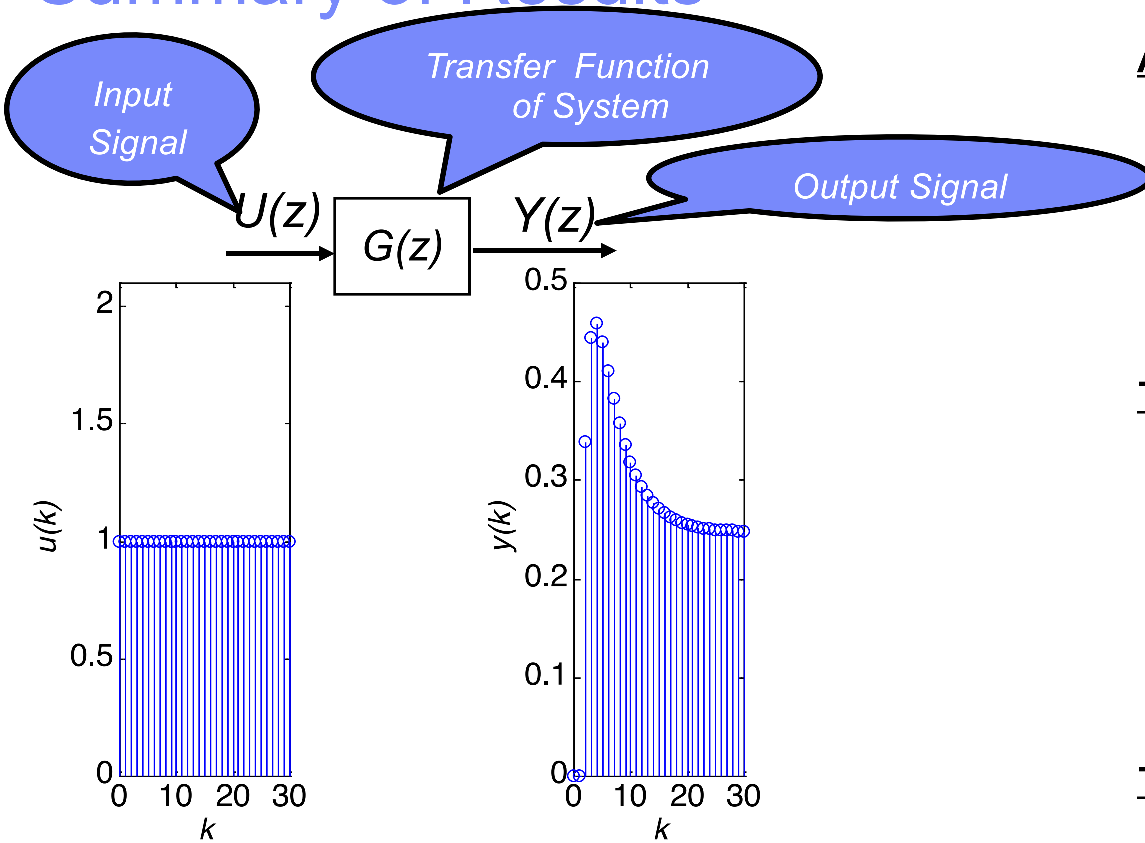
When is the system robust to disturbances and noise? $\boldsymbol{F_D(1)=0=F_N(1)}$

Lab 3: Effect of a Disturbance (Try this later on)



- Model is in file CTShortClass, tab 3 (Notes + Sensor + Disturbance)

Summary of Results

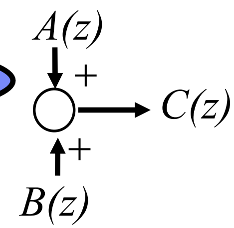


Stable system if |*a*|<*1*, where *a* is the largest pole of *G(z)*

$$\text{Settling time} \approx \frac{-4}{\ln|a|}, \text{ where } |a| \text{ is the largest pole of } G(z)$$

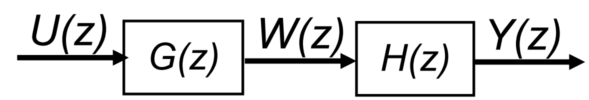
Steady state gain of *G(z)* is $\frac{y(\infty)}{u(\infty)} = G(1)$

Adding signals:

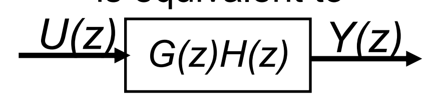


{c(k)=a(k)+b(k)} has Z-Transform *A(z)+B(z)*.

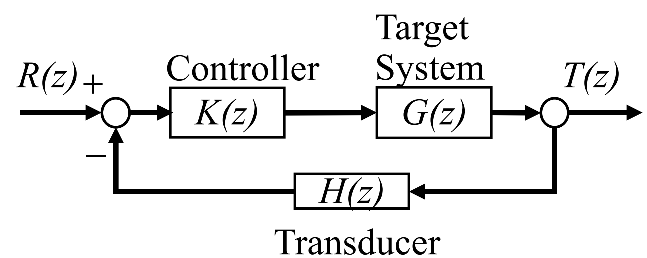
Transfer functions in series



is equivalent to



Transfer function of a feedback loop

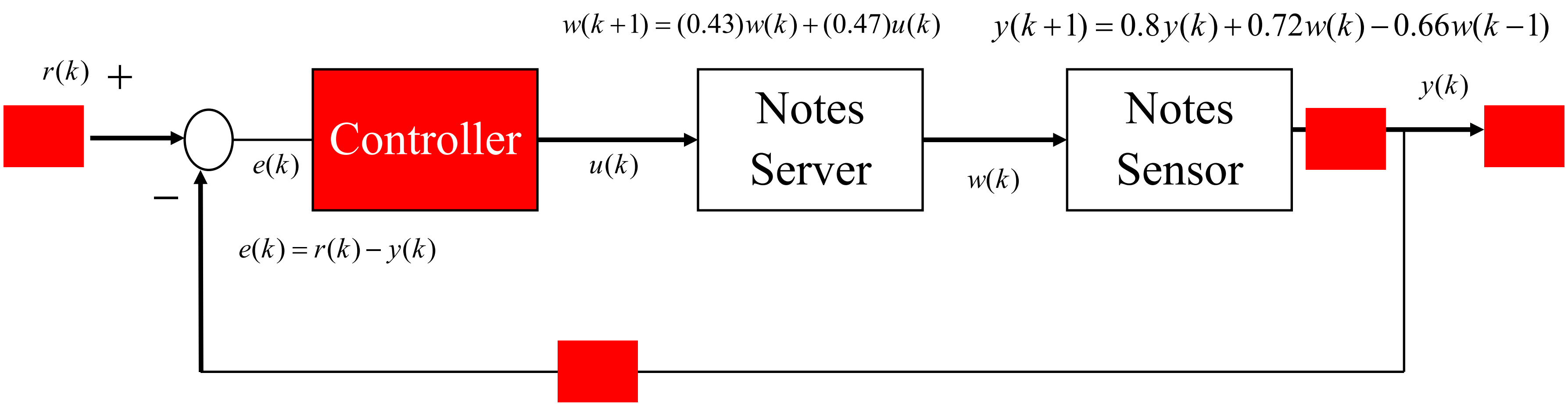


$$F_R(z) = \frac{T(z)}{R(z)} = \frac{K(z)G(z)}{1+H(z)K(z)G(z)}$$

M3 –
Control Analysis

Reference: “Feedback Control of Computer Systems”, Chapters 8,9.

Motivating Example



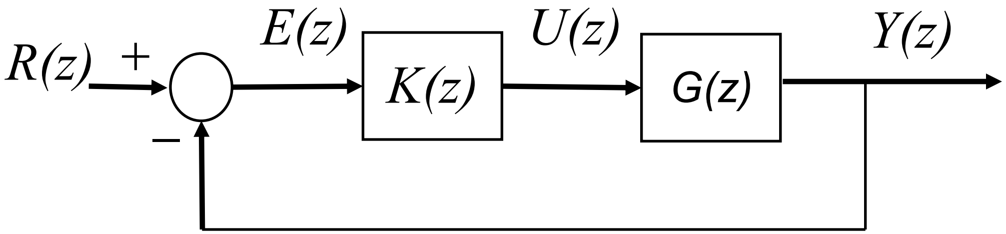
The problem

Design a control system that is stable, accurate, settles quickly, and has small overshoot.

Take a holistic approach

Design a control system, not just a controller

Basic Controllers



Proportional (P) Control

$$u(k) = K_P e(k)$$

$$K(z) = \frac{U(z)}{E(z)} = K_P$$

Integral (I) Control

$$u(k+1) = u(k) + K_I e(k+1)$$

$$zU(z) = U(z) + K_I z E(z)$$

$$K(z) = K_I \frac{z}{z-1}$$

K_P and K_I are called ***control gains***.

Summary of Lab 2: P vs. I Control

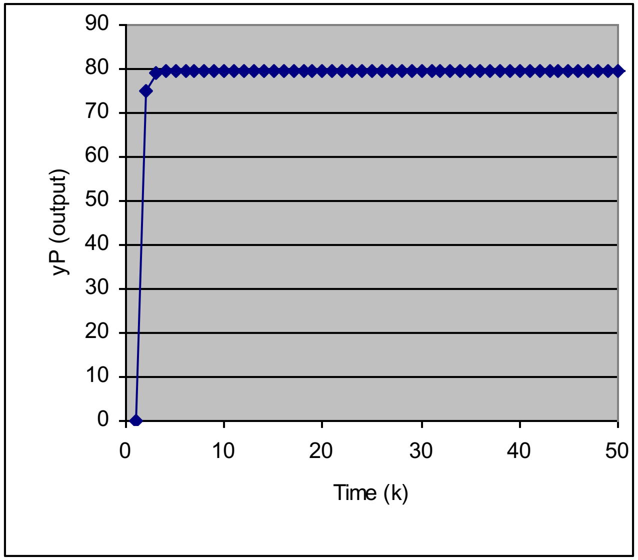
Proportional (P) Control $K(z) = K_P$

eP(k) = r(k)-yP(k)

uP(k)=KP*eP(k)

yP(k+1)=y_coef(1)*yP(k)+y_coef(2)*uP(k)

k	r(k)	eP(k)	uP(k)	yP(k)	KP
0	200	200	160	0	0.8
1	200	124.8	99.84	75.2	

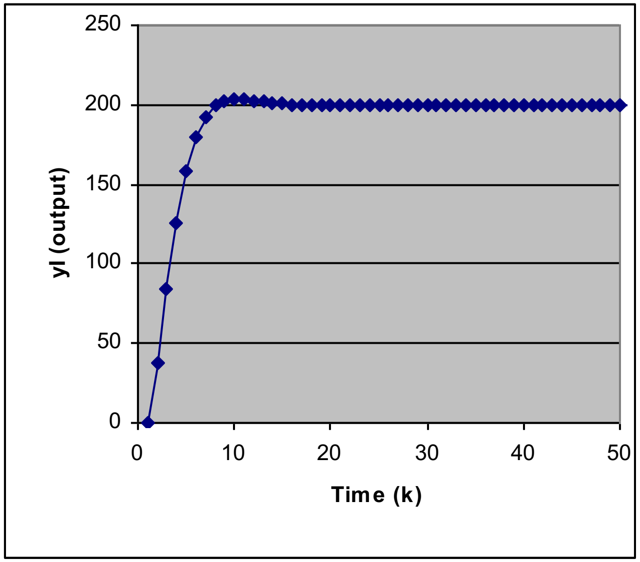


Integral (I) Control $K(z) = \frac{K_I z}{z-1}$

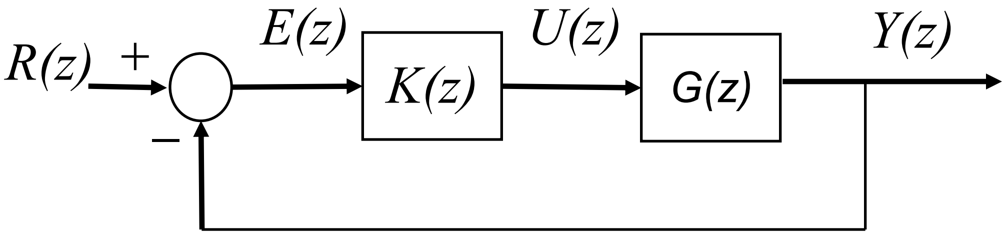
eI(k) = r(k)-yI(k)

uI(k)=uI(k-1)+KI*eI(k)
yI(k+1)=y_coef(1)*yI(k)+y_coef(2)*uI(k)

eI(k)	uI(k)	yI(k)	KI
200	80	0	0.4
162.4	144.96	37.6	



Analysis



$$F_R(z) = \frac{Y(z)}{R(z)} = \frac{K(z)G(z)}{1+K(z)G(z)}$$

$$F_R^P(z) = \frac{Y(z)}{R(z)} = \frac{K_P \frac{0.47}{z-0.43}}{1+K_P \frac{0.47}{z-0.43}} = \frac{K_P}{z-0.43+0.47K_P}$$

$$p_P = 0.43 - 0.47K_P$$

$$F_R^I(z) = \frac{Y(z)}{R(z)} = \frac{K_I \frac{z}{z-1}\frac{0.47}{z-0.43}}{1+K_I \frac{z}{z-1}\frac{0.47}{z-0.43}}$$

$$= \frac{0.47K_I z}{(z-1)(z-0.43)+0.47K_I z}$$

$$= \frac{0.47K_I z}{z^2+(0.47K_I-1.43)z+0.43}$$

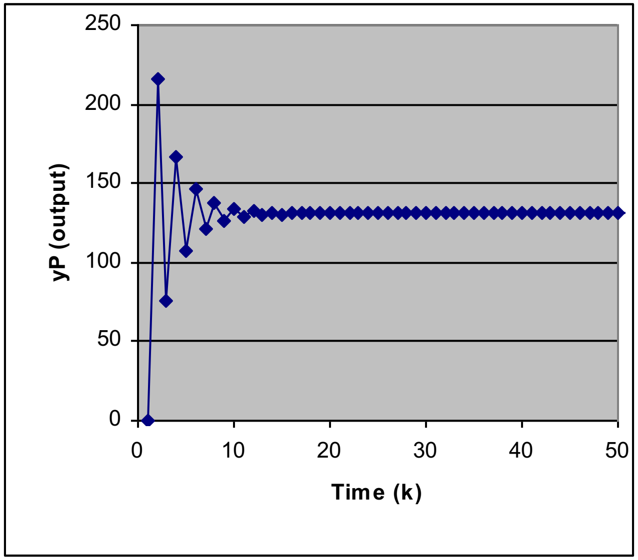
$$p_I = \frac{1.43-0.47K_I \pm \sqrt{(0.47K_I-1.43)^2-1.72}}{2}$$

<u>Settling Times, Steady State Gains</u>

Ctrl Gain	P	I
0.1	5, 0.076	43, 1
0.4	3, 0.25	10, 1
3.0	198, 0.71	10, 1

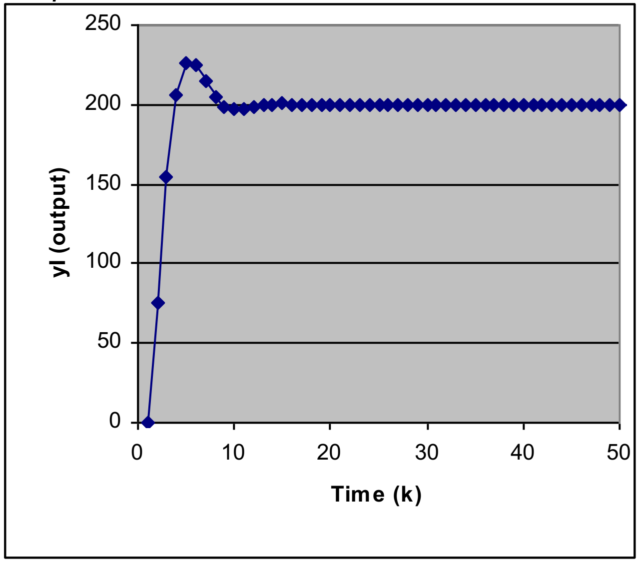
Conclusions from P vs. I Comparison

K_P=2.3



$r(k)$=200

K_I=0.8



Conclusions:

P is fast

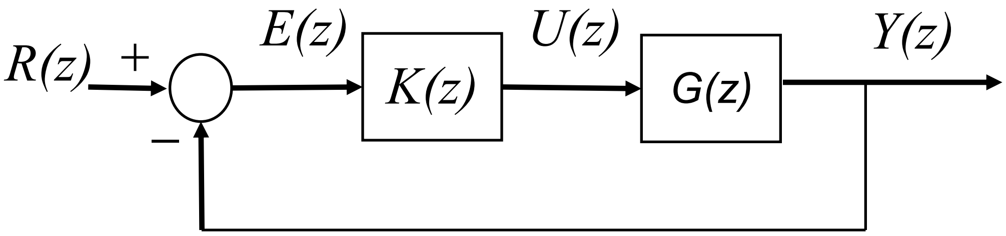
I is accurate and has less overshoot.

Design challenge:

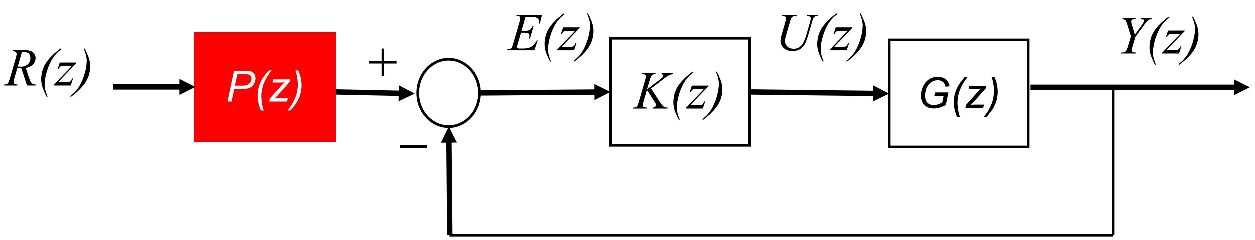
Make P accurate.

Reduce P's overshoot.

Making P Control Accurate



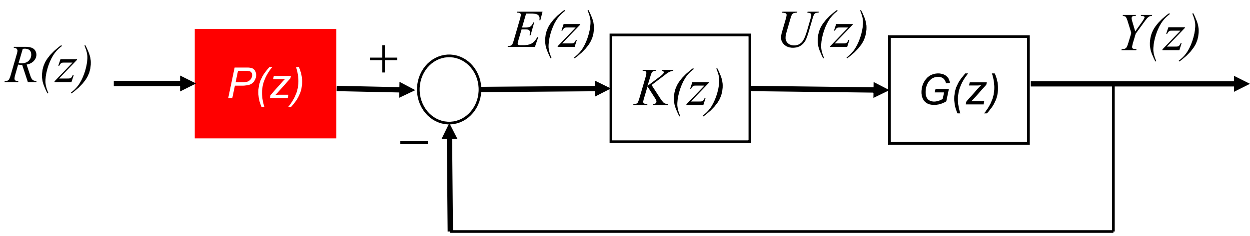
Precompensation: Adjusts the reference input so that the right output is obtained.



Lab 4: Precompensation

- Modify P control to include pre-compensation
- Find a value for the precompensator that makes P control accurate
  - Trial and error
  - Adjust based on ratio between reference and output
- What happens if the reference input changes? What if the control gain changes?
- What is the general rule for the value of the precompensator?

Computing Value of Precompensator



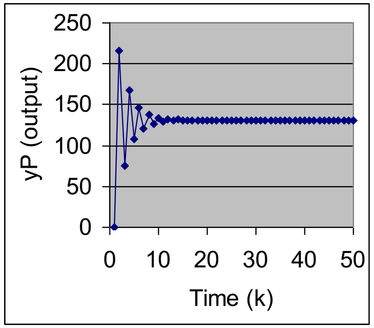
Want $R(1)P(1)F_R(1) = R(1)$

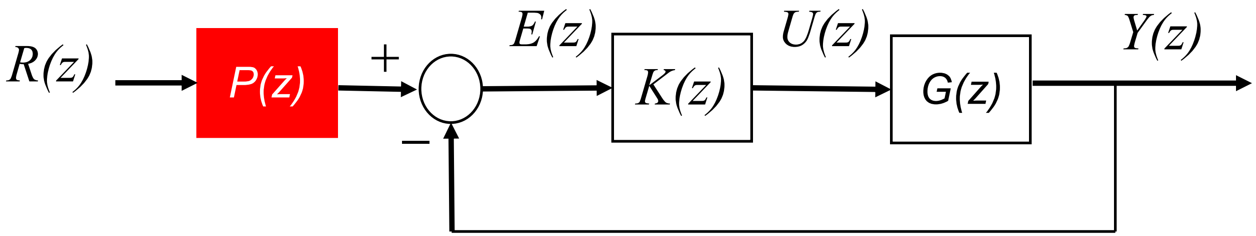
So $P(1) = \dfrac{1}{F_R(1)} = \dfrac{1 - 0.43 + 0.47K_P}{0.47K_P}$

Consider $K_P = 2.3, R(z) = 200$; then $P(z) = 1.53$

Try on spreadsheet. See if it works for other reference inputs.

Reducing P's Overshoot

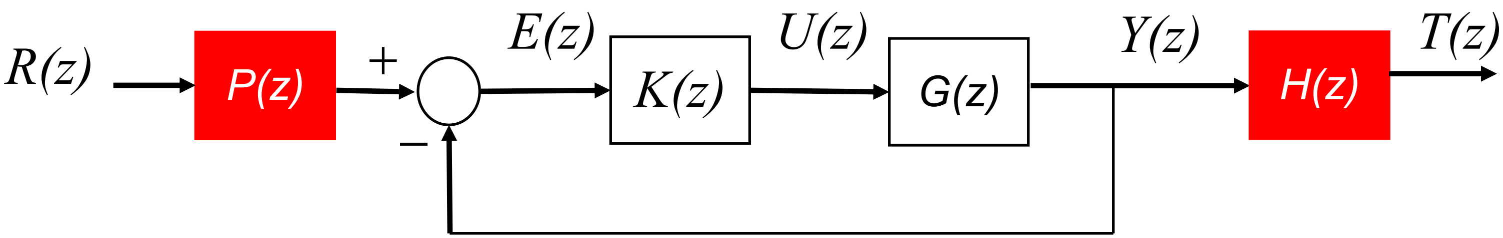




Filter: Smooths values over time.

c – Weight past history (make it smoother)

$t(k+1)=ct(k)+y(k+1)$

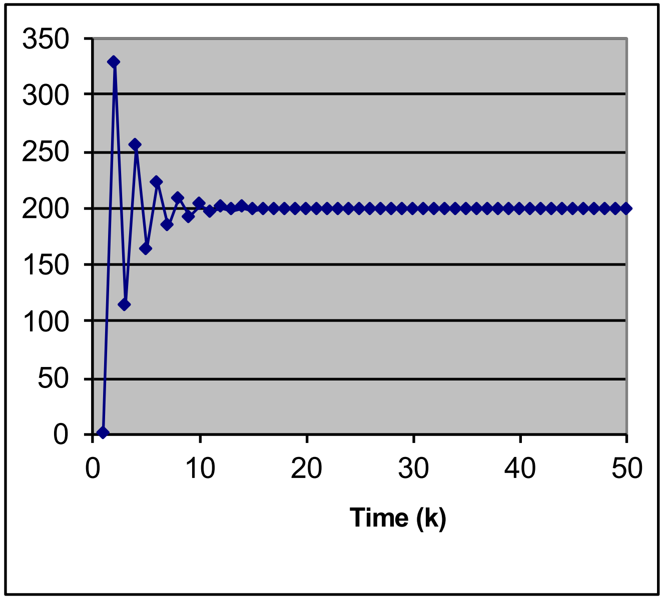


Lab 5: Precompensation + Filter

- Add a filter to precompensated P control
- What values of *c* produce smooth *t*(*k*)?
- What are the other effects of the filter?

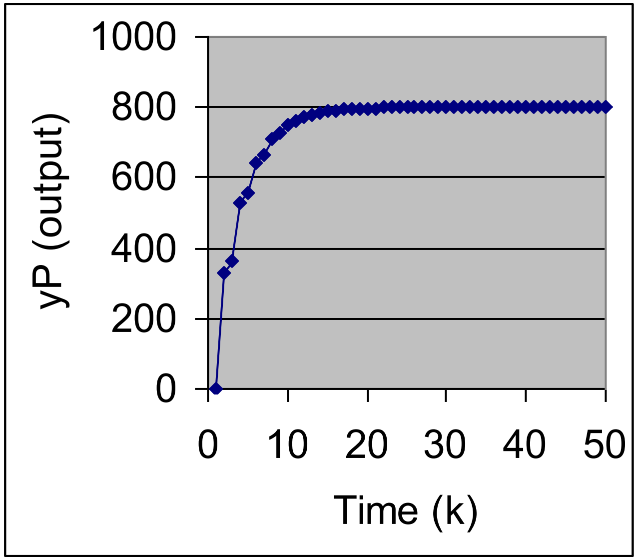
Results of Filter Design

w/o filter



$r(k)$=200

with filter: c = 0.75

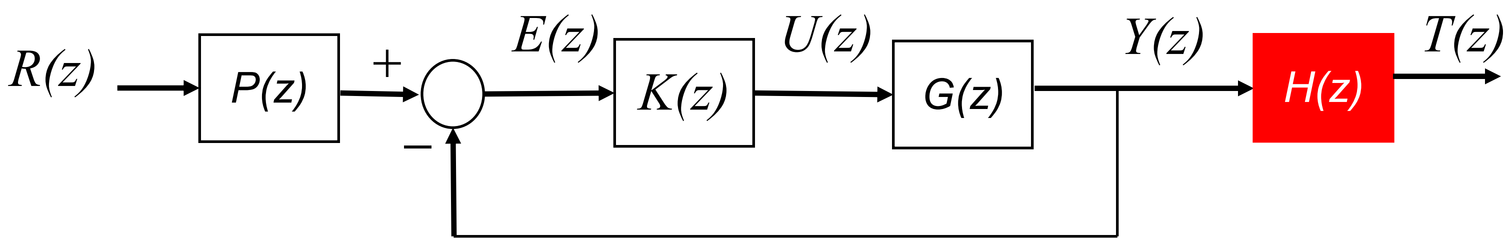


The good news about the filter: Can eliminate overshoot

The bad news: Inaccurate and slower.

Why inaccurate?

Analysis of the Filter



Analysis 1: Why does *H(z)* cause the system to be inaccurate?

Want $P(1)F_R(1)H(1)=1$

We have designed $P(z)$ so that $P(1)F_R(1)=1$. So, it must be that $H(1)\neq 1$.

$$t(k+1)=ct(k)+y(k+1)$$

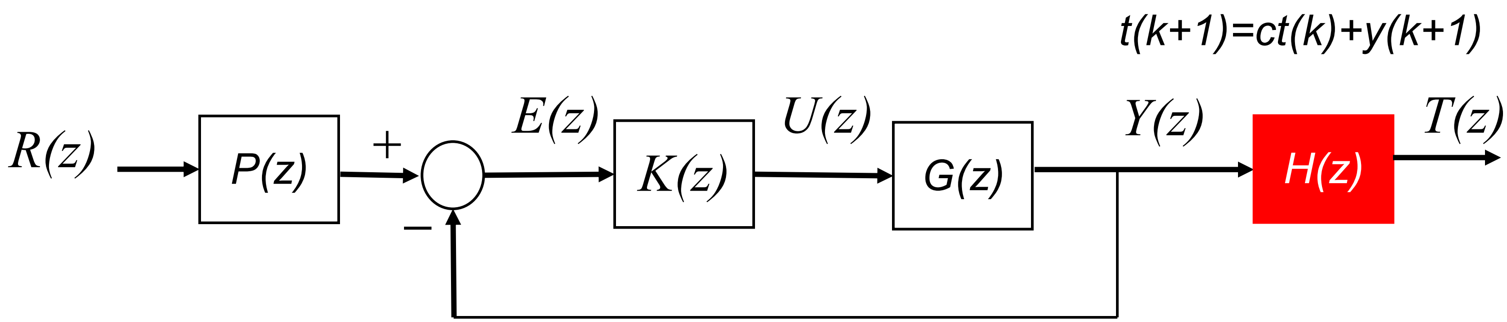
$$zT(z)=cT(z)+zY(z)$$

$$H(z)=\frac{T(z)}{Y(z)}=\frac{z}{z-c}$$

$$H(1)=\frac{1}{1-c}$$

Check the spreadsheet.

Designing a Normalized Filter



Want $H(1) = 1$

Can do this by dividing by multiplying by $1 - c$.

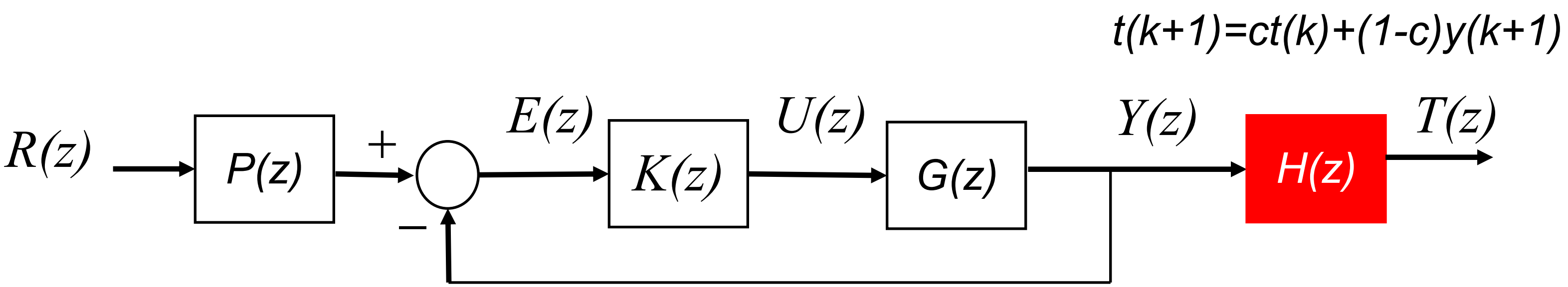
That is, use $H(z) = \dfrac{z(1-c)}{z-c}$

Converting this into a time series model, we have

$t(k+1) = ct(k) + (1-c)y(k+1)$

Check the spreadsheet: Lab 6.

Analysis of the Filter



Analysis 2: Why does *H(z)* cause the system to be slower?

What are the poles of $P(z)F_R(z)H(z)$?

Let $p = \max_{poles}\{P(z), F_R(z), H(z)\}$

$P(z)$ has no poles

$$F_R(z) = \frac{0.47K_P}{z - 0.43 + 0.47K_P}$$

If $K_P = 2.3$, $F_R(z) = \dfrac{1.1}{z - 0.65}$

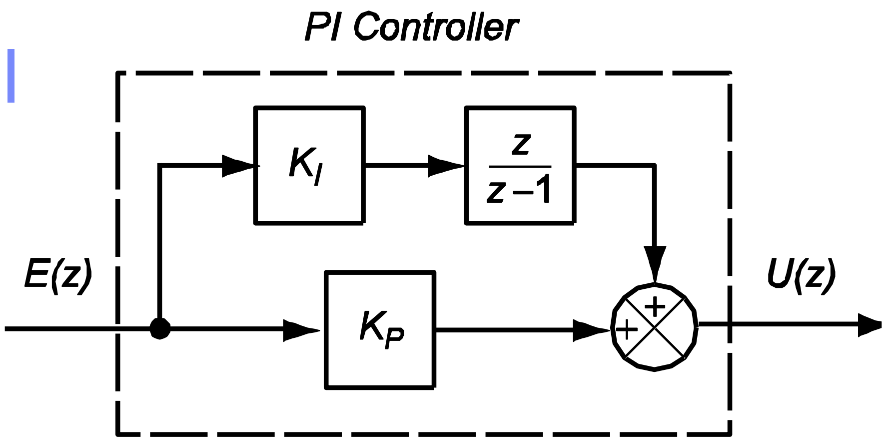
$$H(z) = \frac{T(z)}{Y(z)} = \frac{1-c}{z-c}$$

If $c = 0.75$, then there is a pole at 0.75.

So, the filter adds a closed loop pole at *c*.

Check the spreadsheet.

PI Control



$$u(k) = u_P(k) + u_I(k)$$

$$K(z) = \frac{U(z)}{E(z)} = K_P(z) + K_I(z)$$

$$= K_P + \frac{K_I z}{z-1}$$

$$= \frac{(K_P + K_I)z - K_P}{z-1}$$

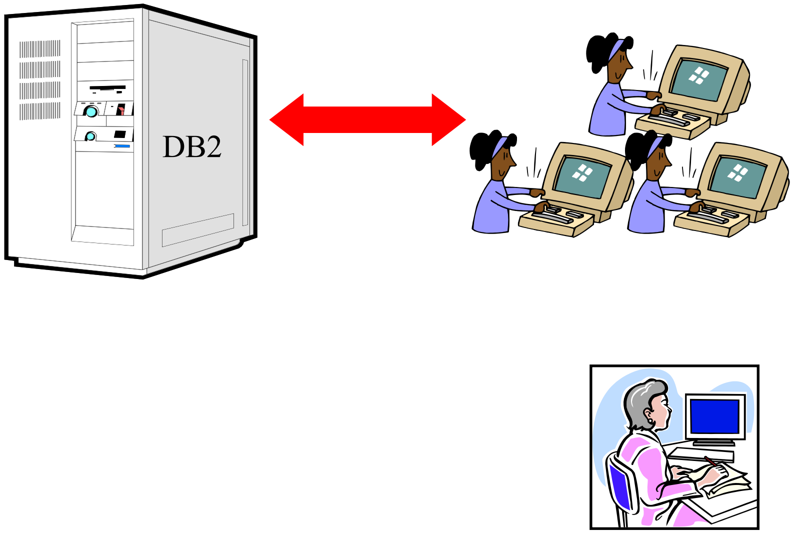
$$u(k) = u(k-1) + (K_P + K_I)e(k) - K_P e(k-1)$$

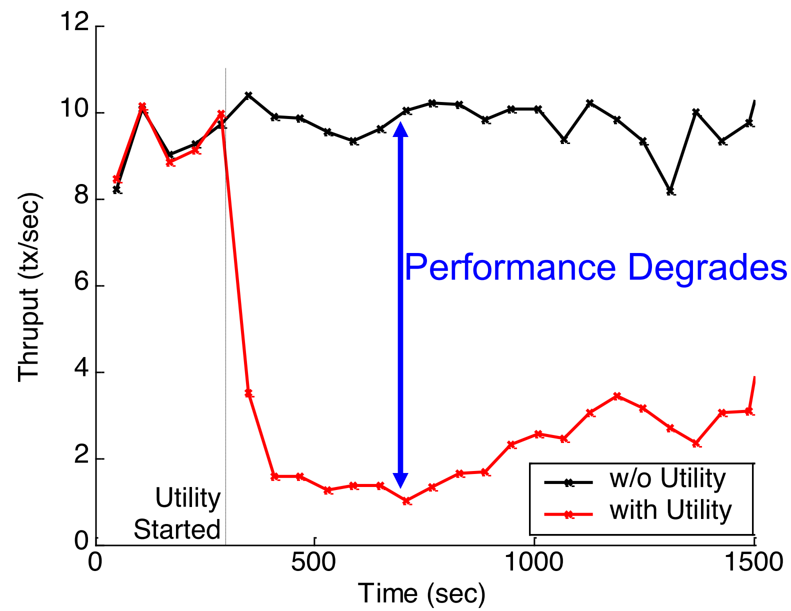
Lab 7: PI Control

M4 – Applications

- DB2 Utilities Throttling (DB2 v8.1)
  - http://www.databasejournal.com/features/db2/article.php/3339041
  - "Throttling Utilities in the IBM DB2 Universal Database Server," Sujay Parekh, Kevin Rose, Yixin Diao, Victor Chang, Joseph L. Hellerstein, Sam Lightstone, Matthew Huras. American Control Conference, 2004.
- Self-tuning memory management
  - "Using MIMO Linear Control for Load Balancing in Computing Systems," Yixin Diao, Joseph L. Hellerstein, Adam Storm, Maheswaran Surendra, Sam Lightstone, Sujay Parekh, and Christian Garcia-Arellano. American Control Conference, 2004.
  - "Incorporating Cost of Control Into the Design of a Load Balancing Controller," Yixin Diao, Joseph L. Hellerstein, Adam Storm, Maheswaran Surendra, Sam Lightstone, Sujay Parekh, and Christian Garcia-Arellano. Real-Time and Embedded Technology and Application Systems Symposium, 2004.

The Utilities Throttling Problem



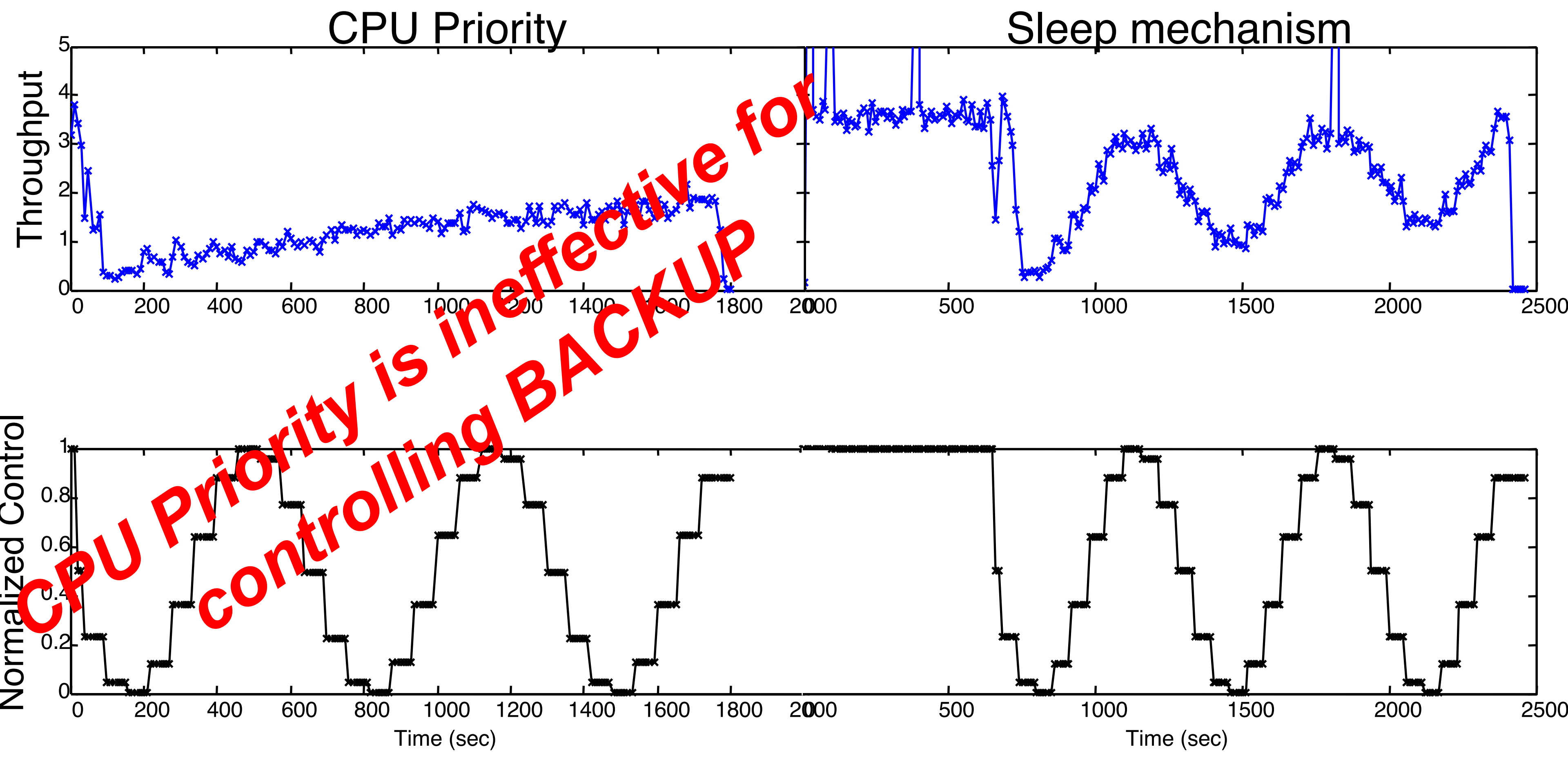


Utilities have a big impact on production performance.

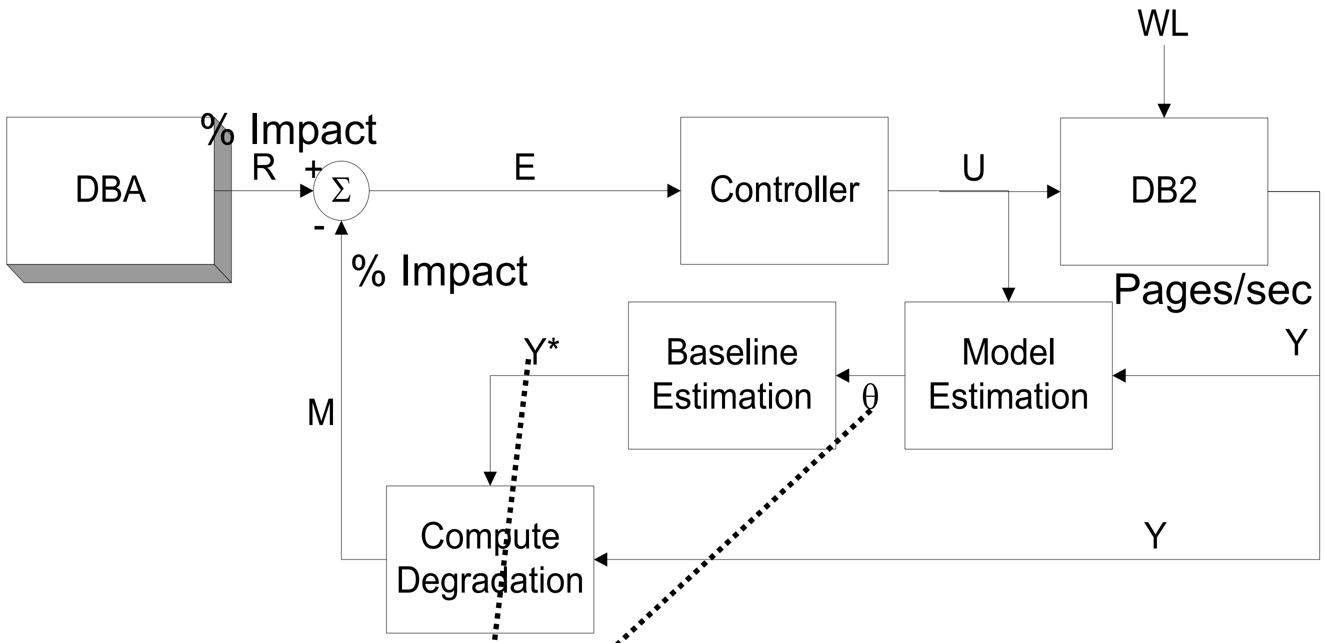
<u>Administrative policy</u>

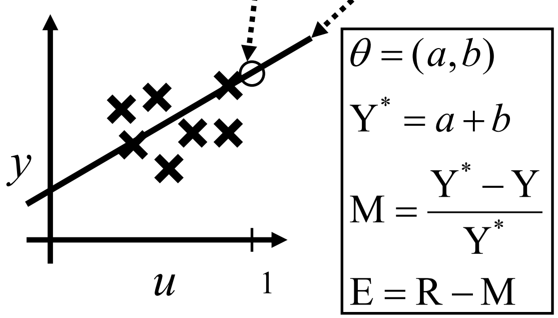
There should be no more than an x% performance degradation of production work as a result of executing administrative utilities

Effector: Priority vs Self-Induced Sleep (SIS)



Block Diagram





R: Impact Limit

E: Error

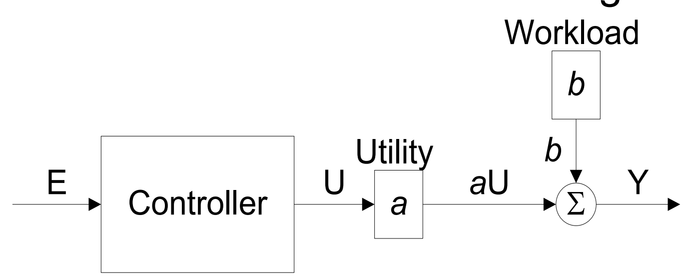
U: Sleep %

Y: Pageometer (pages/sec)

θ: Model parameters

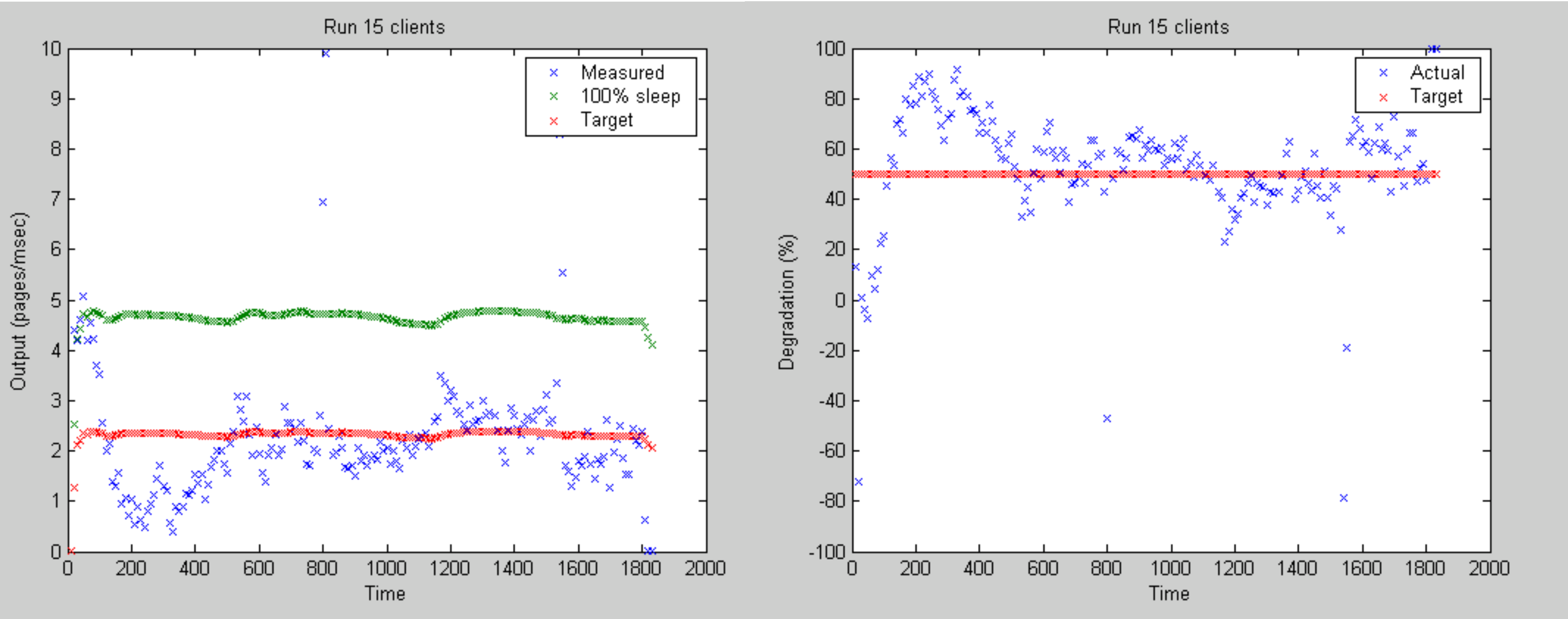
Y*: Baseline perf

Assume linear effect of throttling on Y

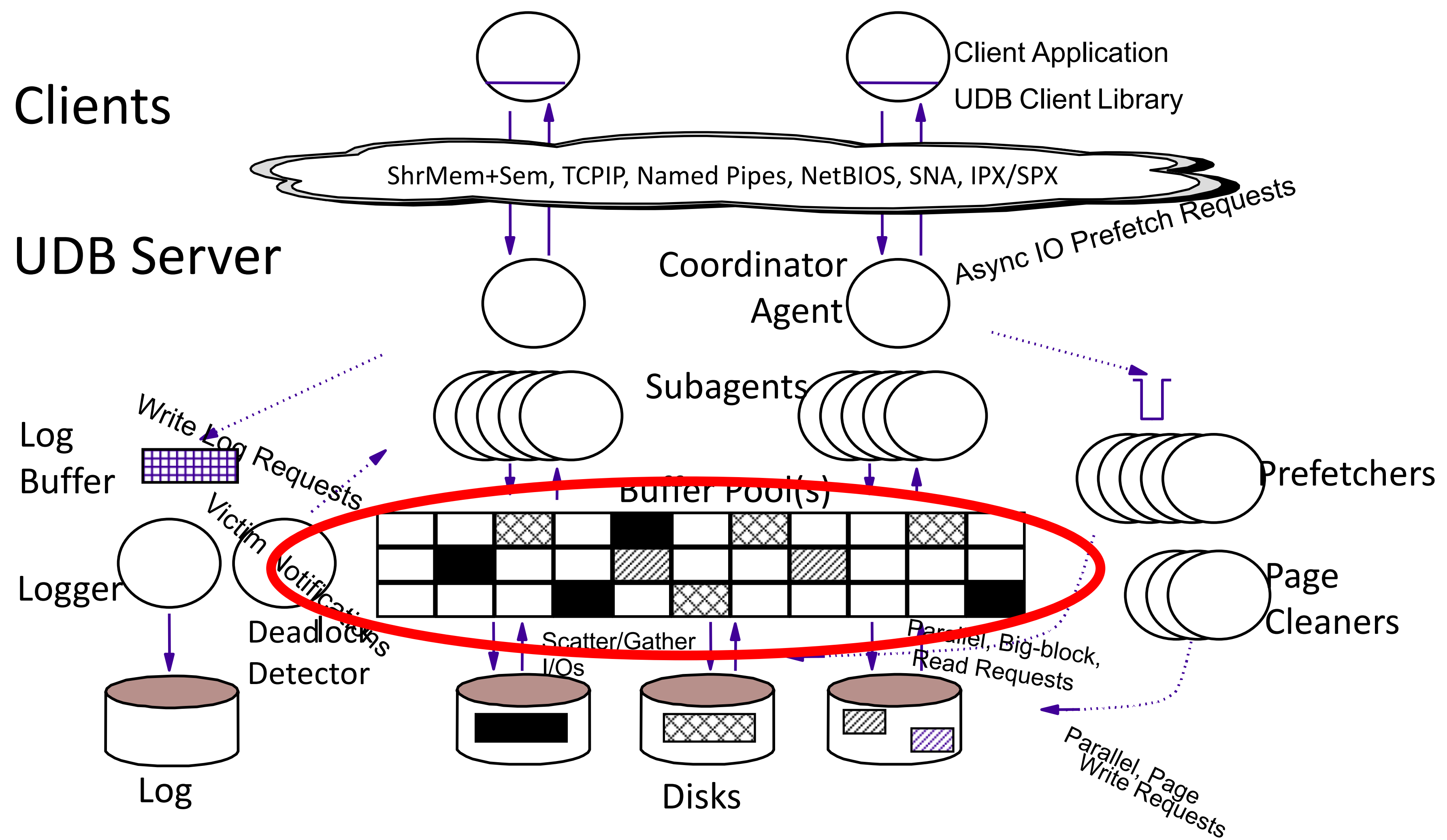


Online Modeling to translate from Pages/sec (Y) to % Impact (M)

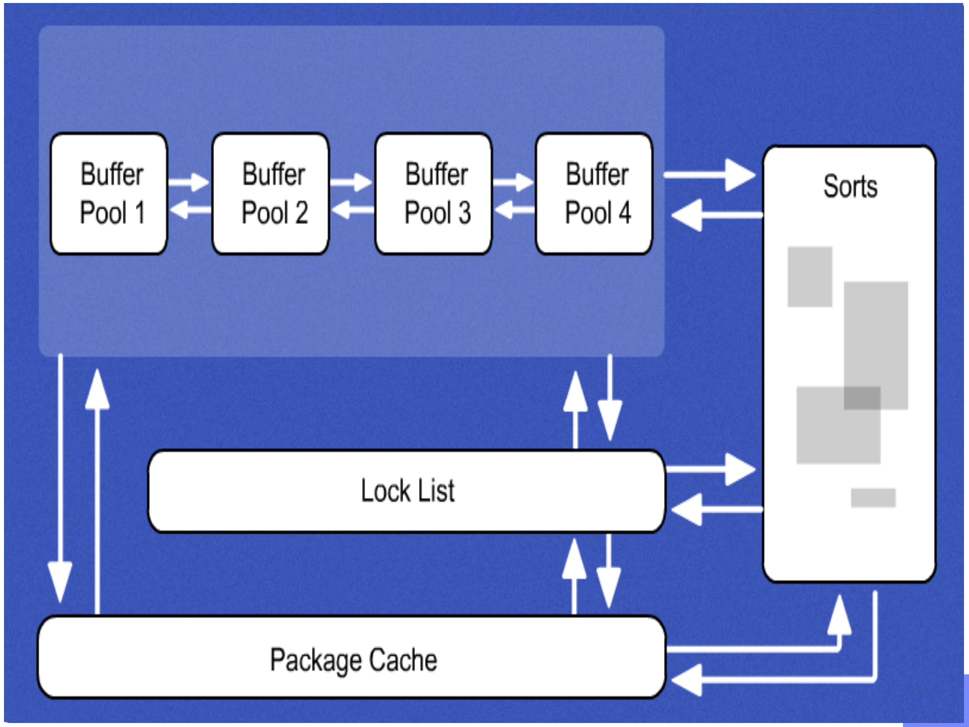
Controller Evaluation



System Description: DB/2 UDB – Self-Tuning Memory

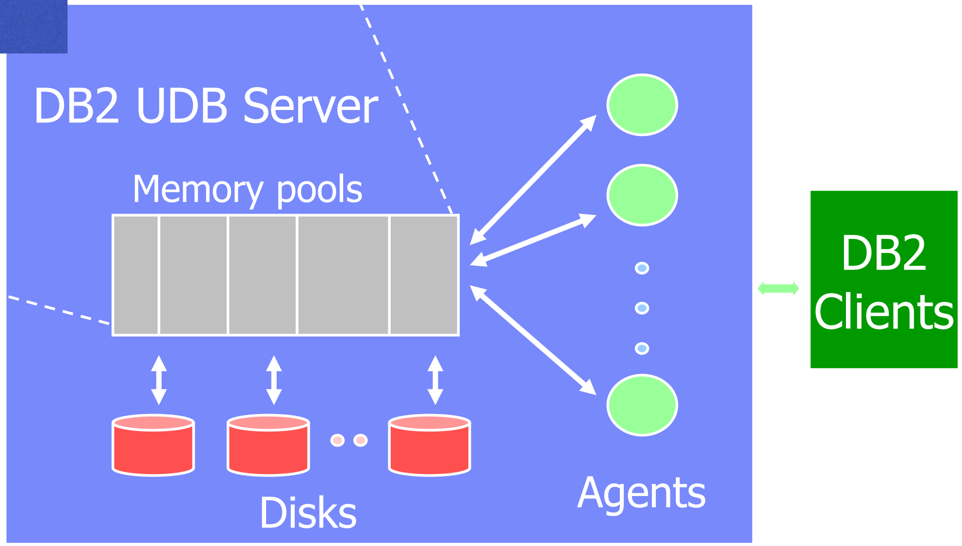


Database Memory Management

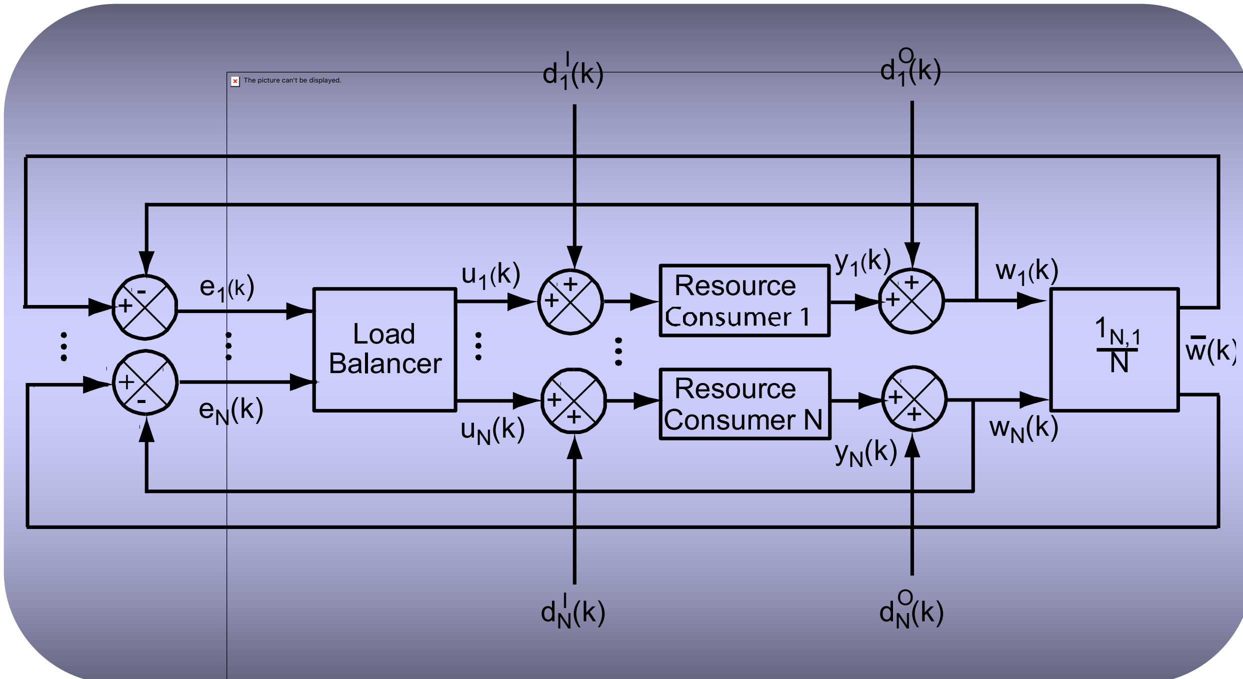


Memory management is key to database performance.

Optimize performance by equalizing loads across the memory pools.



MIMO Controller Design: Linear Quadratic Regulation



$$\min J = q\sum_{k=1}^{\infty}\sum_{i=1}^{N} e_i^2(k) + q\sum_{k=1}^{\infty}\sum_{i=1}^{N} e_{I,i}^2(k) + r\sum_{k=1}^{\infty}\sum_{i=1}^{N} u_i^2(k)$$

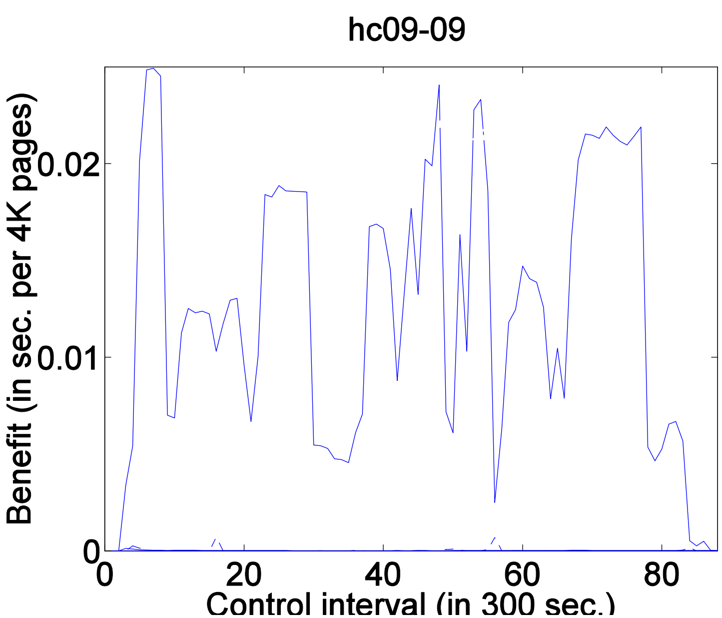
Cost of load imbalance

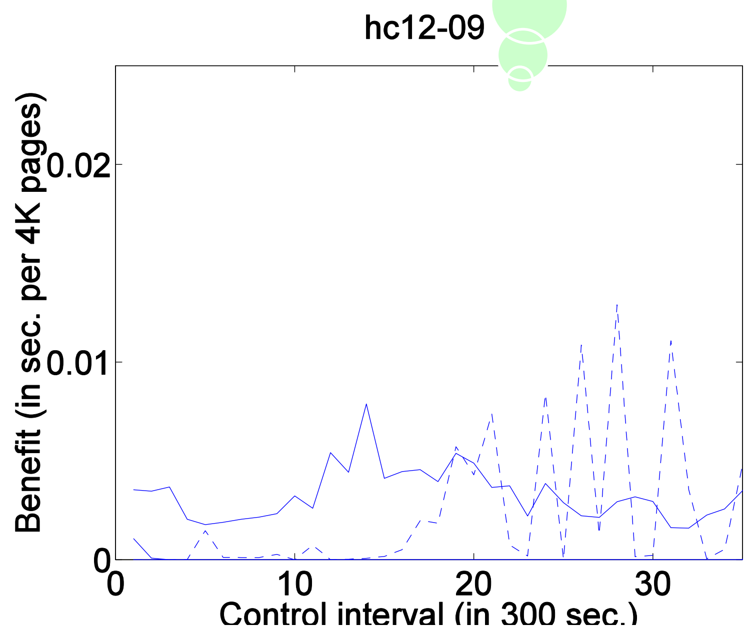
Cost of control

Measured Benefit of Load Balancing

- Decision support workload
  - Long running transactions
  - Resource requirements vary over time
- Study effect on total query response time (T_S) in TPC-H.

59% Reduction in Total RT

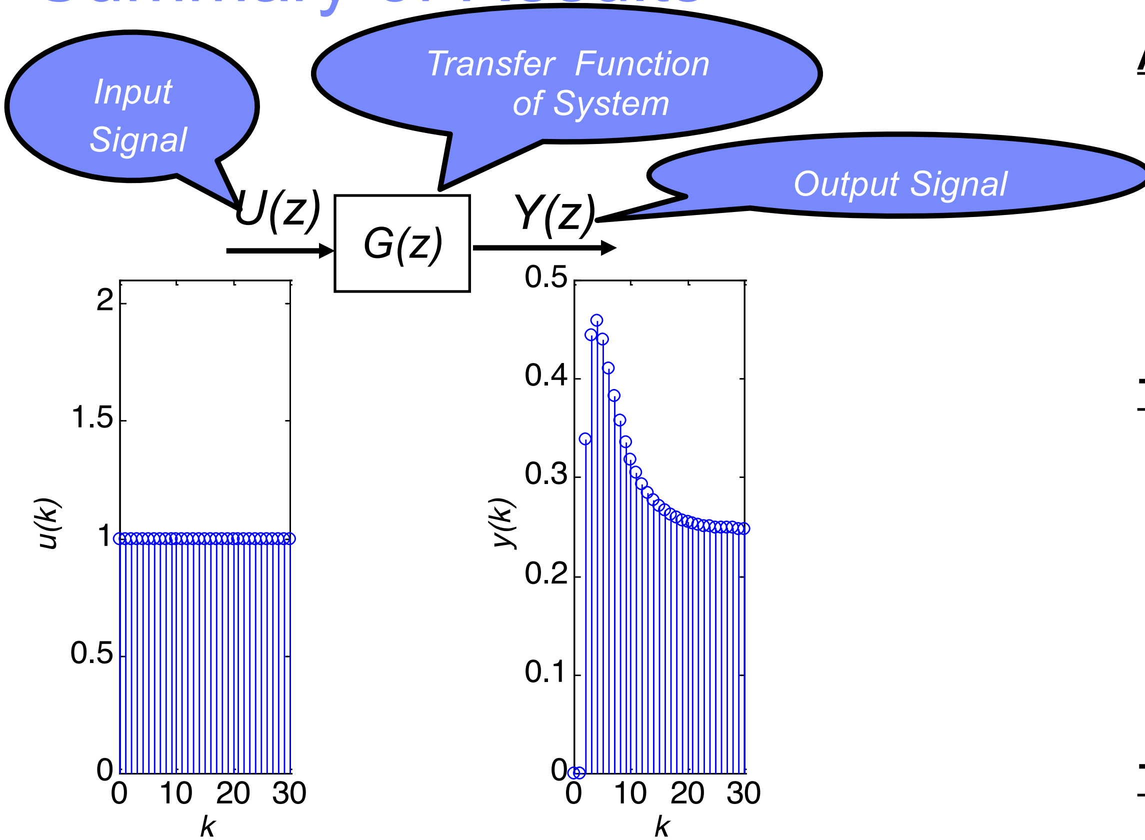




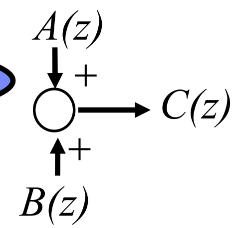
Summary

- Control systems consists of elements
  - Controller, target system, transducer, filter, adapter, …
- Control objectives for computing systems focus on
  - SASO: Stability, accuracy, settling time, overshoot
- Classical control theory builds on linear system theory
  - Signals, transfer functions, composition of systems, use of *z*-transform to encode time related information
- Control analysis involves
  - Constructing ARX models for components
  - Translating these models into the z-domain (transfer functions)
  - Using composition of systems to find the end-to-end transfer functions of interest
  - Analyzing the SASO properties of these systems
- These simple models and analyses have had significant practical at IBM
  - Regulating the execution of administrative utilities
  - Self-tuning memory management

Summary of Results

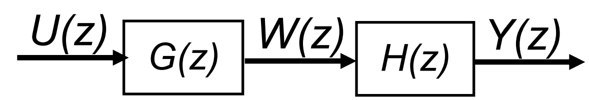


Adding signals:

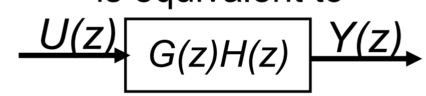


{c(k)=a(k)+b(k)} has Z-Transform *A(z)+B(z)*.

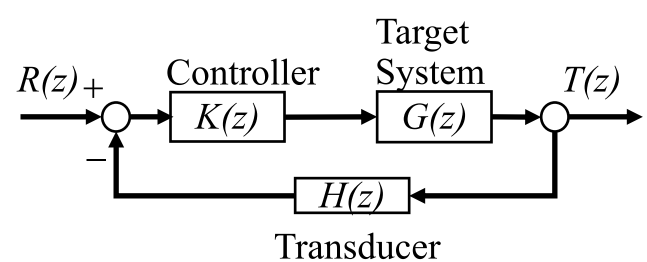
Transfer functions in series



is equivalent to



Transfer function of a feedback loop



Stable system if $|a|<1$, where a is the largest pole of $G(z)$

$$\text{Settling time} \approx \frac{-4}{\ln|a|}, \text{ where } |a| \text{ is the largest pole of } G(z)$$

Steady state gain of $G(z)$ is $\dfrac{y(\infty)}{u(\infty)} = G(1)$

$$F_R(z) = \frac{T(z)}{R(z)} = \frac{K(z)G(z)}{1+H(z)K(z)G(z)}$$

Bibliography – Control Theory Textbooks

- Modern Control Engineering, K Ogata, Prentice Hall, 1996.
- Feedback Control of Dynamic Systems, GF Franklin, JD Powell, and A Emami-Naeini, 1993.
- Discrete-Time Control Systems, K Ogata, Prentice Hall, 1995.
- Adaptive Control, KJ Astrom and B Wittenmark, Addison Wesley, 1995.
- Applied Nonlinear Control, JJE Slotine and W Li, Prentice Hall, 1991.
- Robust Adaptive Control, PA Ioannou and J Sun, Prentice Hall, 1996.
- Introduction to Stochastic Control Theory, KJ Astrom, Academic Press, 1970.
- Fuzzy Control, KM Passino and S Yurkovich, Addison Wesley Longman, 1998.
- Feedback Control of Computing Systems, JL Hellerstein, Y Diao, S Parekh, and DM Tilbury, Wiley, 2004.

Bibliography – Starter Set of Articles

- CV Hollot, V Misra, D Towsley, and WB Gong. A control theoretic analysis of RED. IEEE INFOCOM'01, 2001.
- Y Lu, A Saxena, and TF Abdelzaher. Differentiated caching services: a control-theoretic approach. ICDS, 2001.
- S Parekh, N Gandhi, J Hellerstein, D Tilbury, J Bigus, and TS Jayram. Using control theory to achieve service level objectives in performance management. Real-Time Systems Journal, 23:127-141, 2002.
- A Robertsson, B Wittenmark, and M Kihl. Analysis and design of admission control in Web-server systems. American Control Conference, 2003.
- L Sha, X Liu, Y Lu, and T Abdelzaher. Queueing model based network server performance control. IEEE RealTime Systems Symposium, 2002.
- CE Rohrs, RA Berry, and SJ O'Halek. A control engineer's look at ATM congestion avoidance. GLOBECOM, 1995.
- Yixin Diao, Joseph L. Hellerstein, Adam Storm, Maheswaran Surendra, Sam Lightstone, Sujay Parekh, and Christian Garcia-Arellano. Incorporating cost of control into the design of a load balancing controller. Invited paper, Real-Time and Embedded Technology and Application Systems Symposium, 2004.
- Sujay Parekh, Kevin Rose, Yixin Diao, Victor Chang, Joseph L. Hellerstein, Sam Lightstone, Matthew Huras. Throttling utilities in the IBM DB2 Universal Database Server. American Control Conference, 2004.