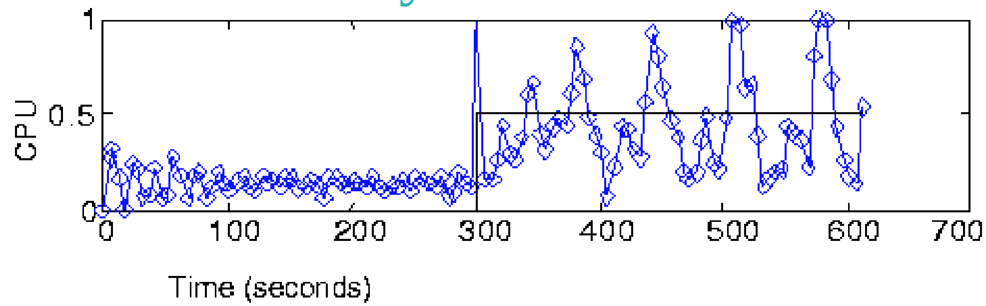
Analysis and Control of Computing Systems Using Linear Discrete-Time System Theory: *Introduction*

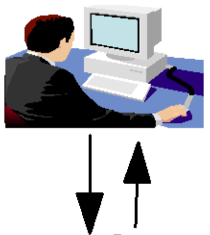
Joseph L. Hellerstein, Jie Liu
Microsoft

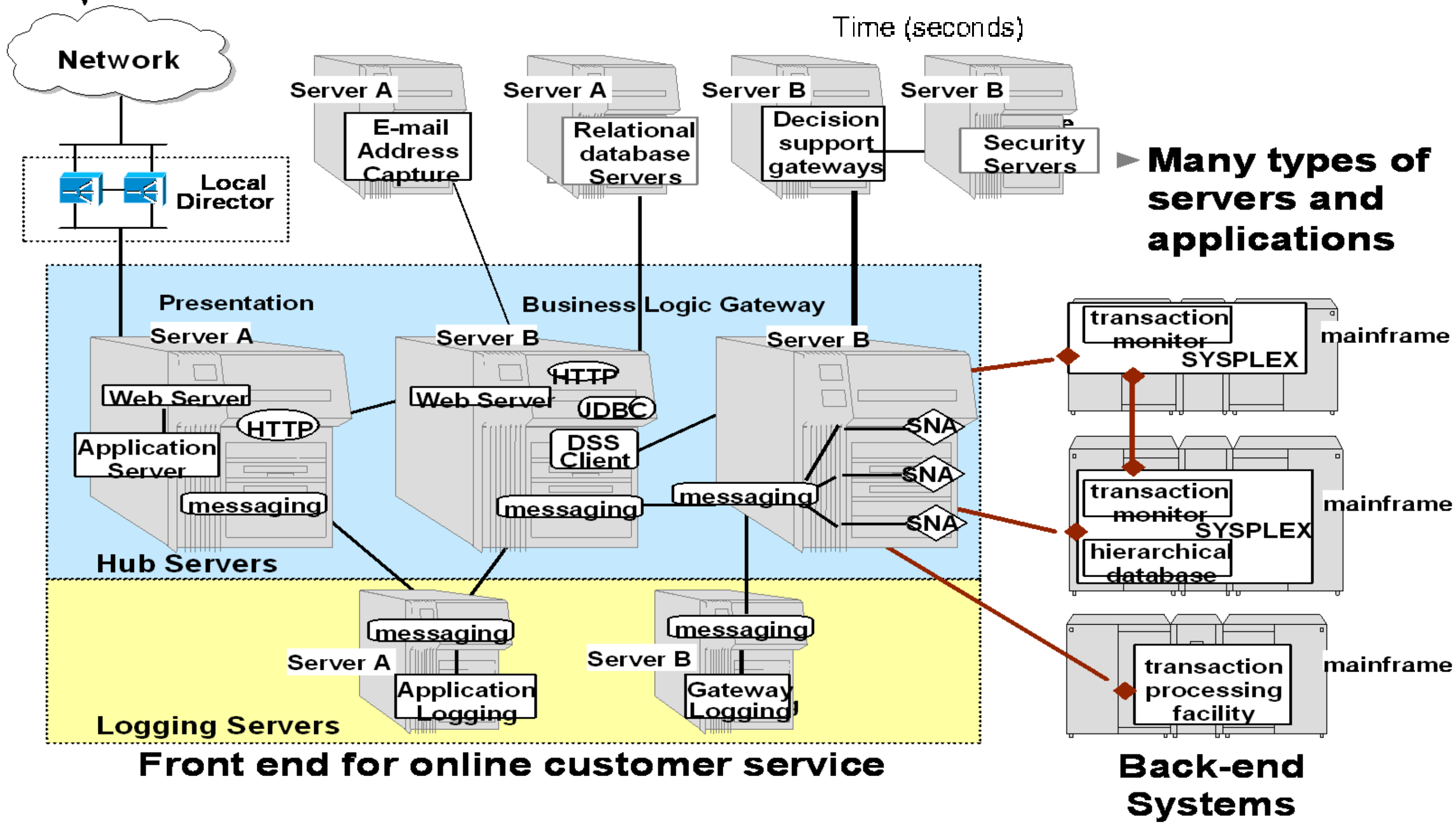
January 7, 2008

Why Control Theory

Unstable System



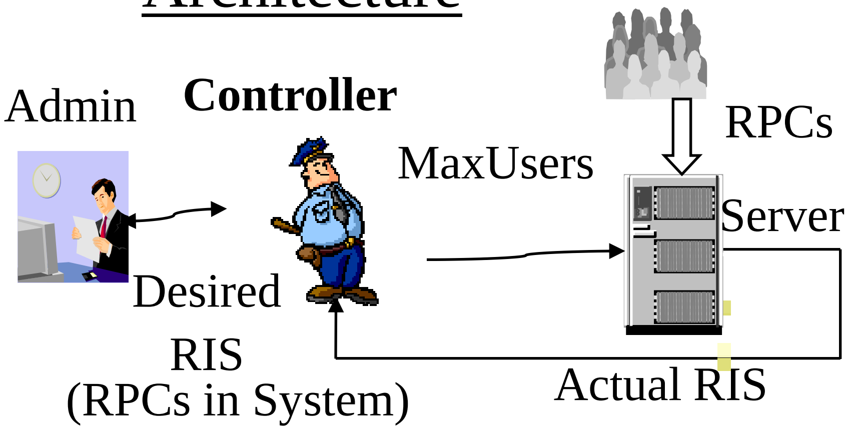


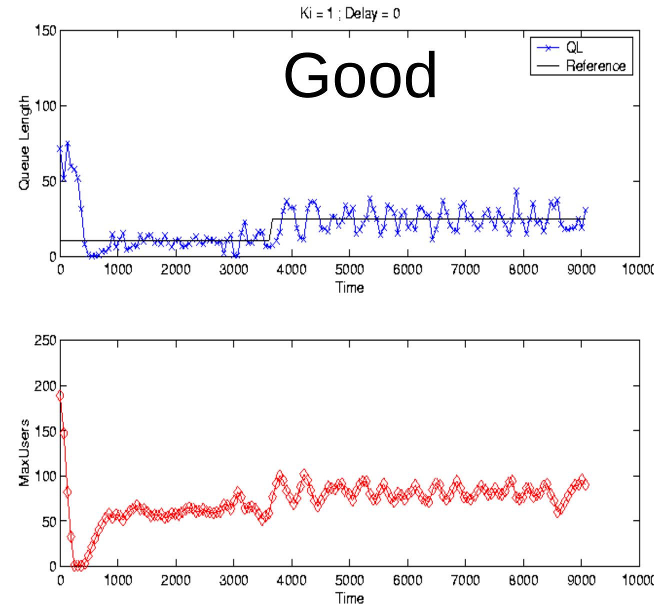


•Stabililty •Accuracy •Settling time •Overshoot

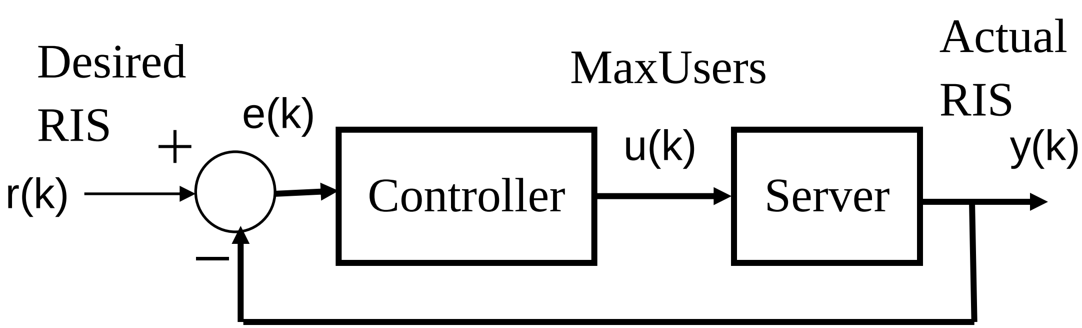
Control Theory By Example – IBM Domino Server

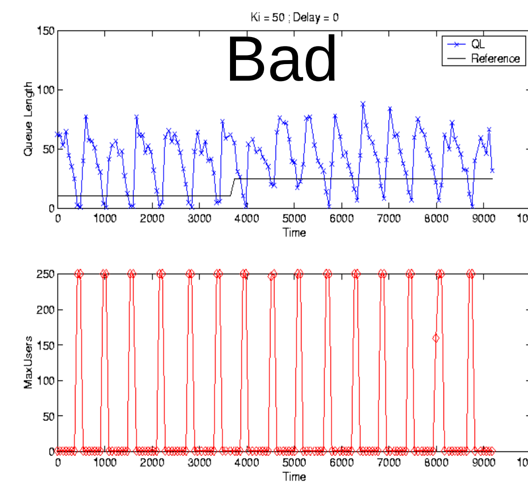
Architecture





Block Diagram





$$u(k) = K_P e(k) \qquad y(k) = (0.43)y(k-1) + (0.47)u(k-1)$$

Agenda

- Course goals, syllabus, & reference materials
- Control theory basics (qualitative control theory)
- “Labs”

Course Objectives & Structure

- Goal
  - Provide computer scientists a practical knowledge of control theory
- Structure of classes
  - 1.5 hours lecture
  - 0.5 hour group problem solving
- References (Willl be on class web page)
  - Class notes
  - Likely reference texts
    - "Feedback Control of Computing Systems," JL Hellerstein, Y Diao, S Parekh, D Tilbury. Wiley, 2004.
    - "Structure and Interpretation of Signals and Systems," Edward A. Lee and Pravin Varaiya, Addison Wesley, 2003
  - Key papers
- Opportunity for students to introduce control design into their research
  - Student presents control problem
  - Group discussion of control design

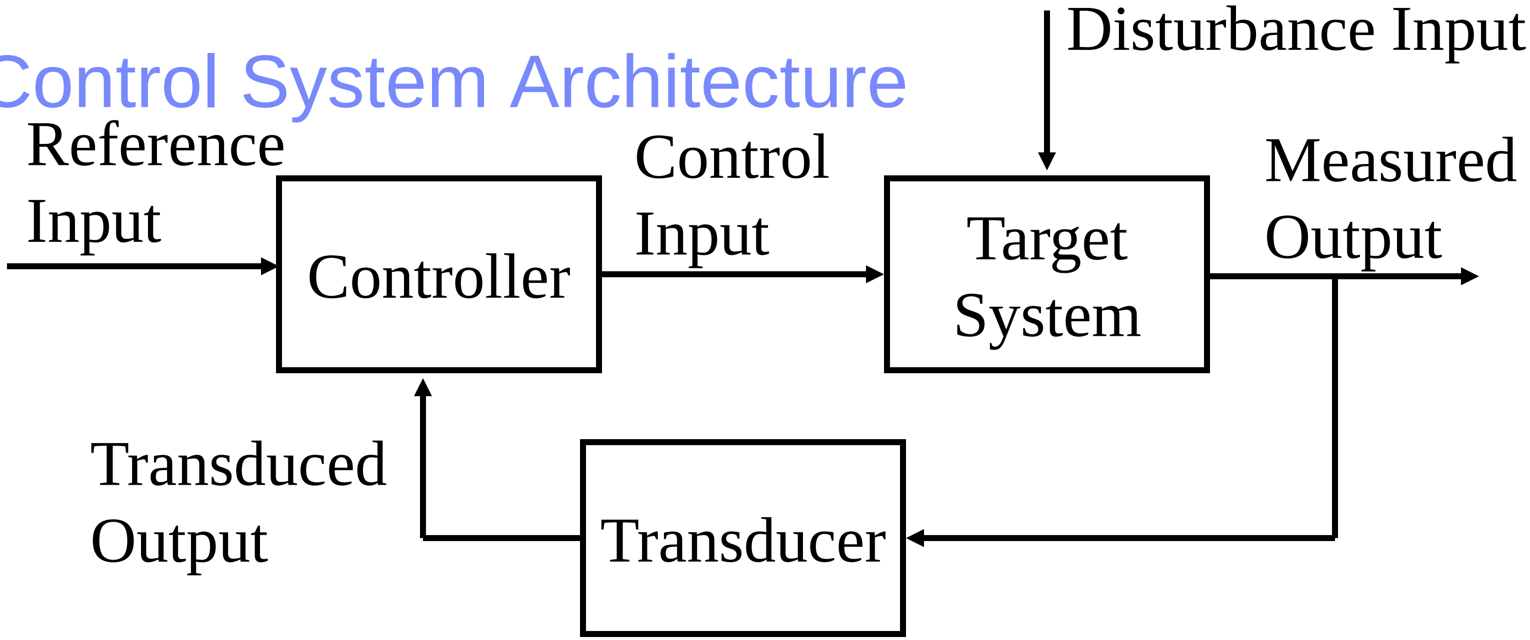
Wk	Topic	Content
1	Introduction	Course structure, objectives of control theory, spreadsheet example, "qualitative control theory", SASO properties.
2	System, modeling, and structures	Modeling systems in the time domain. First principles models using queuing. Modeling with finite state machines.
3	Basics of LTI systems	Signals. Transfer functions. Poles. Steady state gain. Stability. Settling times.
4	Multi-component systems	Block diagrams. Constructing system transfer functions from composition of subsystems. Basic control structures.
5	Controllers, control design, control analysis	PID controllers. Design using pole placement.
6	Case studies	Details of two applications to computing systems: throttling utilities in the IBM DB2 database management system and power management in a data center.
7	State space modeling & control	Multiple-input, multiple-output control. Controllability, observability, hybrid control.
8	Case study	Case study of real-time garbage collection. Possible student presentations.
9	Advanced topics & class conclusion	Nonlinear control; stochastic control; adaptive control.
10	Additional case studies	

Class 1: Introduction

Agenda

- Control architecture & terminology
- Examples
- Kinds of control
- Objectives of control systems
- Goals of control analysis
- Labs

Control System Architecture

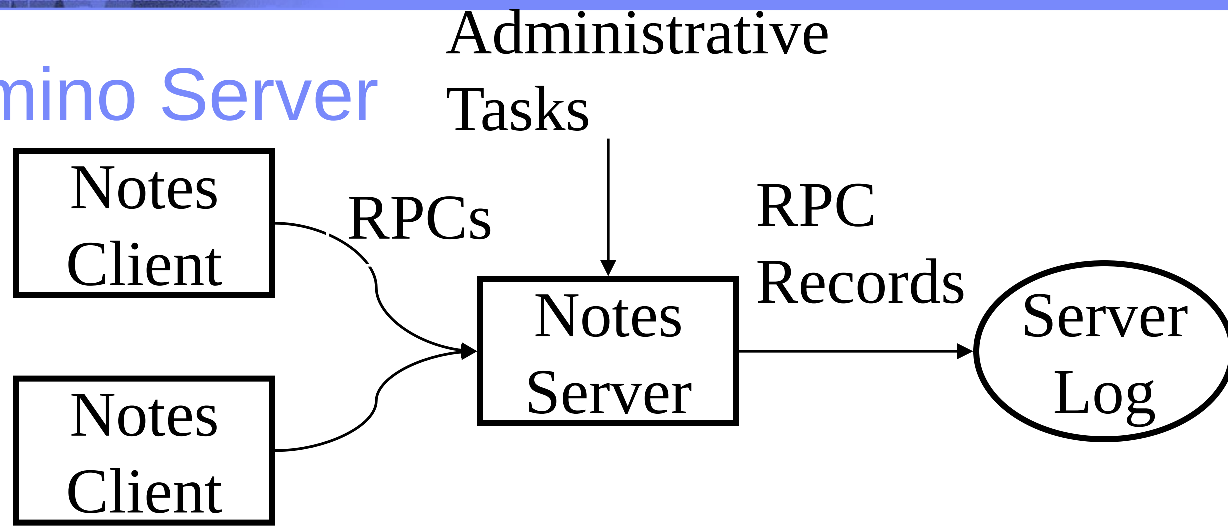


Components
- Target system: what is controlled
- Controller: exercises control
- Transducer: translates measured outputs

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

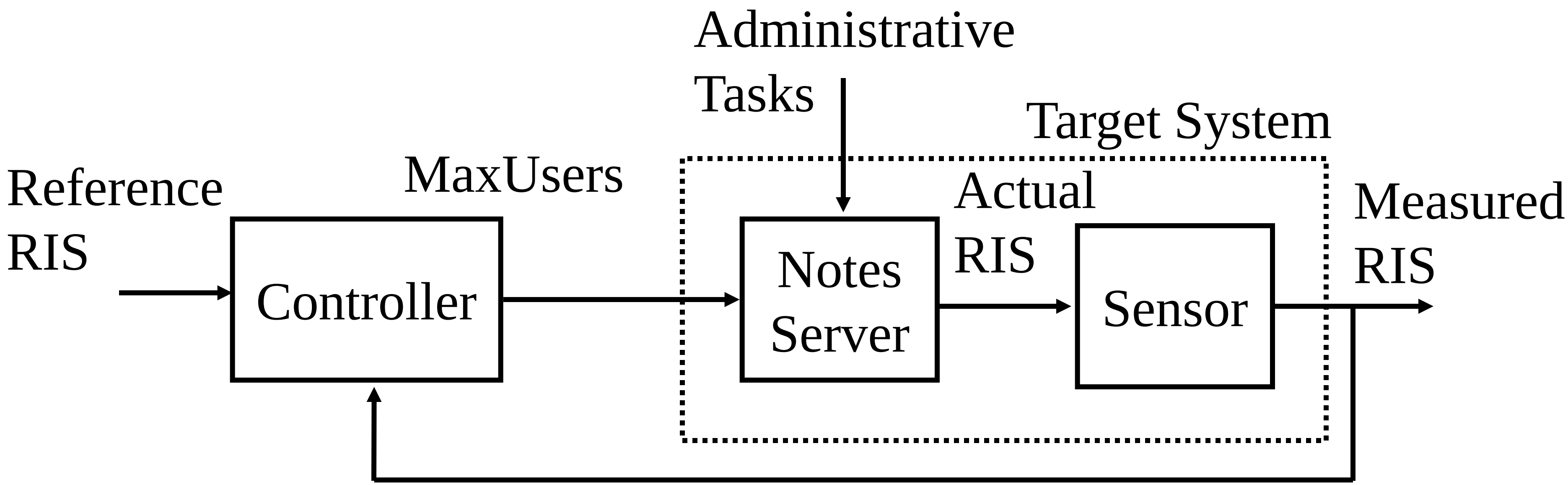
Given target system, transducer
Control theory finds controller
that adjusts control input
to achieve measured
output in the presence of
disturbances.

IBM Lotus Domino Server



Architecture

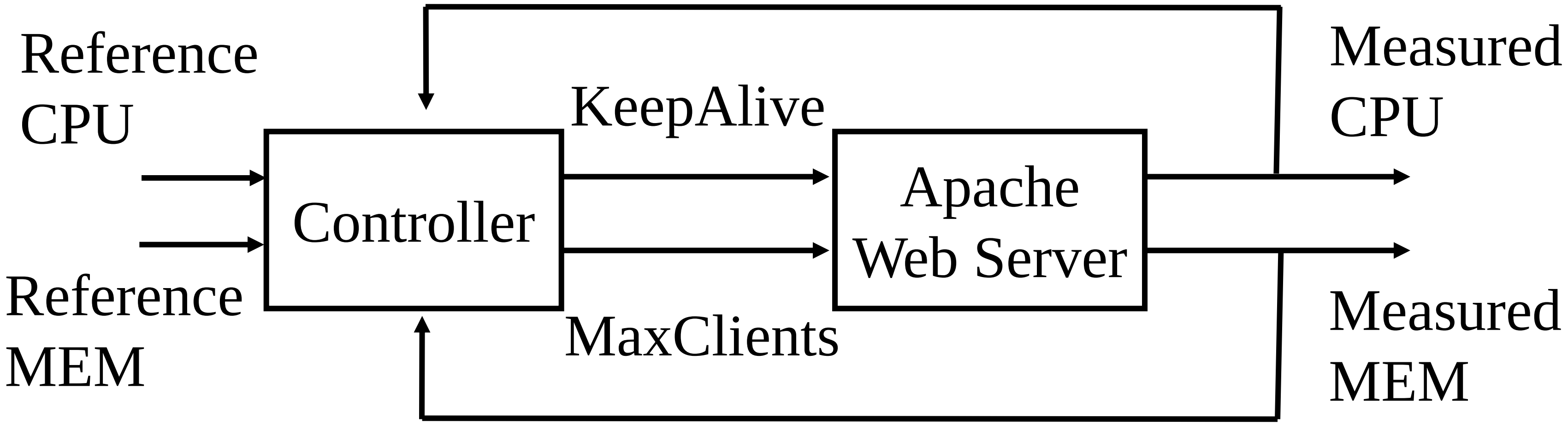
MaxUsers



Block Diagram

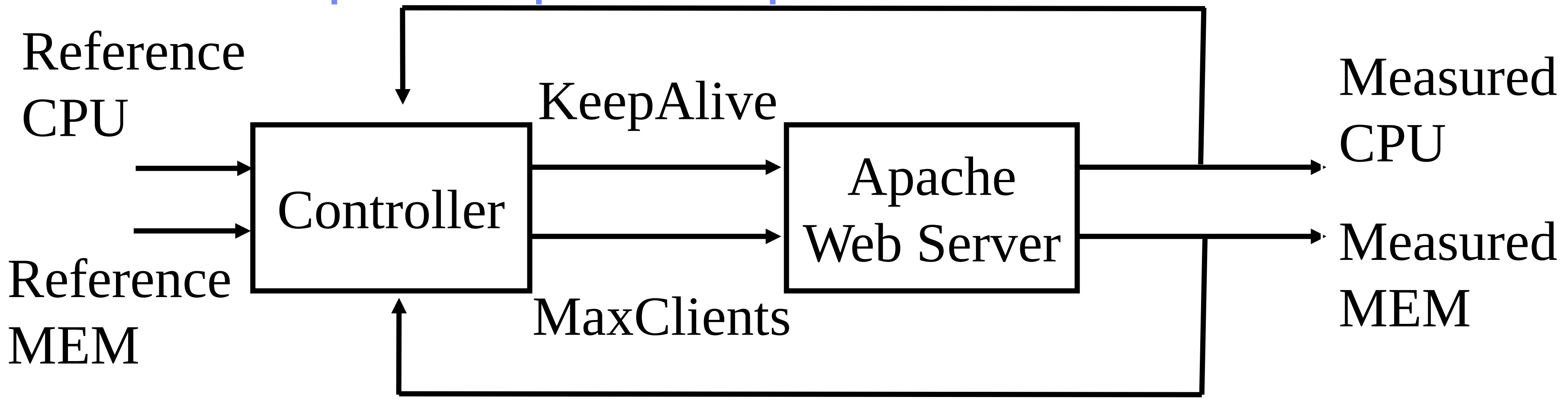
RIS = RPCs in System

Target Systems With Multiple Inputs & Outputs



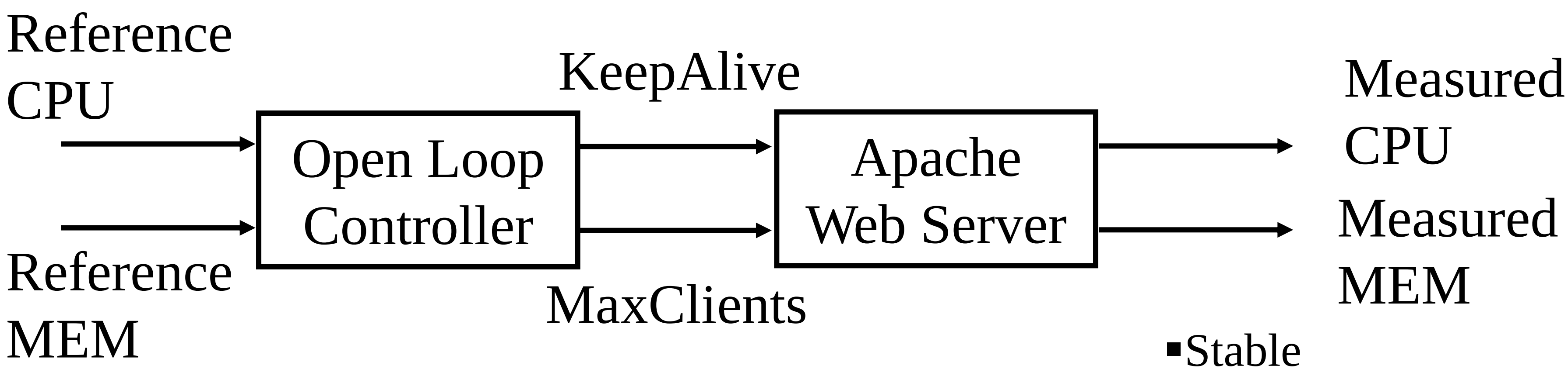
Block Diagram

Closed Loop vs. Open Loop



Apache: Closed Loop Control

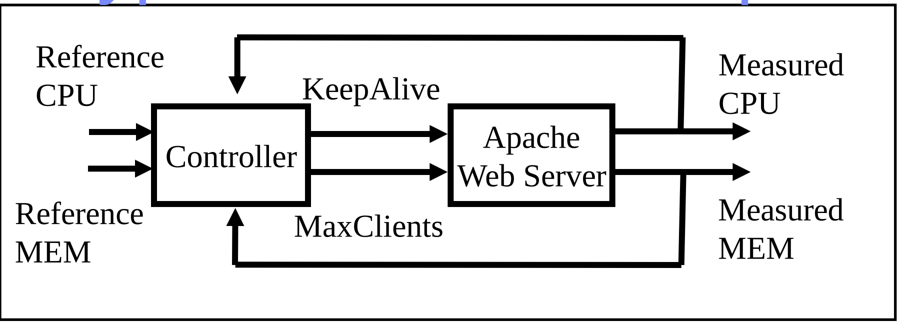
- Adapts
- Simple system model



Apache: Open Loop Control

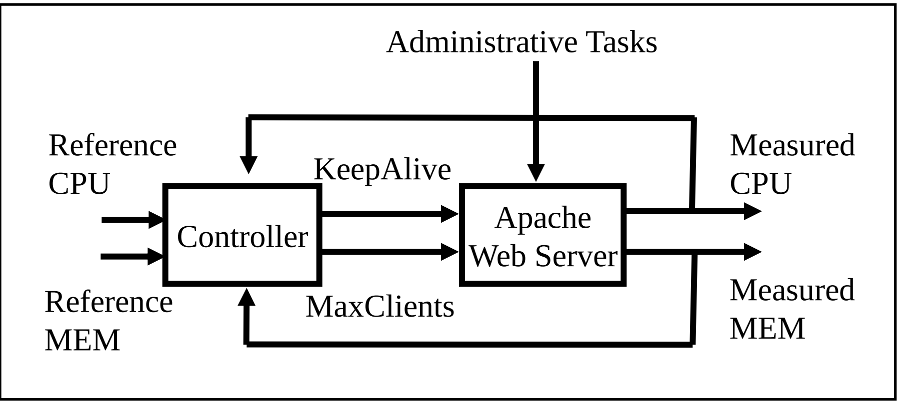
- Stable
- Fast settling

Types of Closed Loop Control Systems



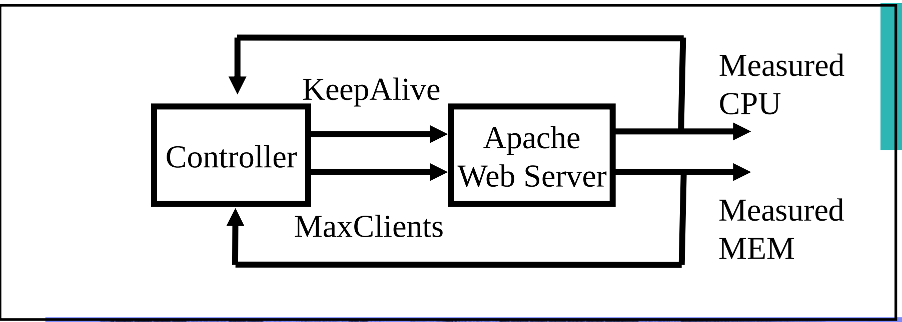
- Manage to a reference value
- Ex: Service differentiation, resource management, constrained optimization

Regulatory Control



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

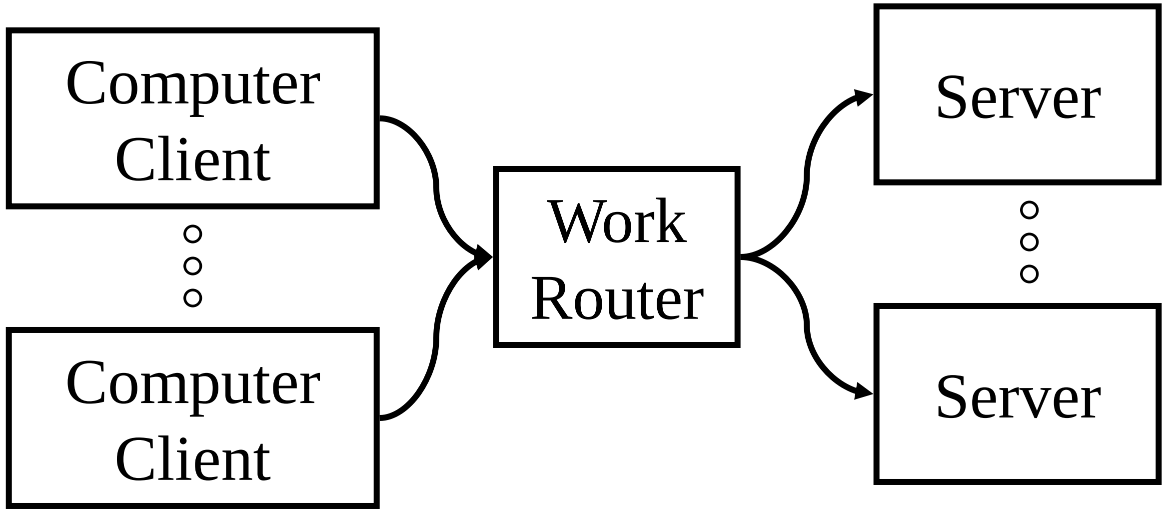
Disturbance Rejection



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

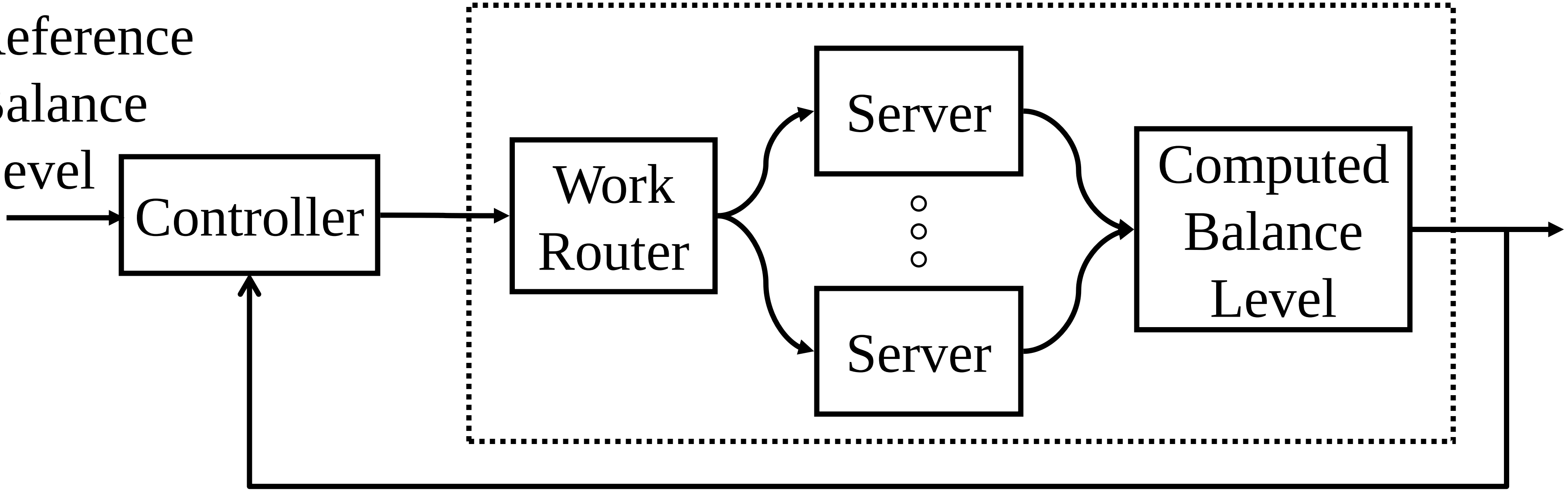
Optimization

Load Balancing



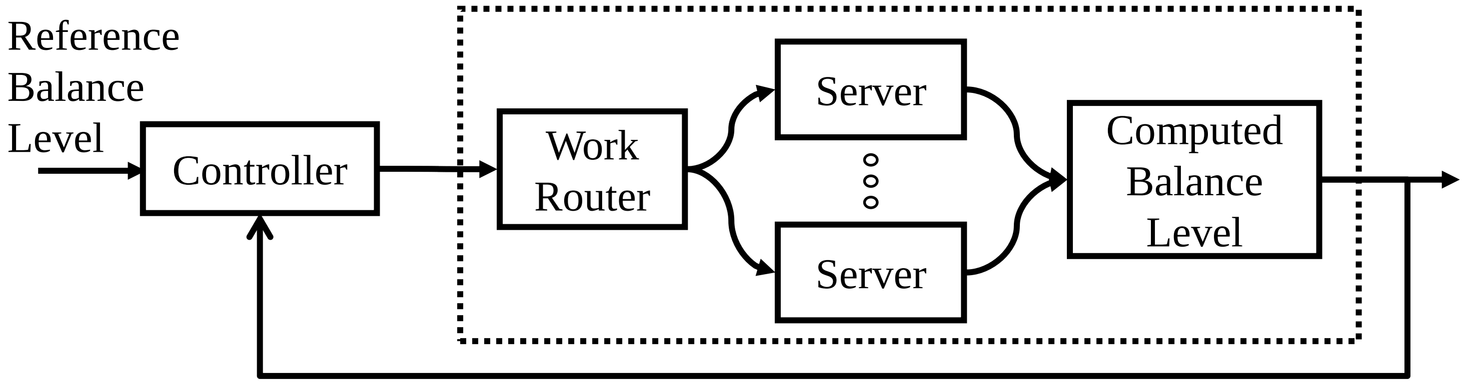
Architecture

Reference Balance Level

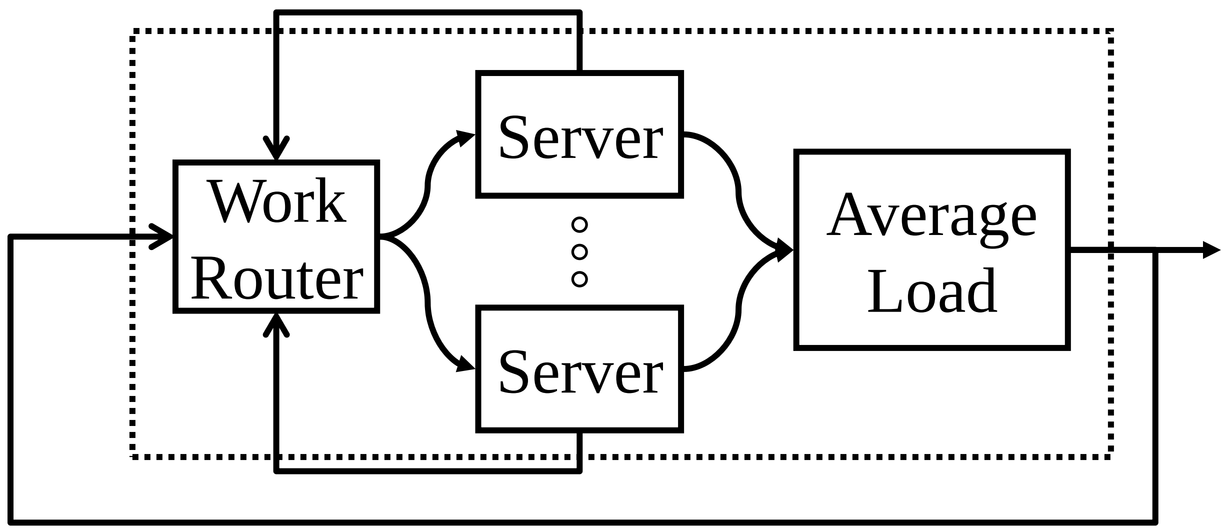


Block Diagram

Load Balancing (continued)



Block Diagram

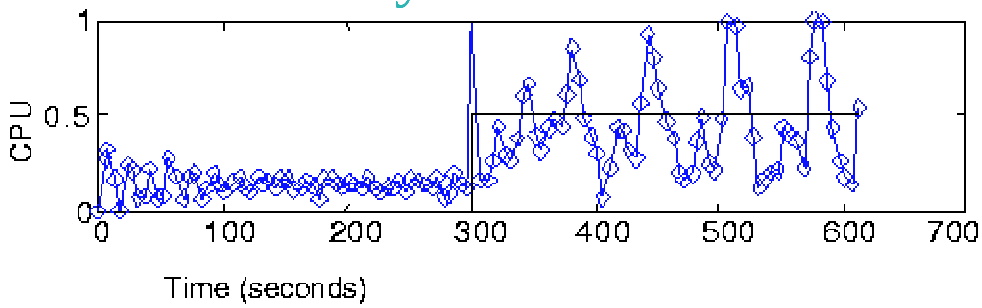


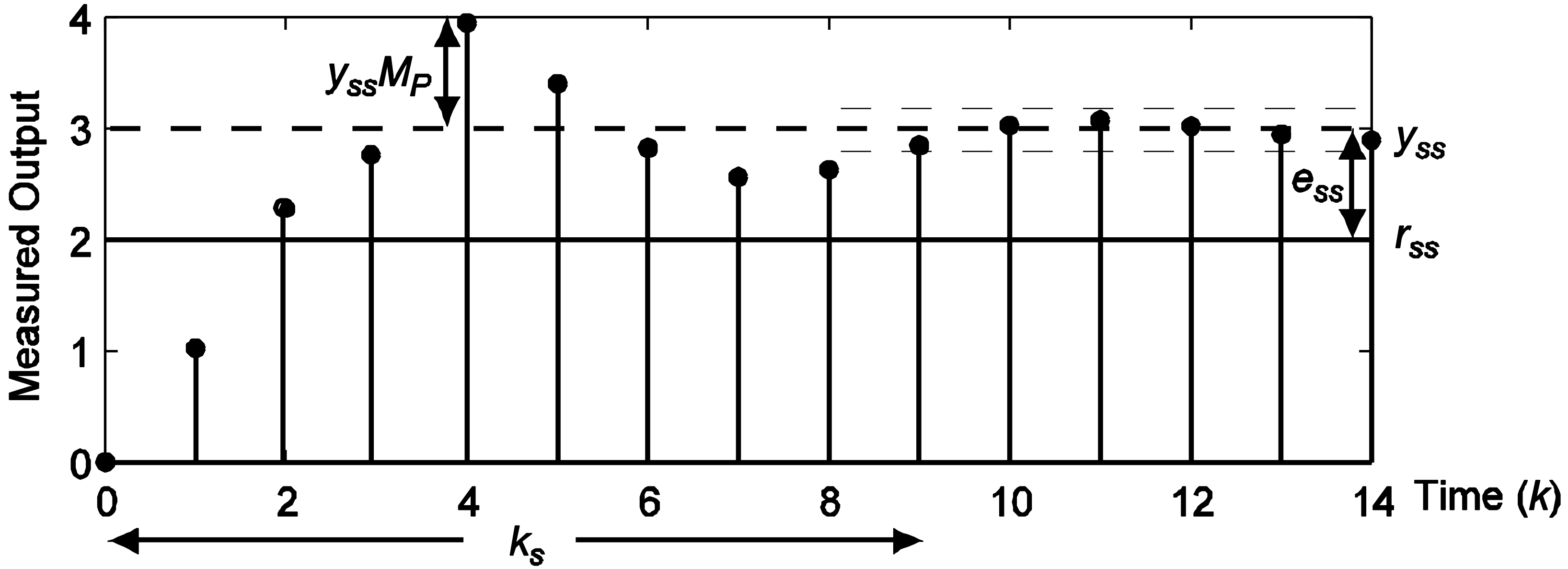
Optimization - No external reference input

SASO Properties of Control Systems

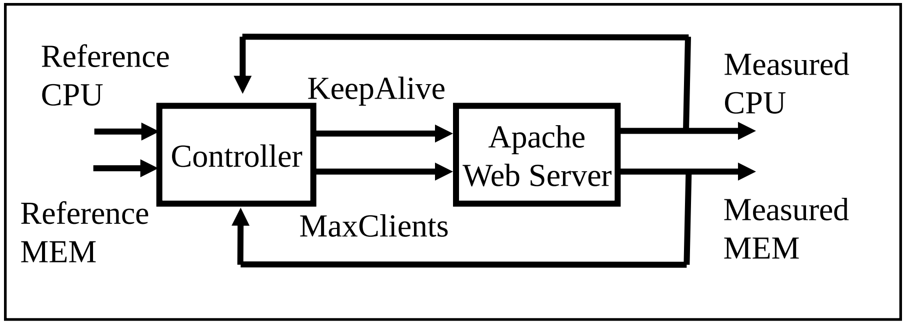
Stability **A**ccuracy **S**hort settling Small **o**vershoot

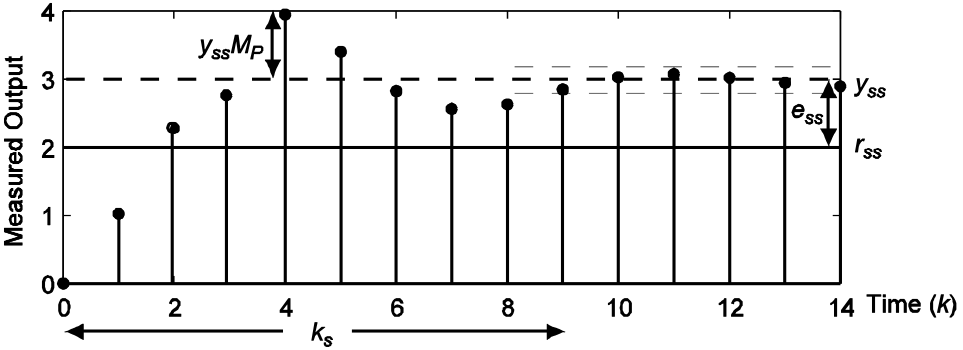
Unstable System





What Is Control Analysis?





- Model input/output relationships of target system
- Design controller to achieve closed loop objectives

Focus on modeling dynamics

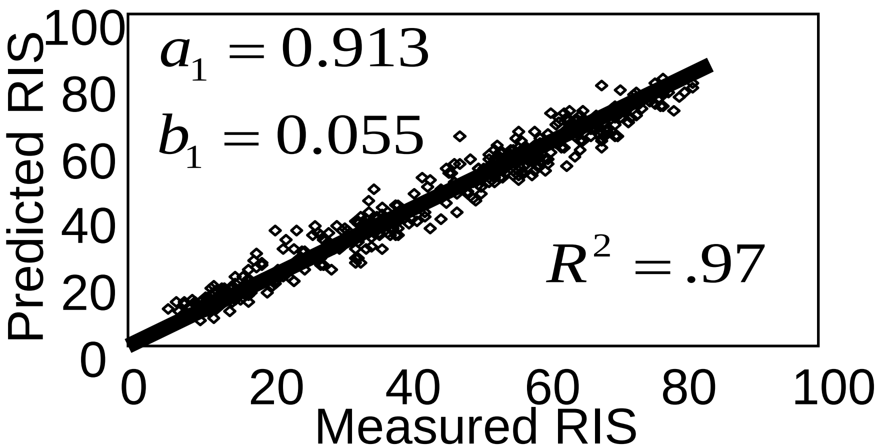
Control Theory In 2 Slides

- System identification
  - Modeling dynamics
- Controller design
  - Choosing the control parameters

System Identification

<u>Model of System Dynamics</u>

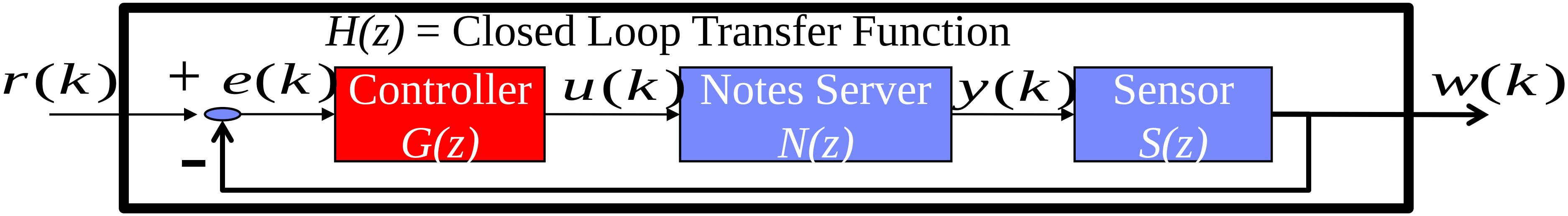
$$y(k) = a_1 y(k-1) + b_1 u(k-1)$$



<u>Transfer Function</u>

$$N(z) = \frac{b_1}{z - a_1}$$

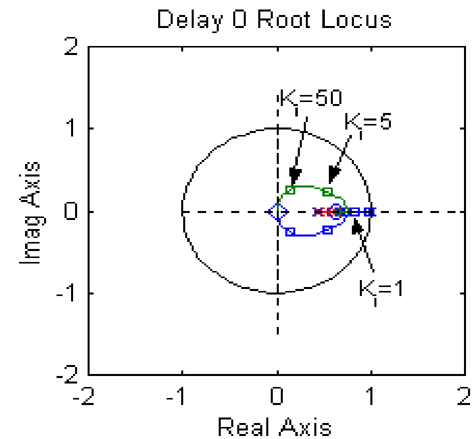
Control Design

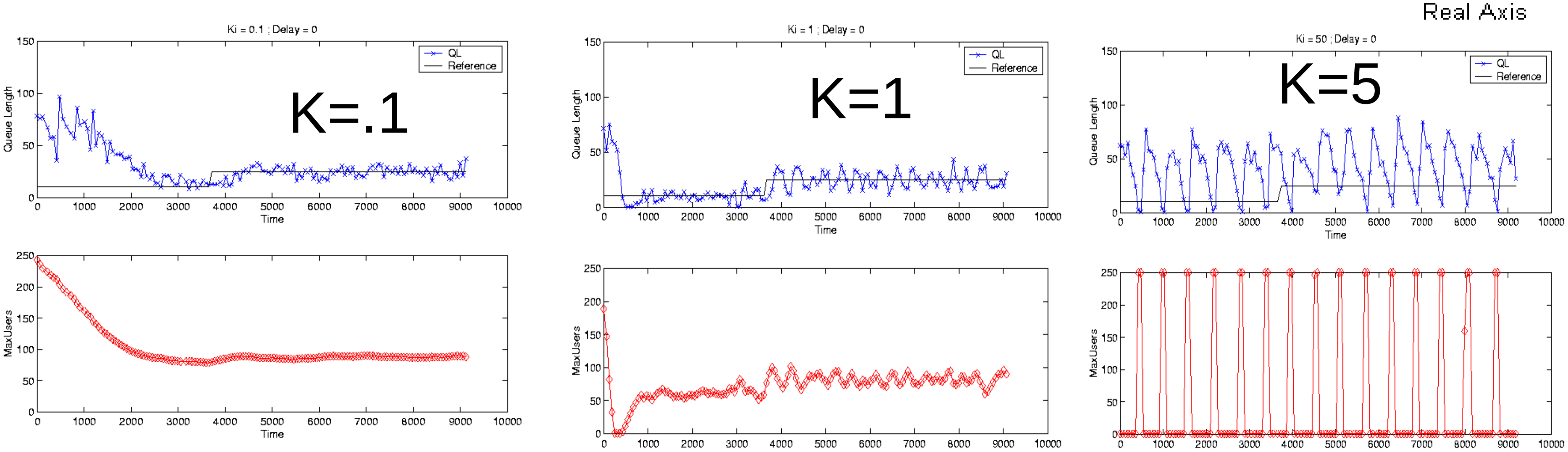


Parameterized Control Law

$$u(k) = u(k-1) + Ke(k)$$

Poles of *H(z)*



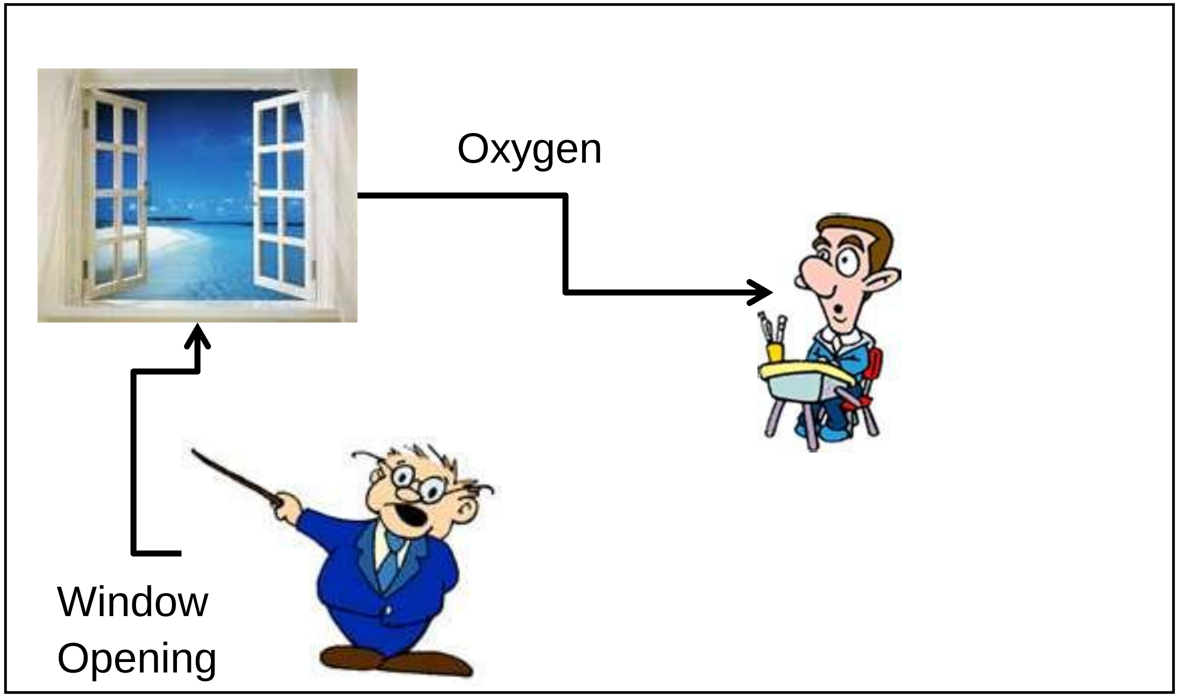


Labs:
Chapter 1

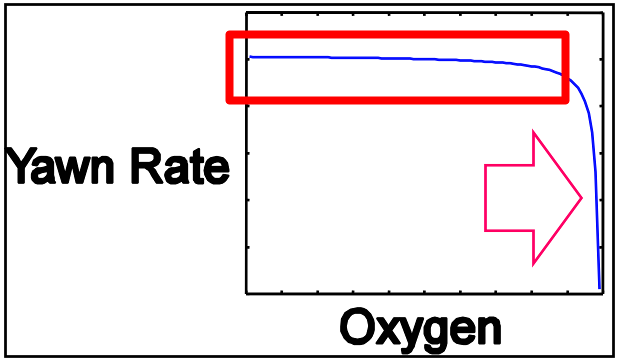
Lab 1: Yawning is Contagious

- Description of system
  - Room full of people
  - Characteristics
    - People yawn because they need more oxygen
    - Yawning consumes more oxygen than normal breathing
  - Can open windows to reduce yawning, but it’s winter
- Control objective
  - Regulate yawning to a desired frequency while maximizing temperature
- Questions
  - What is an architecture of the system? A block diagram?
  - Discuss control policies
  - What does it mean for this system to be unstable? What would make it unstable?

Architecture of "Yawn System"



Oxygen Flow System

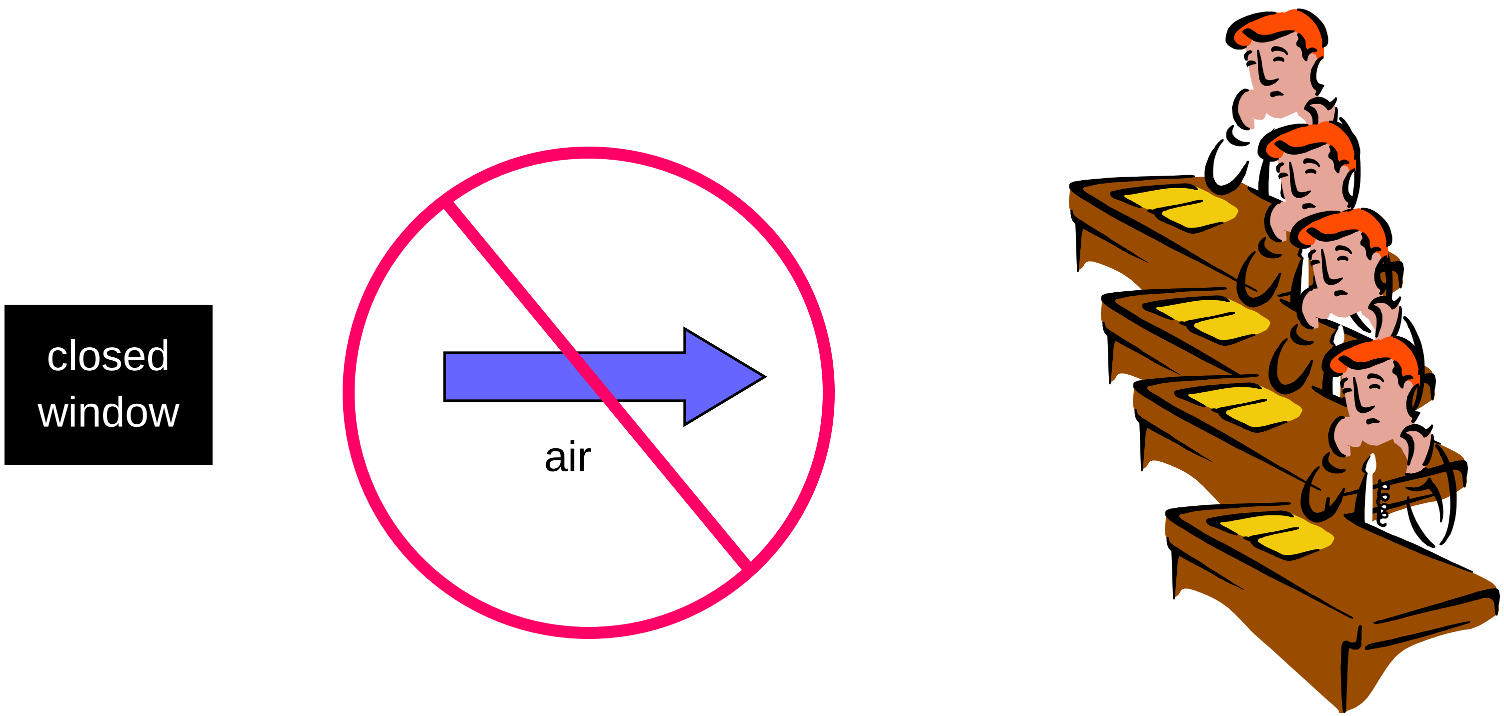


Operation of the Yawn System: Open Window

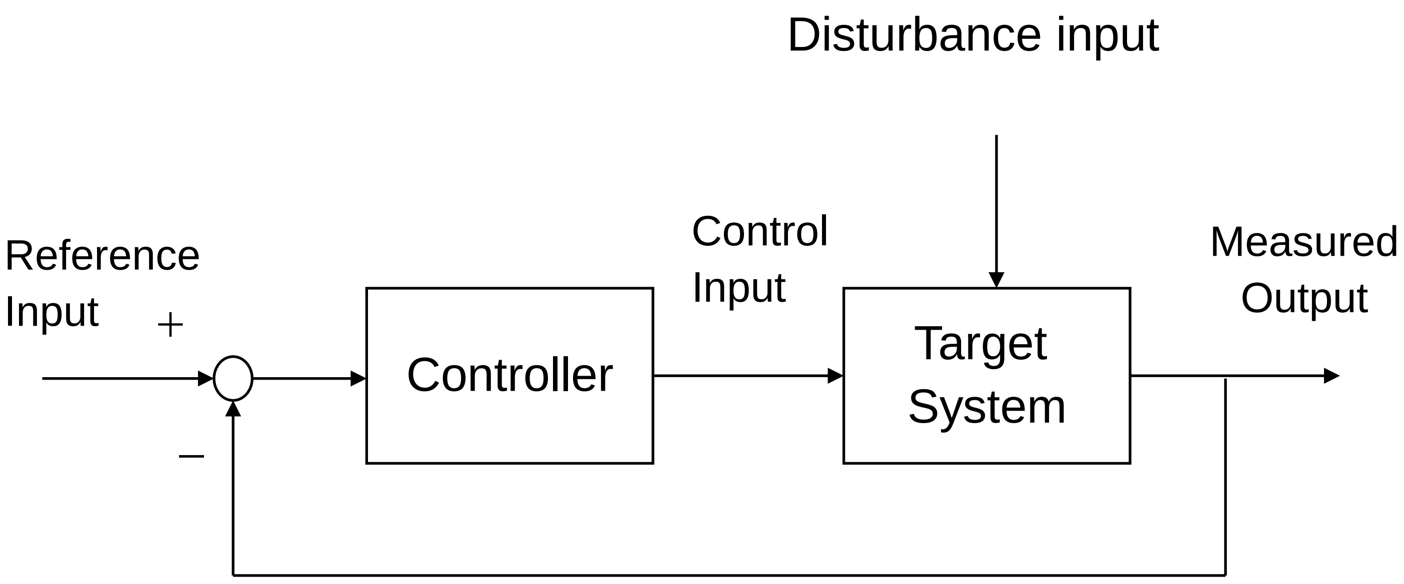
open window

air

Operation of the Yawn System: Closed Window



Feedback For Yawn System



What are

- Reference input
- Target System
- Control input
- Measured output
- Controller
- Disturbance input

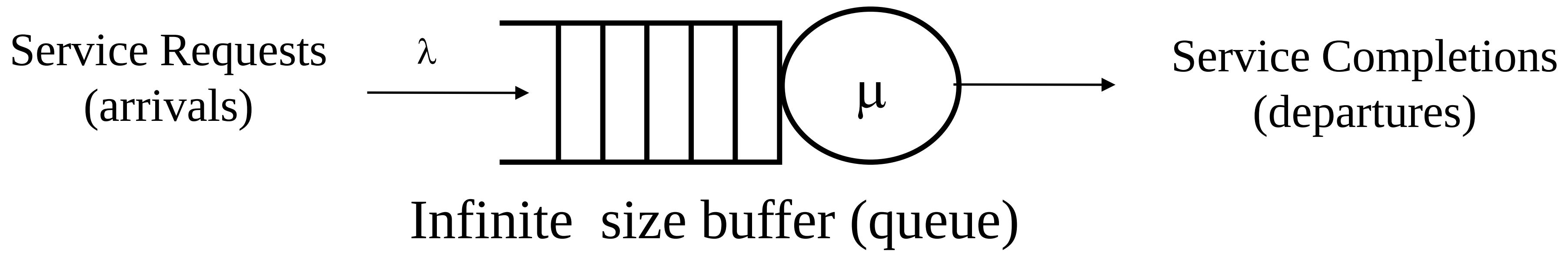
Answers

- Desired yawn rate
- Yawn response (oxygen in; yawn out)
- Window position
- Actual yawn rate
- Person who opens/closes the window
- Add/remove people, opening door

Lab 2: *M/M/1* and *M/M/1/K* Queueing Systems

- Motivation
  - Understand in detail a key model of computing systems
  - Illustrate knowledge required to construct feedback system
- Agenda
  - *M/M/1* statics and dynamics
  - *M/M/1/K* statics and dynamics
  - Open loop control
  - Closed loop control

M/M/1 Queueing System



Operation

- Arrival of a service request
  - Request enters service if buffer is empty
  - Enter queue if server is busy
- Completion of a service request
  - Next request in buffer enters the server
  - If buffer is empty, the system goes idle

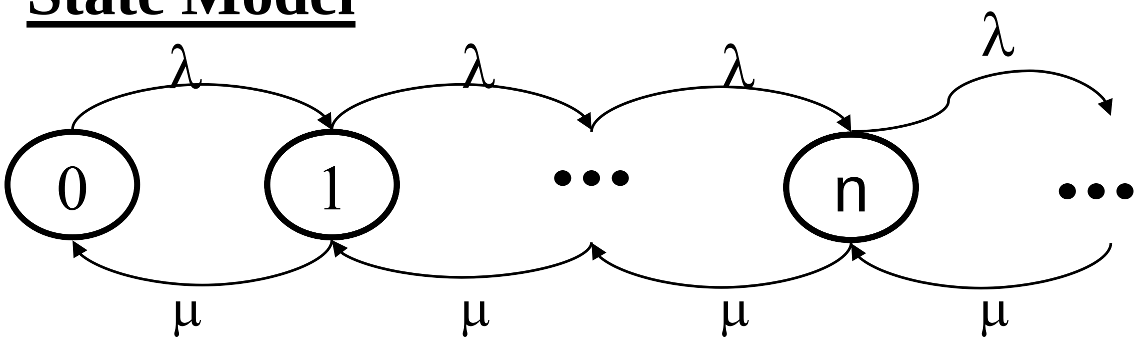
Assumptions & Key Result

- Assumptions
  - Inter-arrival times are exponentially distributed
  - Service times are exponentially distributed
- Key result for steady state
  - N = expected number in system

$$N = \frac{\lambda}{\mu - \lambda}$$

State Analysis of *M/M/1* Queueing System

State Model



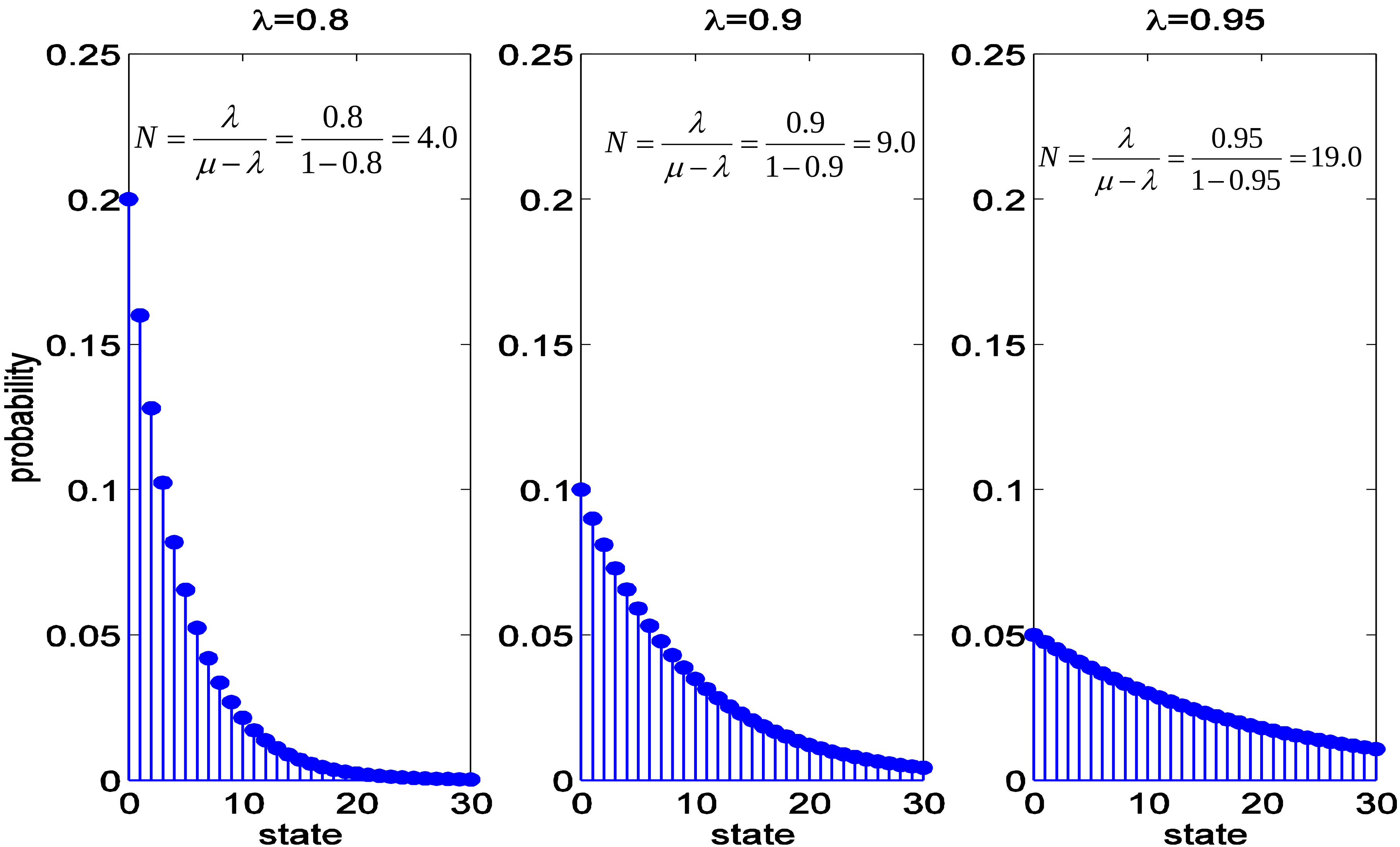
- State is number of customers in the system
- Arrows indicate rate at which transitions occur
- Arrival increases state by 1; departure decreases state by 1
- Probability of being in state *n* is p_n

$$N = \sum_{n=0}^{\infty} n p_n$$

Statics

- Characteristics of system after operating for a long time without any changes
- Examples
  - Expected value of number in system
  - Probability of being in state

M/M/1 Statics: μ=1

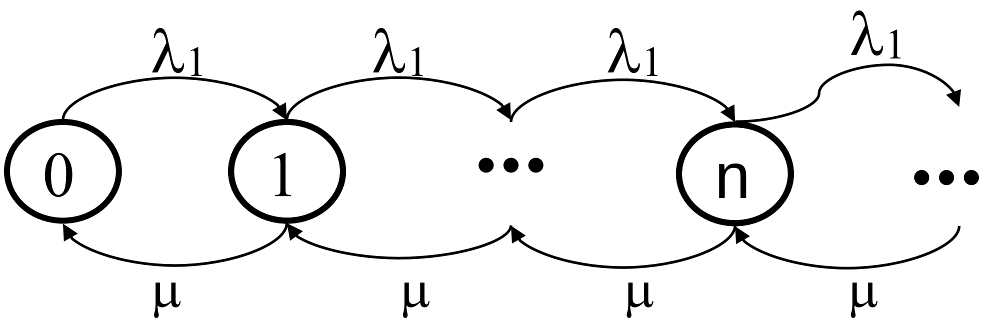


Dynamics

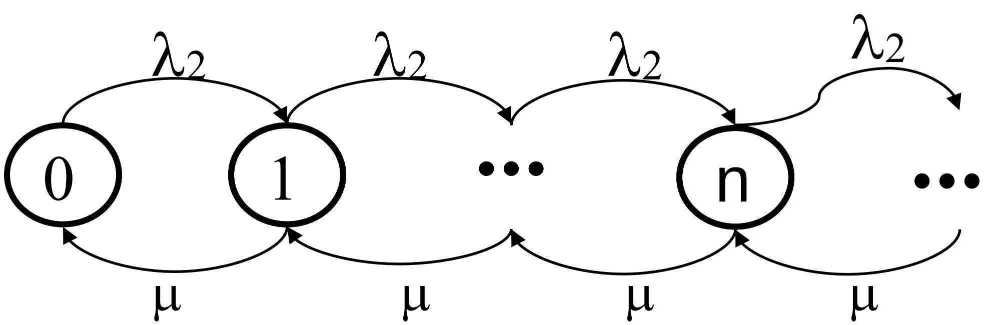
- Characteristics of systems when they change
- Examples
  - Change in request rate due to workload changes
  - Change in service rate due to workload changes and/or due to varying power to CPU in order to control number in system
- ***Feedback control requires an understanding of dynamics***
  - Very interested in settling time—time to reach the new steady state

$M/M/1$ Dynamics

State Model at time t_1



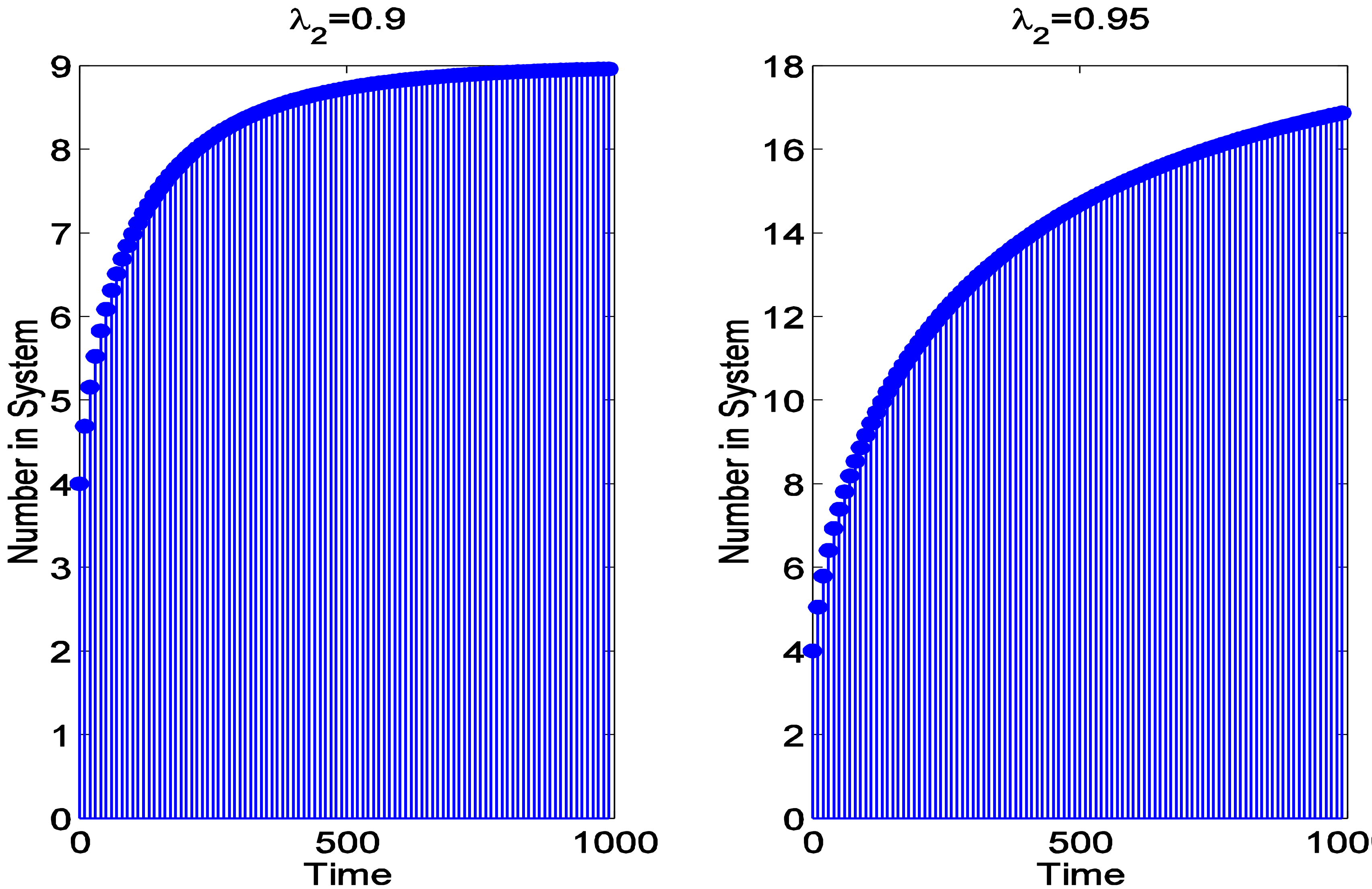
State Model at time t_2



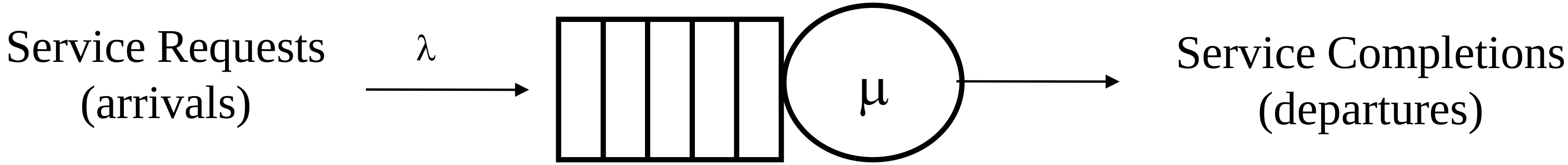
Operation

1. At time t_1, *the* state probability vector is $\boldsymbol{p}_1$
2. When arrival rates change, we use a different Markov chain with $\boldsymbol{p}_1$ as the starting state probability vector
3. The probability vector gradually changes with time, which causes $N(t)$ to change as well.

M/M/1 Dynamics: μ=1, λ_1=0.8 (Expected Number In System)



*M/M/1/**K*** Queueing System



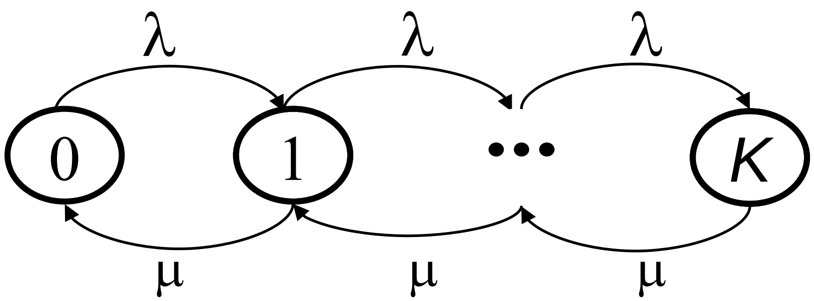
Finite buffer with *K-1* entries

<u>Operation</u>

- Arrival of a service request
  - Request enters service if buffer is empty
  - Enter queue if server is busy **& space exists in buffer**
  - **If buffer is full, request is dropped**
- Completion of a service request
  - Next request in buffer enters the server
  - If buffer is empty, the system goes idle

State Analysis of *M/M/1/K* Queueing System

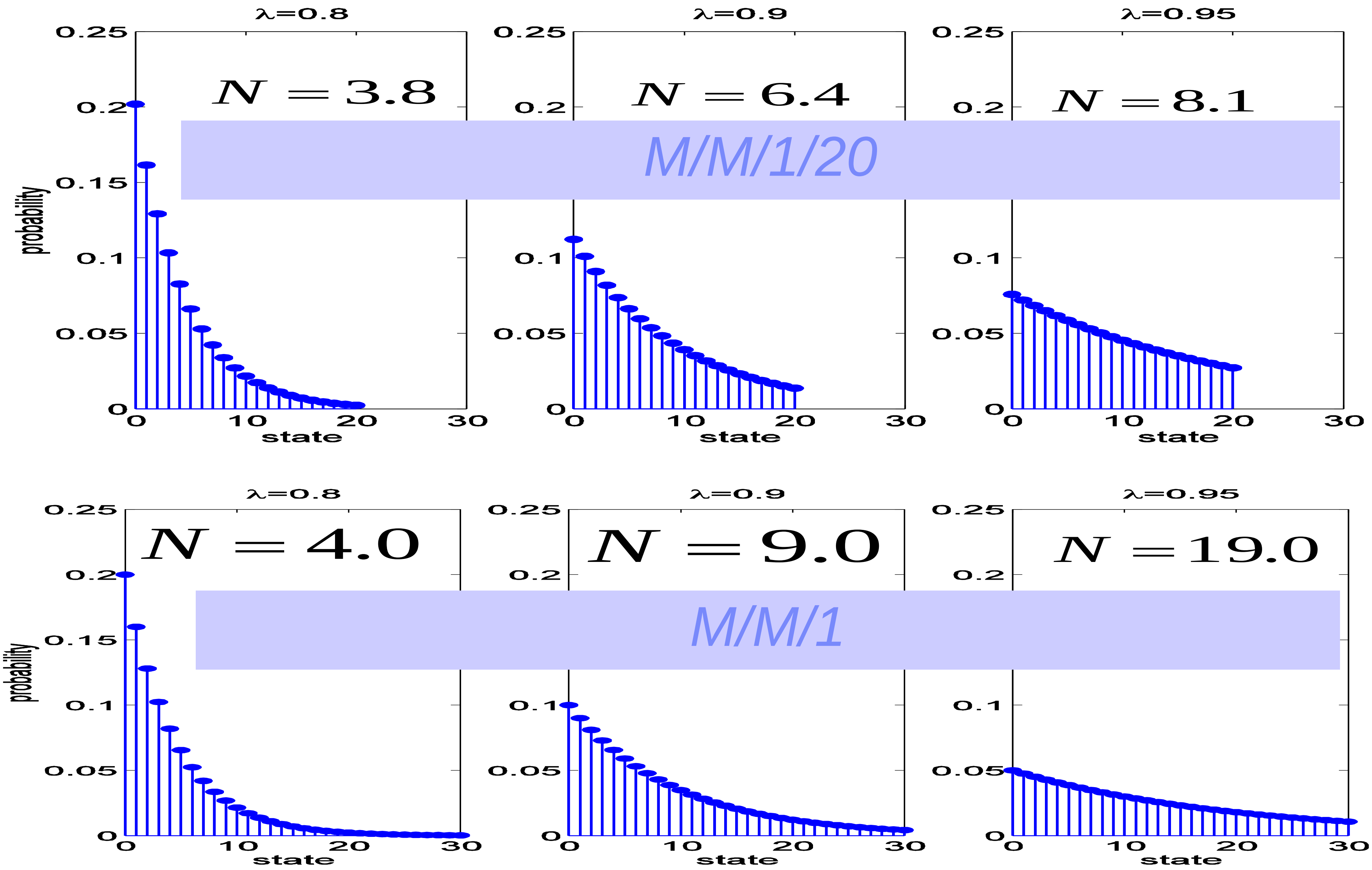
State Model



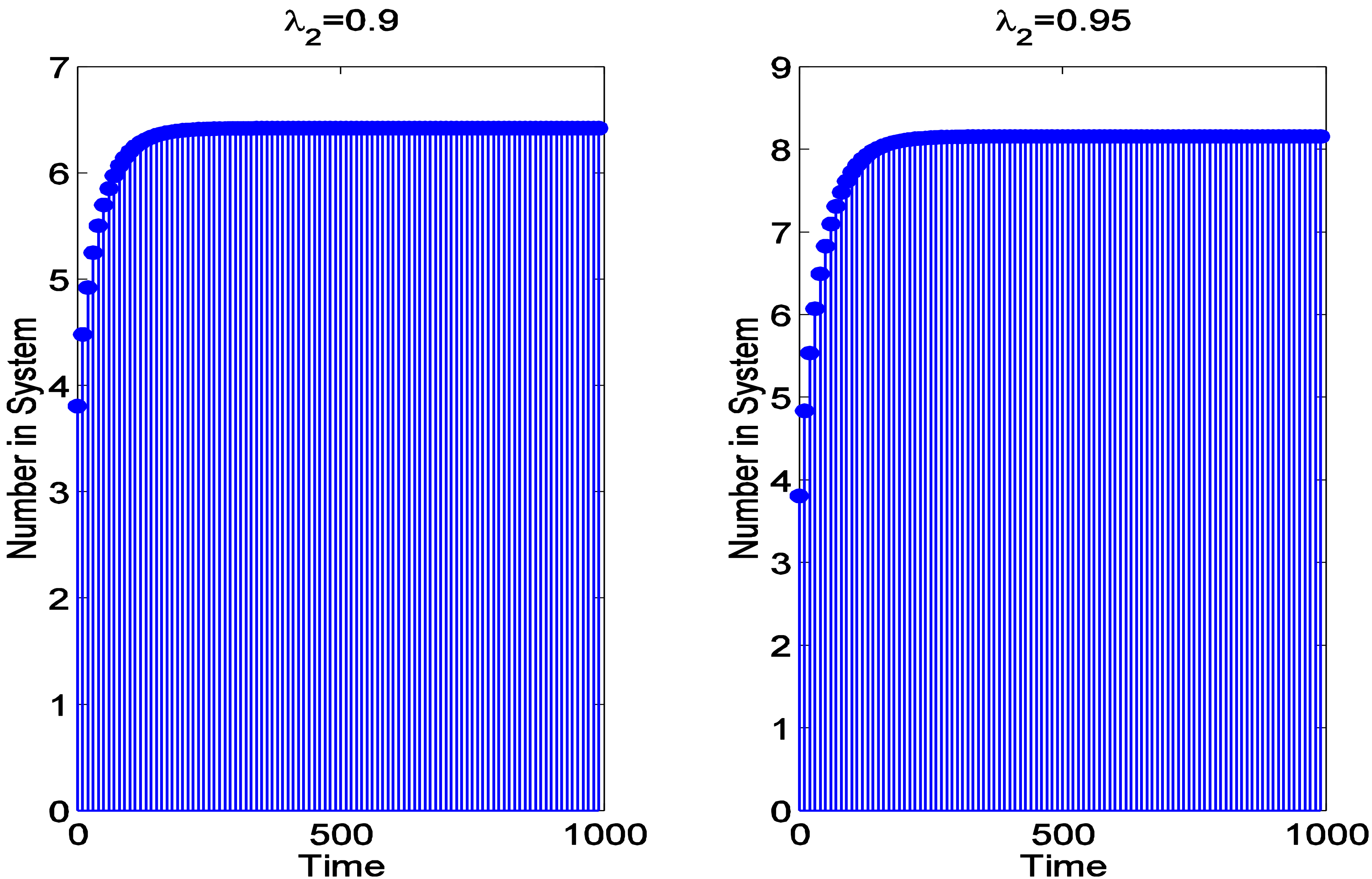
- State is number of customers in the system
- Arrows indicate rate at which transitions occur
- Arrival increases state by 1; departure decreases state by 1

$$N = \frac{\frac{\lambda}{\mu}\left(1-\left(\frac{\lambda}{\mu}\right)^{K} - K\left(\frac{\lambda}{\mu}\right)^{K} + K\left(\frac{\lambda}{\mu}\right)^{K+1}\right)}{\left(1-\frac{\lambda}{\mu}\right)\left(1-\left(\frac{\lambda}{\mu}\right)^{K+1}\right)} \neq \frac{\lambda}{\mu-\lambda}$$

Statics, μ=1


λ=0.8
λ=0.9
λ=0.95
N = 3.8
N = 6.4
N = 8.1
M/M/1/20
probability
state
N = 4.0
N = 9.0
N = 19.0
M/M/1

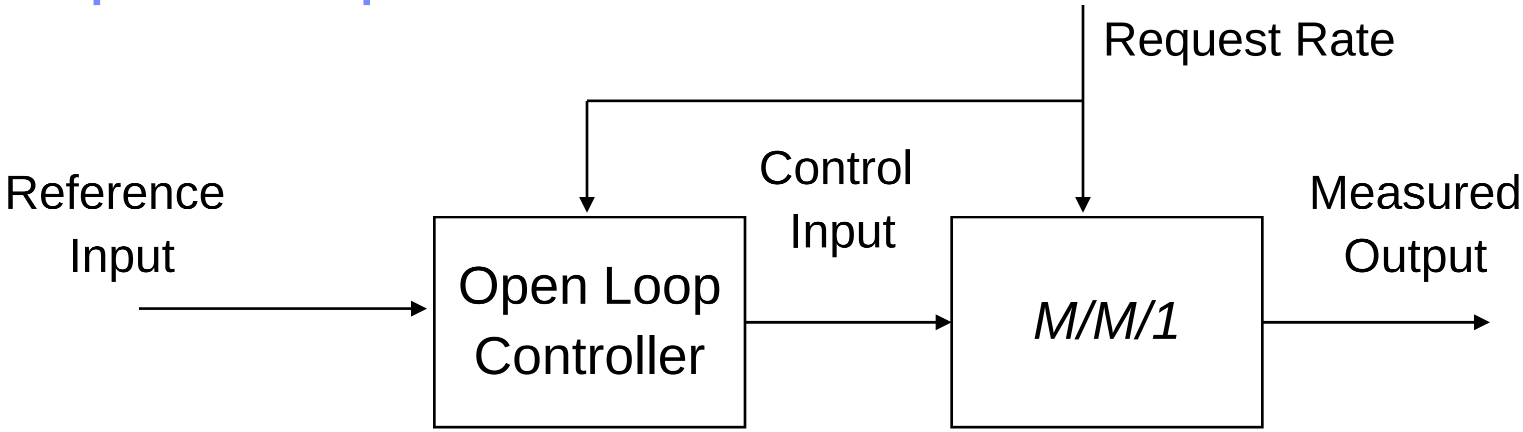
M/M/1/K Dynamics: μ=1, K=20, λ_1=0.8



Summary

- State probability vector
  - Probability of being in each state
- Steady state distribution
  - Long-term distribution if there are no changes in the system
- Transient distribution
  - Distribution during the transition from one steady state to another
- The following are larger for *M/M/1* than for *M/M/1/K*
  - Steady state value of number in system
  - Settling times
- *M/M/1/K* can be approximated by *M/M/1* if
  - Load is light or
  - *K* is large

Open Loop Control of *M/M/1*



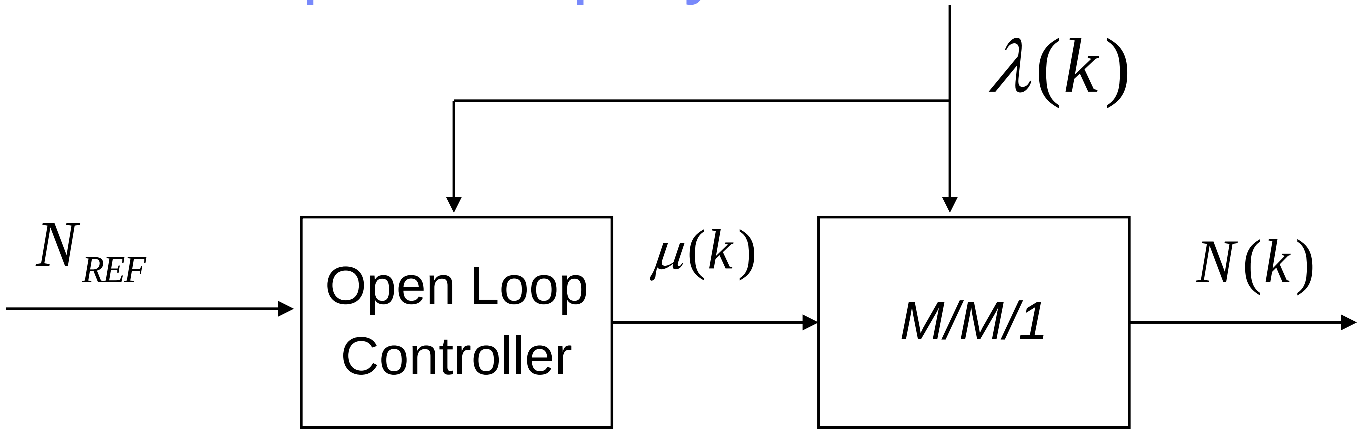
What are

- Reference input
- Control input
- Measured output
- Disturbance

Possible Answers

- Desired number in system: N_{REF}
- Service rate: $\mu(k)$
- Measured number in system: $N(k)$
- Transition from λ_1 to λ_2.

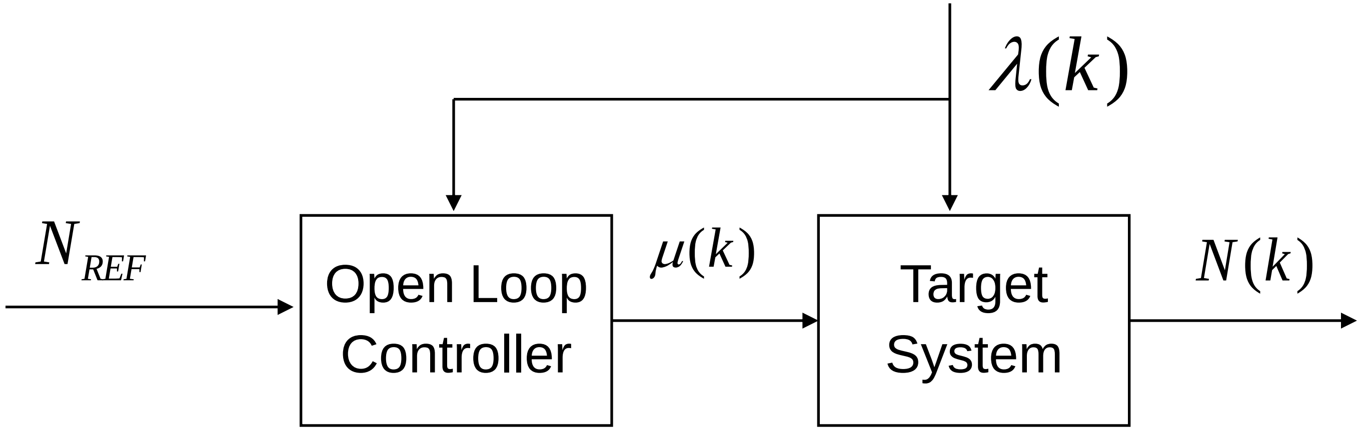
Operation of Open Loop System for *M/M/1*



Operation

1. $k=0$
2. $\mu(k)$=*Open-Loop-Control*(N_{REF}, $\lambda(k)$);
3. $k=k+1$
4. Goto step 2

Open Loop Controller

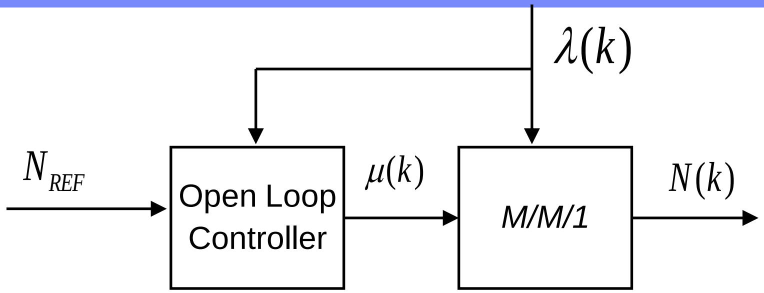


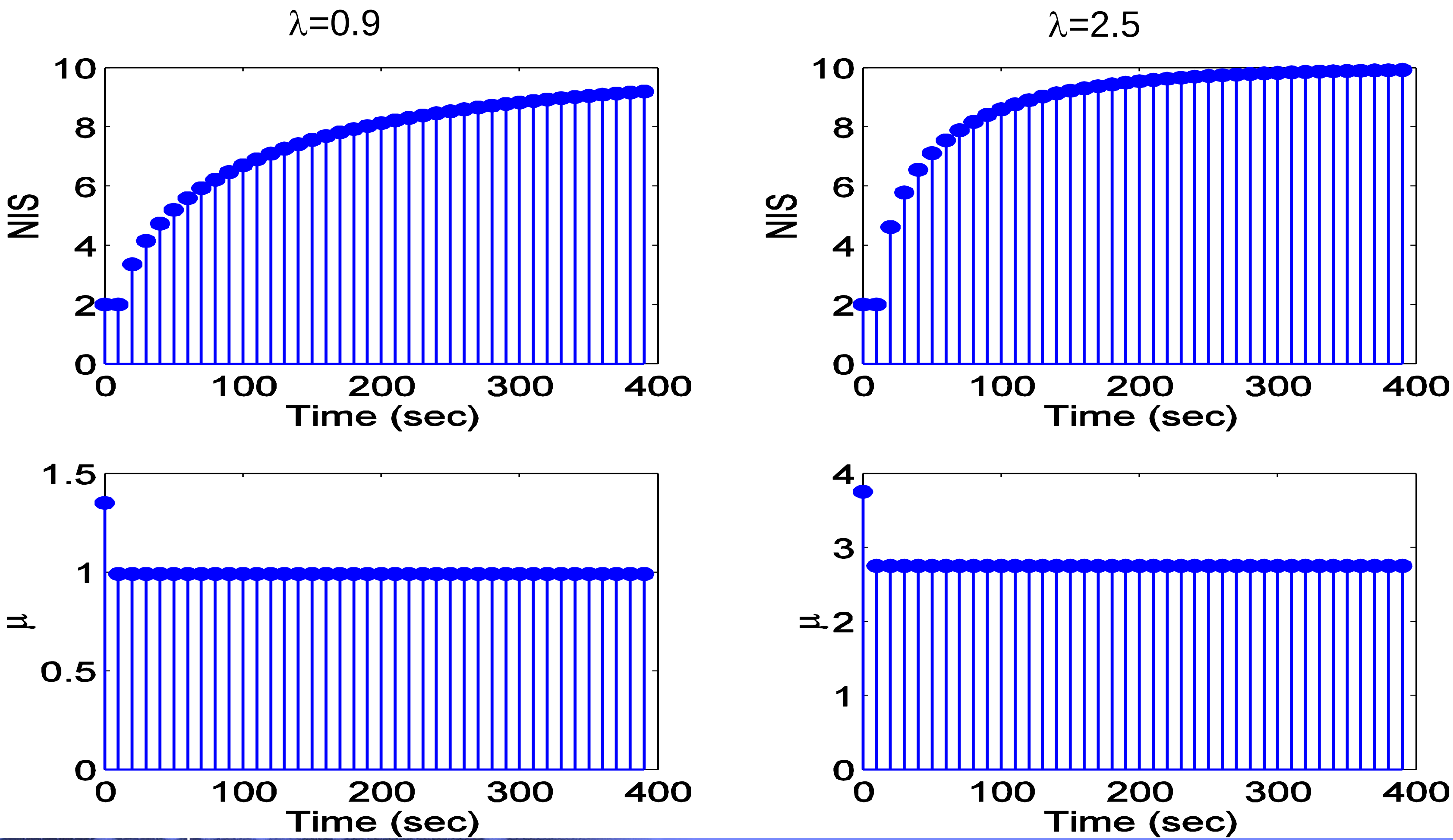
$$M/M/1 \text{ at steady state}: N(k) = \frac{\lambda(k)}{\mu(k) - \lambda(k)}$$

$$\text{So, } \mu(k) = \lambda(k) + \frac{\lambda(k)}{N(k)}$$

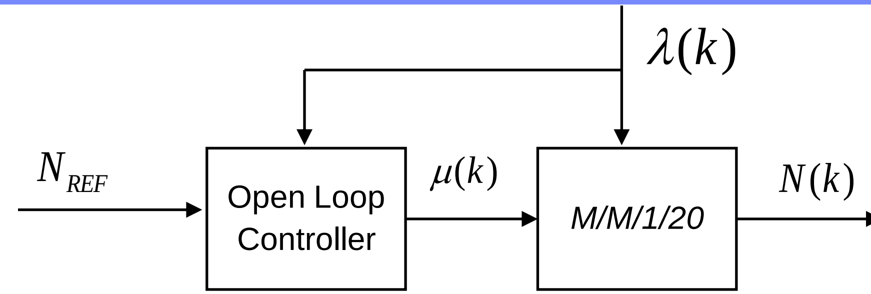
$$\text{Controller policy}: \mu(k) = \lambda(k) + \frac{\lambda(k)}{N_{REF}}$$

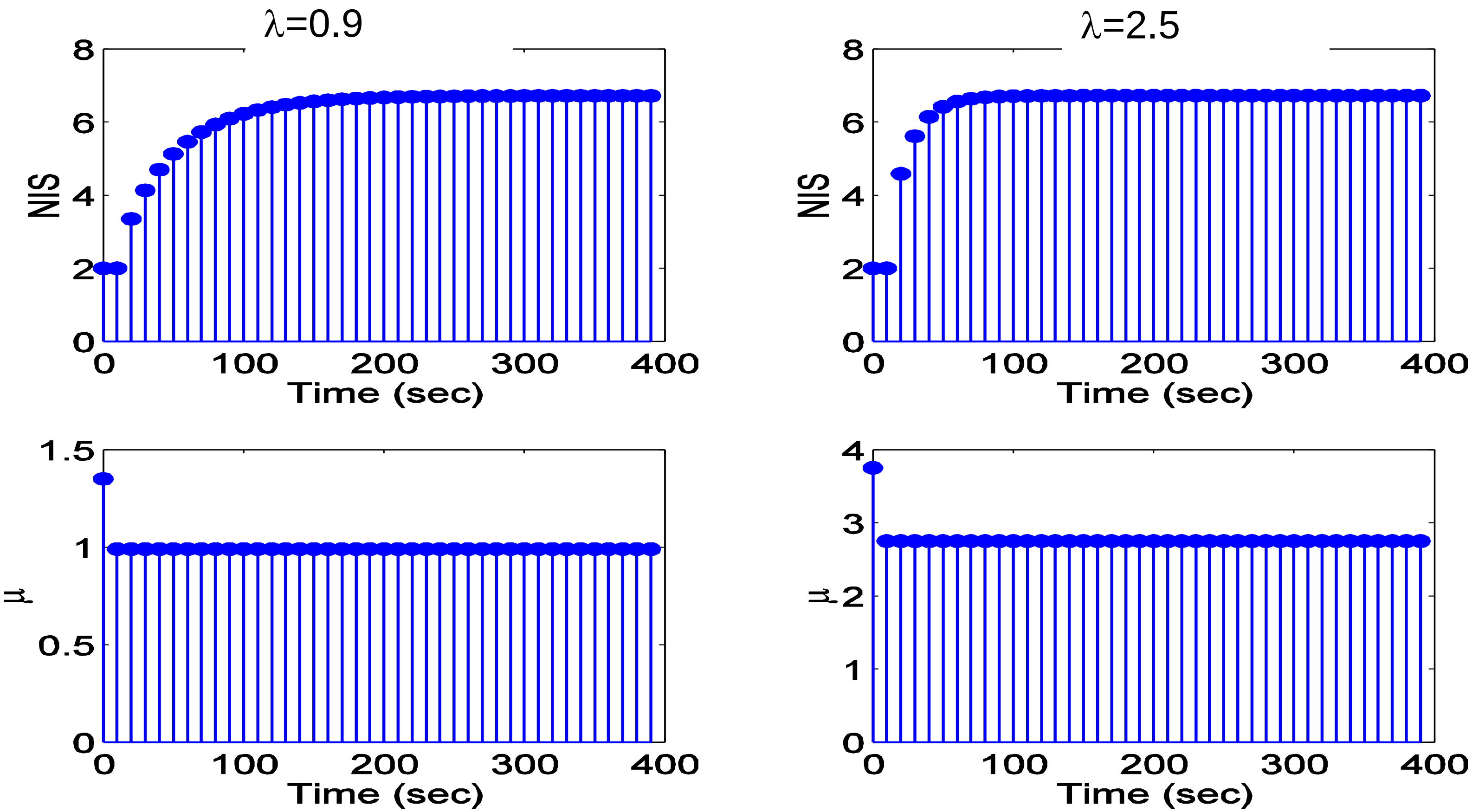
Change in Reference Input
M/M/1: 2→10





Change in Reference Input
M/M/1/20: 2→10



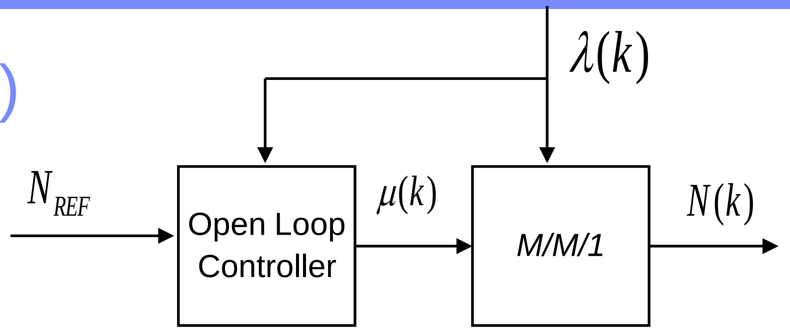


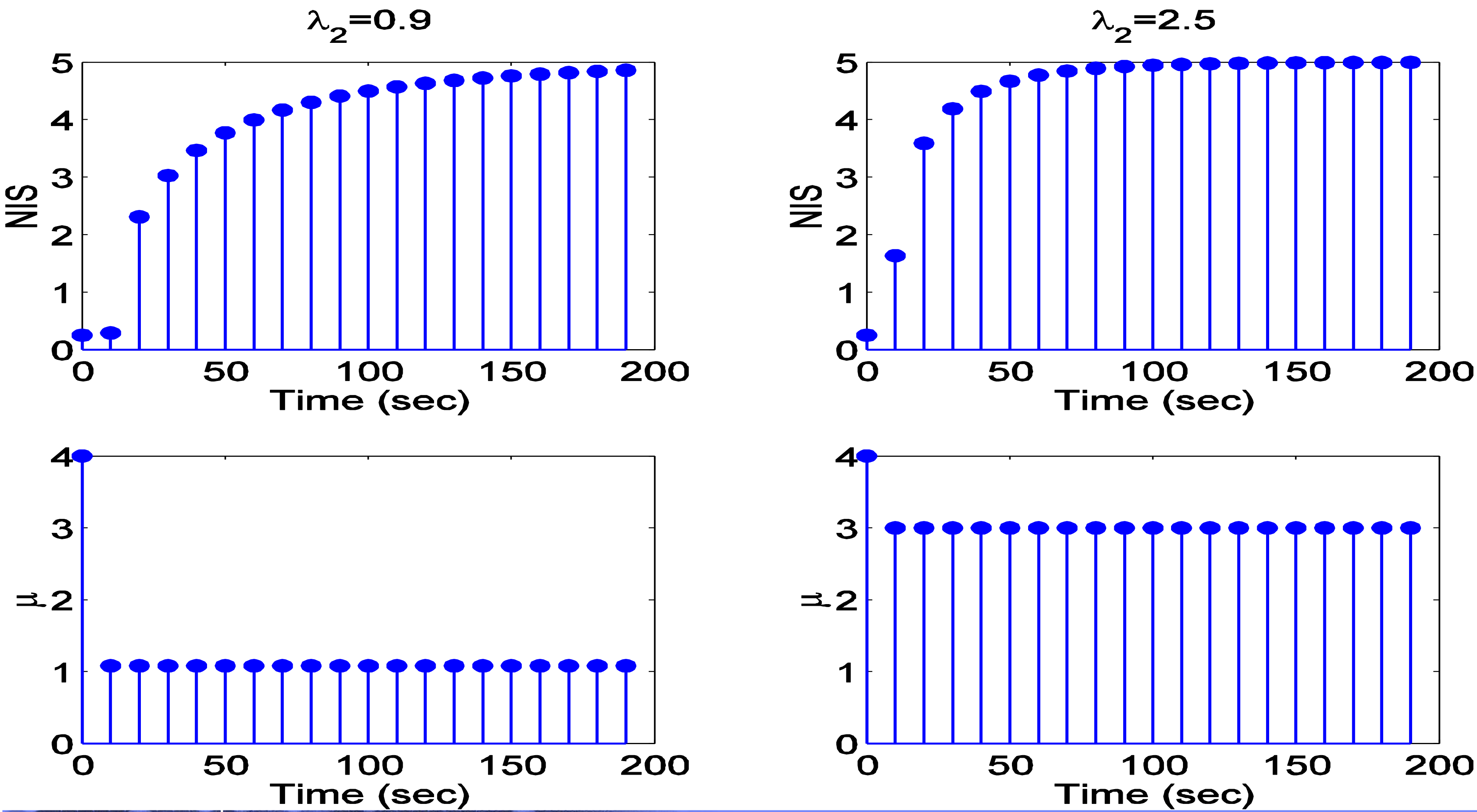
OL Controller does not achieve the desired output since uses the wrong system model

Change in Arrival Rate (Disturbance)

M/M/1: λ_1*=0.8*

N_{REF}*=5*

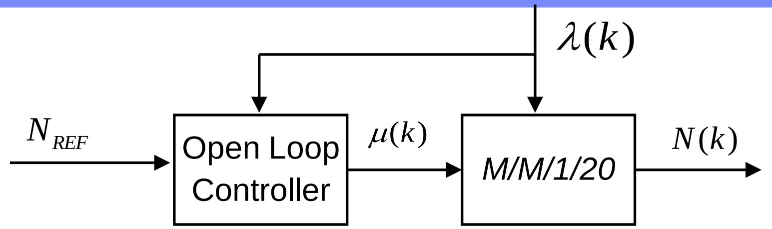


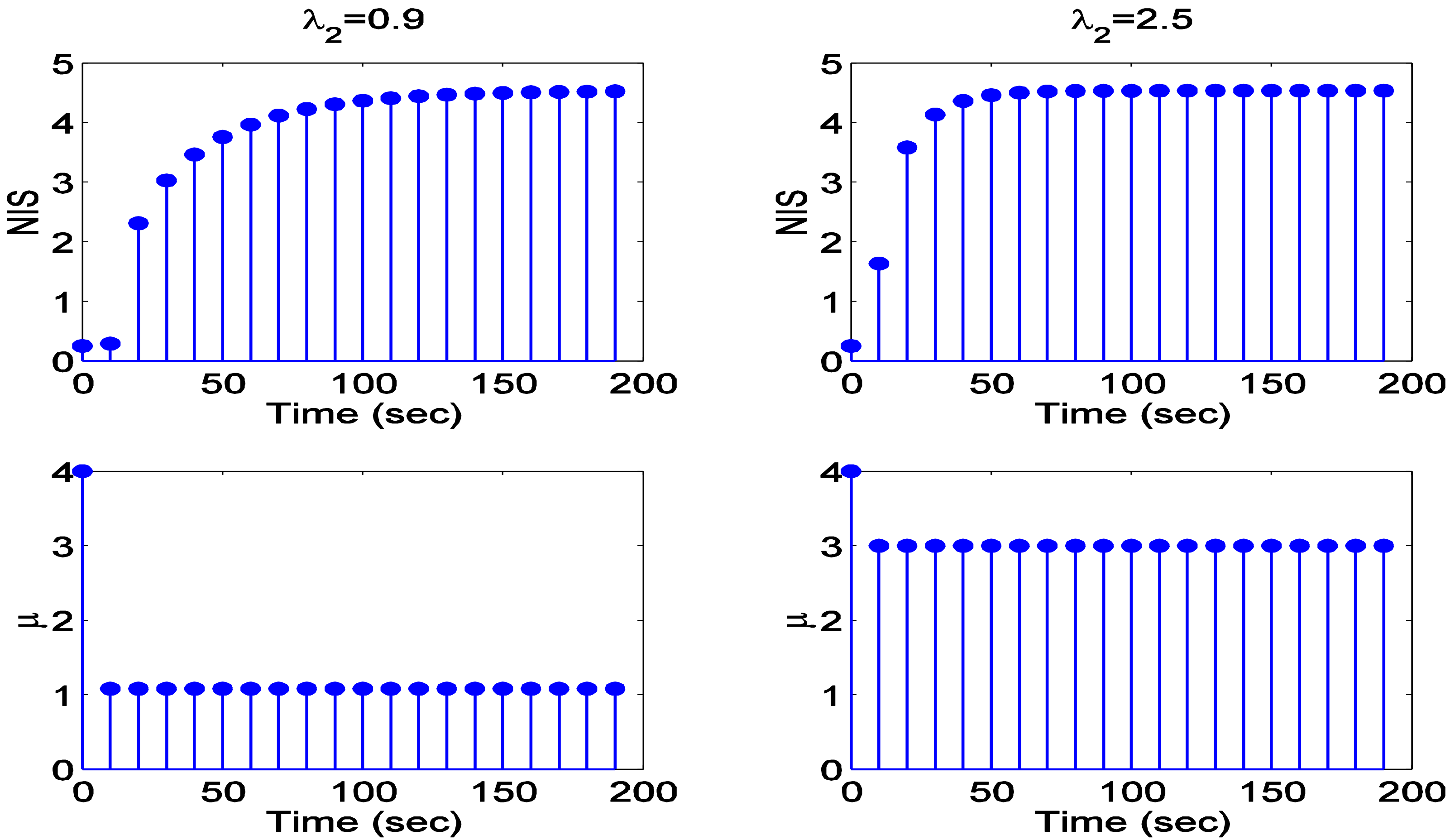


Change in Arrival Rate (Disturbance)

M/M/1/20: λ_1=0.8

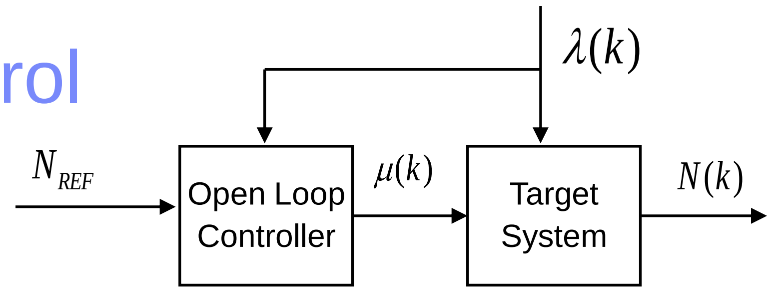
N_{REF}=5





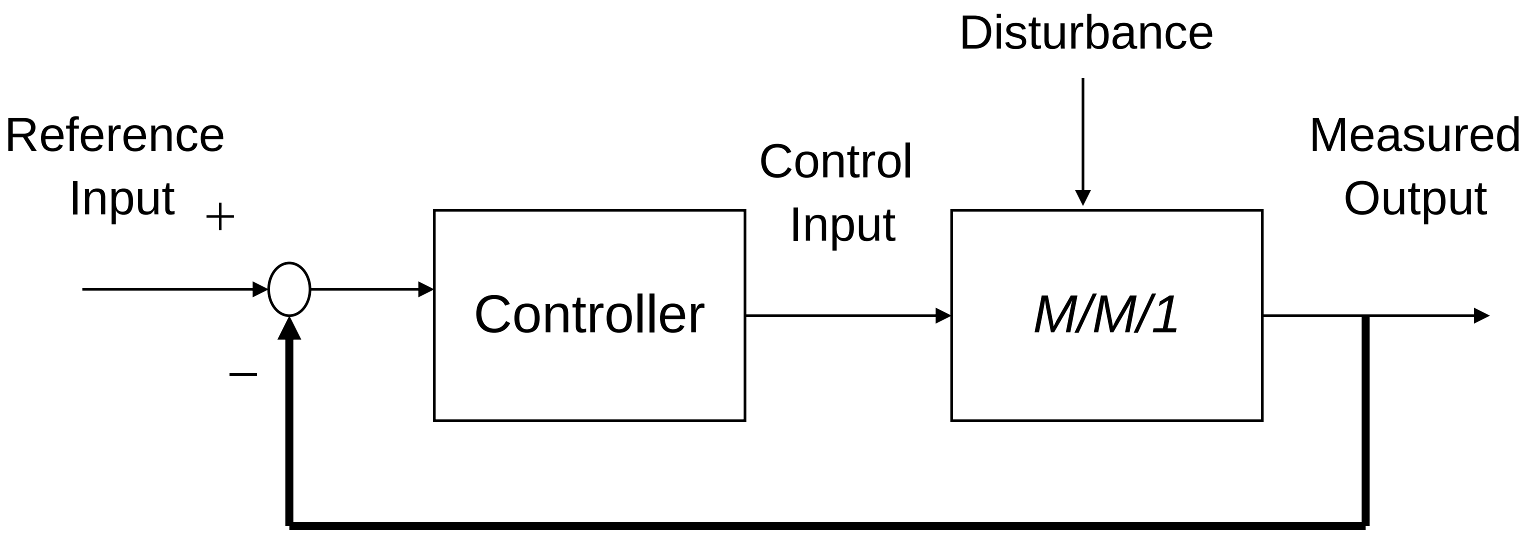
OL Controller does not achieve the desired output since uses the wrong system model

Summary of Open Loop Control



- Good if accurate model of target system
  - Short transients
  - Accurate (reference input = measured output)
- Works poorly if model of target system is inaccurate
- Issue
  - Difficult to get accurate model of target system

Closed Loop (Feedback) System for *M/M/1, M/M/1/K*



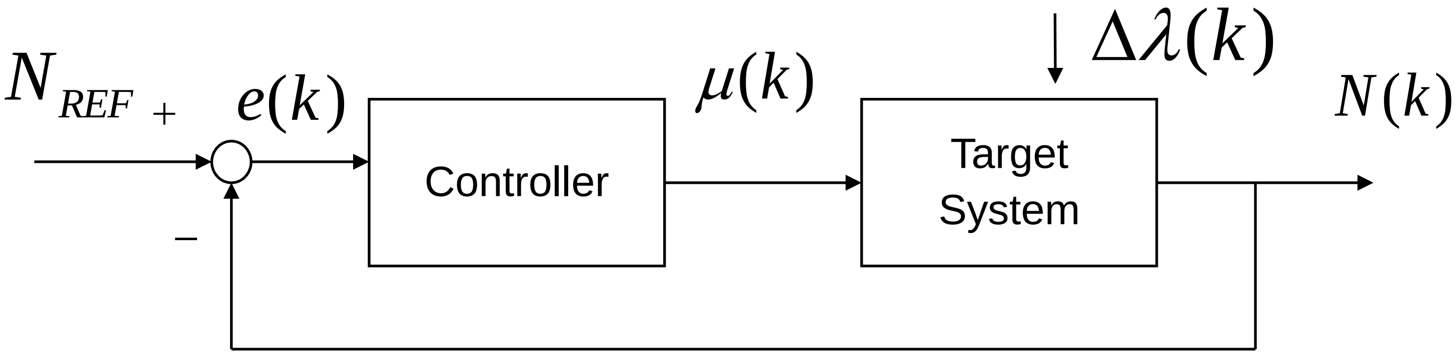
What are

- Reference input
- Control input
- Measured output
- Disturbance

Answers

- Desired number in system: N_{REF}
- Service rate: $\mu(k)$
- Measured number in system: $N(k)$
- Transition from λ_1 to λ_2.

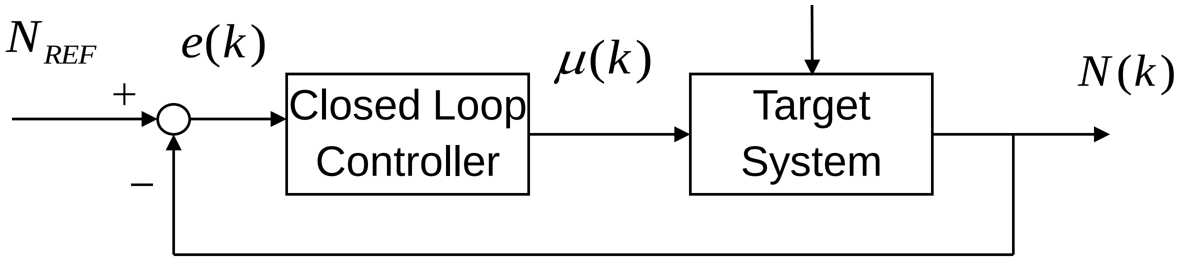
Operation of Feedback System for *M/M/1*



<u>Operation for "zeroth order" controller and *M/M/1*</u>

1. $k=0$
2. $e(k+1)=N_{REF}-N(k)$
3. $k=k+1$
4. $\mu(k)$ is obtained from Controller
5. $N(k)$ is obtained from target system
6. Goto step 2

Control Policy 2

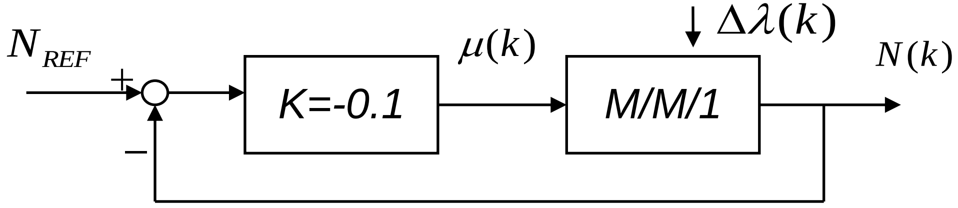


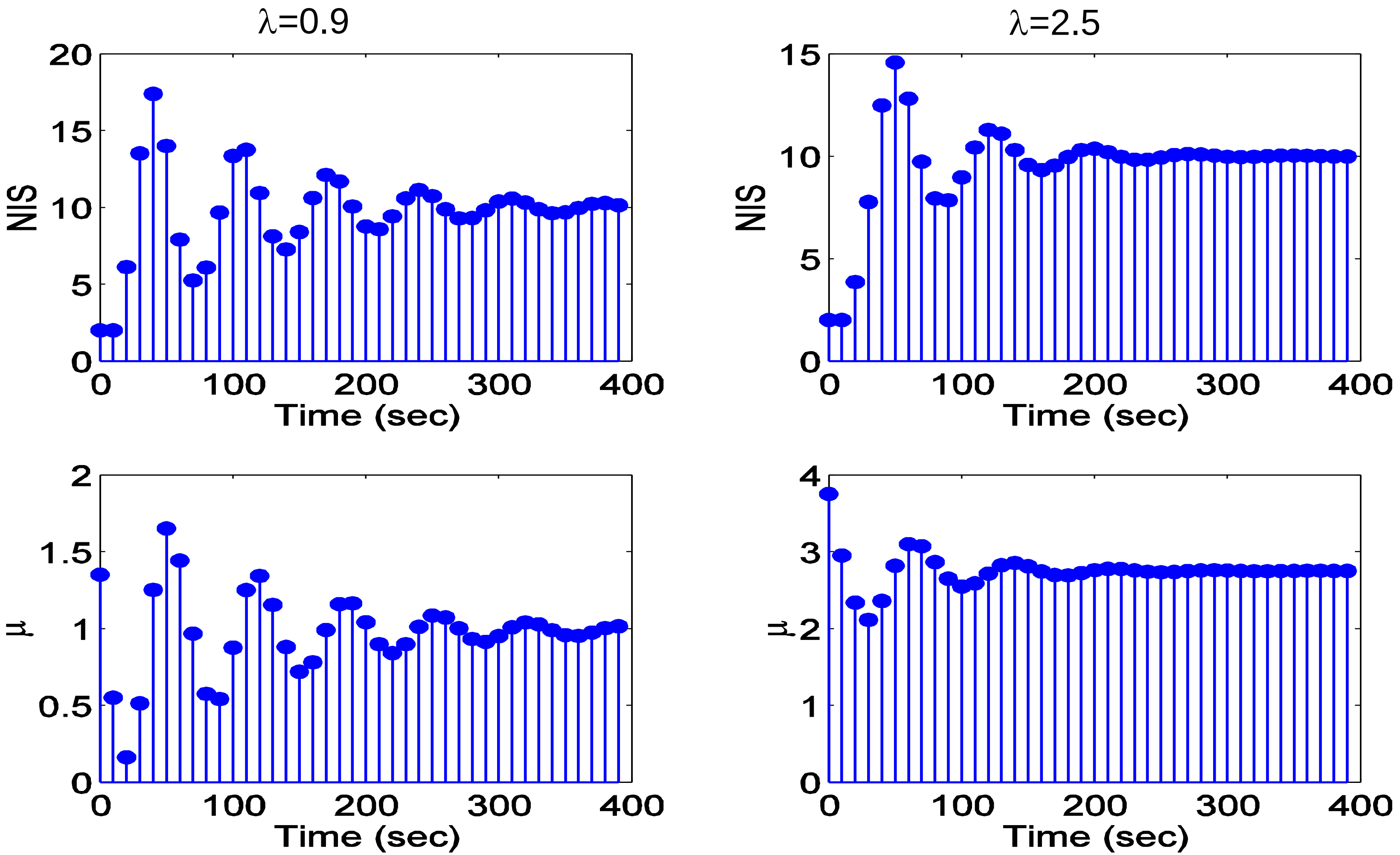
"Iterative adjusment" (integral control)

$$\mu(k) = \mu(k-1) + Ke(k)$$

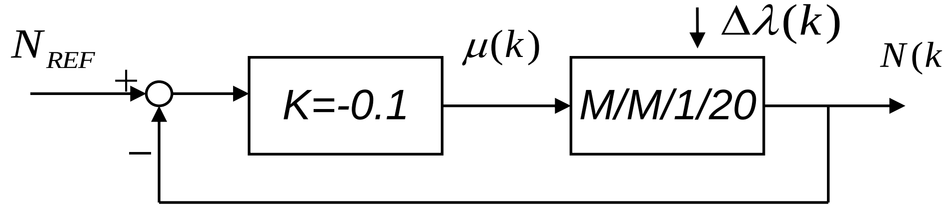
- Appeal: Does not require *M/M/1* model or measurements of *λ(k)*
- Issue: Must choose a value for *K*
- Questions
  - Should *K* be *>0 or <0?*
  - What happens if *K* is the wrong sign?

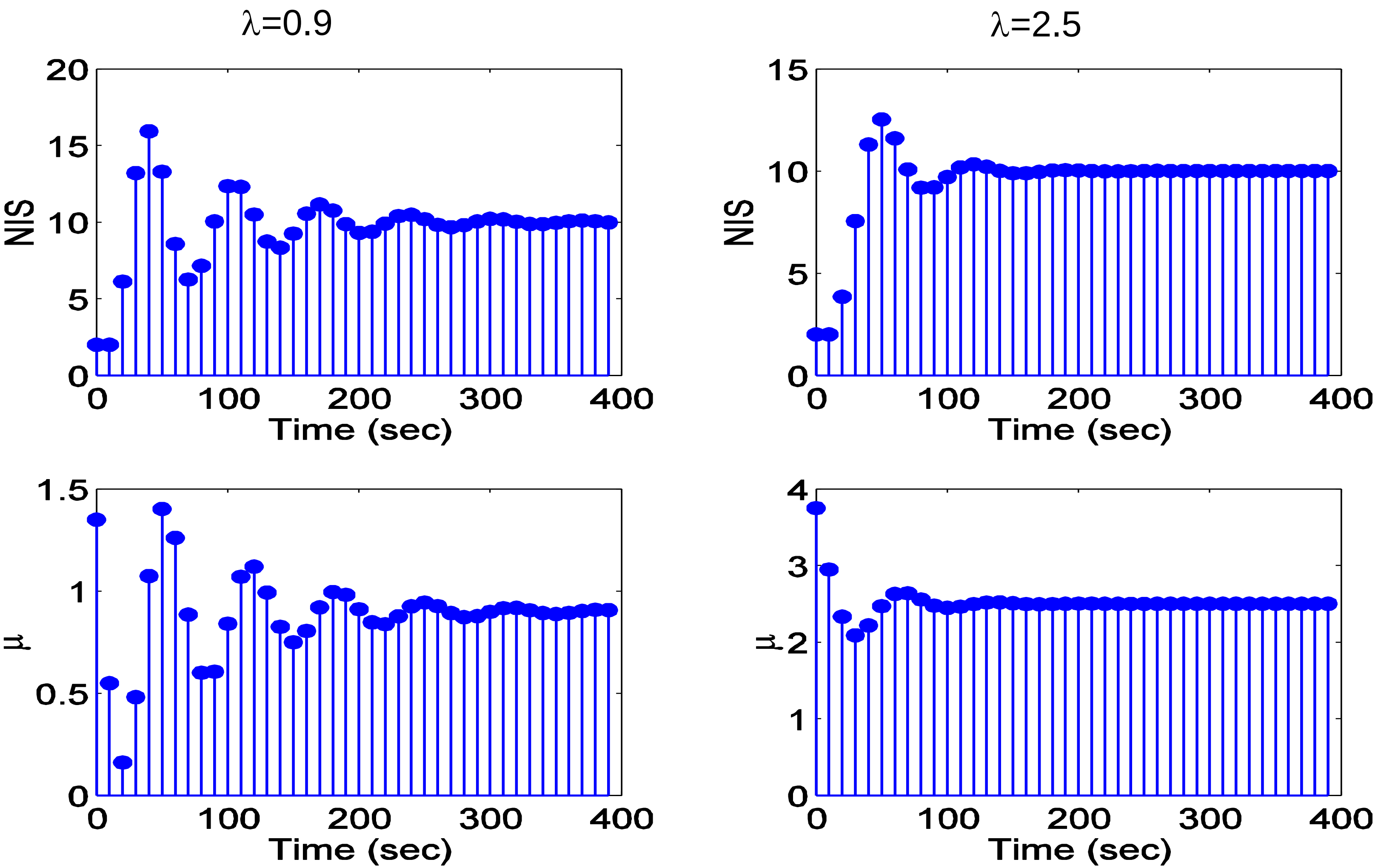
Change in Reference Input
M/M/1: 2→10





Change in Reference Input
M/M/1/20: 2→10

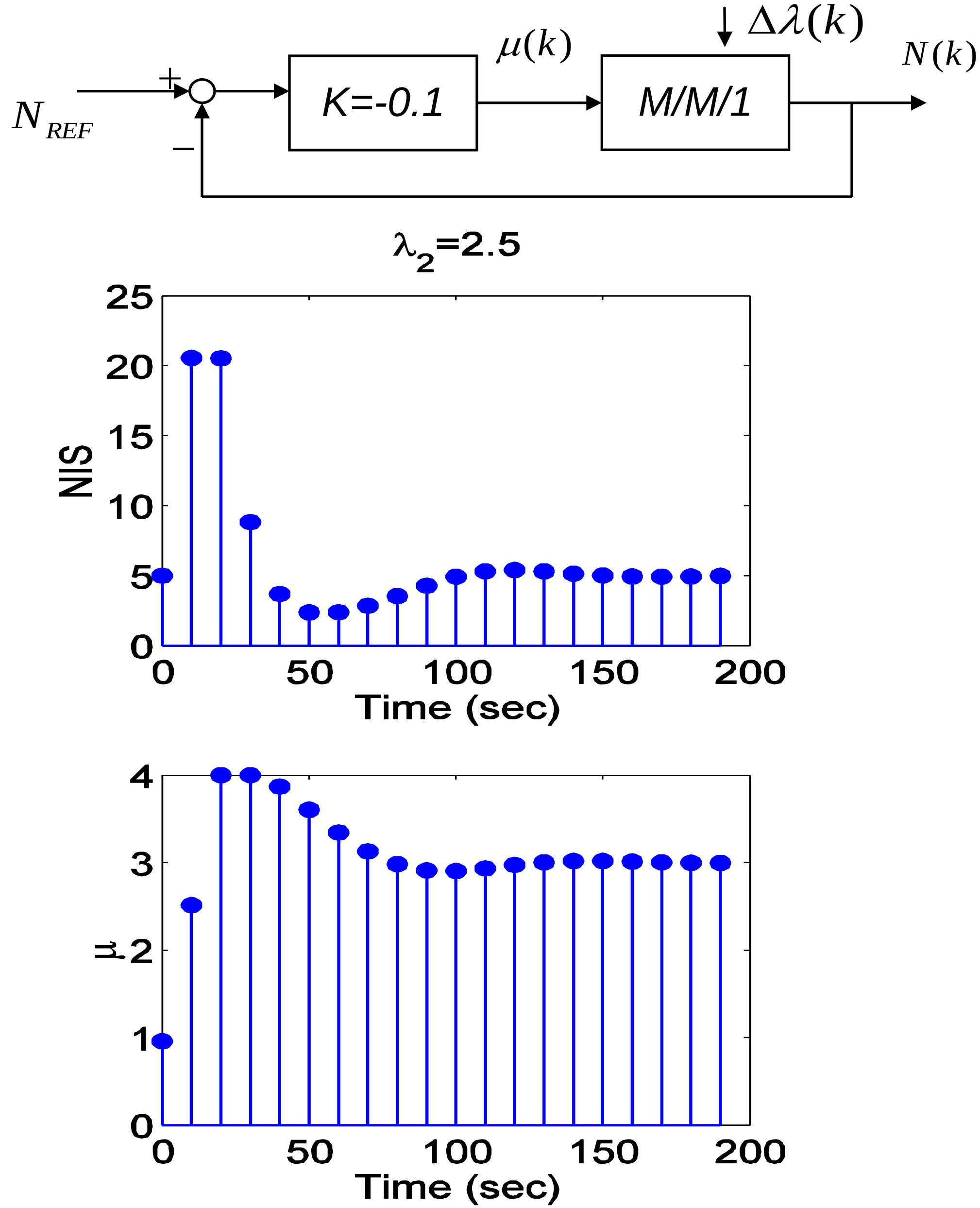




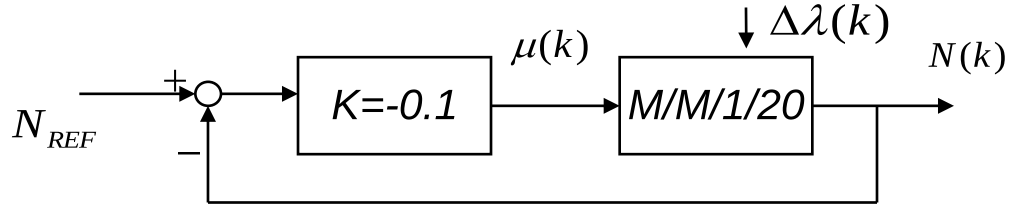
Issue: Controller still works with an inaccurate system model

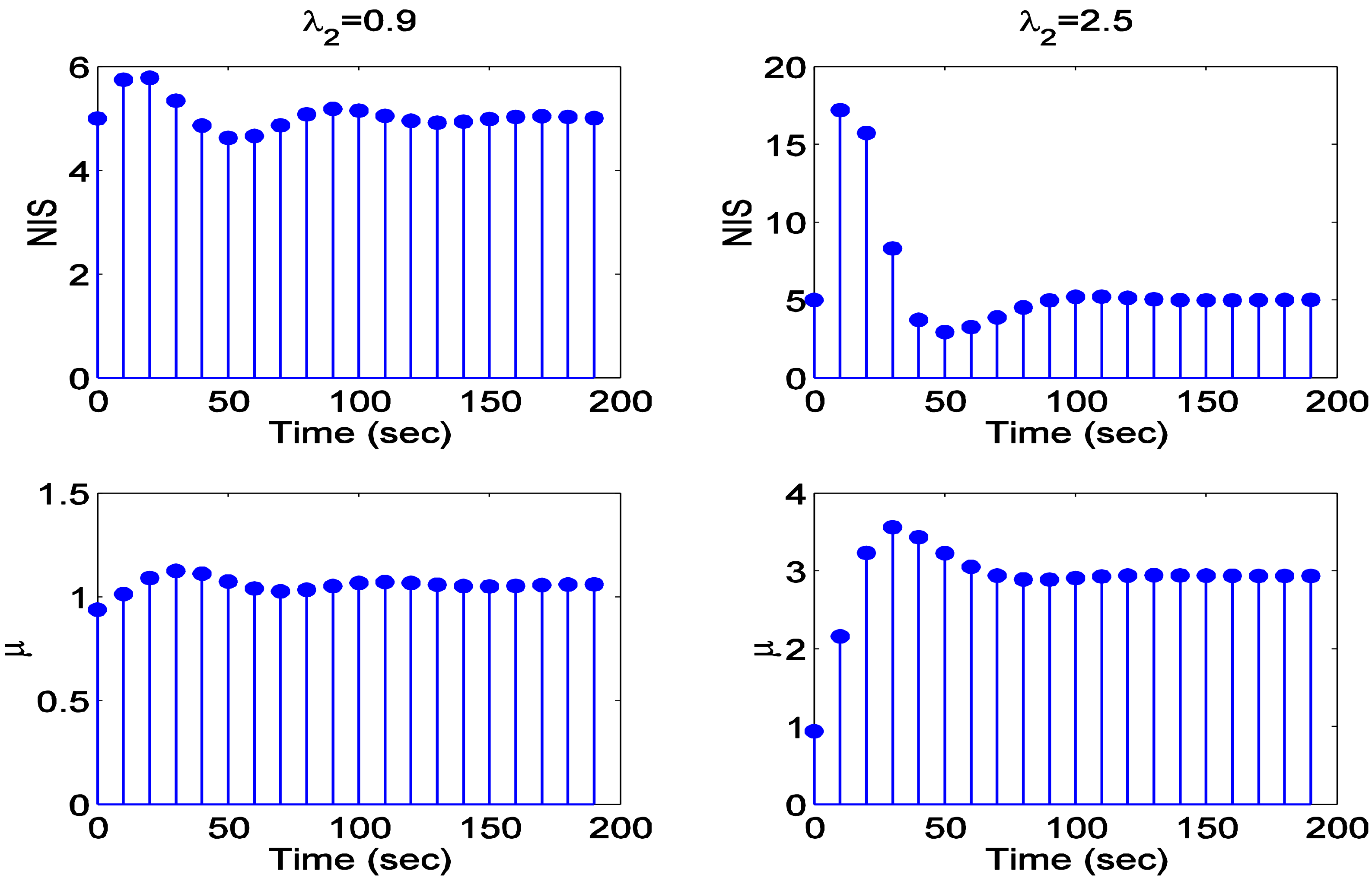
Change in Arrival Rate

M/M/1: 0.8→0.9, 2.5



Change in Arrival Rate
M/M/1/20: 0.8→0.9, 2.5





Issue: CL Controller still works with an inaccurate system model

Questions

- What is a transient? Explain the following:
  - Why the transient is longer if there is a bigger change in the arrival rates
  - Why transients are shorter in M/M/1/K than in M/M/1
- When can open loop control be used?
- How does feedback control avoid requiring an accurate model of the target system?
- How does the choice of K affect the performance of the Integral controller?