Indications in black = Optional elements

Recent Advances in the Application of Control Theory to Network and Service Management

Tarek Abdelzaher, University of Illinois
Yixin Diao, IBM Research
Joseph L. Hellerstein, Google
Chenyang Lu, Washington University
Sharad Singhal, Hewlett Packard Laboratories

June 1, 2009

Indications in black = Optional elements

Tutorial Agenda

- Control theory fundamentals (30 min)
- Managing power and performance in data centers (35 min)
- Self-tuning memory management in IBM’s DB2 (35 min)

BREAK

- Control of real-time systems using model-predictive control (35 min)
- Automated workload management in virtualized data centers (35 min)
- Research challenges (30 min)

Indications in black = Optional elements

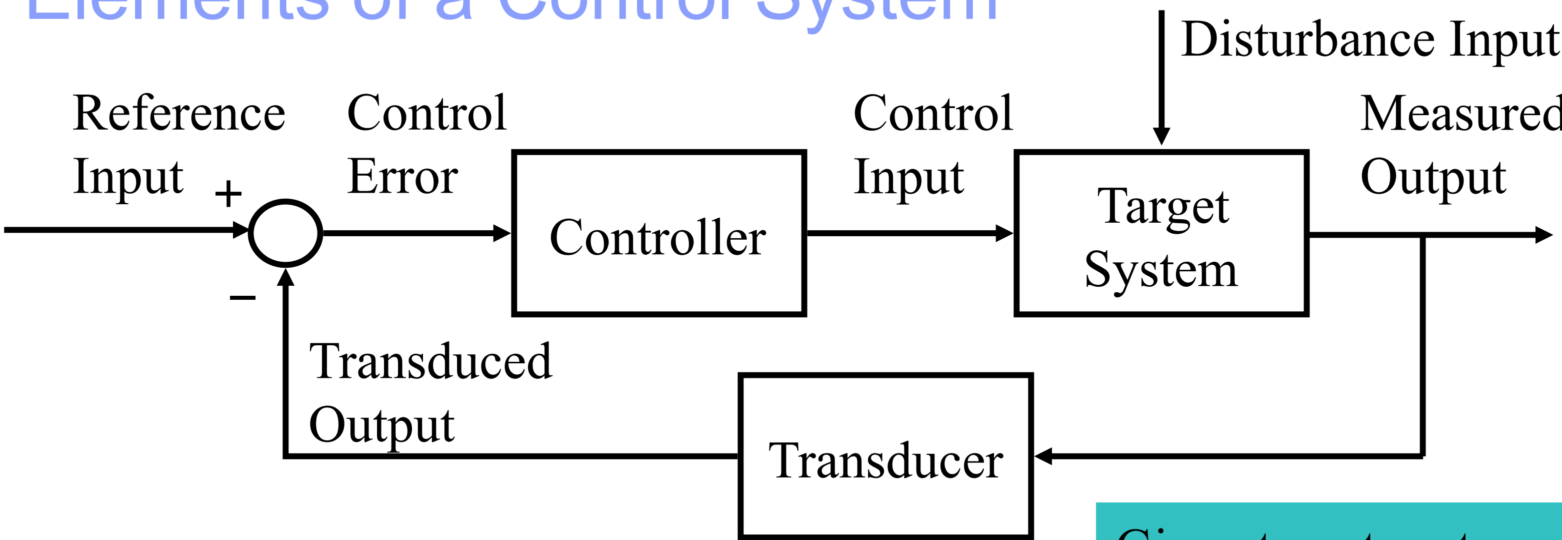
Tutorial Agenda

- Control theory fundamentals
  - Control architecture and taxonomy
  - Simple analytics
  - Application summaries
    - Regulating load for IBM's Lotus Domino email server
    - Optimizing throughput for Microsoft's .NET thread pool
- Self-tuning memory management in IBM's DB2
- Control of real-time systems using model-predictive control
- Automated workload management in virtualized data centers
- Managing power and performance in data centers
- Research challenges

Optional slide number:

Indications in black = Optional elements

Elements of a Control System



Components

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

Data

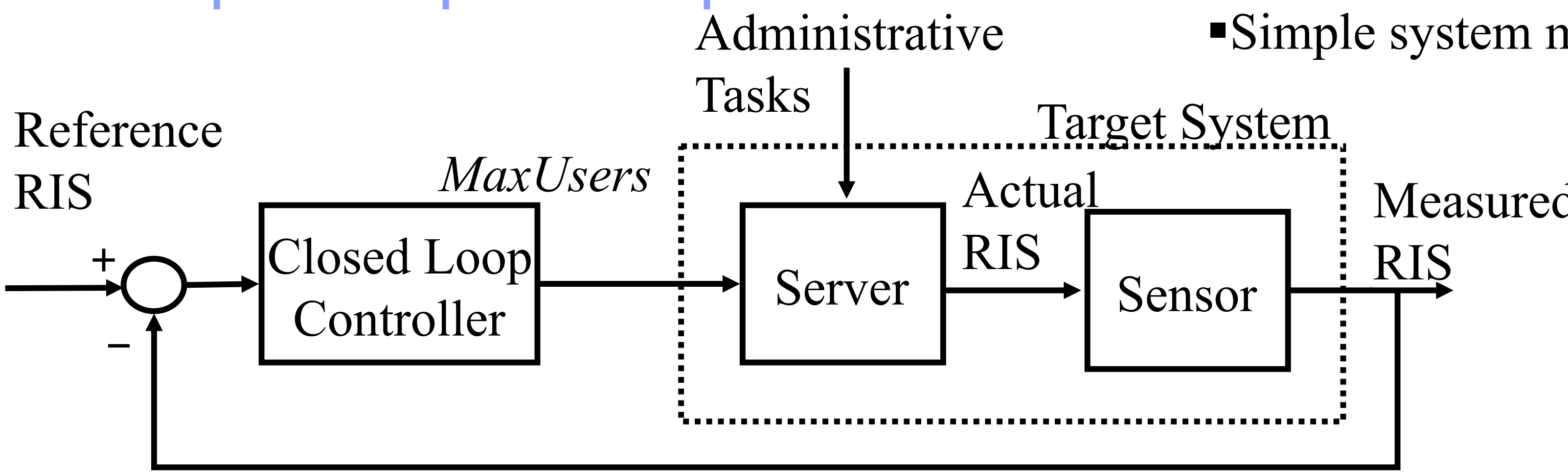
- Reference input: objective
- Control error: reference input minus measured output
- Control input: manipulated to affect output
- Disturbance input: other factors that affect the target system
- Transduced output: result of manipulation

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

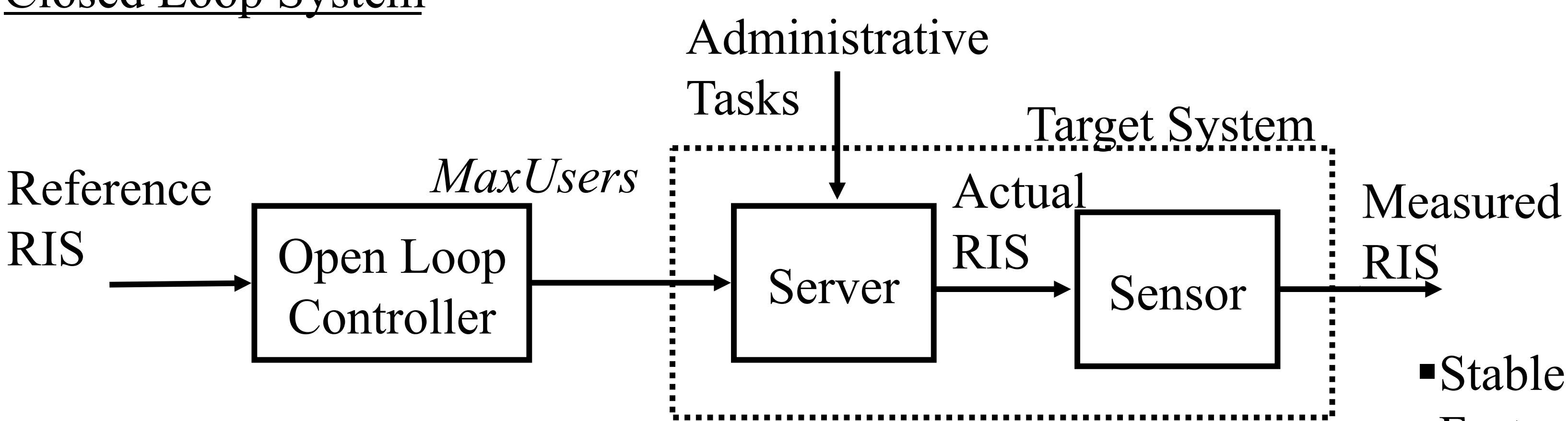
Indications in black = Optional elements

Closed Loop vs. Open Loop

- Adapts
- Simple system model



Closed Loop System



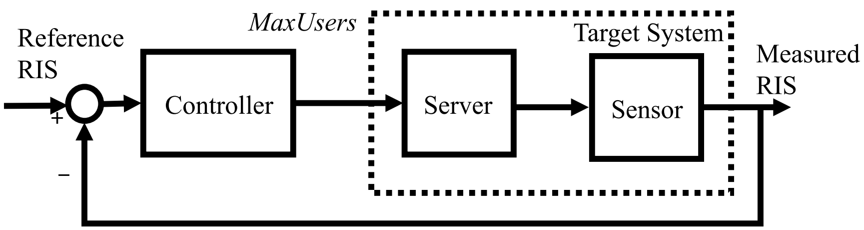
- Stable
- Fast settling

Open Loop System

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal

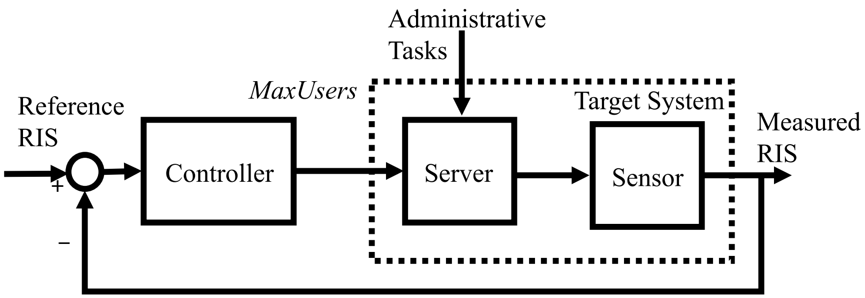
Indications in black = Optional elements

Types of Control



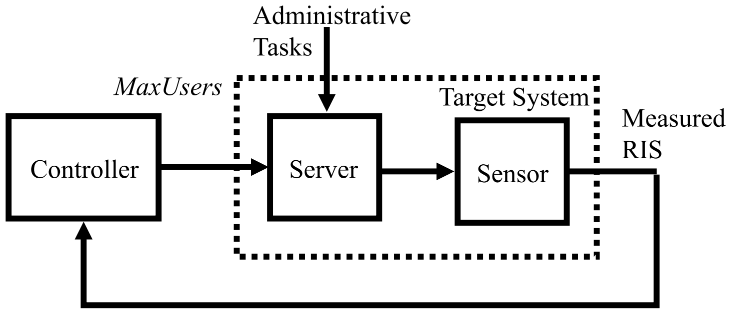
- Manage to a reference value
- Ex: Service differentiation, resource management, constrained optimization

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

IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal

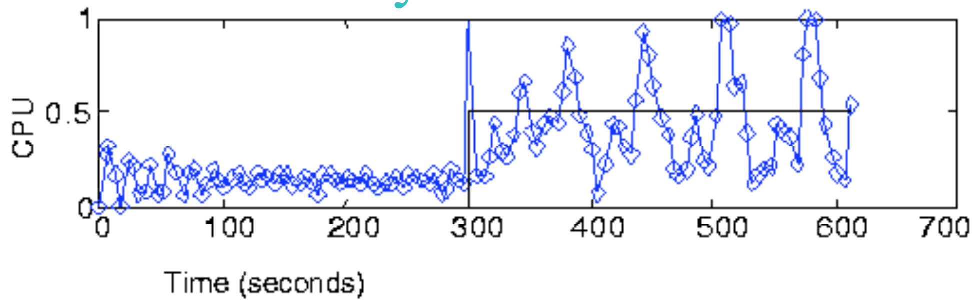
Optional slide number:

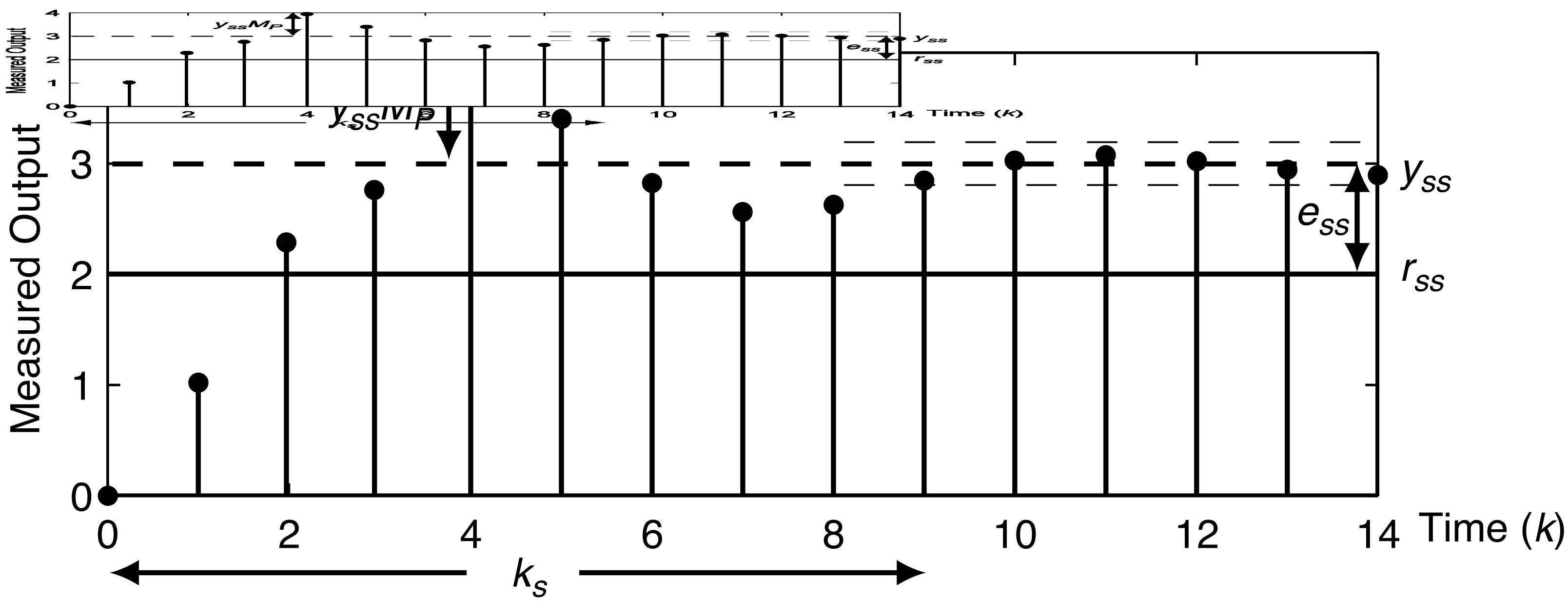
Indications in black = Optional elements

The SASO Properties of Control Systems

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

Unstable System

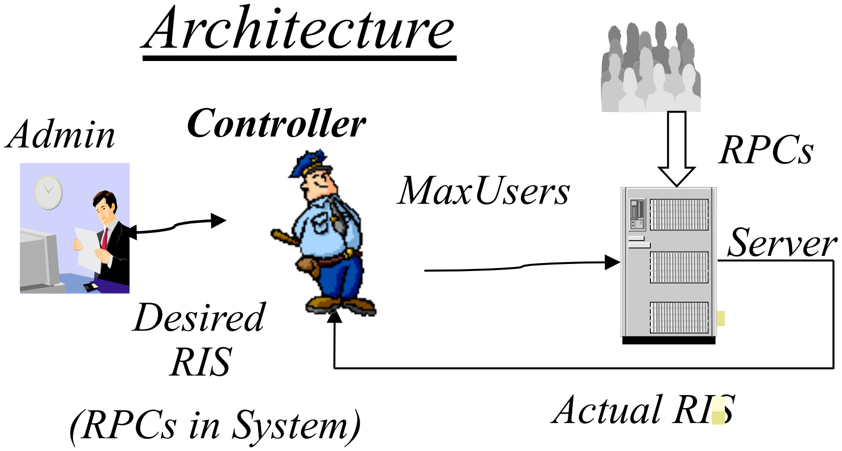


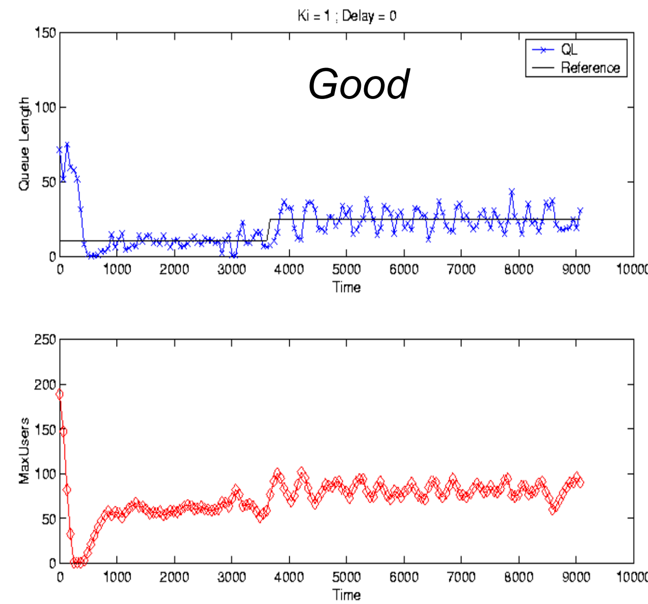


IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal

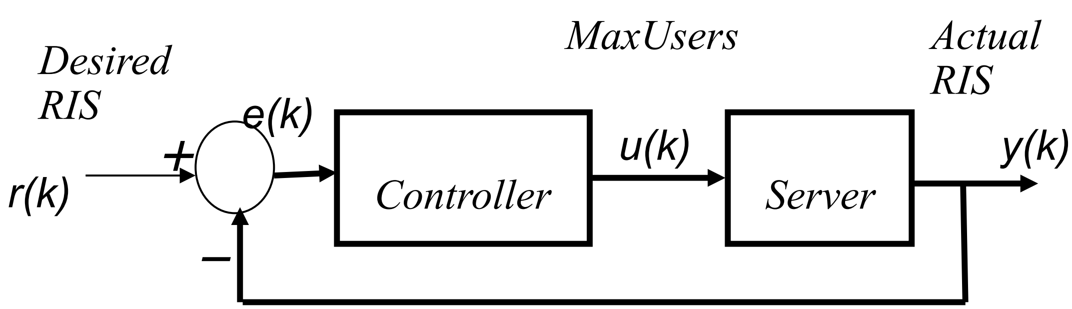
Control Theory By Example – IBM Domino Server

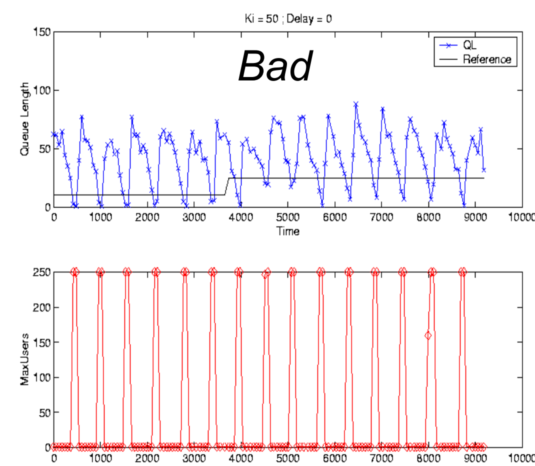
Architecture





Block Diagram

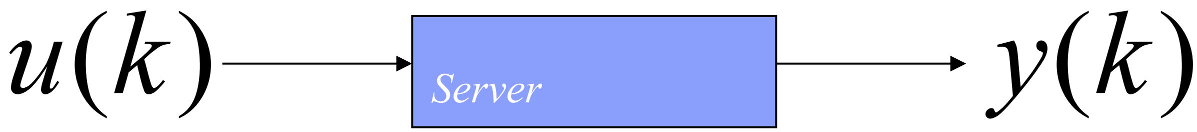




IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal

Dynamical Analysis of Discrete Time Systems

$$y(k+1) = ay(k) + bu(k)$$



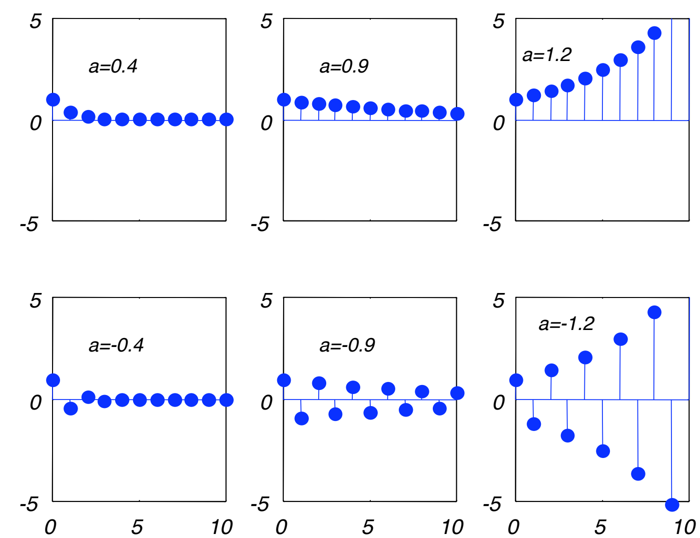
Z-Transform

$$y(k+1) = ay(k) + bu(k) \Leftrightarrow zY(z) = aY(z) + bU(z)$$

Transfer Function (TF)

$$F(z) = \frac{Y(z)}{U(z)} = \frac{b}{z-a} \Leftrightarrow (ba^0, ba^1, ba^2, \ldots$$

Pole: *Output at time k is proportional to a^k, for pole a.*

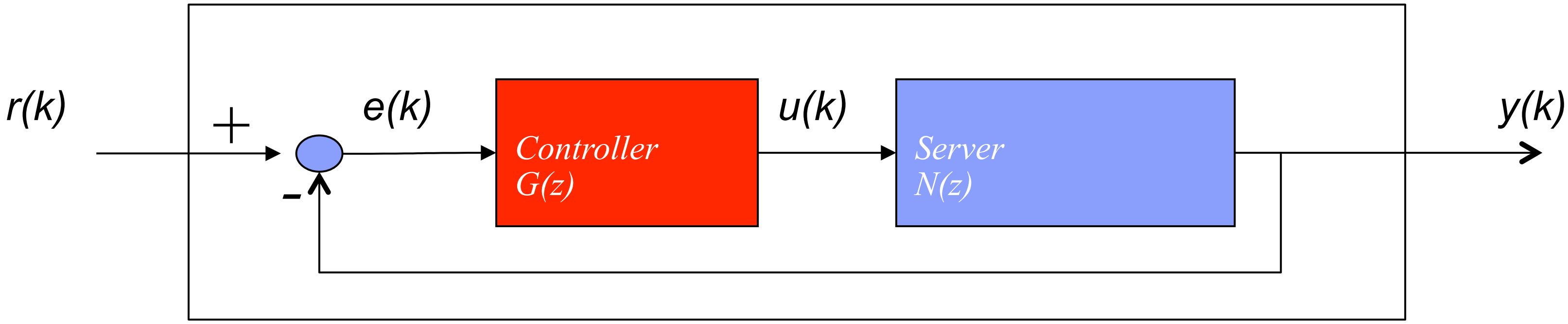


Fast systems have small poles

Oscillations result if neg or im poles

Gain: *Ratio of steady state output to steady state input*

Control Design

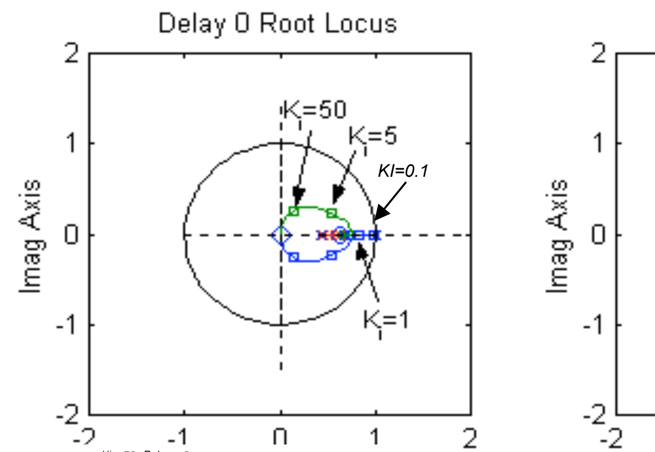


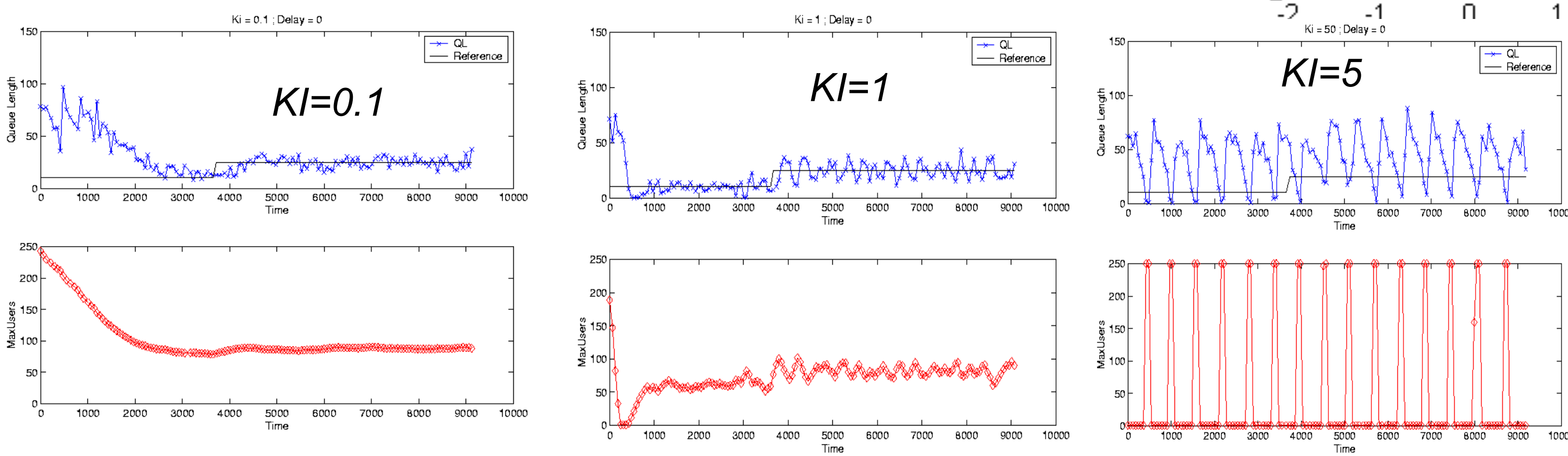
$$F(z) = \frac{Y(z)}{R(z)} = \textit{Closed Loop Transfer Function}$$

<u>*Example: Control Law*</u>

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

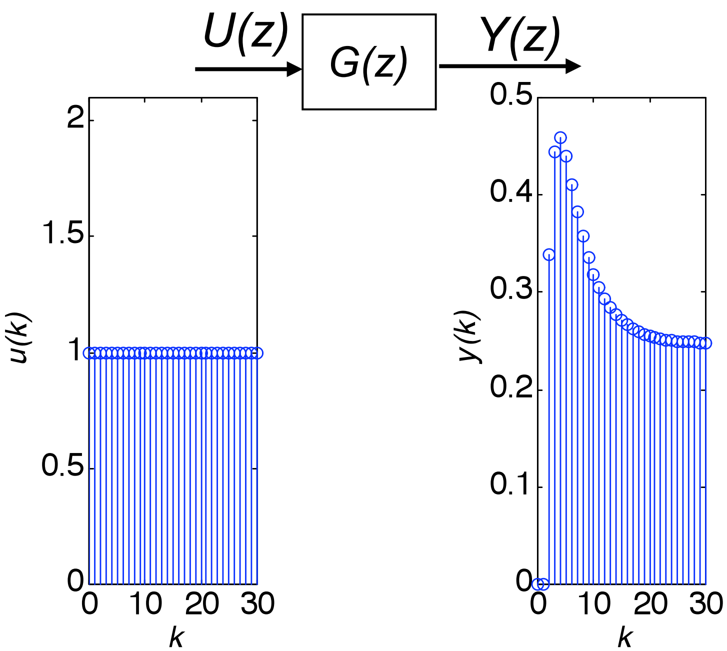
Poles of FR(z)





Indications in black = Optional elements

Key Results From Linear Systems

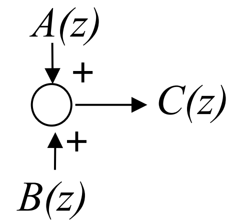


Stable if $|a|<1$, where a is the largest pole of $G(z)$

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

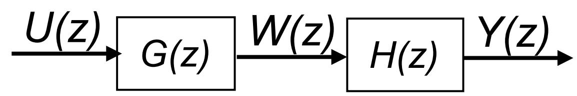
Steady state gain of $G(z)$: $\dfrac{y(\infty)}{u(\infty)} = G(1)$

Adding signals:

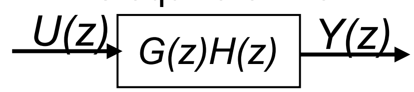


$\{c(k)=a(k)+b(k)\}$ has Z-Transform $A(z)+B(z)$.

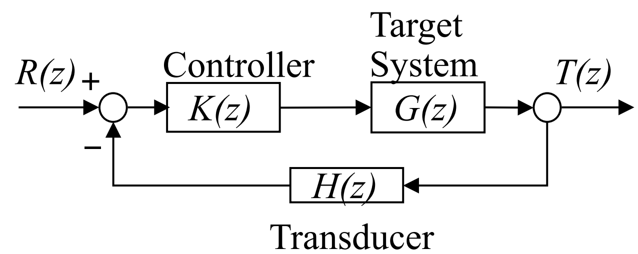
Transfer functions in series



is equivalent to

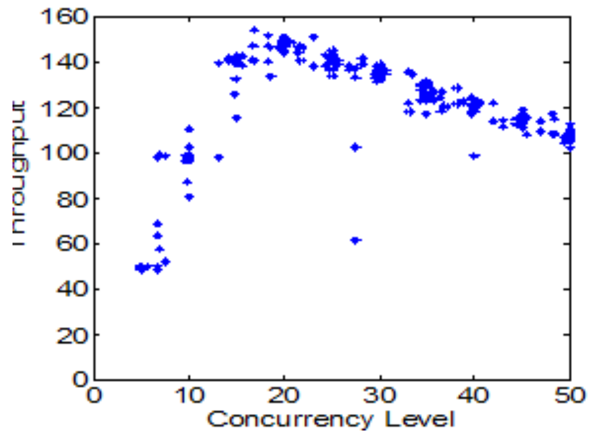


Transfer function of a feedback loop



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

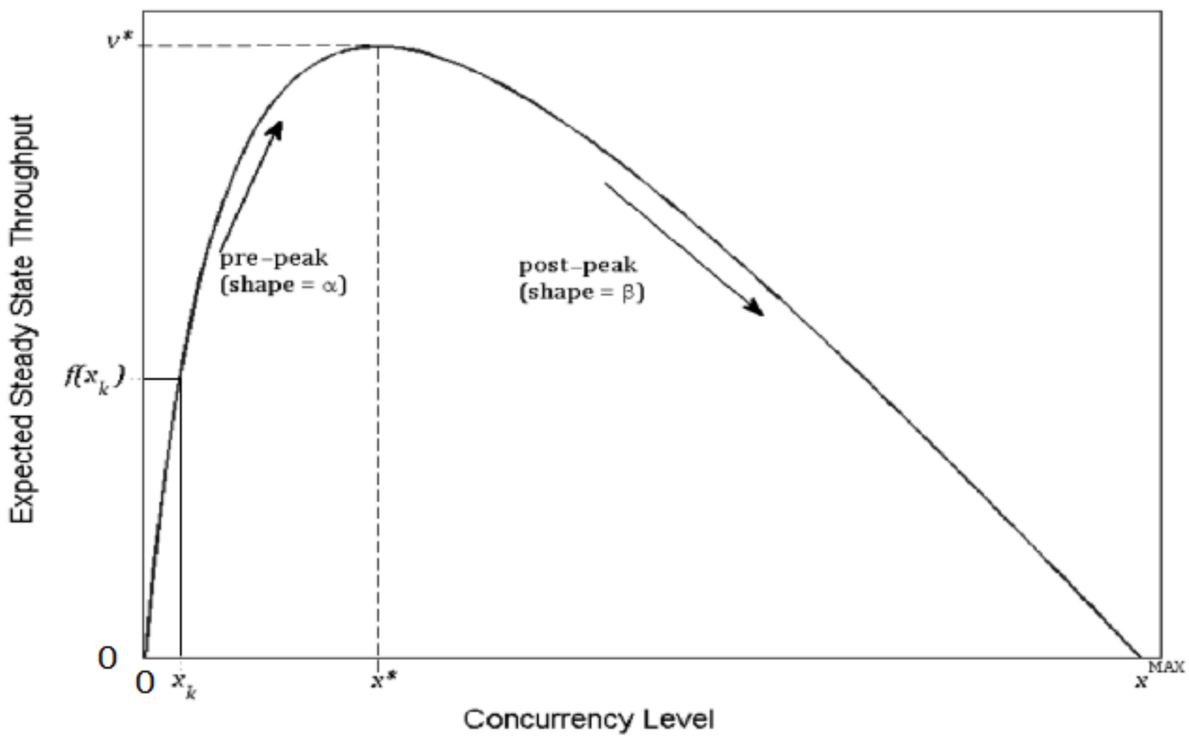
Optimizing the Microsoft .NET Thread Pool



- Optimization Objective
  - Choose concurrency level that maximizes throughput
- Intuition: Use “hill climbing”
- But specifics (and effectiveness) depend on
  - Curve shape: concave vs. just unimodal?
  - Transients when concurrency level changes
  - Measurement variability

Indications in black = Optional elements

Steady State System Identification Model



Properties

$$f_a(x^*) = v^* = f_b(x^*)$$

$$\frac{df_a}{dx} > 0 \; if \; \alpha > 0 \qquad \frac{df_b}{dx} < 0 \; if \; \beta > 0$$

$$f_a(f_b) \; is \; linear \; if \; \alpha = 1 \, (\beta = 1)$$

$$f_a(f_b) \; is \; constant \; if \; \alpha = 0 \, (\beta = 0)$$

Concave if $\alpha, \beta > 1$

Parameters: $x^*, x^{MAX}, v^*, \alpha, \beta$

$$\log(1 - \frac{f_a(x)}{v^*}) = \alpha \log(1 - \frac{x}{x*})$$

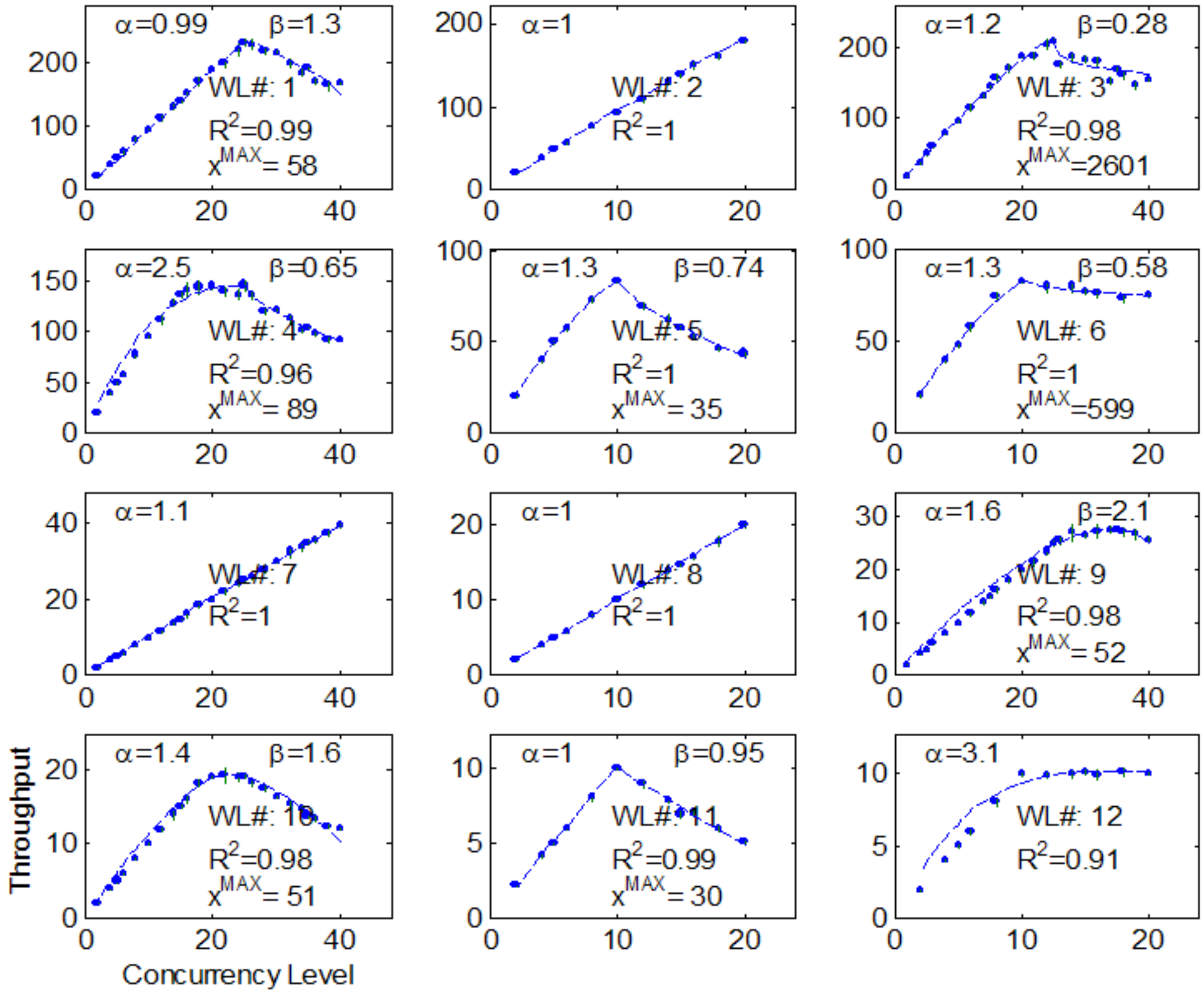
$$f_a(x) = v^*(1 - (1 - \frac{x}{x^*})^{\alpha})$$

$$f_b(x) = v^*(1 - (1 - \frac{x - x^*}{x^{MAX} - x^*})^{\beta})$$

$$f(x) = f_a(x), x \le x^*, f_b(x), x \ge x^*$$

Indications in black = Optional elements

Steady State System Identification

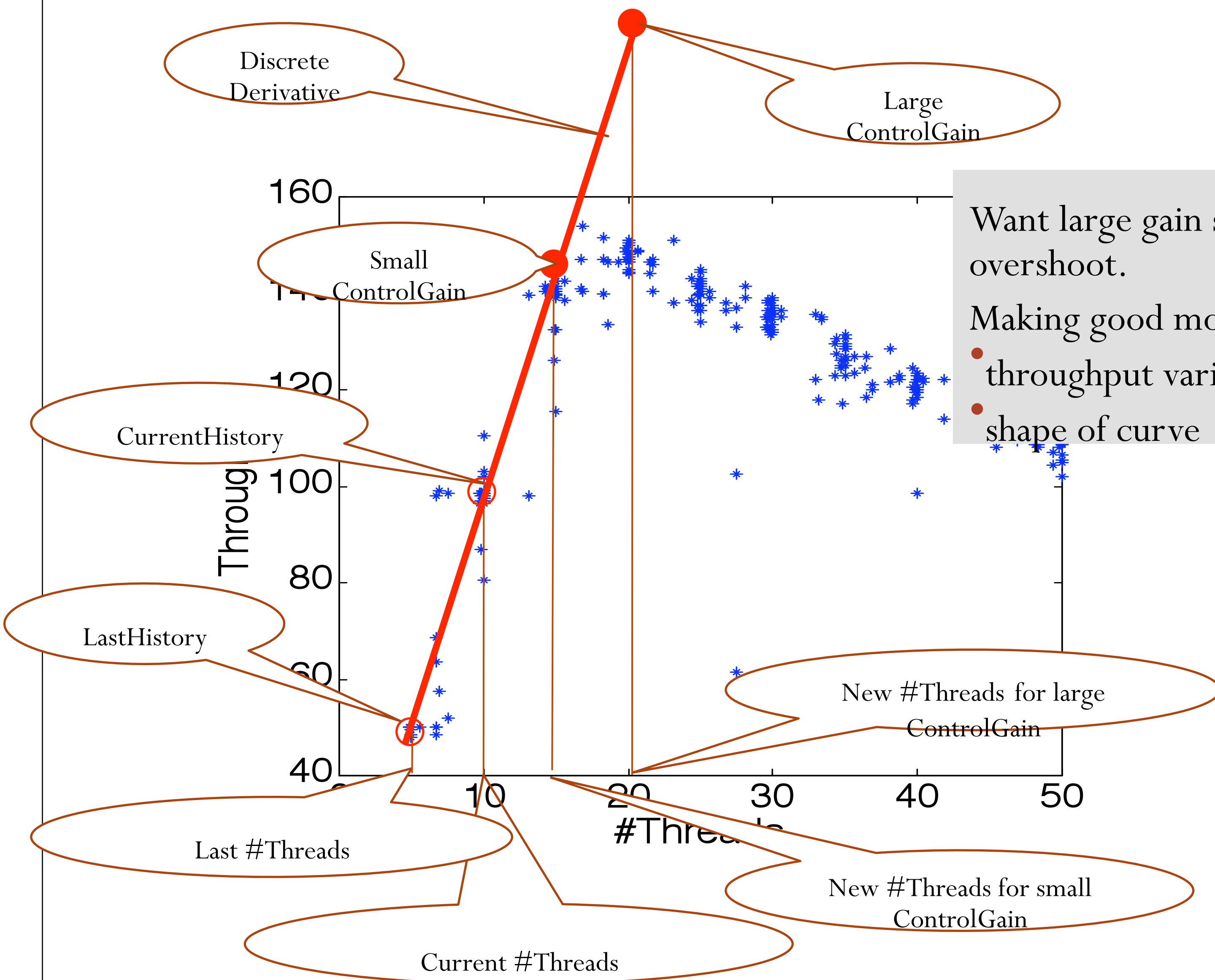


IM 2009: Recent Advances in Control Theory. Abdelzaher, Diao, Hellerstein, Lu, Singhal

Optional slide number:

Copyright: 10pt Arial

Hill Climbing Using Stochastic Gradient Approximation



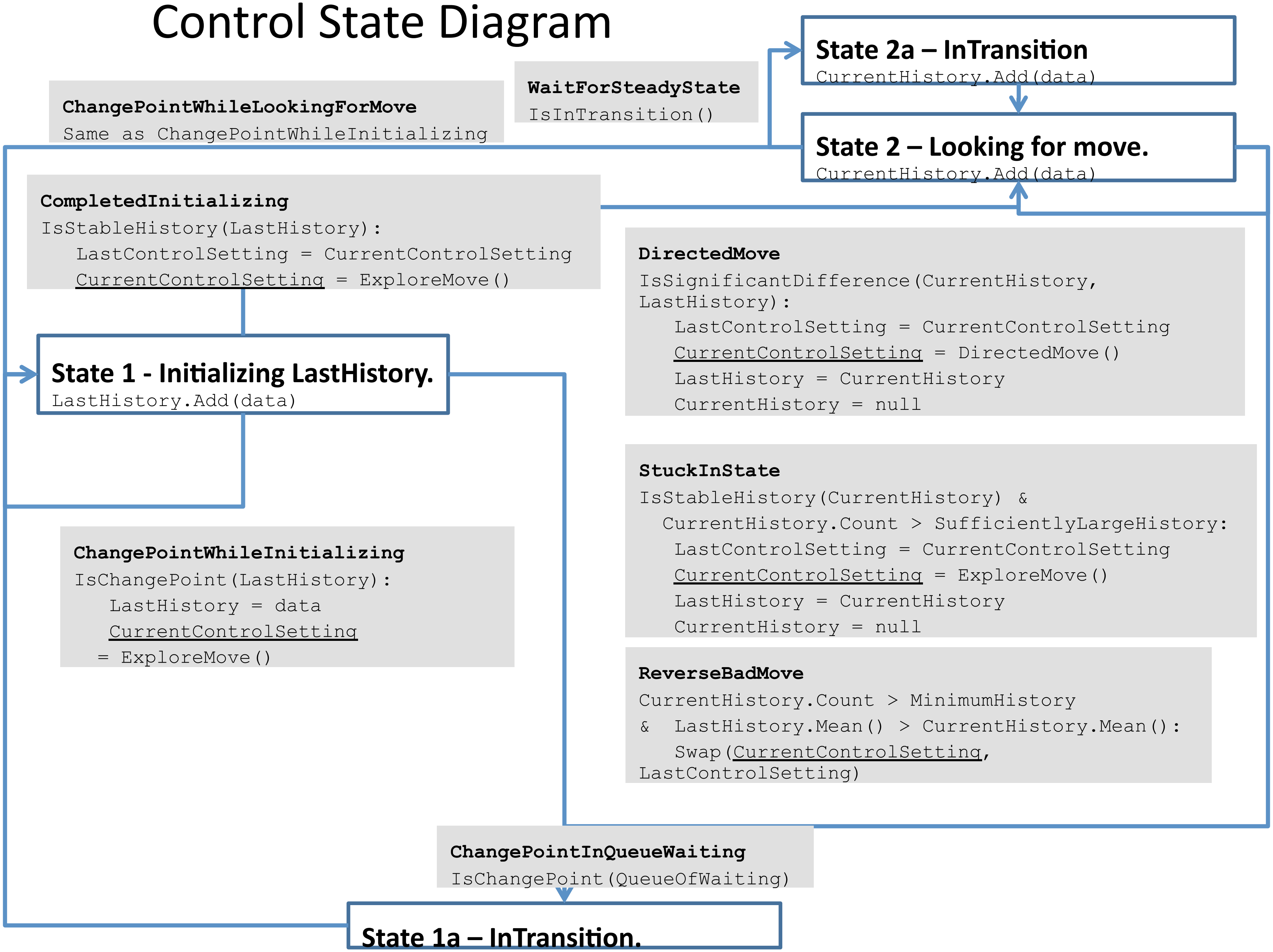
Want large gain so move quickly, but not overshoot.

Making good moves depends on

- throughput variance
- shape of curve

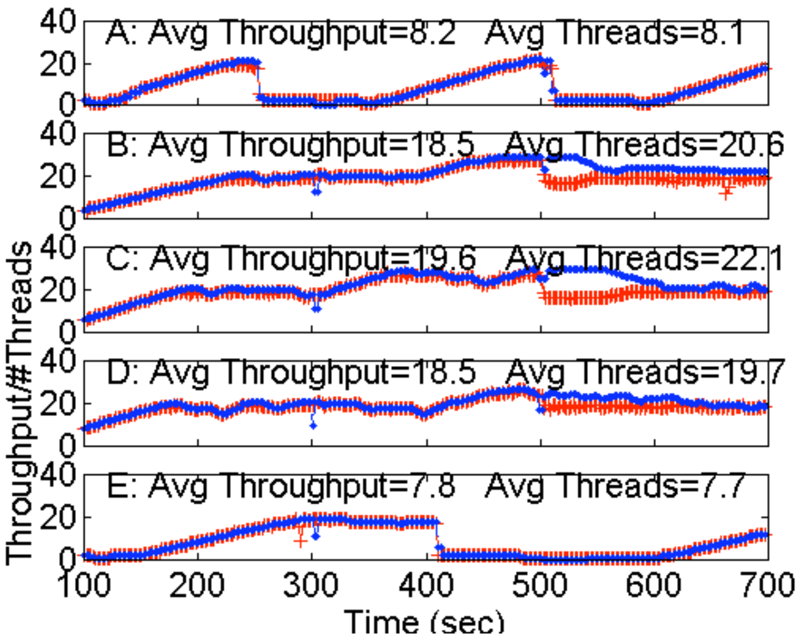
○ = History mean

Control State Diagram



Example of Controller Assessments

throughput *#threads*



Controller	thruput	#threads
A	8.2	8.1
B	18.5	20.6
C	19.6	22.1
D	18.5	19.7
E	7.8	7.7

Challenge: assess tens of thousands of cases

- Use system ID to generate synthetic workloads efficiently as concurrency-throughput curves
- Real controller operates on synthetic workloads
- Can assess optimality of controller since know peak of curve.