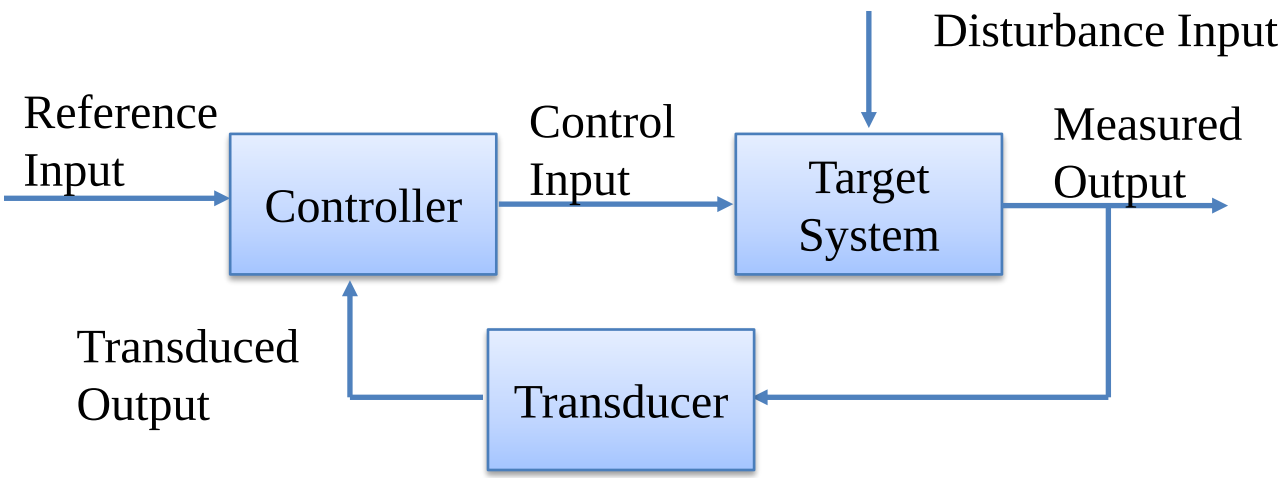
Modeling and System Identification

Joe Hellerstein and Jie Liu
Microsoft

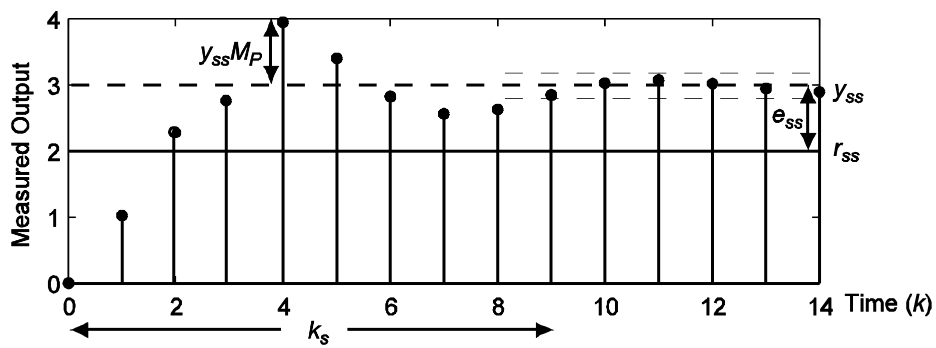
Jan. 14, 2008

Last Time

- Basic Control System Architecture



- Typical Control Goals:
  - Regulatory control
  - Disturbance rejection
  - Optimization
- SASO properties:
  - Stability, Accuracy, Settling, Overshot



Queuing System

Service Requests
(arrivals)

Service Completions
(departures)

Infinite size buffer (queue)

<u>Operation</u>

- 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

<u>*M/M/1* Assumptions & Key Result</u>

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

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

Today: Modeling

Signals, Systems, and Models

- Purpose of modeling
- Types of models

Model Construction

- Modeling from first principles
- Modeling from data

Hybrid System Models

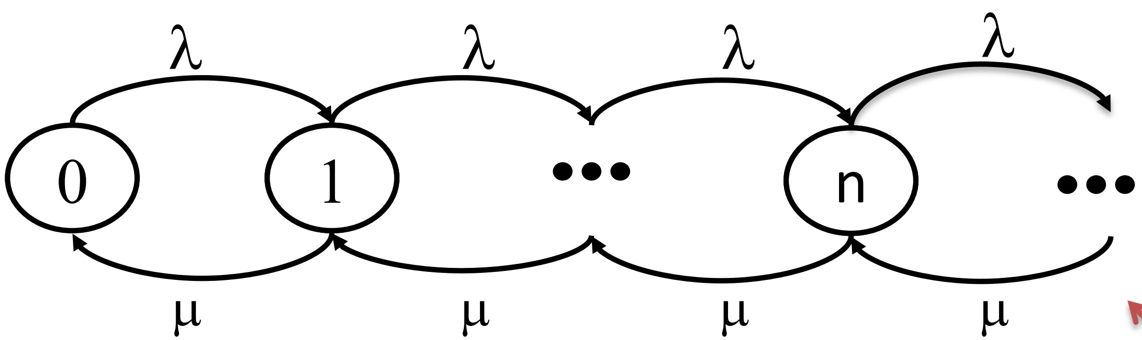
Paper Discussion

Why bother modeling?

- Analysis
  - prove formal properties (e.g. stability)
- Prediction
- Diagnostics
- Simulation

Queuing Model Revisit

Infinite size queue

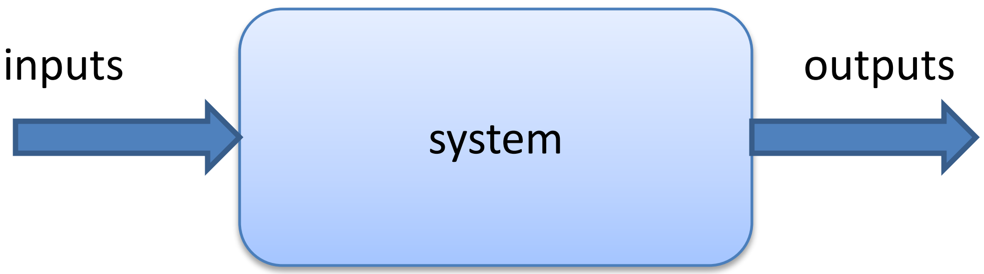


Good for steady-state analysis

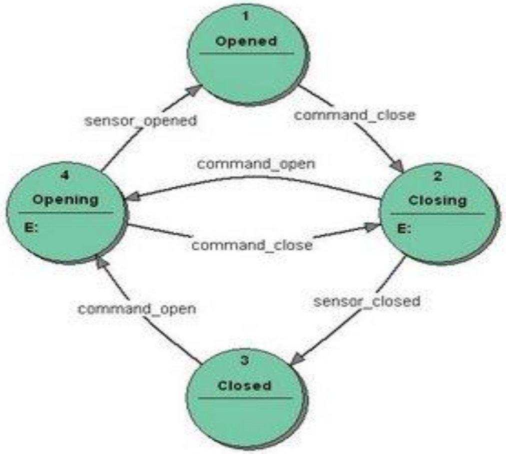
- State is number of customers in the system
- Arrows indicate rate at which transitions occur
- Arrival increases state by 1; departure decreases state by 1
- Probability of being in state *n* is p_n

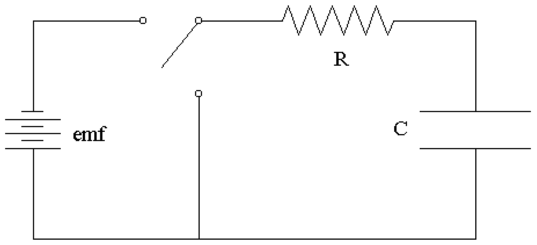
$$N = \sum_{n=0}^{\infty} n p_n$$

Input-Output Models

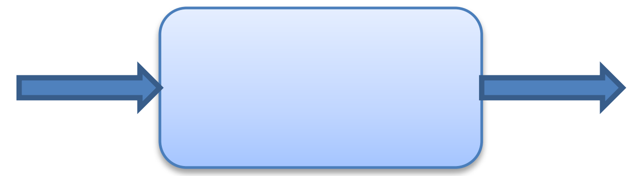


- Examples of non-IO models
  - -- Automata
  - -- Circuits

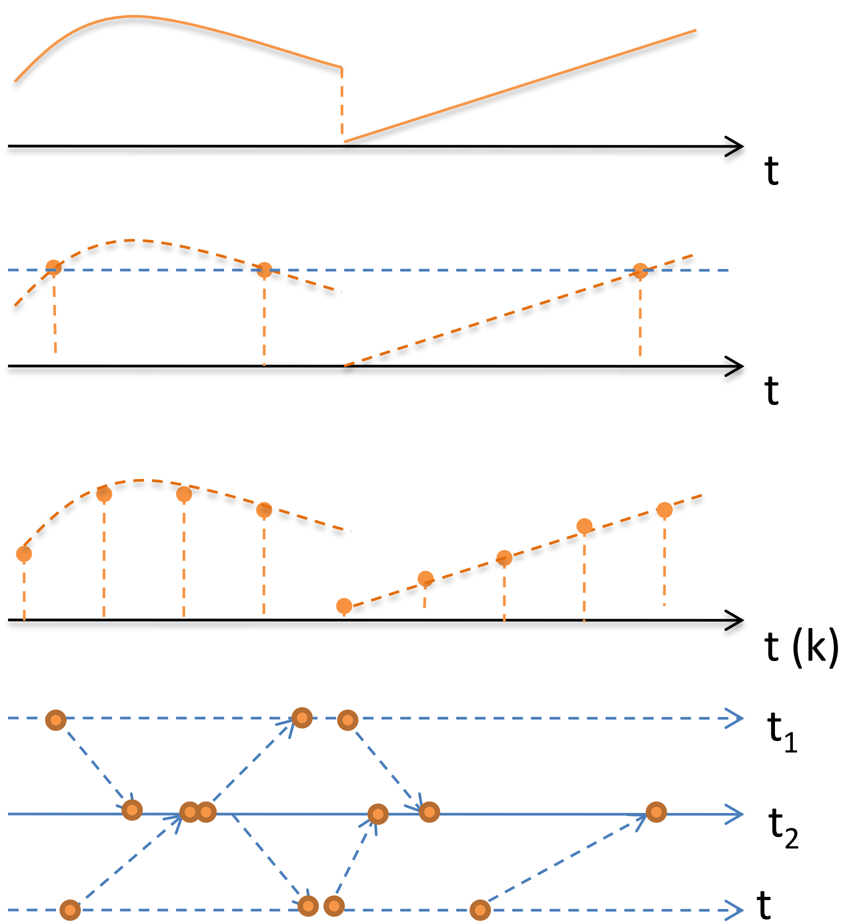




Signals



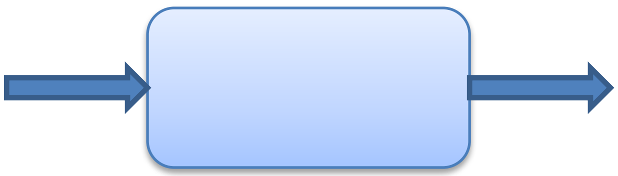
- A signal is a function on a (usually ordered) set. Some signals may be partial functions.



Examples

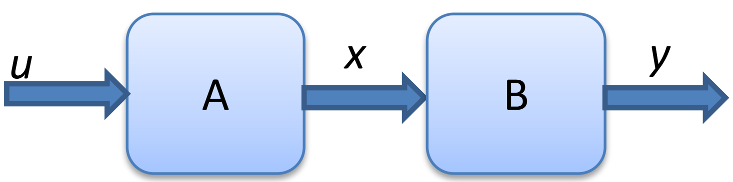
- Continuous-Time Signals
  - Functions on R^+
- Discrete Events
  - Partial functions on R^+
- Discrete-Time signals
  - Partial functions on R^+
  - Functions on N
- Signals on partially ordered sets

Systems

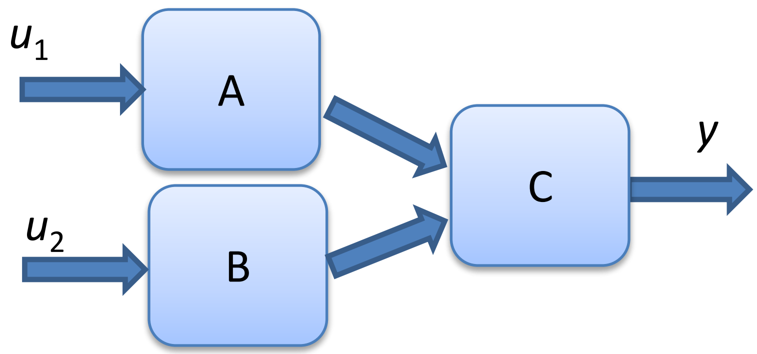


- Systems are functions from signals to signals.
  - Note: Input and output signals do not necessarily have the same domain or type.

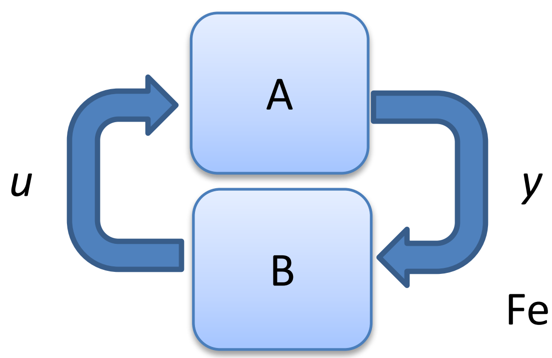
- System composition



Serial: $y = B(A(u))$

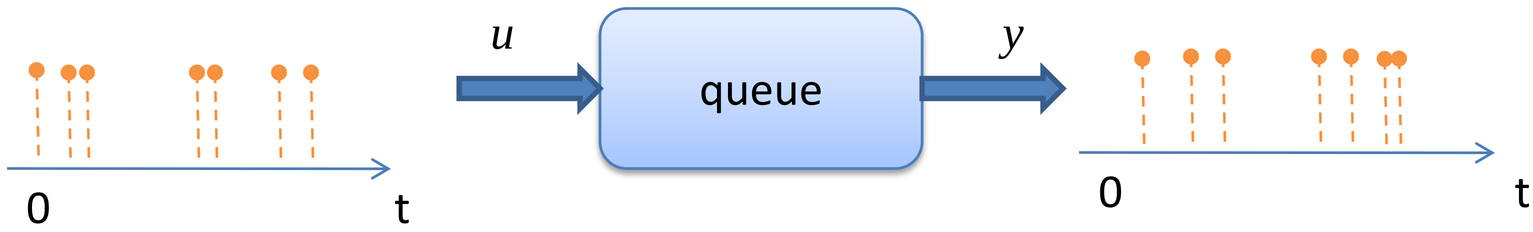


Parallel: $y = C(A(u_1), B(u_2))$



Feedback: $y = A(B(y))$

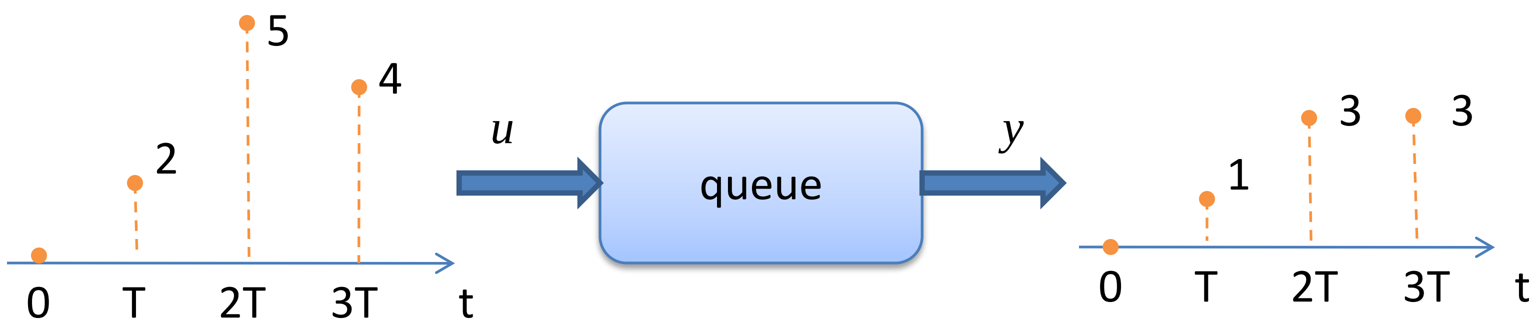
An I/O Model for a Queue



Is this model good for the purpose of controlling queue length?

What is it good for?

Another I/O Model for a Queue



Check the queue length every T seconds.

Difference equation

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

Is this right?

General View of Difference Equations

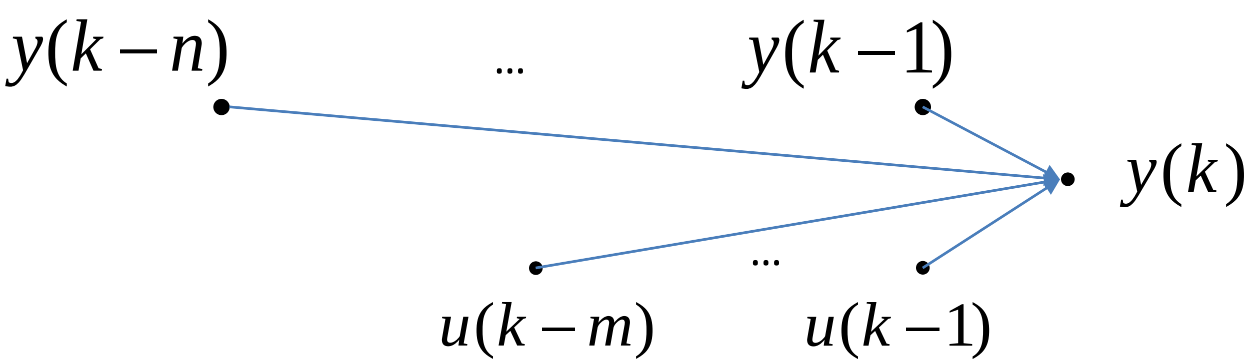
Term for the input-output models used General form

Linear Time-Invariant

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

Order of a model: *max(n,m)*

Relates current output to past outputs and inputs



Linear Time-Invariance

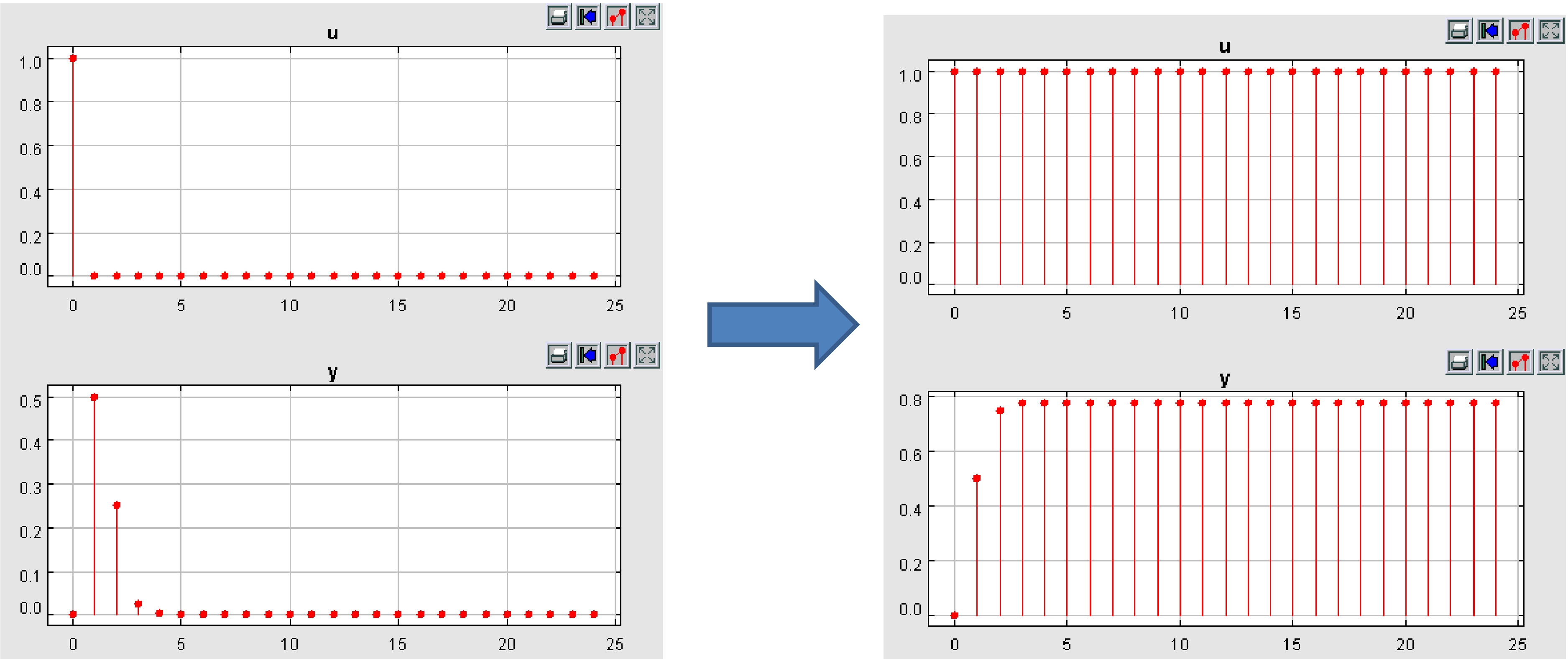
- Linearity:
  - $f(x + y) = f(x) + f(y)$
  - $f(\alpha x) = \alpha f(x)$ for all α
- Time-Invariance
  - f(x(k+d))(k)=f(x)(k+d)
- Check

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

LTI only if *y* starts from 0!

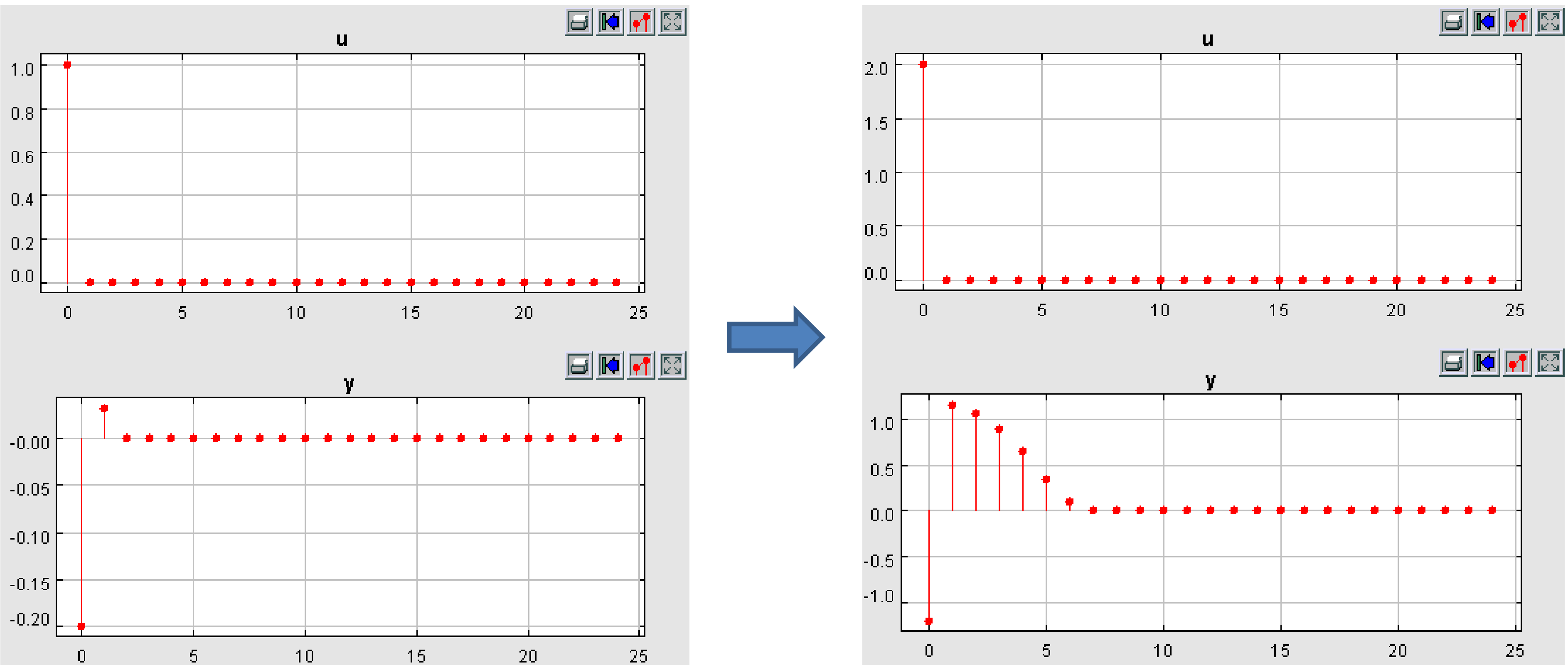
Impulse Response

- The behavior of linear time invariant system is uniquely defined by it impulse response



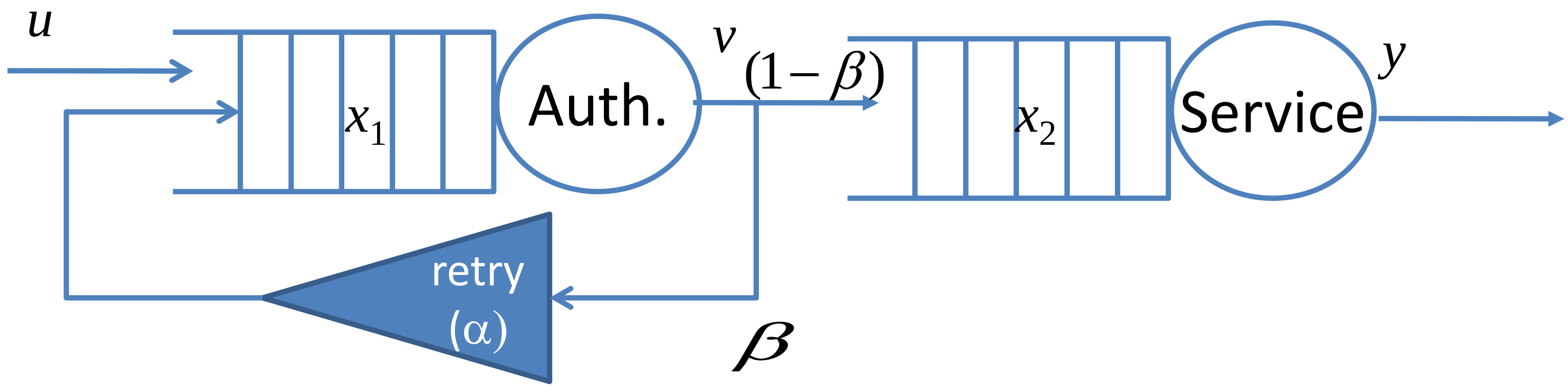
Non-Linear System Example

$$y(k+1) = ay^2(k) - y(k)u(k)$$



Modeling Using First Principles

- Construct systems from components
  - E.g. two queues



$$x_1(k+1) = x_1(k) + u(k) - (1-\alpha\beta)v(k)$$

$$x_2(k+1) = x_2(k) + (1-\beta)v(k) - y(k)$$

Modeling Using First Principles

- Pros:
  - Can be accurate
  - Have strong system implications
- Cons:
  - Requires strong domain knowledge
  - Can be complicated

Modeling Using Data

- There is a whole field called machine learning!
- Pros:
  - Weak dependency on domain knowledge
  - Can be adaptive
- Cons:
  - Requires data
  - Only as good as data

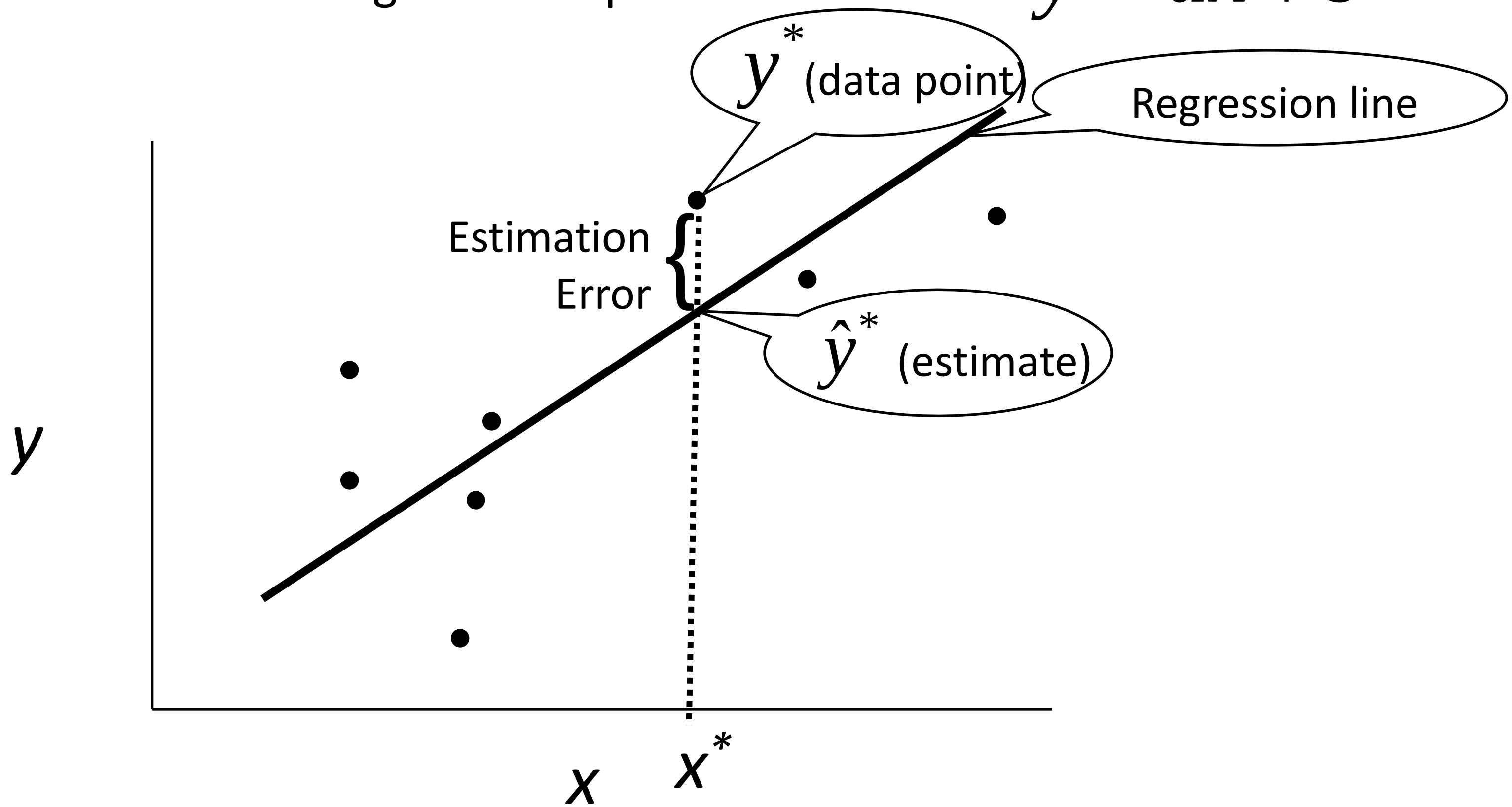
Estimating Parameters of Difference Equations

- Statistical approach---Use linear least squares regression
  - Computations are simple
  - Lots of software computes regression estimates (e.g., MatLab, Excel)

- Not a purely mechanical procedure
  - Need to determine a model structure (e.g. order)
  - Need to validate inputs
  - Need checks to ensure that models make sense
  - Plots are very important tools

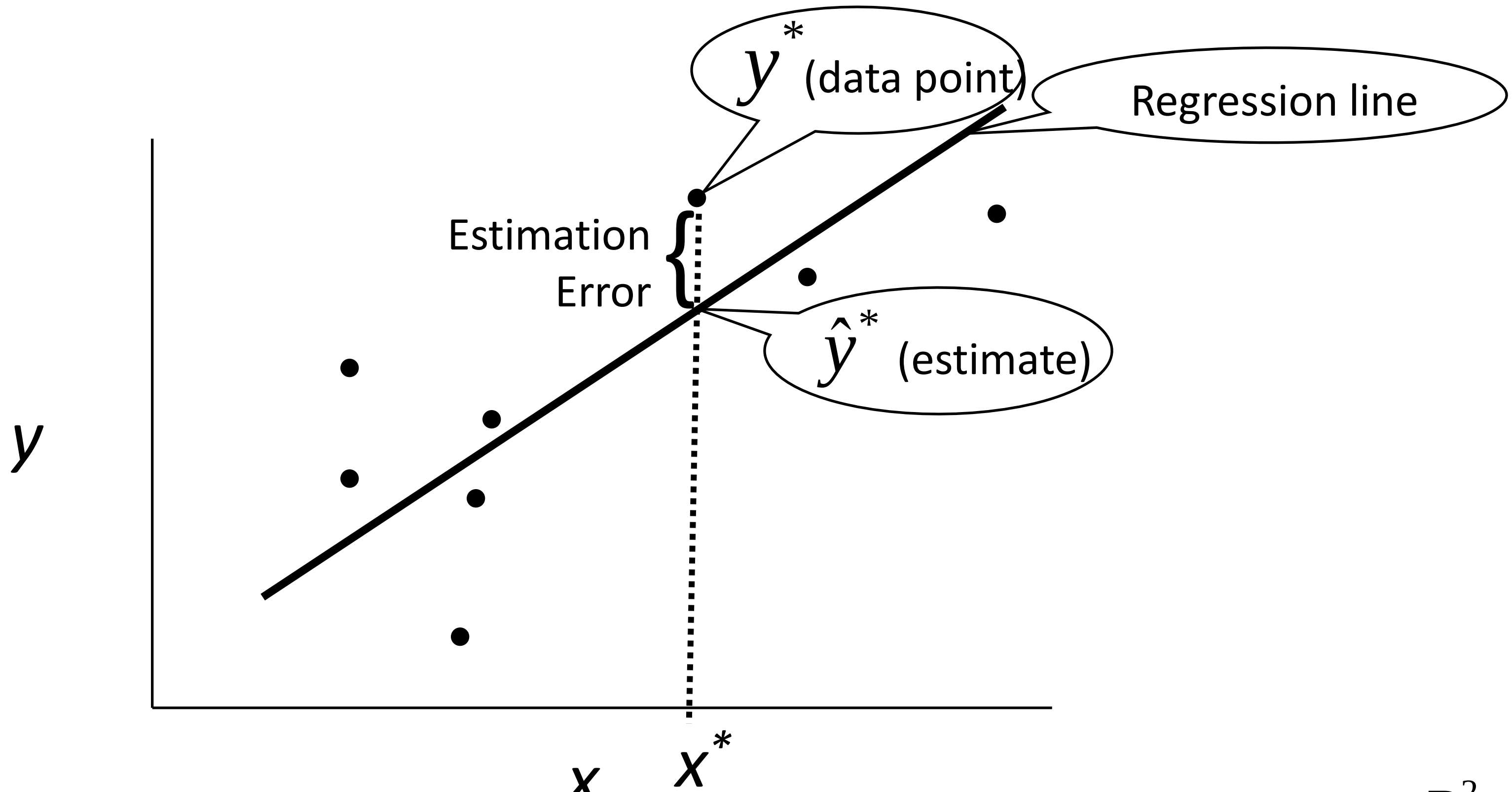
Linear Least Squares Regression Basics

Univariate linear regression equation: $y = ax + b$



LSR chooses *a* and *b* so as to minimize the sum of the square of the distances from data to line

Regression Metrics

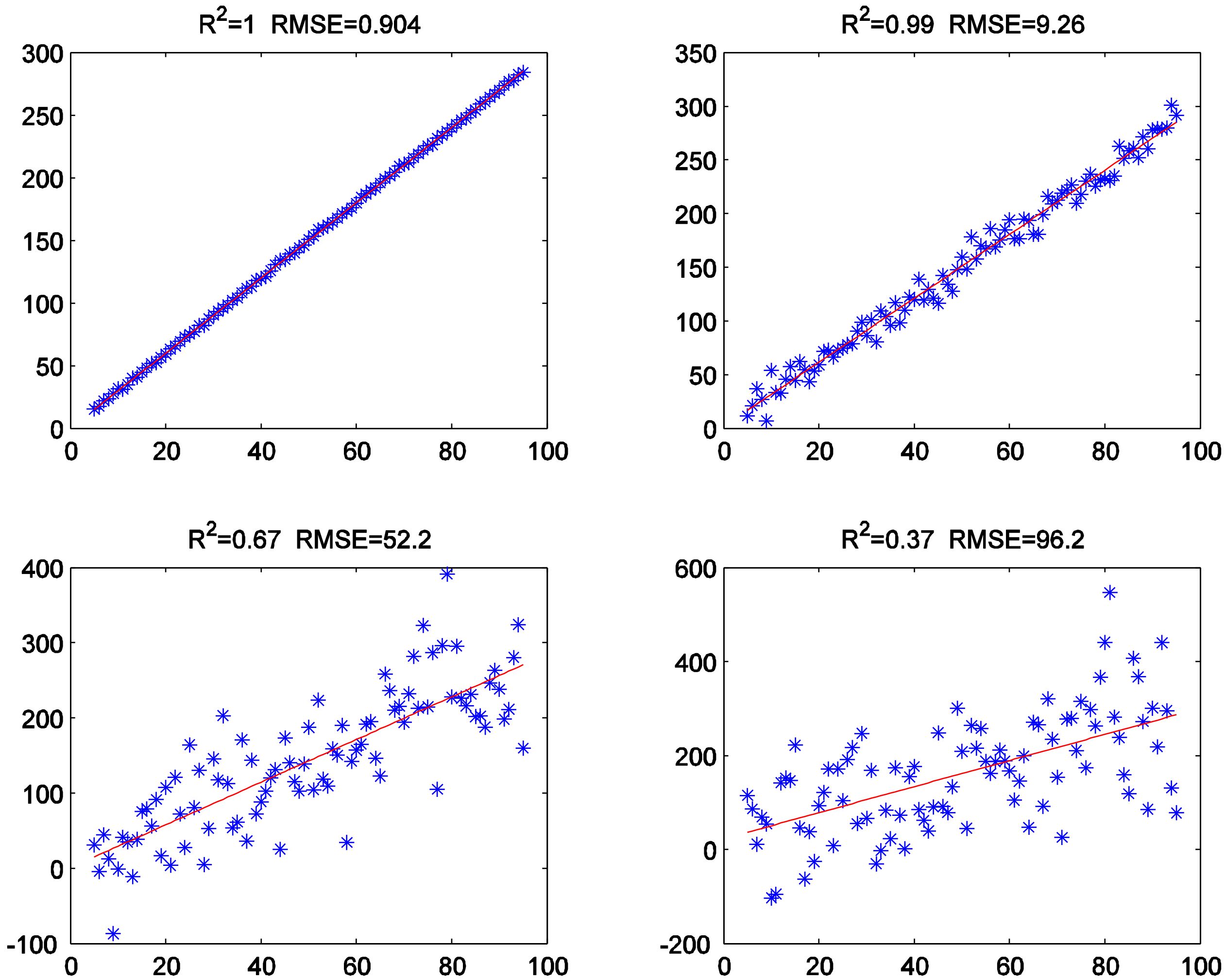


$$R^2 \in [0,1]$$

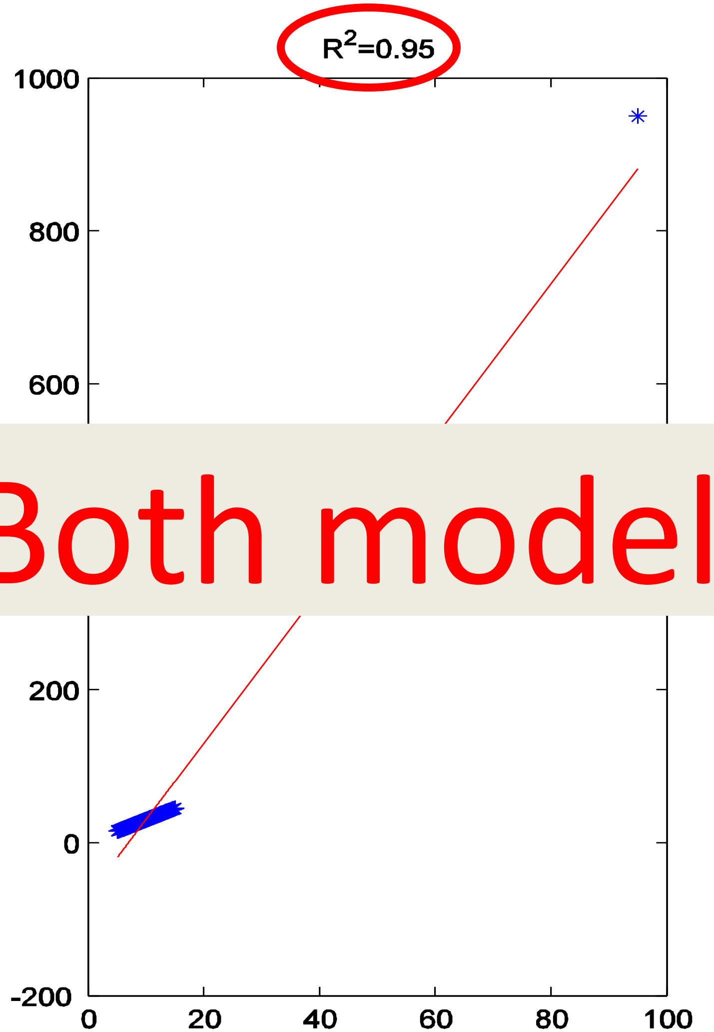
$$R^2 = \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} =$$ Fraction of variance in data explained by the regression line.

RMSE = Square root of the mean square of the estimation error

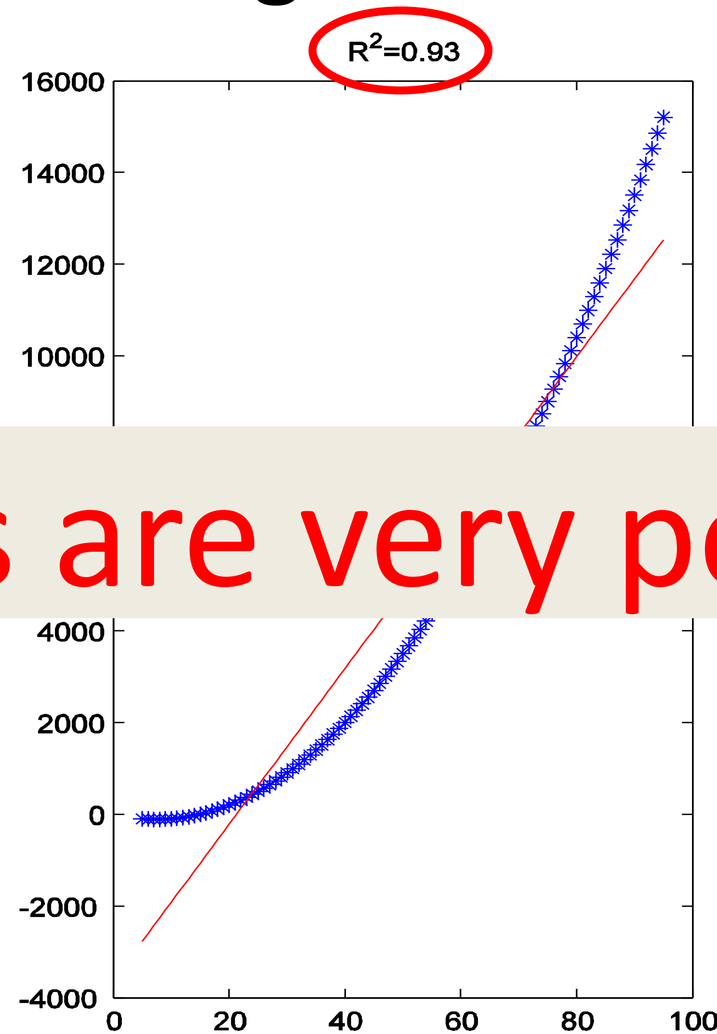
Examples of Regressions and Regression Metrics



Lies, Damn Lies, and Regression Metrics



Outlier distorts regression line



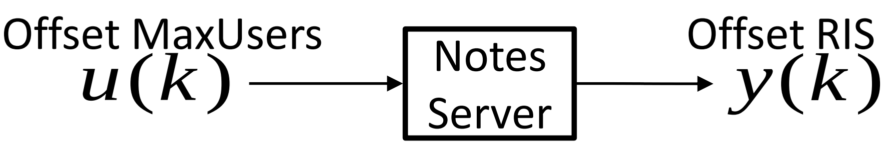
Functional bias

Both models are very poor.

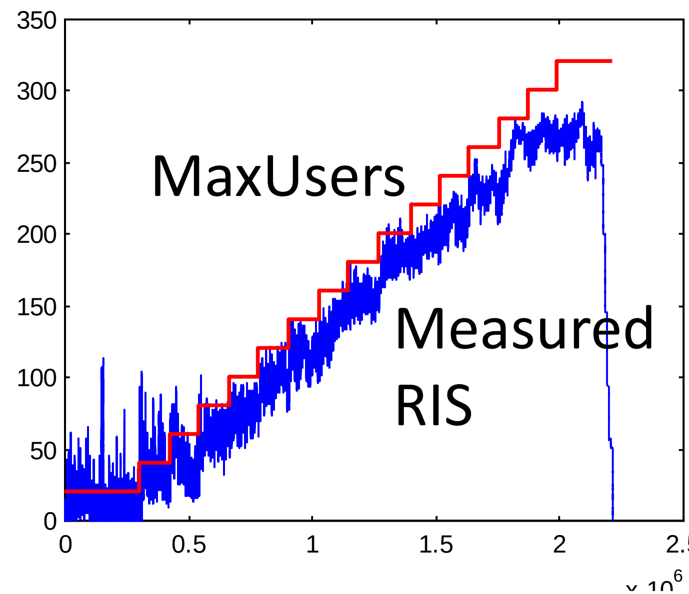
Estimating Parameters

1. Choose order of model
   Typically requires a multivariate regression model
2. Run experiments in which control input is varied systematically
3. Use least squares regression to estimate model parameters
4. Assess the results

Notes Example



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



$$a_1 = 0.43,\ b_1 = 0.47$$

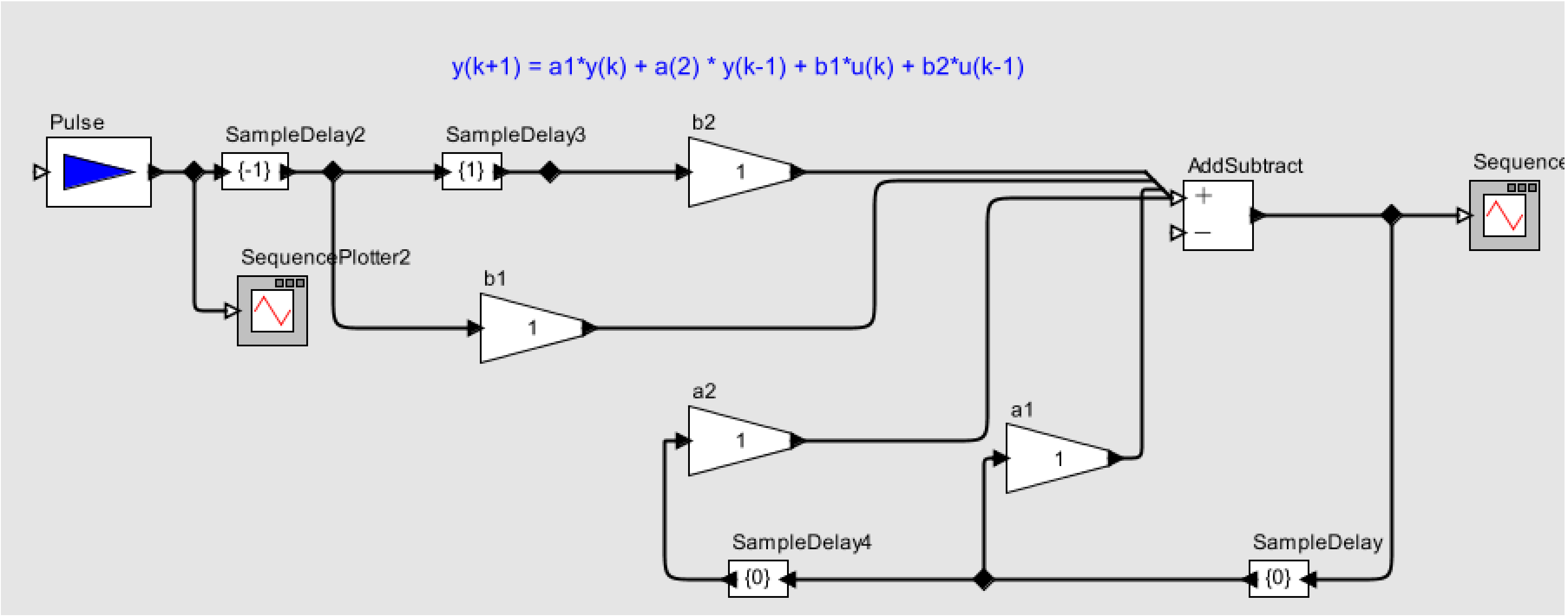
Probe Further

- Recursive Least Square
  - On-line parameter estimation
- Closed-loop system identification

- Reference:

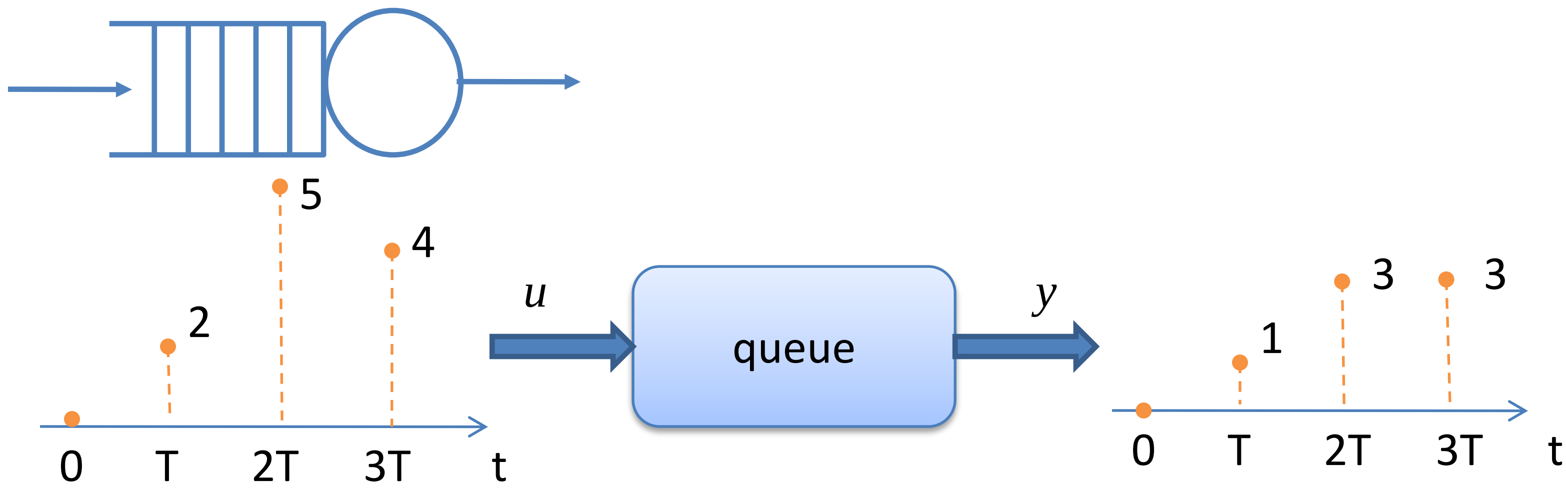
  Lennart Ljung, *System Identification: Theory for the User*, Prentice Hall, 1999

Example



y(k+1) = a1*y(k) + a(2) * y(k-1) + b1*u(k) + b2*u(k-1)
Pulse
SampleDelay2
{-1}
SampleDelay3
{1}
b2
1
AddSubtract
+
−
Sequence
SequencePlotter2
b1
1
a2
1
a1
1
SampleDelay4
{0}
SampleDelay
{0}

Hybrid System Models



Check the queue length every T seconds.

Difference equation

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

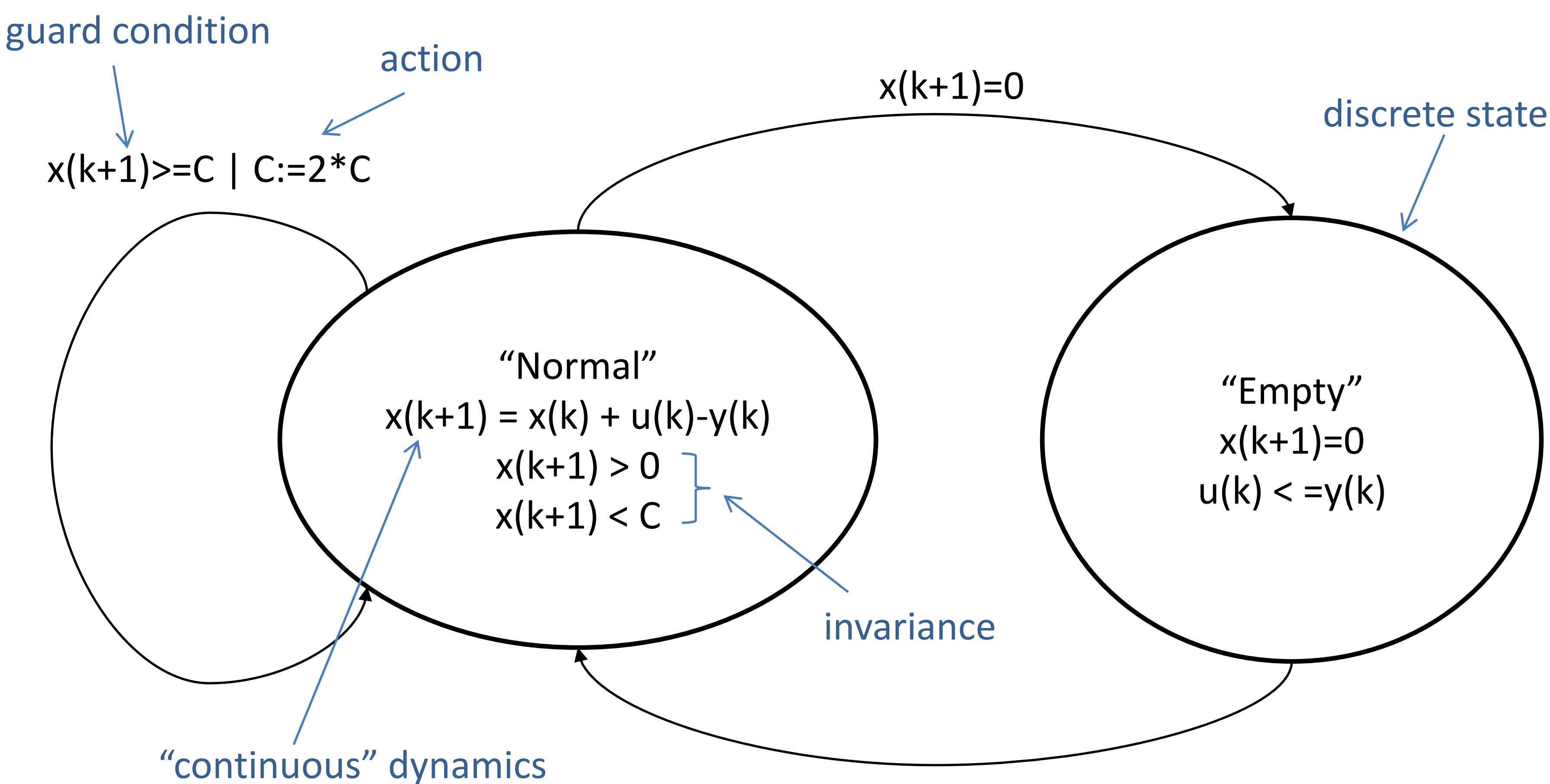
Is this right?

Only if $x(k) + u(k) - y(k) \geq 0$

Otherwise $x(k+1) = 0$

Hybrid System Models

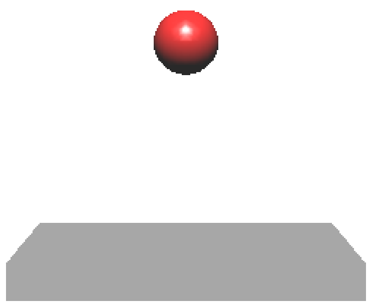
- Composition of state machines and differential(difference) equations.

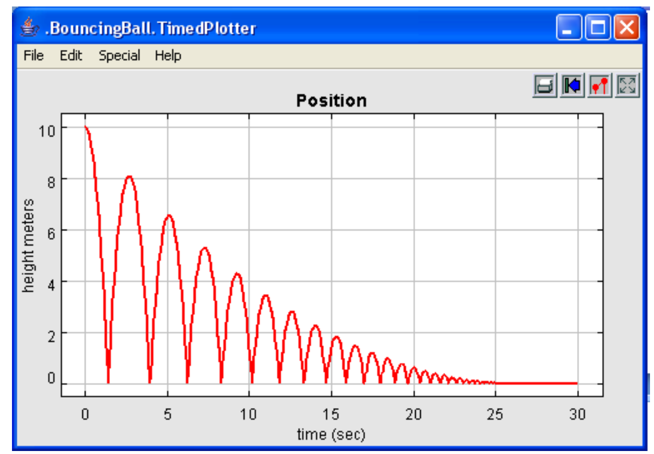


A queue with increasing buffer size C

Behavior of Hybrid Systems

- a sequence of flows and jumps

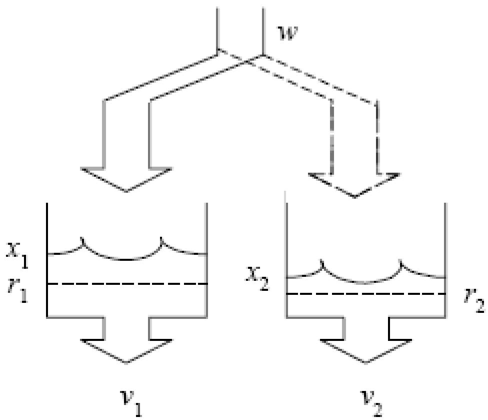




- Properties about behavior
  - Safety: Do not enter “bad” states
  - Liveness: Behavior extends to infinity
  - Stability:
- Many properties are undecidedly in general

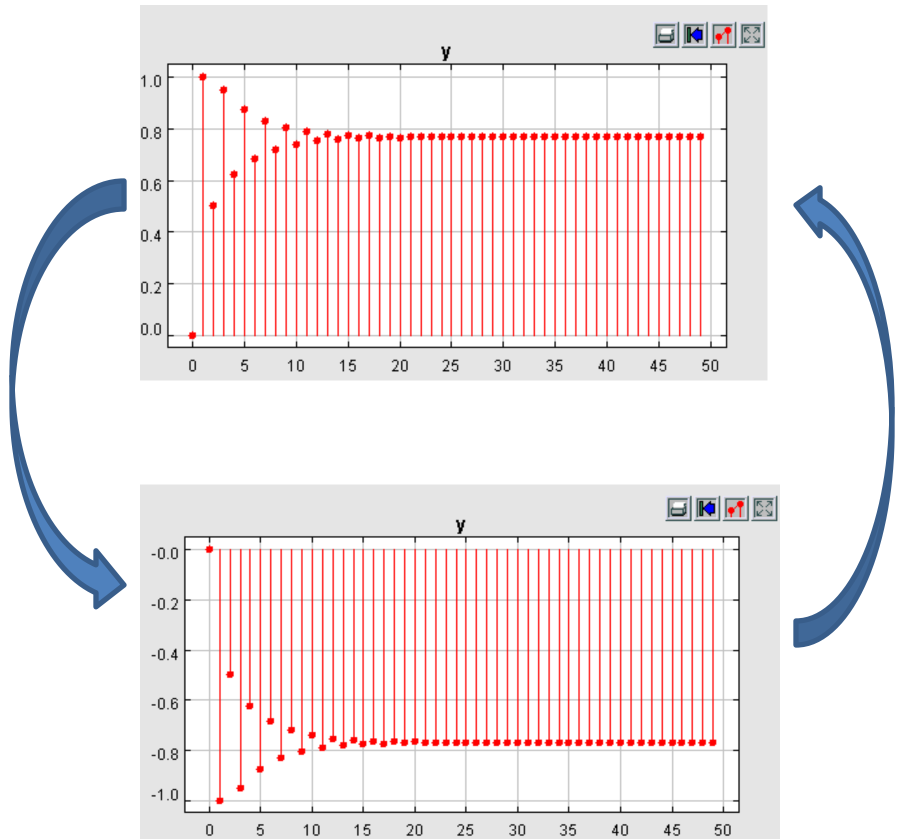
Be Careful about Hybrid Systems

- Zeno behavior
  - Never-empty water tanks



$$w < v_1 + v_2$$

- Stability is not composable
  - switching between two stable systems can be unstable



Probe Further

- Hybrid system lecture notes: http://robotics.eecs.berkeley.edu/~sastry/ee291e/lygeros.pdf

- Hybrid system modeling and simulation:
  Google: HyVisual

Summary

- Many model structures for difference purposes

- Models can be constructed from first principles or data

- System identification for LTI systems

- Hybrid systems

Paper discussion

- Lu, Lu, Abdelzaher, Stankovic, Son, “Feedback Control Architecture and Design Methodology for Server Delay Guarantees in Web Servers” IEEE Tran. on Parallel and Distributed Systems, 17(9), Sept. 2006, pp.1014~1027