LET'S CONTROL EVERYTHING!

C. G. Cassandras
Boston University

cgc@bu.edu
http://vita.bu.edu/cgc

LET'S CONTROL **EVERYTHING**

LET'S **CONTROL** EVERYTHING

LET **US** CONTROL EVERYTHING

CENTRAL THEMES OF THIS TALK...

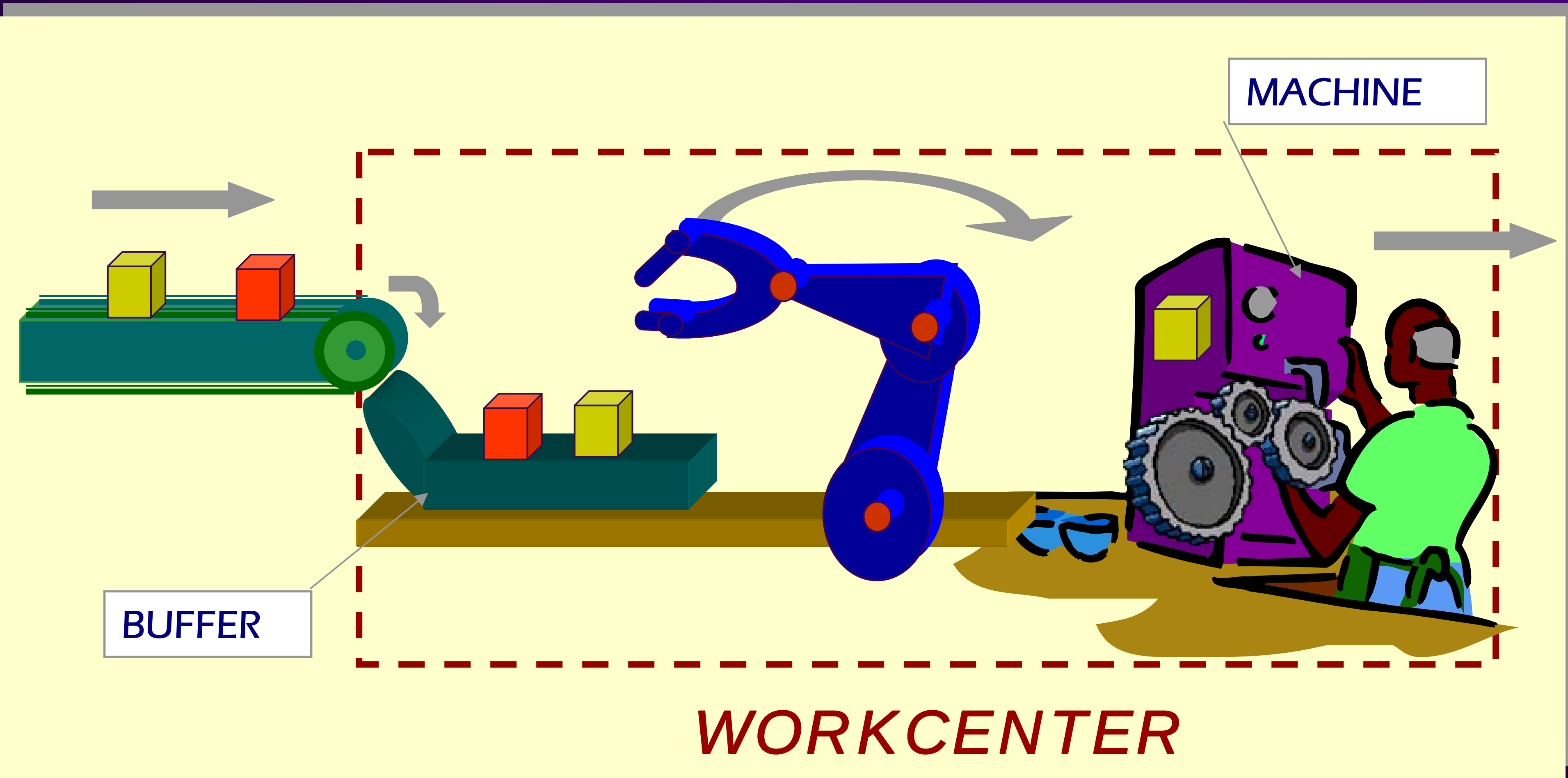
- *CONTROL* is pervasive
- *CONTROL* is not just for systems $\dot{x}(t) = f\left(x, u, t\right)$
  (*if you are a hammer, everything looks like a nail…*)
- *CONTROL* doesn't have to always be "sophisticated"
- *Principles* of *CONTROL THEORY* (*FEEDBACK*, *DYNAMICS*, etc) matter more than specific methodologies
- "When our classifications start breaking down, we know we are learning something exciting…"

OUTLINE

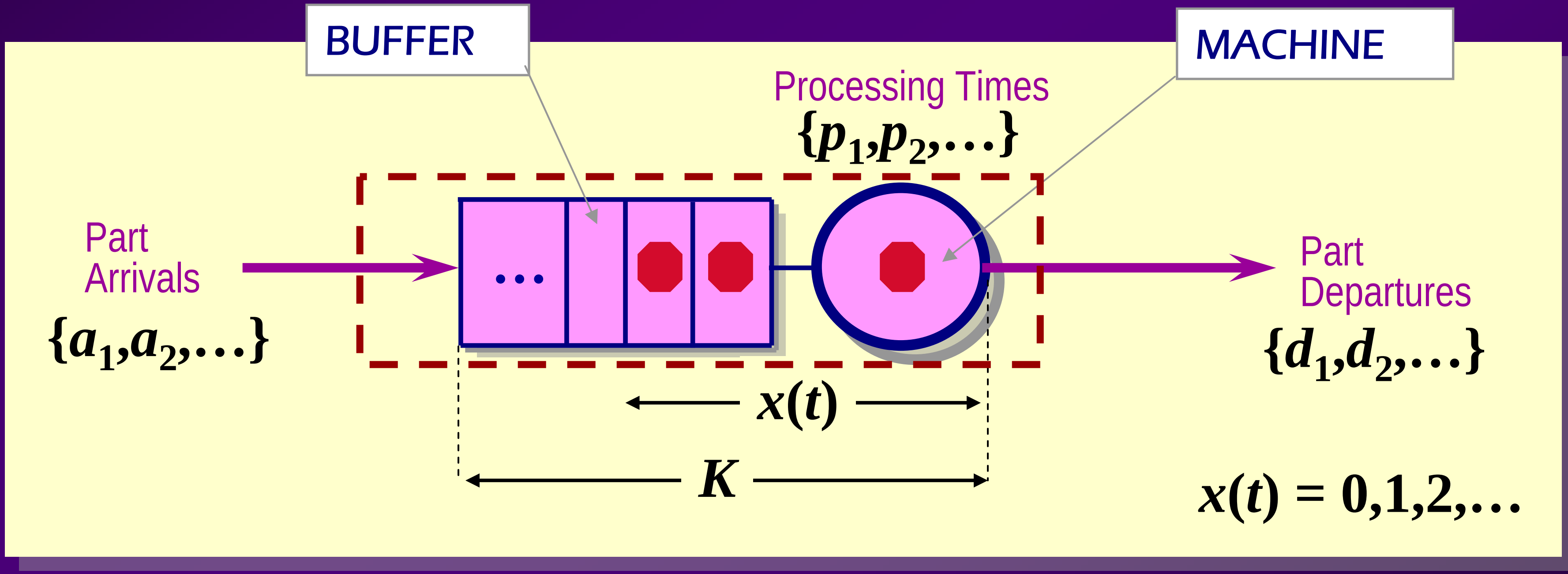
A somewhat personal account of how the "control mindset" can lead to solutions of intriguing, unconventional problems, which in turn stimulate new theoretical developments

- *1980's* – Manufacturing systems, kanban control
  - ➡ the roots of Discrete Event Systems
- *1990's* – Learning from sample paths of complex systems
  - ➡ controlling elevators, Australian mines, air traffic, communication networks, command-control systems
- *FUTURE* – Hybrid systems, complexity, computation…

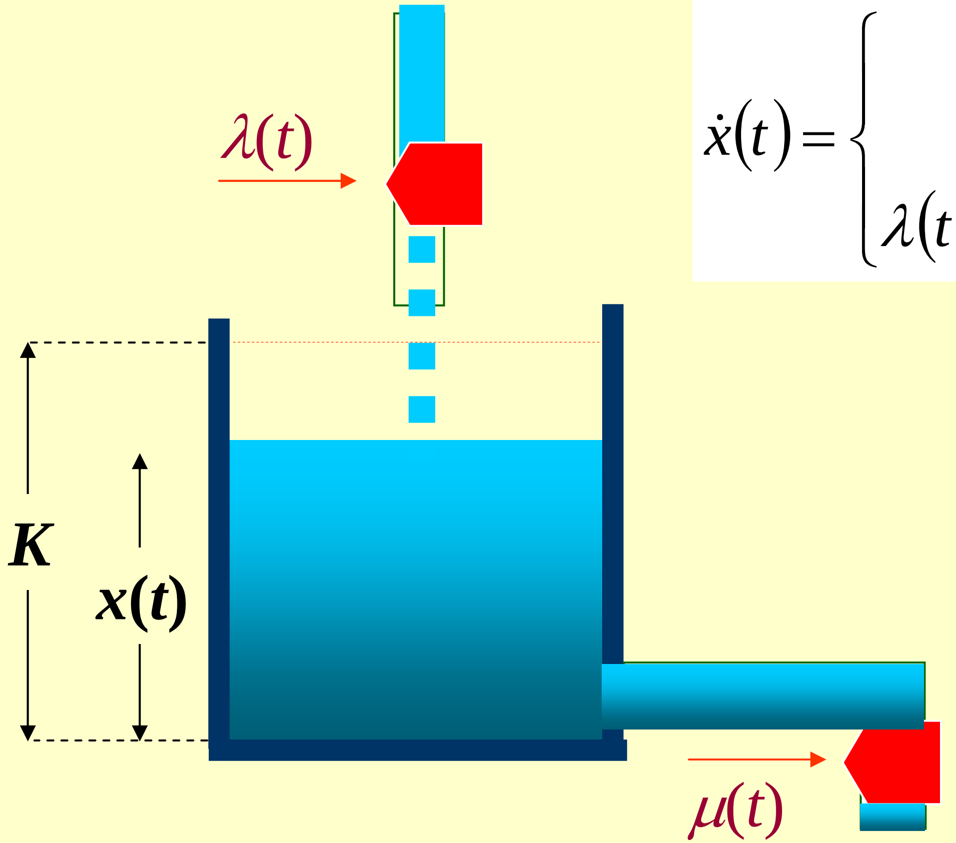
CONTROL IN MANUFACTURING SYSTEMS



A model for a manufacturing workcenter capturing part-by-part behavior:



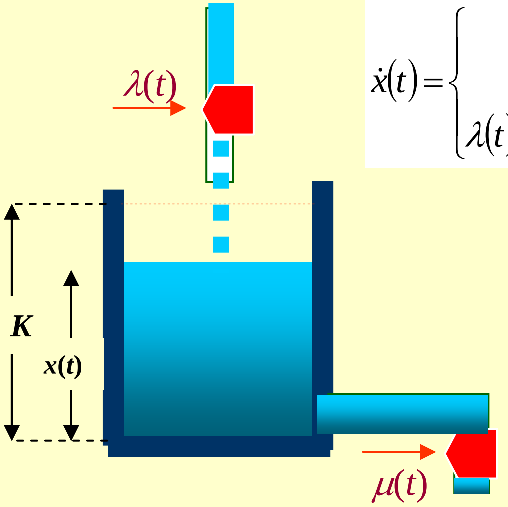
A continuous flow model for a manufacturing workcenter:



$$\dot{x}(t)=\begin{cases} 0 & \text{if } [x(t)=0 \text{ and } \lambda(t)\le\mu(t)] \\ 0 & \text{if } [x(t)=K \text{ and } \lambda(t)\ge\mu(t)] \\ \lambda(t)-\mu(t) & \text{otherwise} \end{cases}$$

CONTROL IN MANUFACTURING SYSTEMS *CONTINUED*

$$\dot{x}(t) = f\big(x,u,t\big) \text{ ???}$$

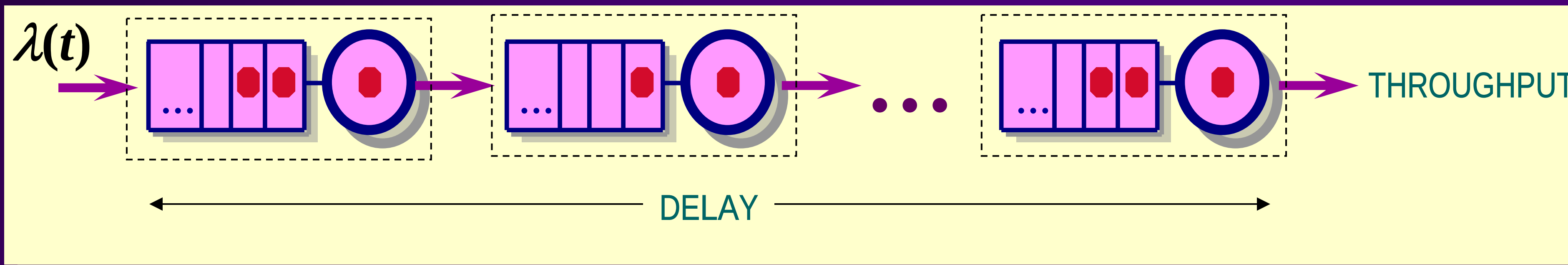


$$\dot{x}(t) = \begin{cases} 0 & \text{if } [x(t)=0 \text{ and } \lambda(t) \le \mu(t)] \\ 0 & \text{if } [x(t)=K \text{ and } \lambda(t) \ge \mu(t)] \\ \lambda(t)-\mu(t) & \text{otherwise} \end{cases}$$

Limitations of the continuous flow model:

- Can only analyze *averages*
- Cannot deal with part-by-part control issues such as:
  - Is *n*th part guaranteed to be served within *T* time units?
  - If a part is within *T* time units from its due-date, serve it next
  - Prioritize RED parts over YELLOW parts

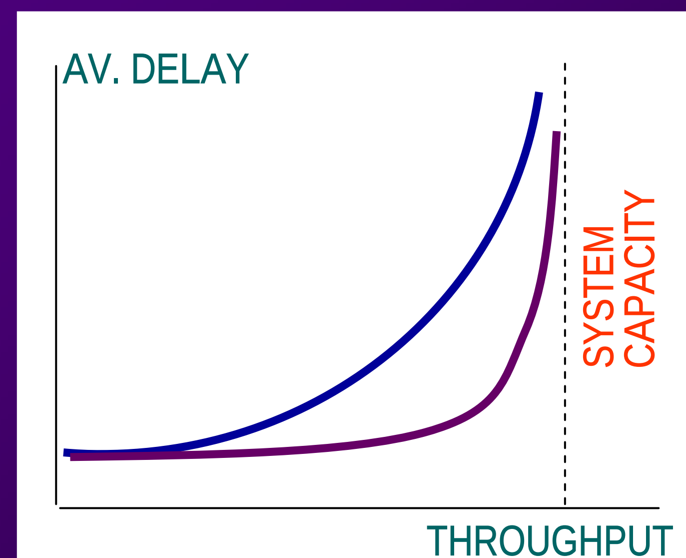
Manufacturing system with *N* sequential operations:



INCREASE $\lambda(t)$

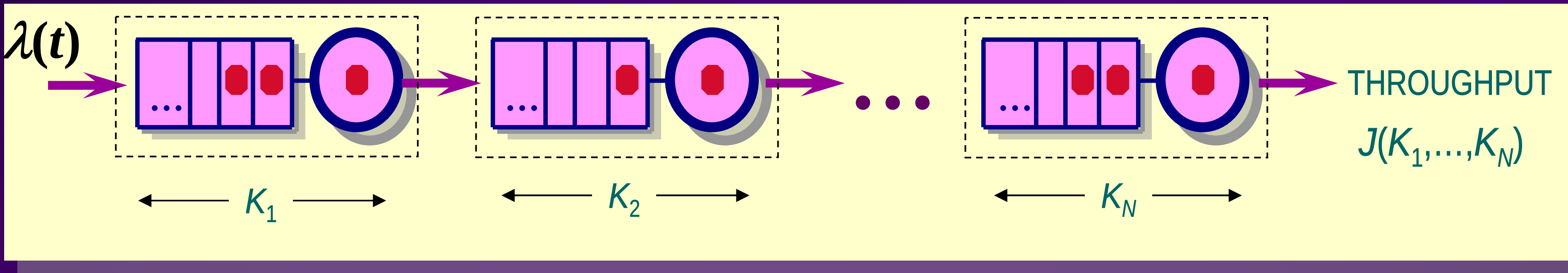
THROUGHPUT increases (GOOD)

average DELAY increases (BAD)



TWO PROBLEMS

1. BUFFER ALLOCATION (FIAT, circa 1979)

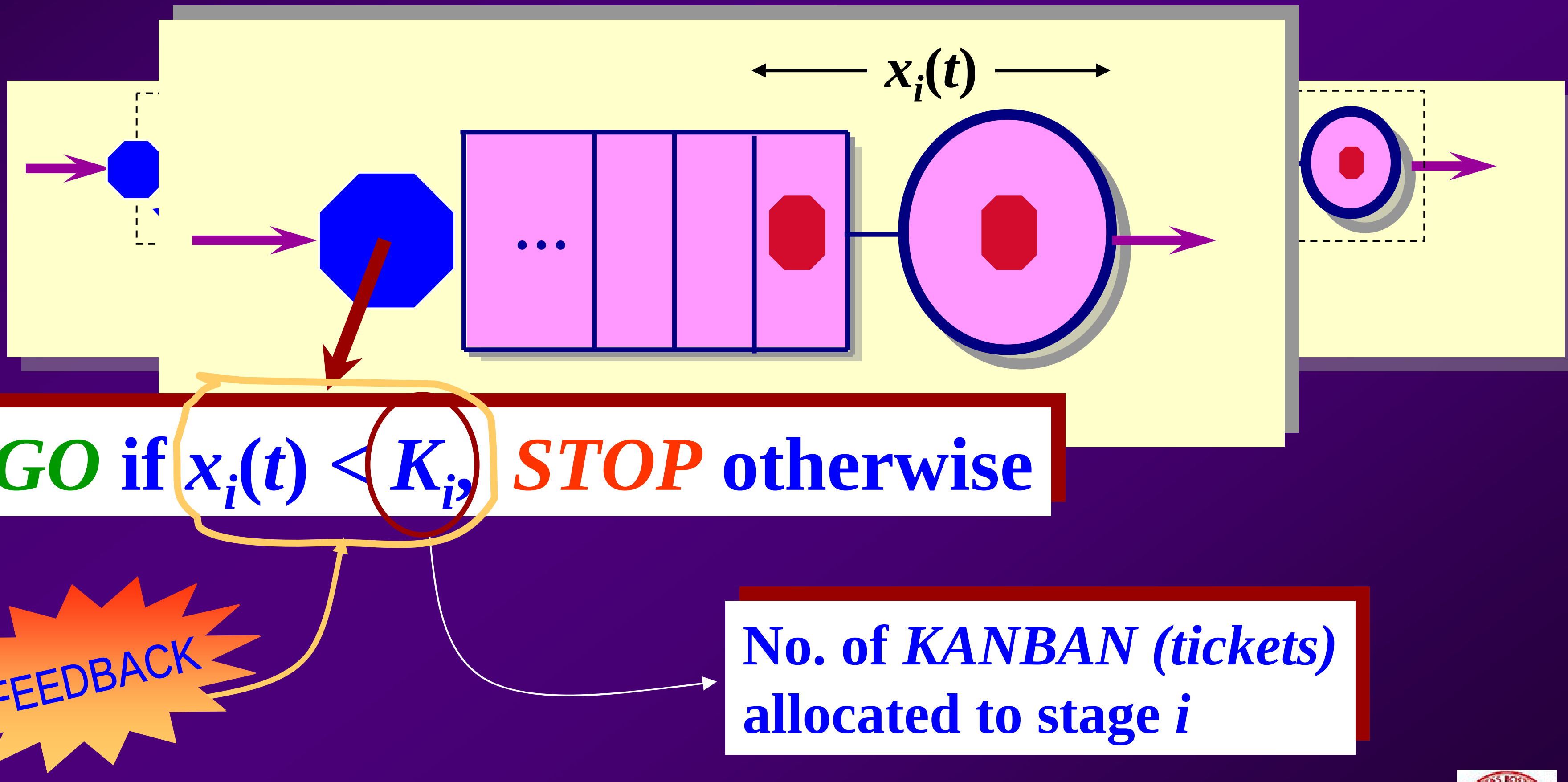


$$\max_{K_1,\ldots,K_N} J(K_1,\ldots,K_N) \text{ s.t.} \sum_{i=1}^{N} K_i = C$$

PARAMETRIC OPTIMIZATION

TWO PROBLEMS

2. *FLOW CONTROL*

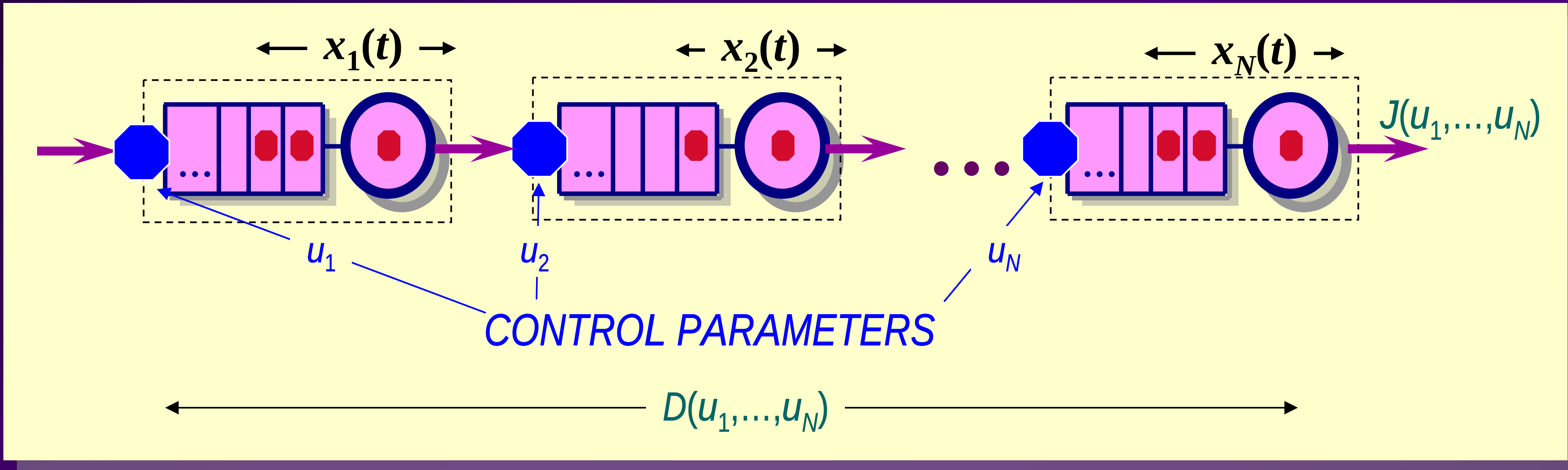


GO **if** $x_i(t) < K_i$, ***STOP*** **otherwise**

No. of *KANBAN (tickets)* allocated to stage *i*

TWO PROBLEMS

CONTINUED

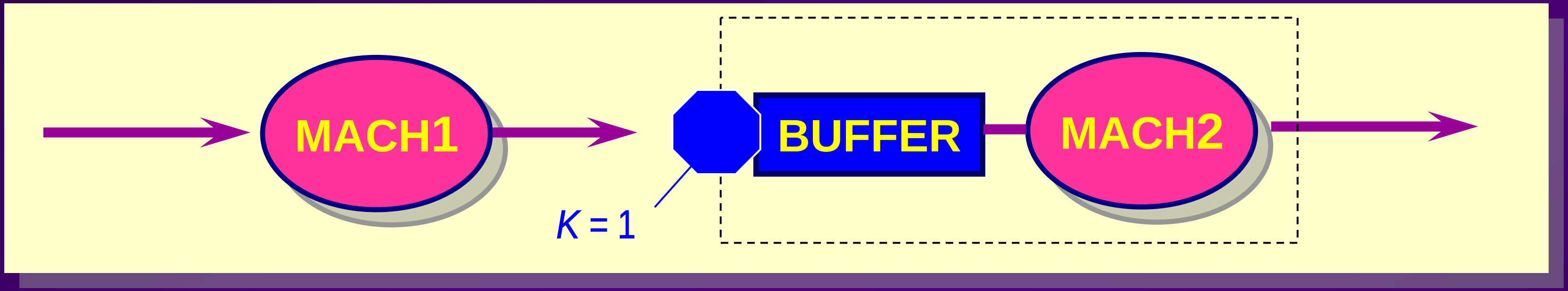


$$\max_{u_1,\ldots,u_N} J(u_1,\ldots,u_N) \ \text{s.t.} \begin{cases} D(u_1,\ldots,u_N) \le C \\ \text{system dynamics} \end{cases}$$

DYNAMIC OPTIMIZATION

SUPERVISORY CONTROL PROBLEM

Supervise the proper execution of simple kanban-based control



1. MACH1 can only start when BUFFER is empty.
2. MACH2 can only start when BUFFER is full.
3. MACH1 cannot start when MACH2 is down.
4. If both MACH1 and MACH2 are down, then MACH2 is repaired first

THE STUDY OF THESE PROBLEMS HAS LED TO…

- New modeling frameworks paralleling $\dot{x}(t) = f\left(x, u, t\right)$

- Supervisory Control theory: *enable/disable controllable events*

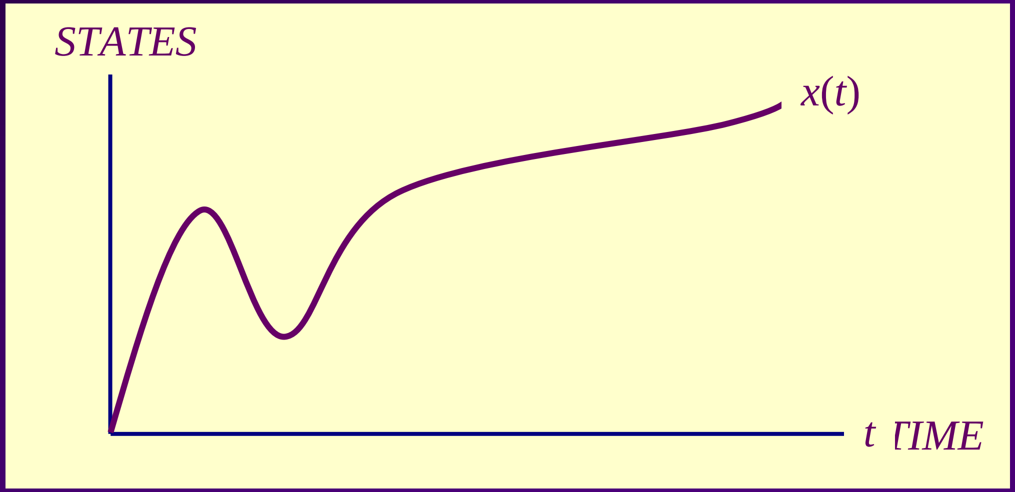
Ramadge, Wonham, Krogh, Lin, Rudie, Lafortune, etc.

- Perturbation Analysis theory: *learning from state trajectories*

Ho, Cassandras, Cao, Glasserman, Gong, etc.

TIME-DRIVEN vs *EVENT-DRIVEN* SYSTEMS

TIME-DRIVEN SYSTEM



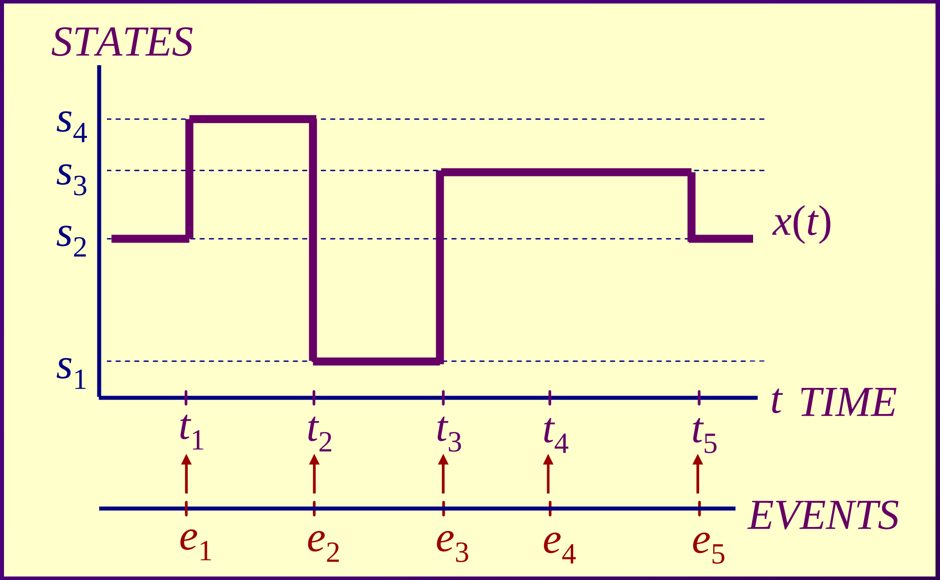
STATE SPACE:

$$X = \Re$$

DYNAMICS:

$$\dot{x} = f(x,t)$$

EVENT-DRIVEN SYSTEM



STATE SPACE:

$$X = \{s_1, s_2, s_3, s_4\}$$

DYNAMICS:

$$x' = f(x,t)$$

MODELING FOUNDATIONS

AUTOMATON: (E, X, Γ, f, x_0)

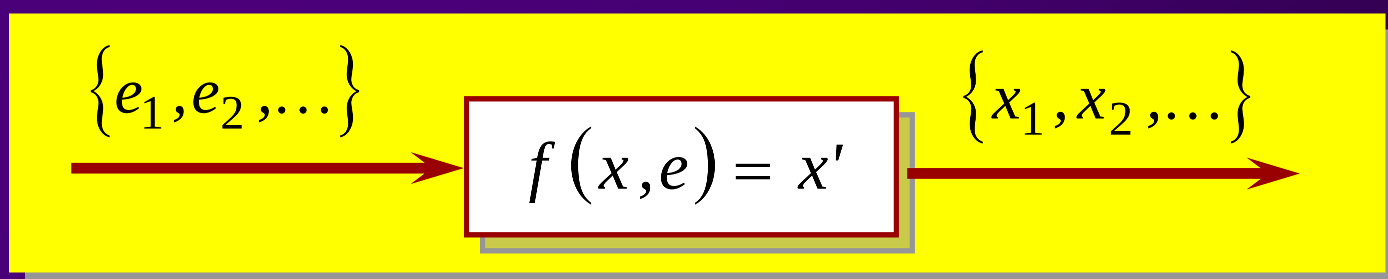
E : Event Set

X : State Space

$\Gamma(x)$: Set of *feasible* or *enabled* events at state *x*

f : State Transition Function $f : X \times E \to X$
(undefined for events $e \notin \Gamma(x)$)

x_0 : Initial State, $x_0 \in X$



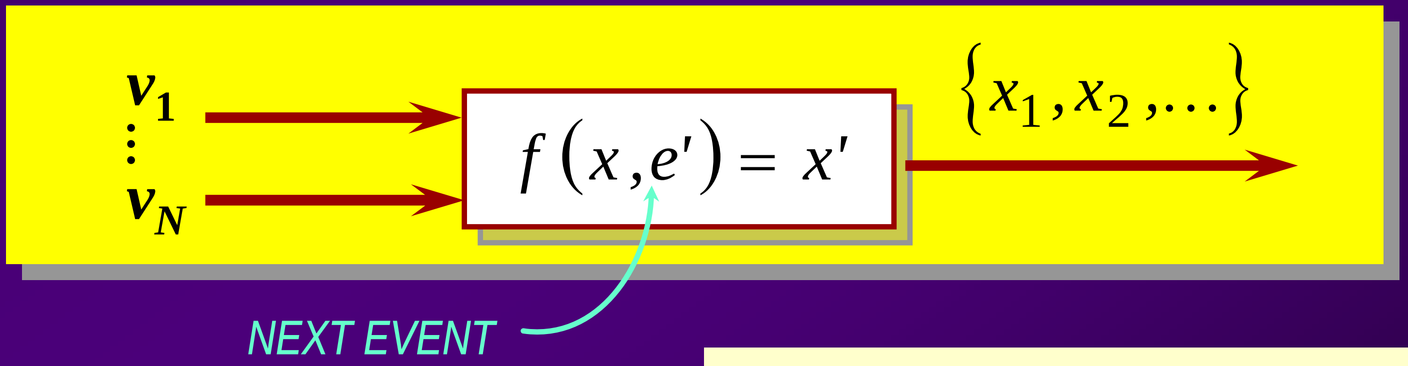
TIMED AUTOMATON

Add a ***Clock Structure V*** to the automaton: $(E, X, \Gamma, f, x_0, \mathbf{V})$

where:

$$\mathbf{V} = \{\mathbf{v}_i : i \in E\}$$

and $\mathbf{v}_i$ is a ***Clock or Lifetime sequence***: $\mathbf{v}_i = \{v_{i1}, v_{i2}, \ldots\}$
one for each event *i*



Need an *internal mechanism* to determine *NEXT EVENT e´* and hence *NEXT STATE* $x' = f(x, e')$

HOW THE TIMED AUTOMATON WORKS...

- **CURRENT STATE**

  $x \in X$ with feasible event set $\Gamma(x)$

- **CURRENT EVENT**

  e that caused transition into x

- **CURRENT EVENT TIME**

  t associated with e

Associate a
CLOCK VALUE/RESIDUAL LIFETIME y_i
with each feasible event $i \in \Gamma(x)$

HOW THE TIMED AUTOMATON WORKS... *CONTINUED*

- **NEXT/TRIGGERING EVENT *e'* :**

$$e' = \arg \min_{i \in \Gamma(x)} \{y_i\}$$

- **NEXT EVENT TIME *t'* :**

$$t' = t + y^*$$

$$\text{where:} \quad y^* = \min_{i \in \Gamma(x)} \{y_i\}$$

- **NEXT STATE *x'* :**

$$x' = f(x, e')$$

HOW THE TIMED AUTOMATON WORKS... *CONTINUED*

Determine new *CLOCK VALUES* y_i'
for every event $i \in \Gamma(x)$

$$y_i' = \begin{cases} y_i - y^* & i \in \Gamma(x'), i \in \Gamma(x), i \neq e' \\ v_{ij} & i \in \Gamma(x') - \{\Gamma(x) - e'\} \\ 0 & \text{otherwise} \end{cases}$$

where: v_{ij} = new lifetime for event i

EVENT CLOCKS ARE STATE VARIABLES

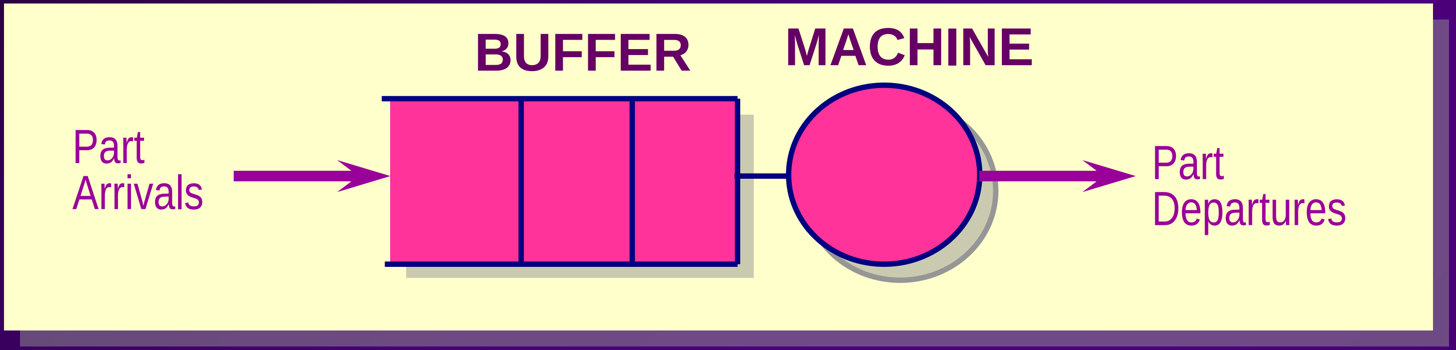
$\mathbf{v}_1$ ⋮ $\mathbf{v}_N$ →

$$x' = f(x, e'), \quad e' = \arg \min_{i \in \Gamma(x)} \{y_i\}$$

$$\mathbf{y}' = \mathbf{g}(\mathbf{y}, x, V)$$

→ $\{x_1, x_2, \ldots\}$

TIMED AUTOMATON - AN EXAMPLE



$$E = \{a, d\}$$
$$X = \{0,1,2,\ldots\}$$

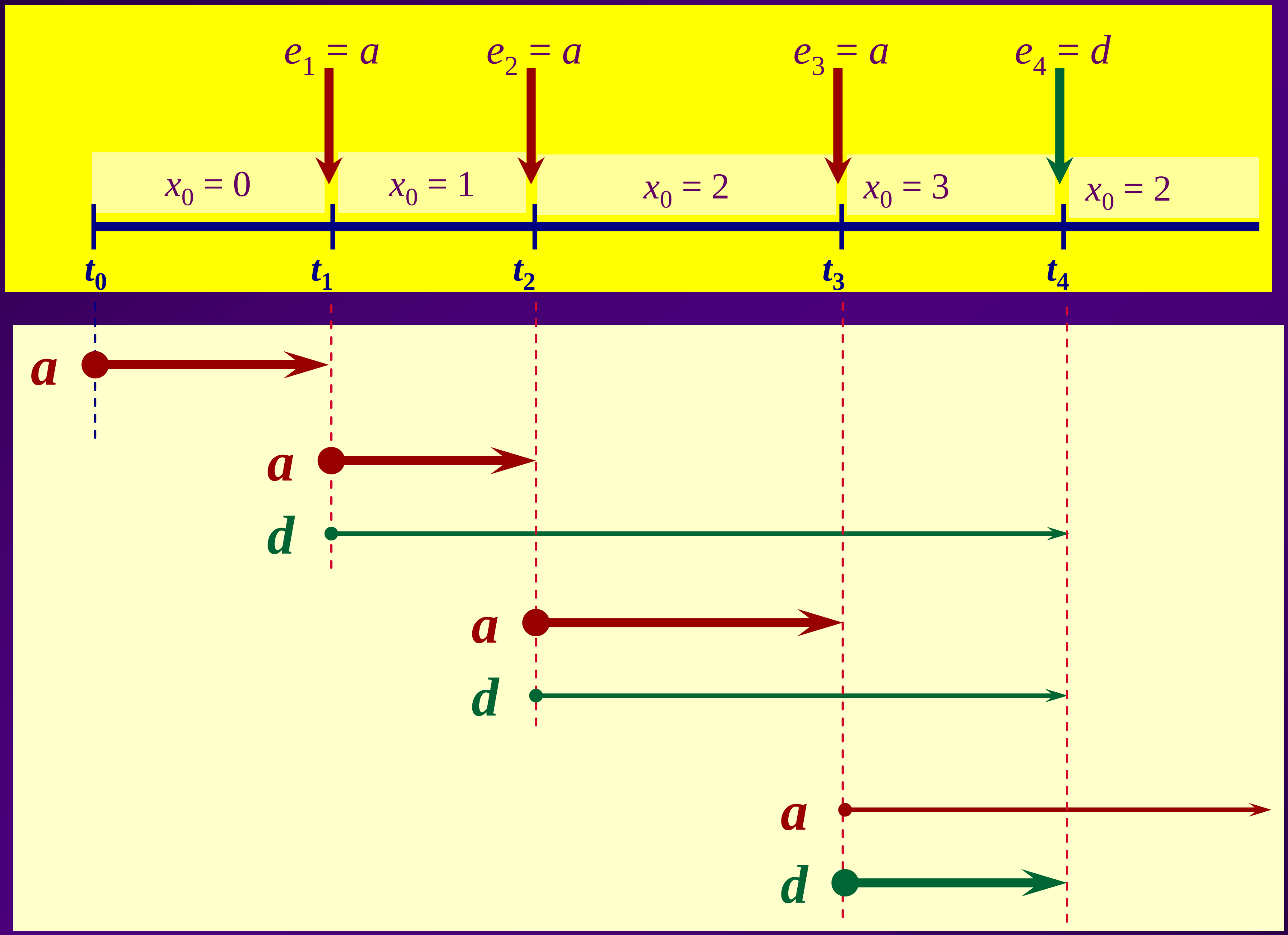
$$\Gamma(x) = \{a, d\}, \quad \text{for all } x > 0$$
$$\Gamma(0) = \{a\}$$

$$f(x, e') = \begin{cases} x+1 & e' = a \\ x-1 & e' = d,\ x > 0 \end{cases}$$

$$\text{Given input}: \quad \mathbf{v}_a = \{v_{a1}, v_{a2}, \ldots\}, \ \mathbf{v}_d = \{v_{d1}, v_{d2}, \ldots\}$$

TIMED AUTOMATON - A TYPICAL SAMPLE PATH

CONTINUED



STOCHASTIC TIMED AUTOMATON

- Same idea with the Clock Structure consisting of ***Stochastic Processes***

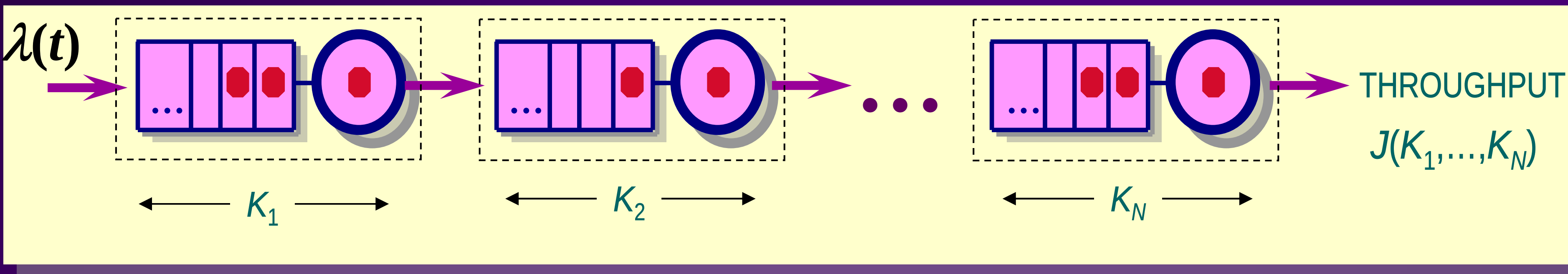
- Associate with each event *i a* ***Lifetime Distribution*** based on which $\boldsymbol{v}_i$ is generated

↓

Generalized Semi-Markov Process
(GSMP)

In a simulator, $\boldsymbol{v}_i$ is generated through a pseudorandom number generator

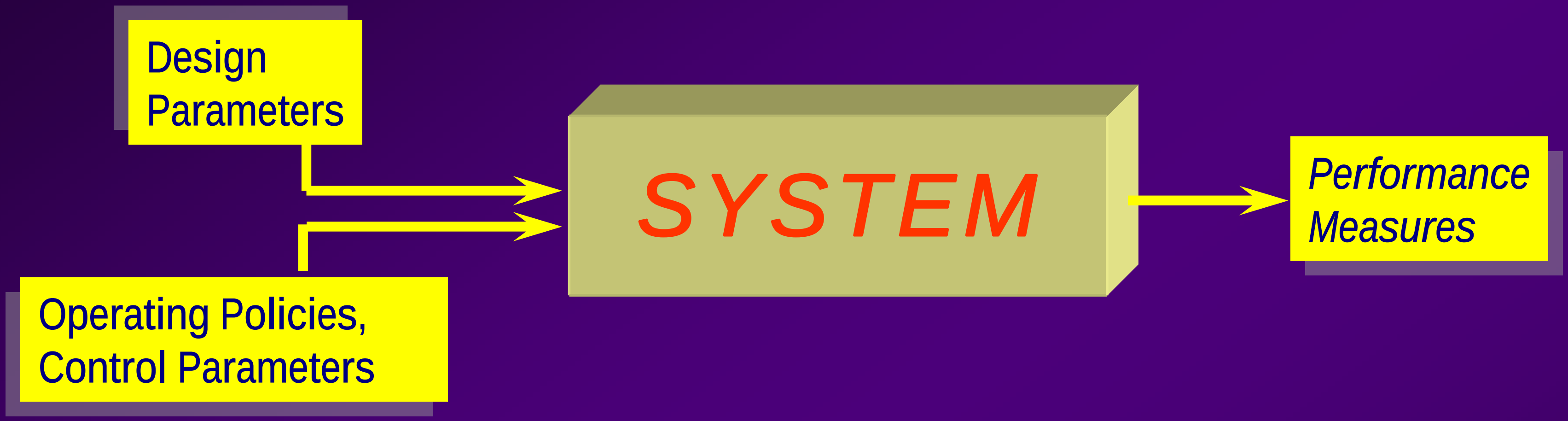
BACK TO THE *BUFFER ALLOCATION* PROBLEM



$$\max_{K_1,\ldots,K_N} J(K_1,\ldots,K_N) \;\text{ s.t.} \sum_{i=1}^{N} K_i = C$$

PARAMETRIC OPTIMIZATION

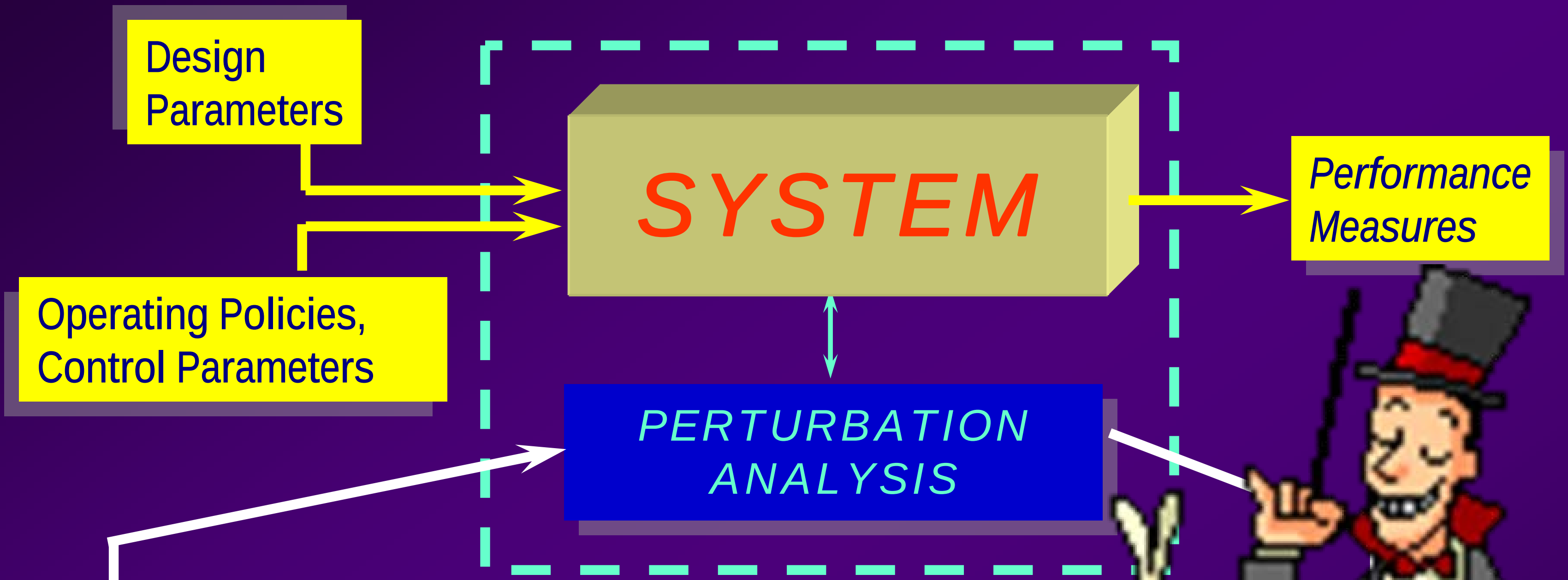
LEARNING BY TRIAL AND ERROR



CONVENTIONAL TRIAL-AND-ERROR ANALYSIS (e.g., simulation)

- *Repeatedly change parameters/operating policies*
- *Test different conditions*
- *Answer multiple WHAT IF questions*

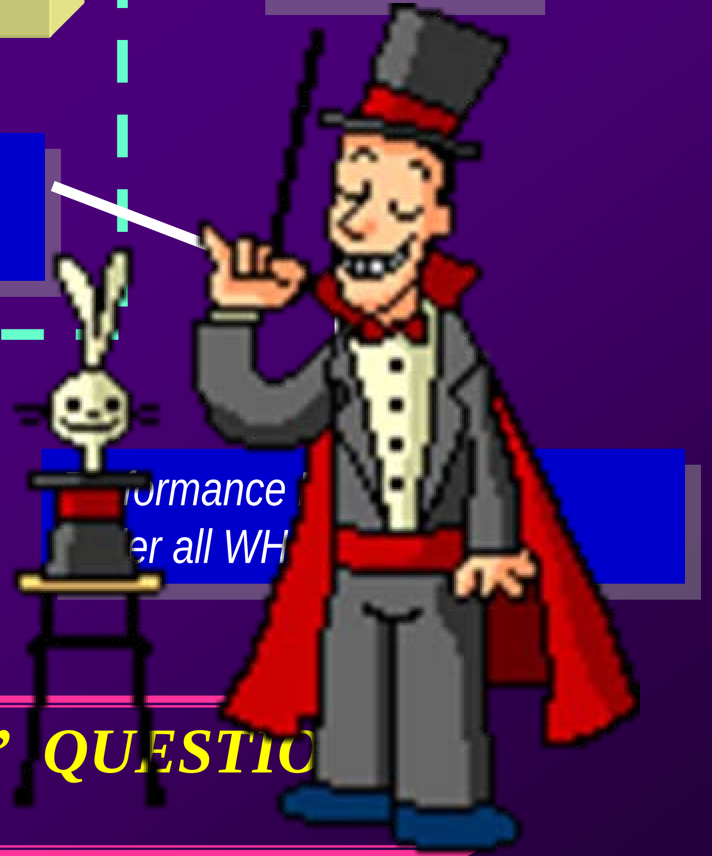
LEARNING THROUGH *PERTURBATION ANALYSIS*



WHAT IF…

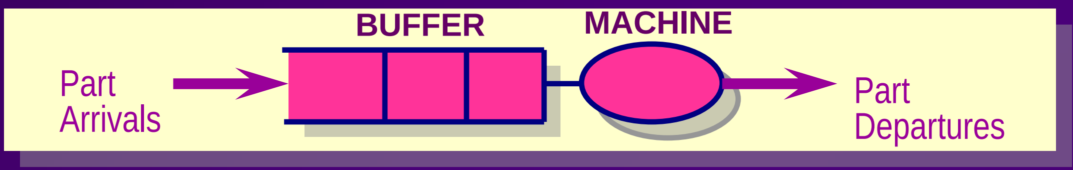
- Parameter $p_1 = a$ were replaced by $p_1 = b$
- Parameter $p_2 = c$ were replaced by $p_2 = d$

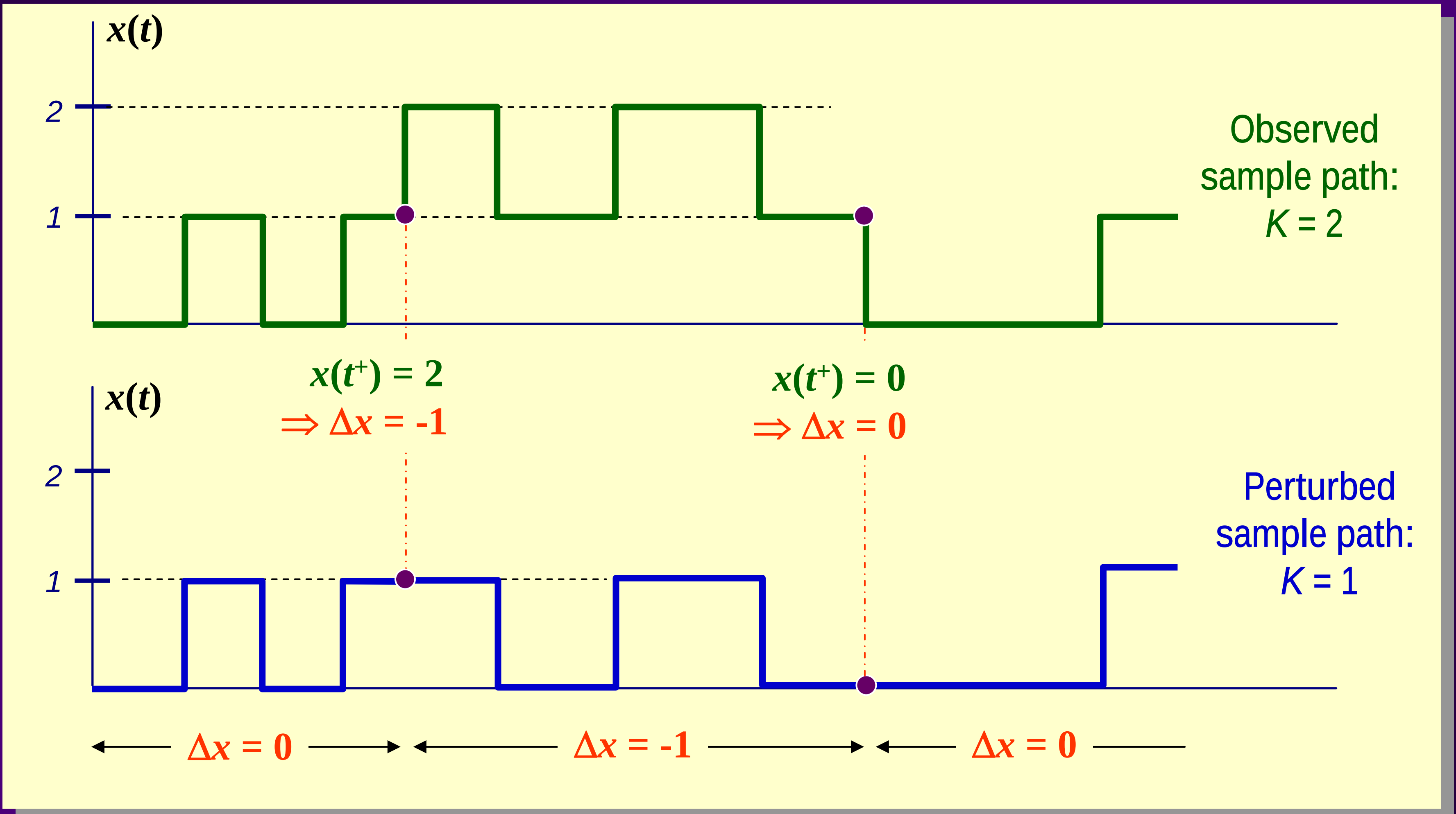
  ⋮



ANSWERS TO MULTIPLE "WHAT IF" QUESTIO
AUTOMATICALLY PROVIDED

PERTURBATION ANALYSIS





PERTURBATION ANALYSIS

CONTINUED

PERTURBATION DYNAMICS OBTAINED
FROM OBSERVED NOMINAL SAMPLE PATH ONLY!

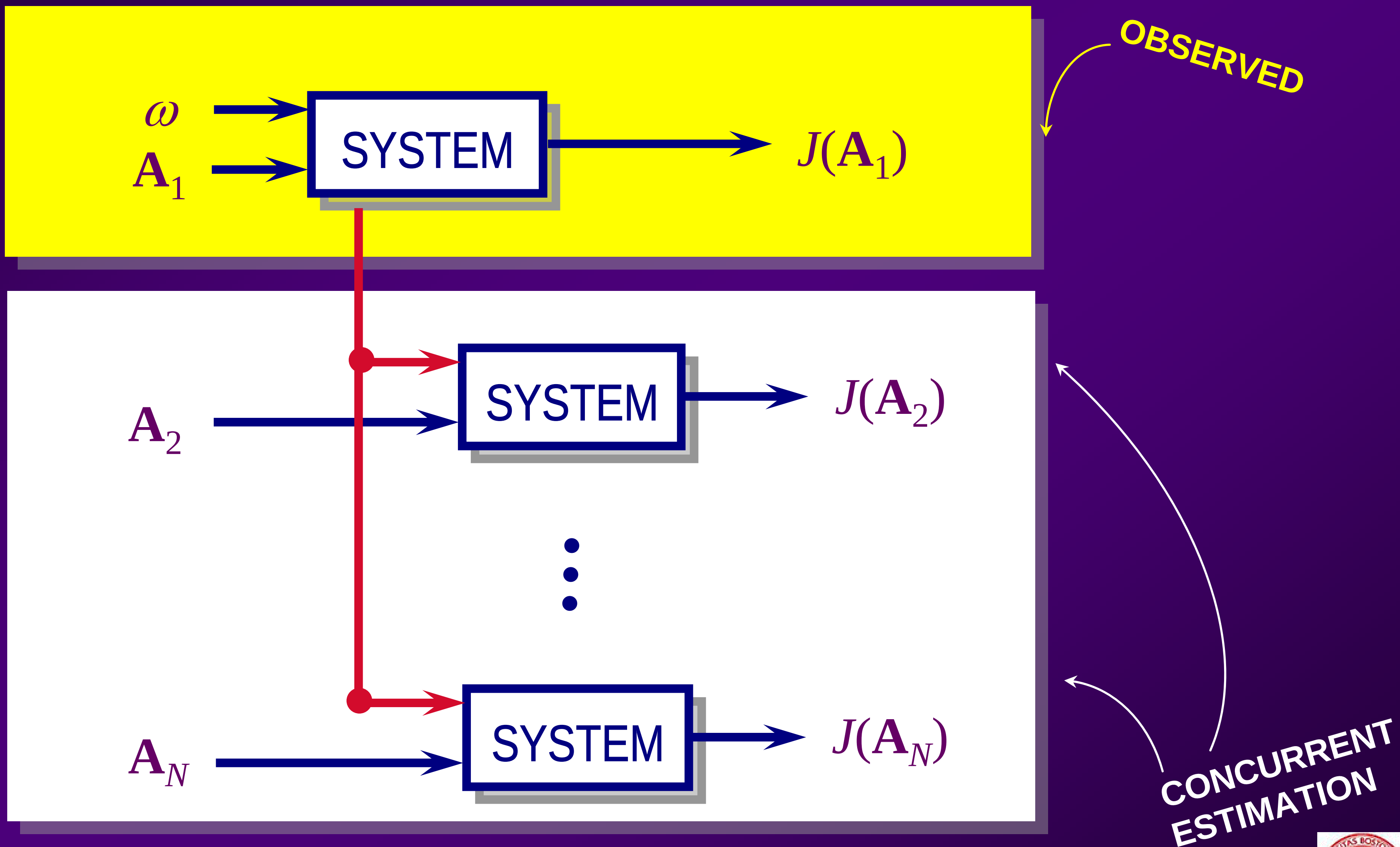
$$\Delta x(t+\delta;\theta,\Delta\theta) = f[\Delta x(t;\theta,\Delta\theta), x(t;\theta); \theta, \Delta\theta]$$

Why does this work?
Because structural knowledge of *nominal system dynamics* is also used

- Constructability Theory: Conditions under which this is possible and methods for constructing perturbed sample paths
- Performance Analysis: $$\Delta J(t+\delta;\theta,\Delta\theta) = f[\Delta J(t;\theta,\Delta\theta), x(t;\theta); \theta, \Delta\theta]$$
- Perturbation Analysis: Obtaining unbiased, consistent estimators for $\dfrac{dJ}{d\theta}$

BACK TO THE *BUFFER ALLOCATION* PROBLEM

ALLOCATION VECTOR: $\mathbf{A} = [K_1, \ldots, K_N]$



"LET'S CONTROL EVERYTHING" IN THE 1980's…

Anonymous referee comments for 1983 papers on Supervisory Control:

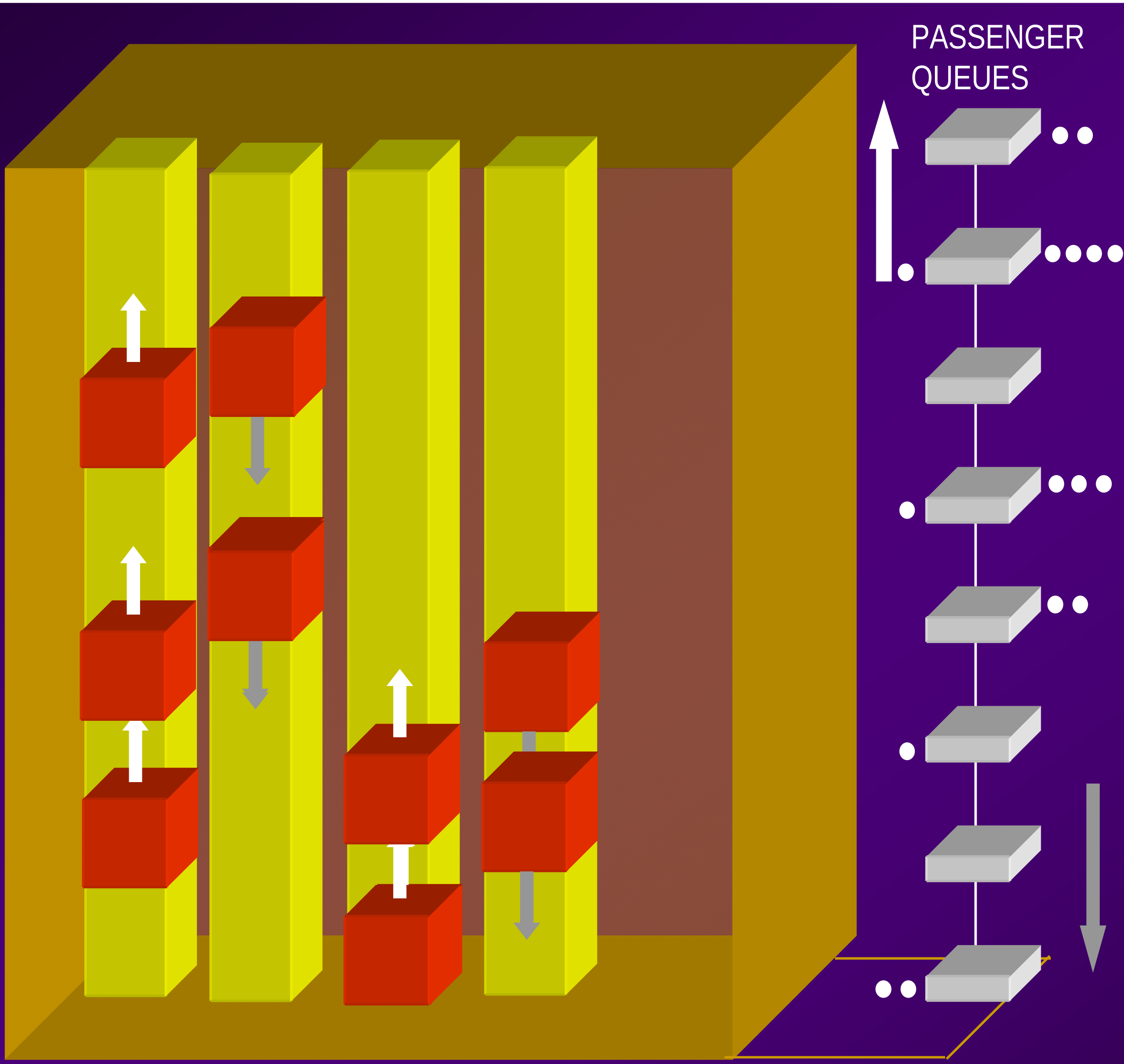
(courtesy W.M. Wonham)

- *Automatica* (reject)
  "Automata have no place in control engineering"
- *Math. Systems Theory* (reject)
  "FSM and regular languages are nothing new at best and trivial at worst"
- *SIAM J. Control and Optimization* (accept)
  "If it's optimal control we'll take it"

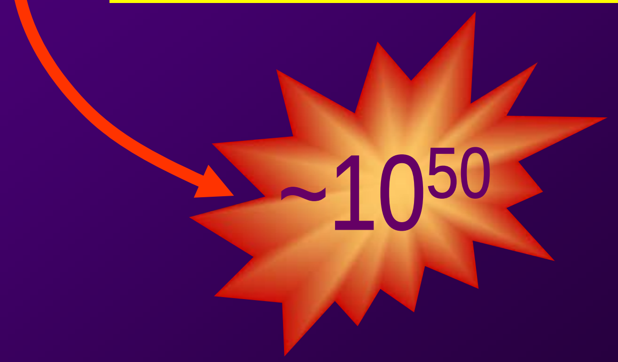
W.M. Wonham's conclusion:

"Crossing cultural divides can be a chilly business"

ELEVATOR DISPATCHING CONTROL



COMPLEXITY:
Huge state space,
Movement constraints,
incomplete state info.,
etc.



ELEVATOR DISPATCHING CONTROL

CONTINUED

PROBLEM (*OTIS Elevator, circa 1987*)

- How should an elevator respond to calls?
- At each floor: *STOP*, *GO*, or *SWITCH DIRECTION*?
- *Goal*: *Reduce waiting times and preserve fairness over all floor response times*

Can you outperform existing patented elevator dispatch control?

Challenge posed to…

- A Computer Scientist
- A Control Engineer

How would you proceed ?

Computer Scientist

Build a database…

… collect data…

… *design a user interface*…

Control Engineer

Build a model…

… formulate control/optimization problem…

Does a solution exist ?

…*and, if so, is it unique* ?

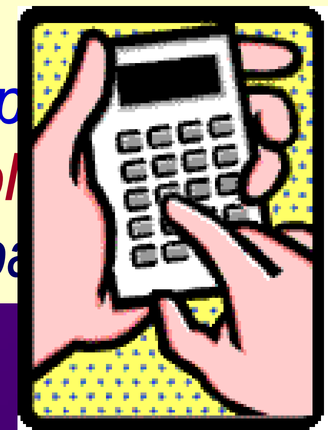
ELEVATOR DISPATCHING CONTROL *CONTINUED*

2 weeks later, they are ready to propose…

Computer Scientist

$0.5M, 6-month project

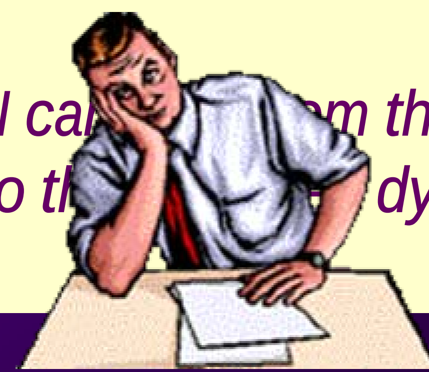
Will deliver a database-driven Exp… using AI methods and a java-based full-col… nning on a LAN with agent-based HLA-compa… erability



Control Engineer

$50K, 1-year project

Let me investigate what insight I ca… m the system dynamics, and seek a solution to th… dynamic optimization problem…



ELEVATOR DISPATCHING CONTROL *CONTINUED*

6 months later: *Can you outperform existing technology?*

Computer Scientist	Success!... Expert system finds solutions… not always good, but that's because we need a faster CPU for the neural net training… Need $200K more to upgrade.
Control Engineer	Well, here is an algorithm that improves performance by 20%... But I can prove convergence of my solution to the optimal only "in probability" -- not "with prob. 1"…

ELEVATOR DISPATCHING CONTROL *CONTINUED*

2 years later:

Computer Scientist

Still adding rules to the Expert System and training the neural net...
...but that User Interface is a real beauty!

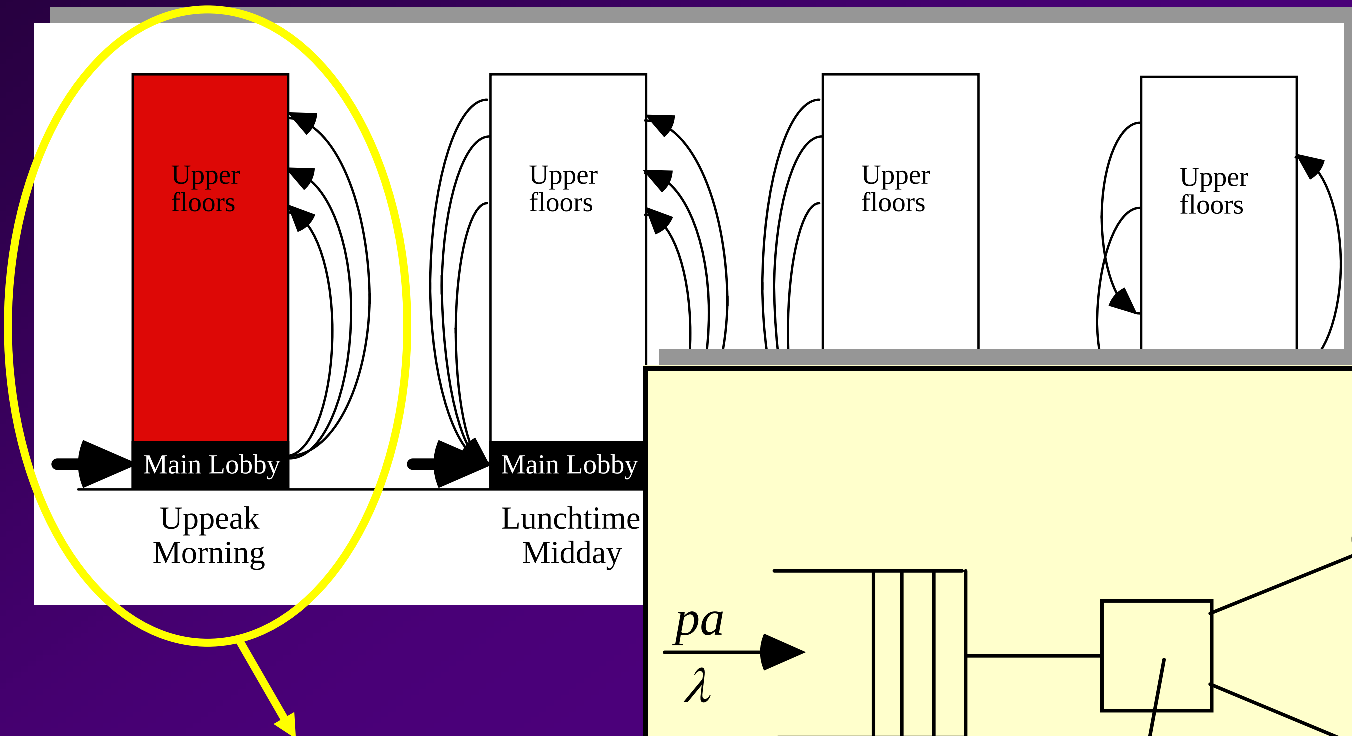
Control Engineer

Algorithms developed now outperform existing solutions by 30%...
...Asking for $0.5M to add...
java-based full-colored GUI running on a LAN with agent-based HLA-compatible interoperability

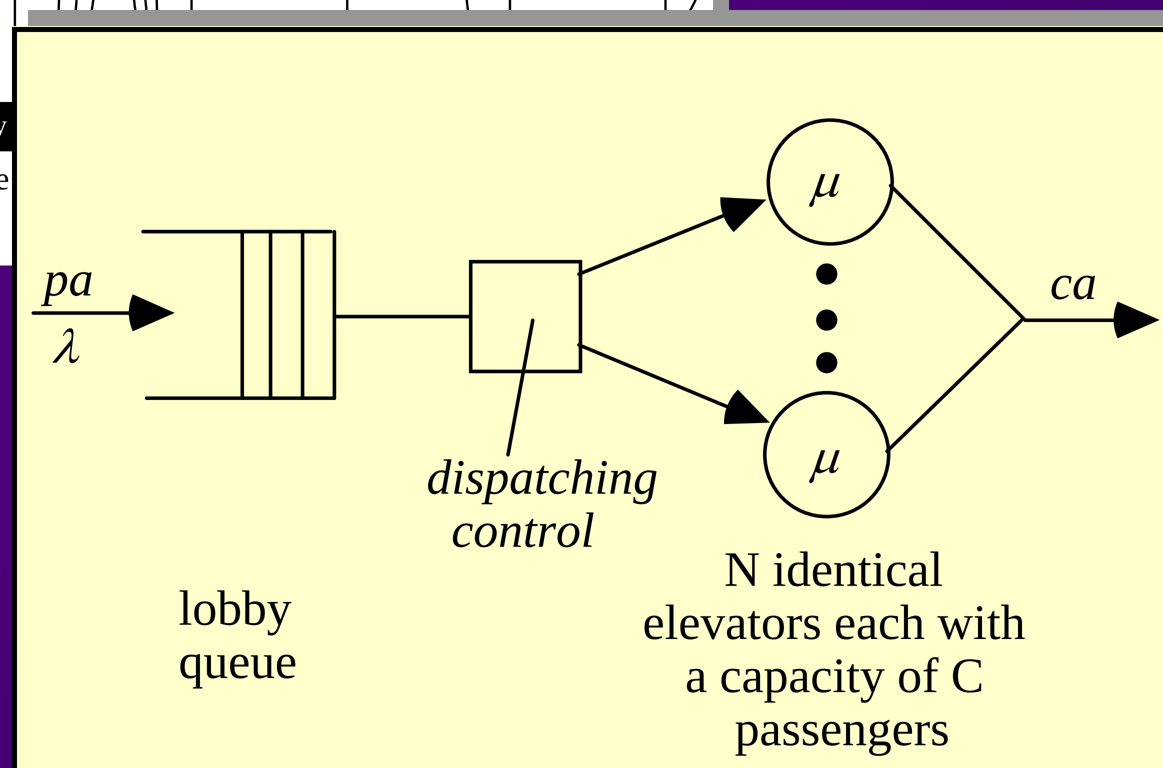
A glimpse of the Control Engineer's approach…

ELEVATOR DISPATCHING

CONTINUED

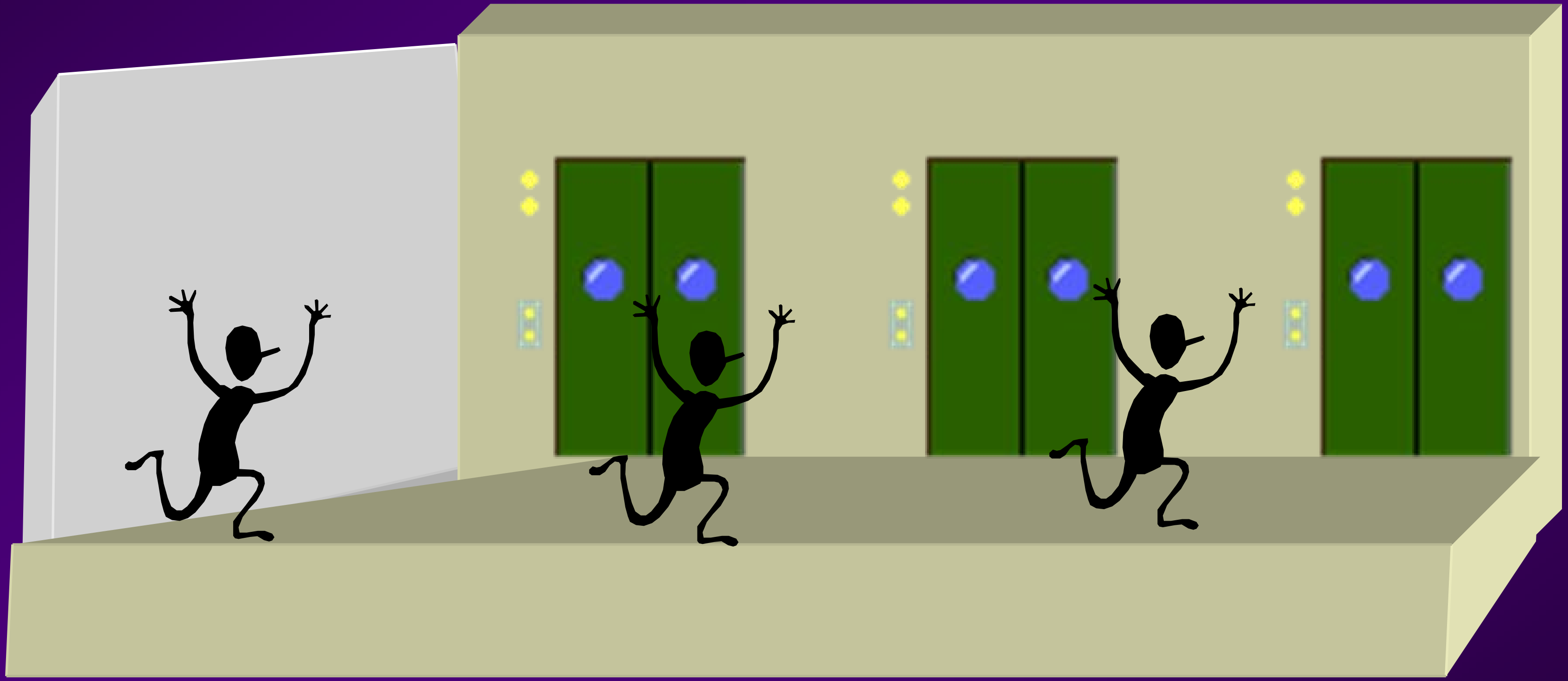


UPPEAK DISPATCHING CONTROL PROBLEM:



HOW *NOT* TO CONTROL…

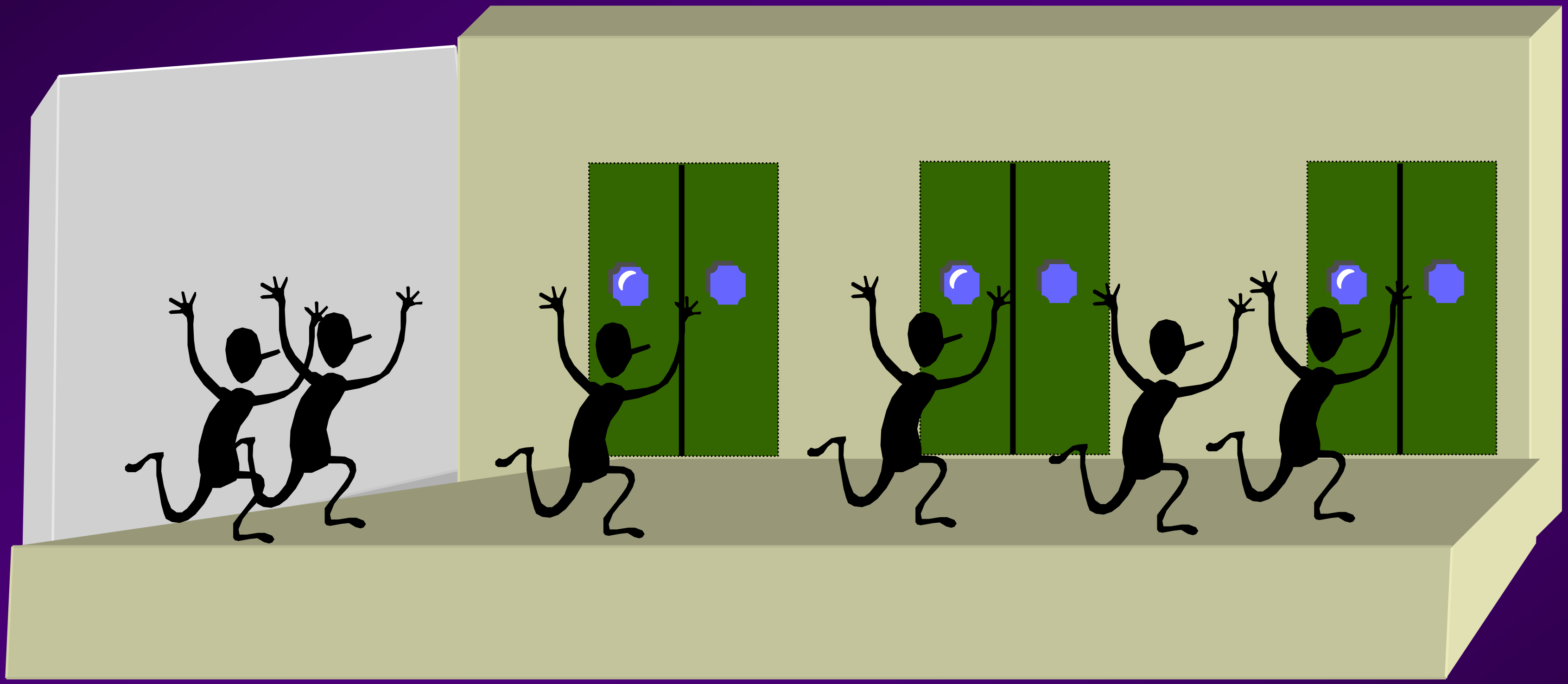
3 elevators available at lobby…



Each person takes one and goes

HOW *NOT* TO CONTROL…

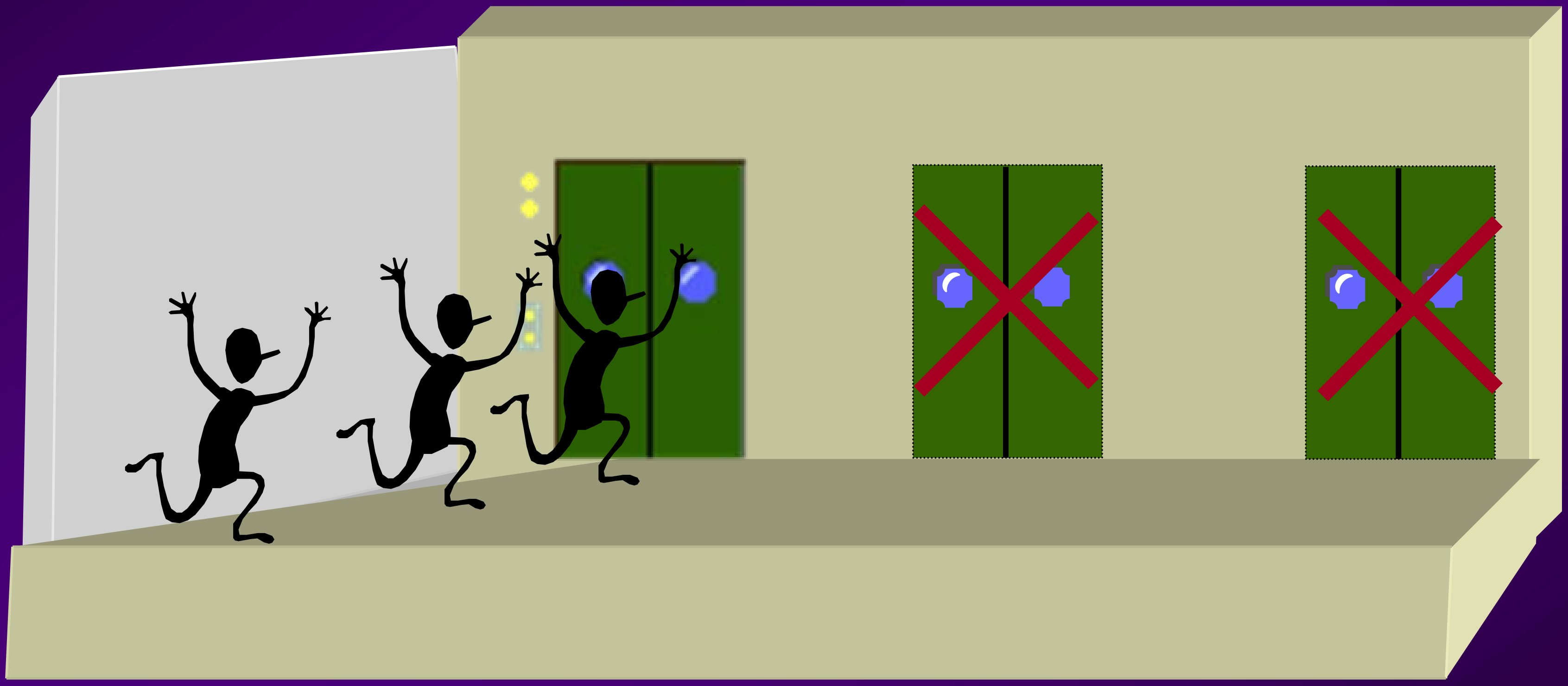
CONTINUED



Long waiting results…

A BETTER WAY TO CONTROL…

Force only 1 of the 3 elevators to be available



ELEVATOR DISPATCHING

Can formally show (*TCST, 1997*) that a Markov Decision Problem formulation leads to a *threshold-based optimal policy* minimizing average waiting time.

Threshold parameters depend on
- *passenger arrival rate*
- *car service rate*

CONTROLLER:

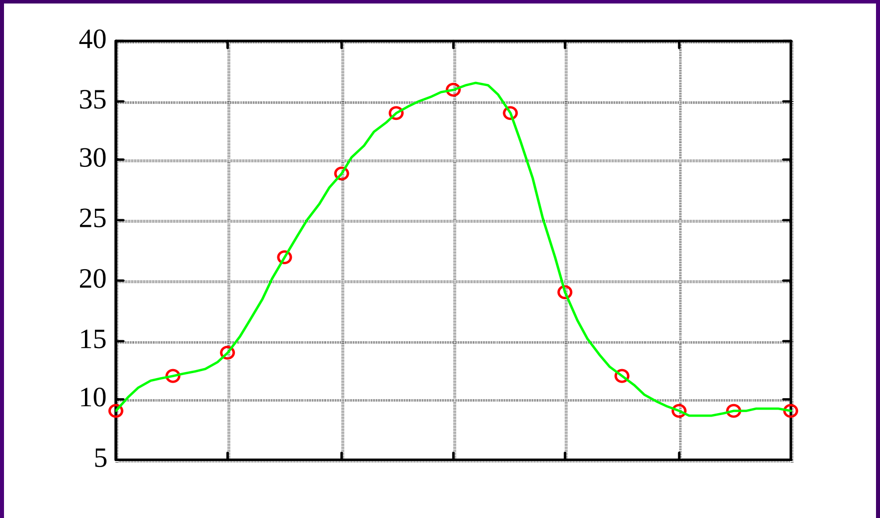
- Load one car at a time
- Dispatch this car when

number of passengers inside car $\geq \theta(\lambda,\mu)$

ELEVATOR DISPATCHING

CONTINUED

Variation in λ over 12 5-min. intervals for 1 hour uppeak traffic (*courtesy B. Powell, OTIS Elevator*)



PROBLEM:

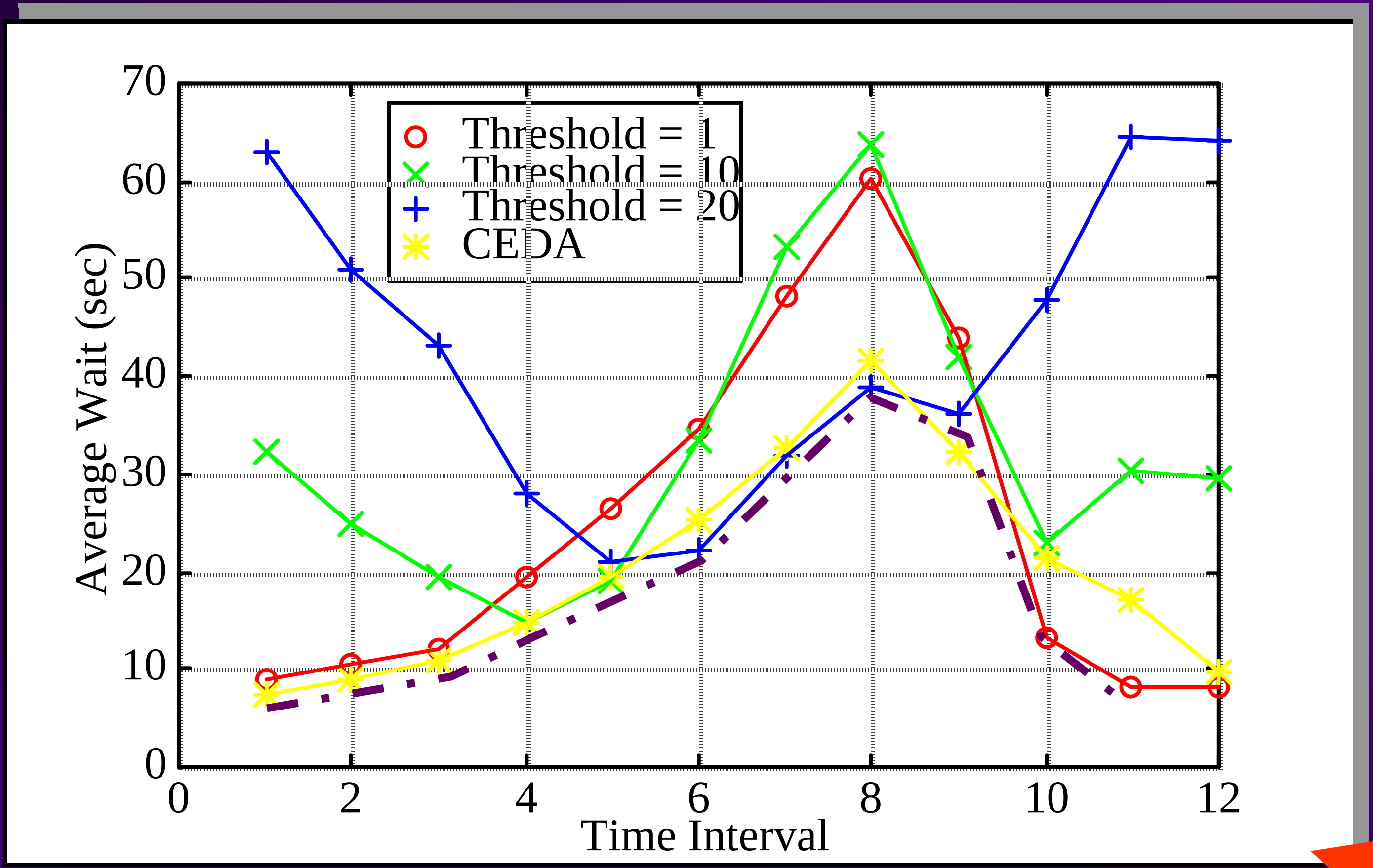
- How to determine 12 thresholds, one for each 5 min. interval of fixed traffic rate?
- How to automatically adjust them on line?

CONCURRENT ESTIMATION APPROACH:

- Choose any set of 12 thresholds
  (one for each 5-min. interval)
- Observe system under given thresholds
- Apply Concurrent Estimation to *"learn"* effect of all other feasible thresholds
  (*i.e., infer performance under hypothetical threshold values*)
- Optimize thresholds

ELEVATOR DISPATCHING

CONTINUED



CEDA:
*C*oncurrent
*E*stimation
*D*ispatching
*A*lgorithm

How fast did the CEDA learn optimal thresholds?

- Approximately 5 uppeak periods to converge (i.e., 5 real days)

THE FUTURE

- *COMPLEXITY*
- *HYBRID SYSTEMS*
- *COMPUTATIONAL CHALLENGES*
- *NEW OPTIMIZATION METHODS*

OPPORTUNITIES…

- *MIXED INITIATIVE CONTROL OF AUTOMA-TEAMS (MICA)*

THREE FUNDAMENTAL *COMPLEXITY LIMITS*

$1/T^{1/2}$ LIMIT

NP-HARD LIMIT

one order increa

$\Big|$ *INFO. SPACE* $\Big|$

Tradeoff between **GENERALITY** and **EFFICIENCY** of an algorithm
[*Wolpert and Macready, 1997*]

NO-FREE-LUNCH LIMIT

THREE FUNDAMENTAL *COMPLEXITY LIMITS*

1/$T^{1/2}$ LIMIT

NP-HARD LIMIT

NO-FREE-LUNCH LIMIT

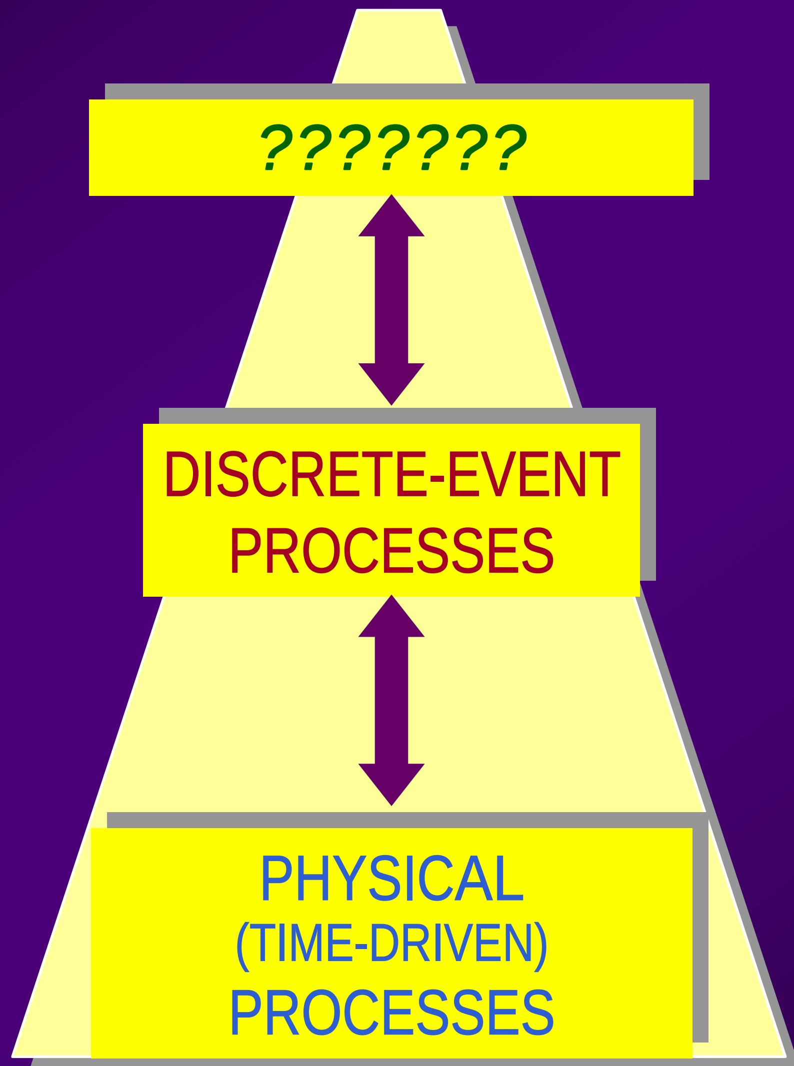
HIERARCHICAL DECOMPOSITION OF COMPLEX SYSTEMS

TIME SCALE

Weeks - Months

Minutes - Weeks

*m*sec - Hours



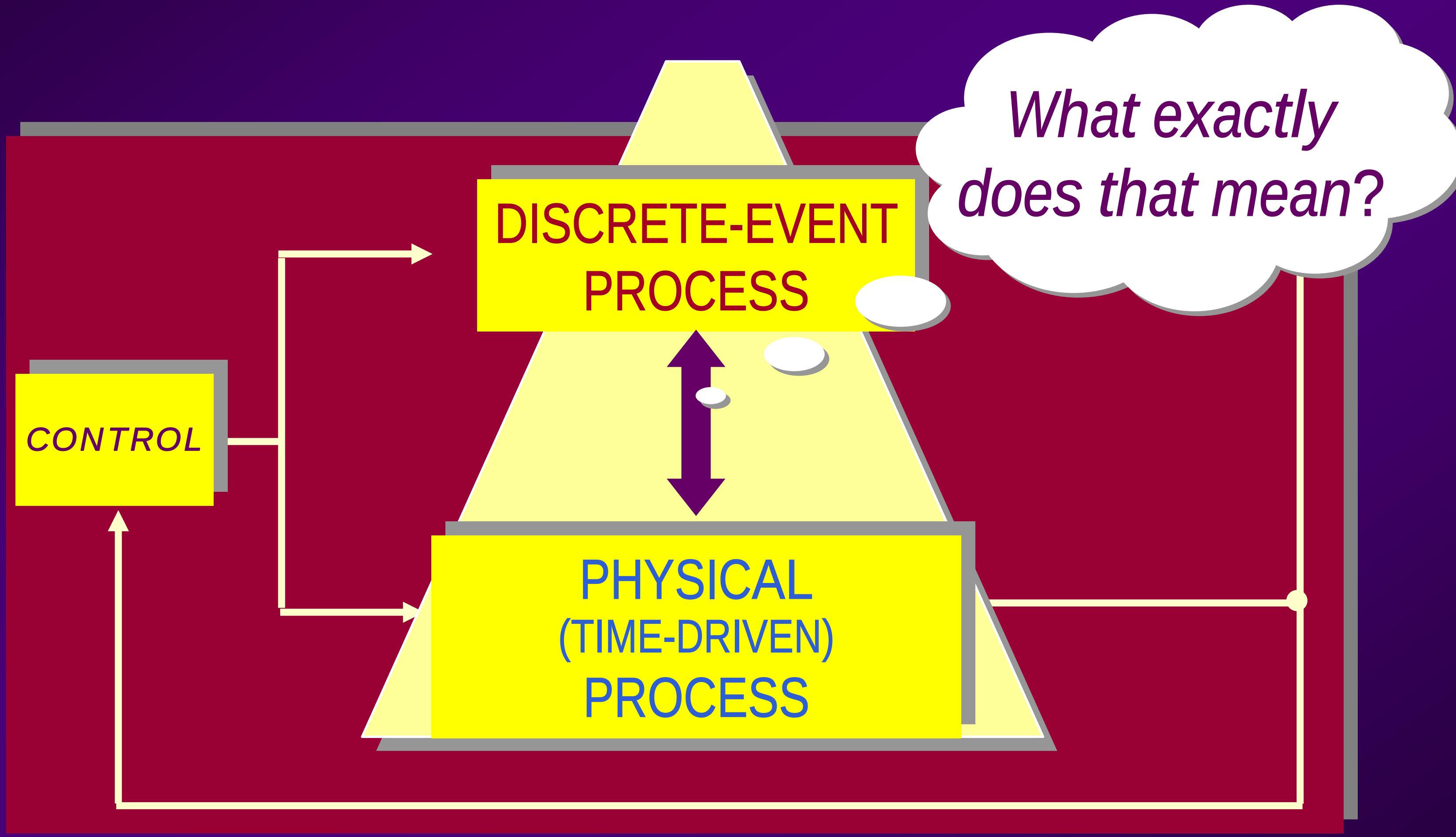
MODEL

Diff. Eq’s, LP

Automata, Simulation, Queueing

Diff. Eq’s, Detailed Simulation

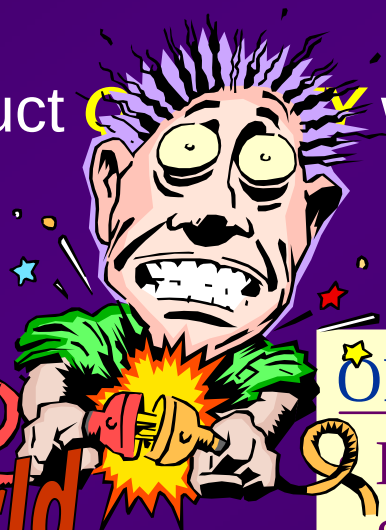
HYBRID SYSTEMS



HYBRID SYSTEMS IN MANUFACTURING

Manufacturing system integration *(ALCOA, 1997)* **:**

- How to integrate *'process control'* with *'operations control'* ?
- How to improve product ***QUALITY*** within reasonable ***TIME*** ?



PROCESS CONTROL

- Physicists
- Material Scientists
- Chemical Engineers
- …

Time-Driven World

OPERATIONS CONTROL

- Industrial Engineers, OR
- Schedulers
- Inventory Control
- …

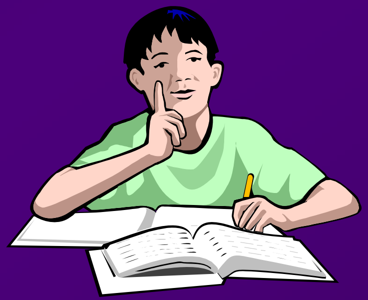
Event-Driven World

ACKNOWLEDGEMENTS

Thanks to several co-workers whose contributions are reflected in this talk…

… All 13 Former

+ Current PhD Students…



…Colleagues…

Y.C. (Larry) Ho
Wei-Bo Gong
Yorai Wardi
Young Cho

Liyi Dai
Xiren Cao
Ted Djaferis
Doug Looze
Felisa Vazquez-Abad

…and Sponsors…

NSF, AFOSR, AFRL, DARPA, NASA, EPRI, ARO, UT, ALCOA, NOKIA, GE

TO CONCLUDE…

…LET'S APPLY CONTROL PRINCIPLES LIKE FEEDBACK, etc. IN EVERYTHING…

…BUT…

LET'S NOT GET CARRIED AWAY EITHER…