COMPLEXITY MADE SIMPLE *

* AT A SMALL PRICE

C. G. Cassandras

Division of Systems Engineering
Dept. of Electrical and Computer Engineering
Center for Information and Systems Engineering
Boston University

cgc@bu.edu, http://people.bu.edu/cgc

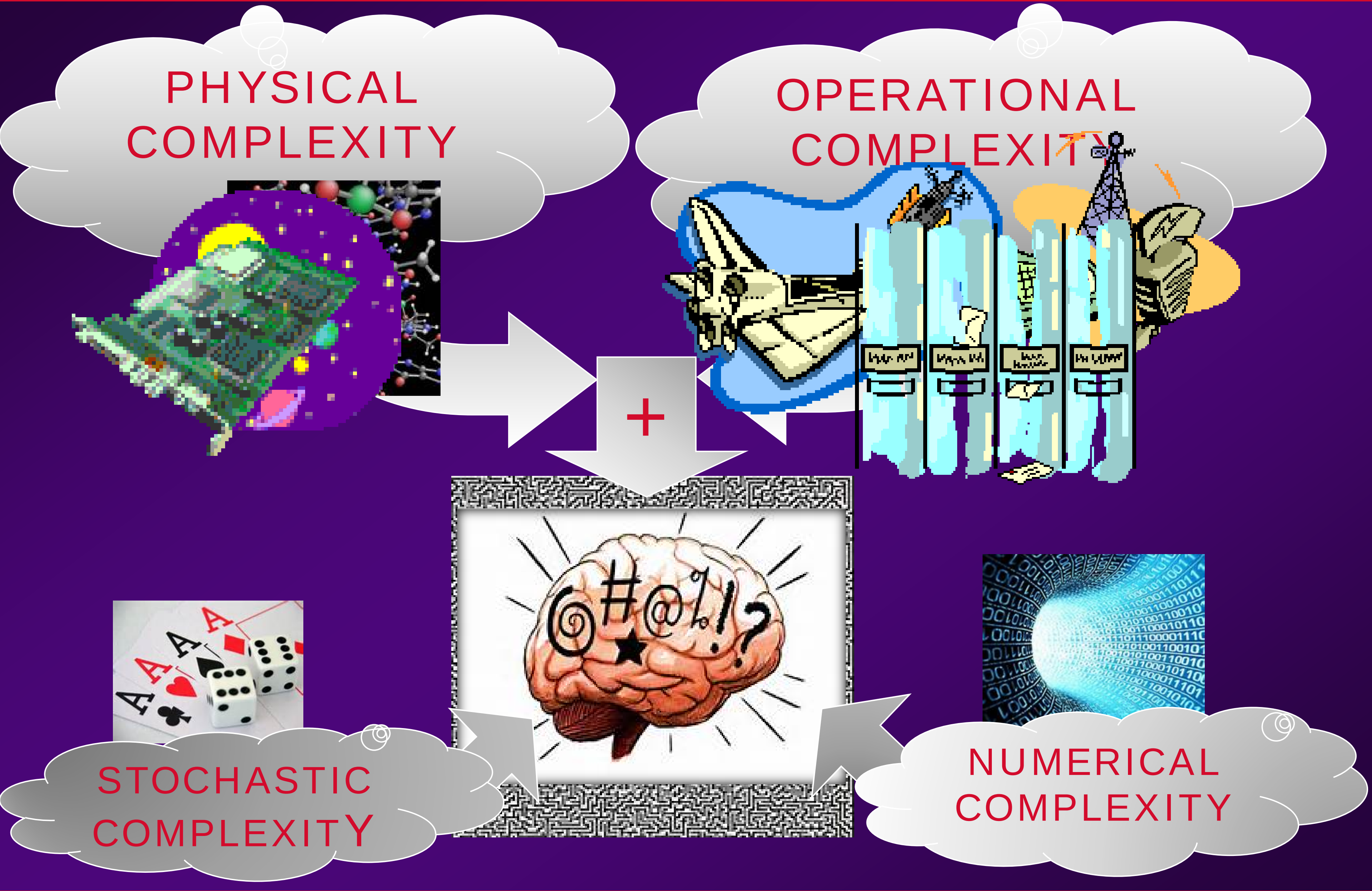
There are THREE FUN
constrain the design,
management of COMP

But we can often (
around these limit

- ...by exploiting the INTERNAL STRUCTURE of a system (avoid “brute-force” analysis)

- ...by asking the “RIGHT” QUESTIONS (get the most “bang for the buck”)

COMPLEXITY


PHYSICAL COMPLEXITY
OPERATIONAL COMPLEXITY
+
STOCHASTIC COMPLEXITY
NUMERICAL COMPLEXITY

THREE FUNDAMENTAL *COMPLEXITY LIMITS*

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

NP-HARD LIMIT

Tradeoff between
GENERALITY and EFFICIENCY
of an algorithm

[*Wolpert and Macready, IEEE TEC, 1997*]

uncertainty decreases as $1/\sqrt{N}$)

NO-FREE-LUNCH LIMIT

THREE FUNDAMENTAL *COMPLEXITY LIMITS*

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

NP-HARD
LIMIT

NO-FREE-LUNCH
LIMIT

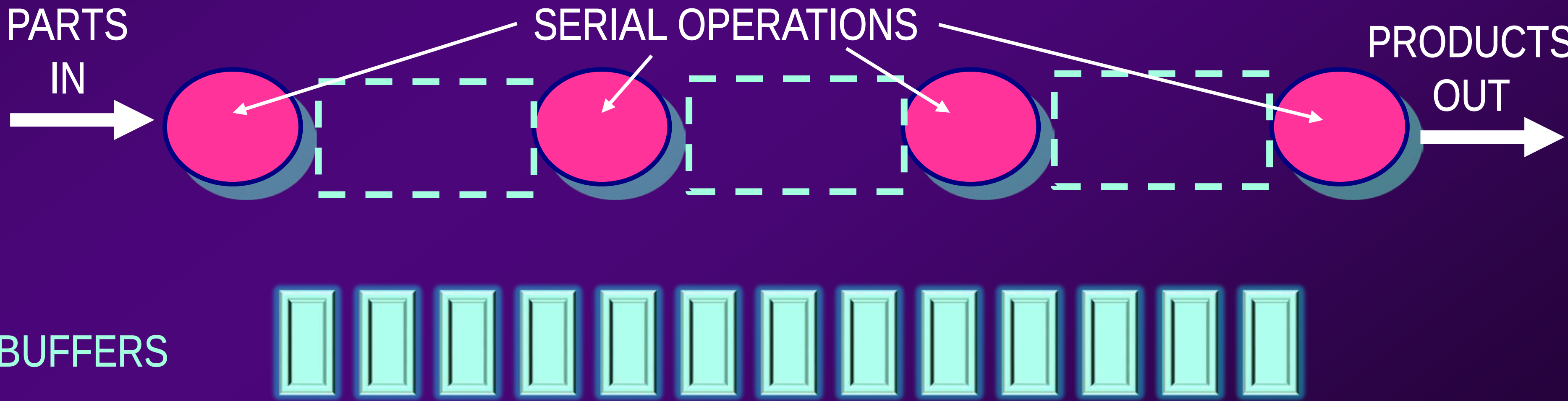
EXPLOITING STRUCTURE TO LEARN COMPLEX SYSTEM BEHAVIOR *FAST*

- Discrete Event Dynamic Systems
- Perturbation Analysis Theory

THE FIAT PROBLEM (circa 1978)

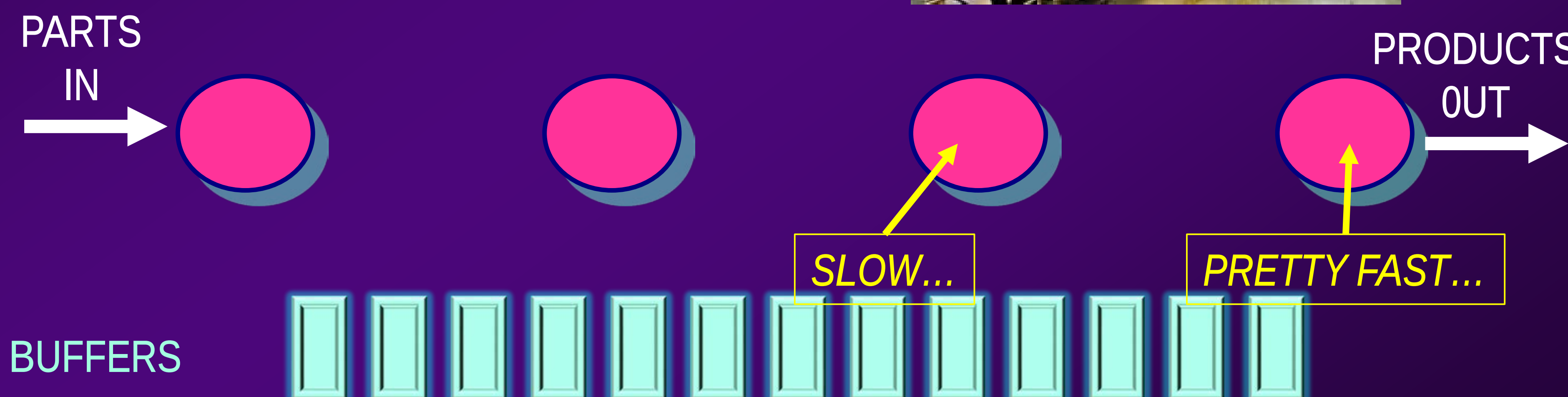
What is the best way to distribute BUFFER capacity in a manufacturing transfer line?

Ho, Eyler, Chien, Cassandras, Cao,...



THE FIAT PROBLEM (circa 1978)

What is the best way to distribute BUFFER capacity in a manufacturing transfer line?



THE FIAT PROBLEM (circa 1978)

Complexity of this buffer allocation process (***K*** buffers, ***N*** stages)

$$\binom{K+N-1}{K}$$

Example: $K = 24$, $N = 6 \longrightarrow$ **118,755** possible allocations

- "Brute Force" trial-and-error: test each allocation for about a week to get statistically meaningful results
  (that's if the manager allows you to mess with the system…)

  $\longrightarrow$ about **2300 YEARS…**

- Suppose you can reduce to only 1000 "promising" allocations:

  $\longrightarrow$ about **19 YEARS…**

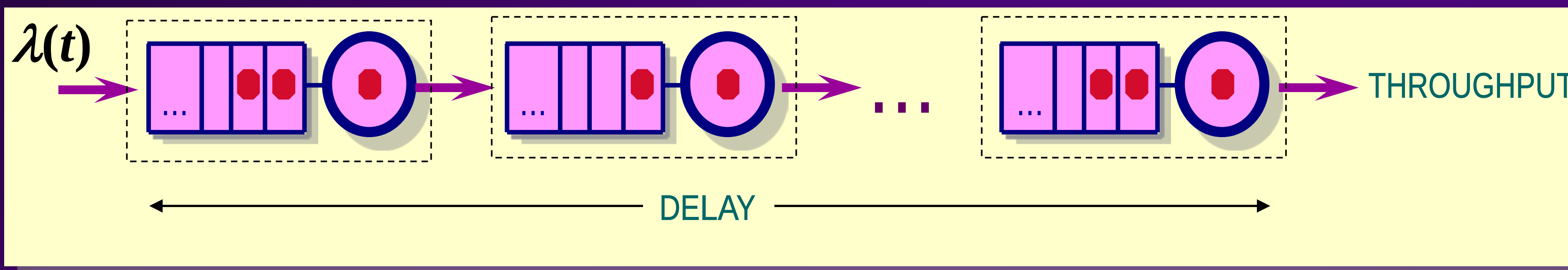
- Using a simulated transfer line, about 3 minutes per tri computing technology…)

  $\longrightarrow$ about **250 DAYS…**

Slow and painful…

WHY IS THIS PROBLEM IMPORTANT ?

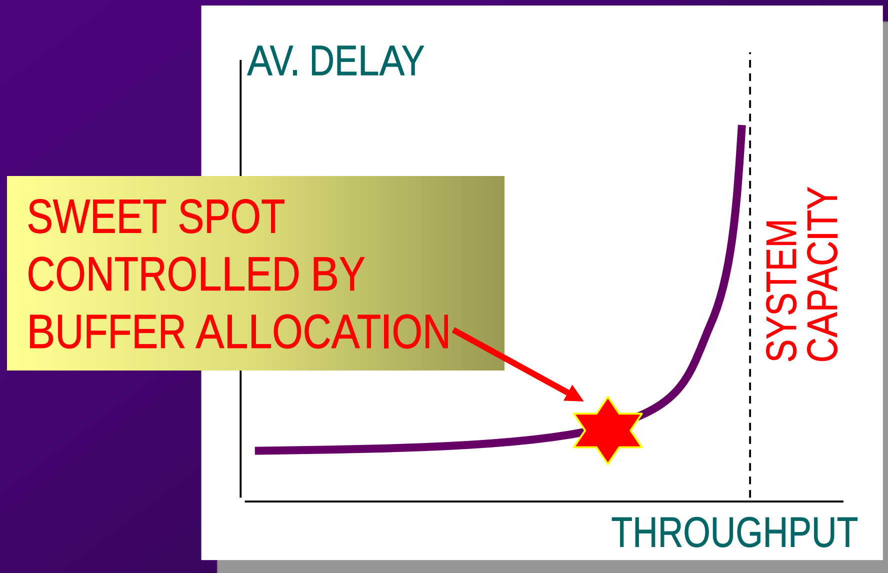
Manufacturing system with N sequential operations:



INCREASE $\lambda(t)$

THROUGHPUT increases (GOOD)

average DELAY increases (BAD)



TWO KEY OBSERVATIONS

1. This is a dynamic system.

   But it's not like the usual TIME-DRIVEN ones, i.e., described by differential equations

$$\frac{dx}{dt} = f(x,u,t)$$

   Need a NEW modeling framework for these EVENT-DRIVEN systems

   → *DISCRETE EVENT DYNAMIC SYSTEMS*

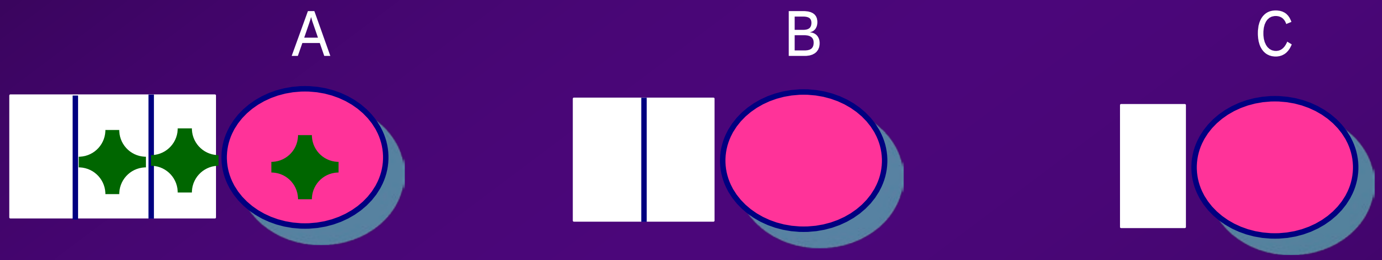
2. You don't need brute-force trial-and-error for each allocation…

   Once the system dynamics are understood, you can predict what happens by changing allocations (*adding*, *removing*, *moving* buffers)

   → *PERTURBATION ANALYSIS THEORY*

SYSTEM DYNAMICS:

HOW ONE COMPONENT OF THE SYSTEM AFFECTS OTHER COMPONENTS



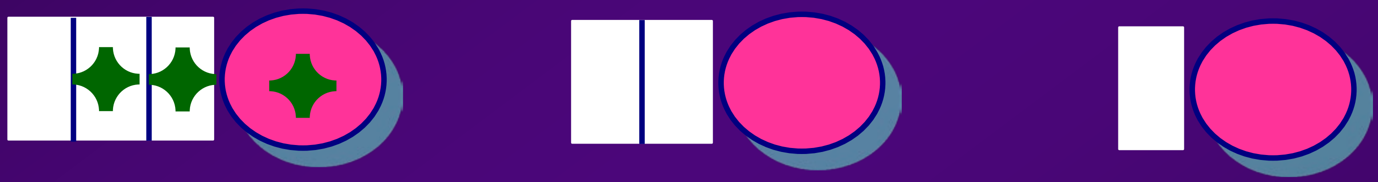
ARRIVAL 1

ARRIVAL 2

ARRIVAL 3

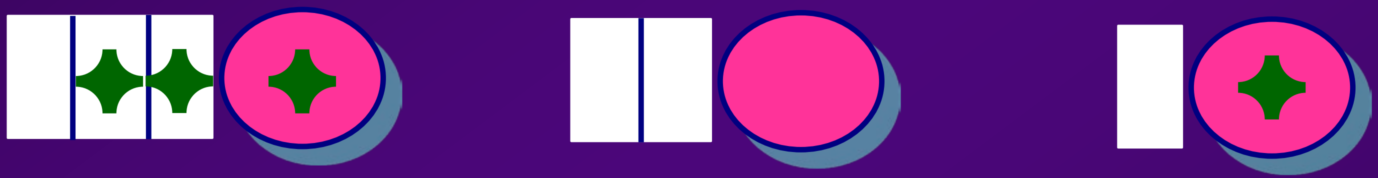
THE FIAT PROBLEM: SYSTEM DYNAMICS

DEPARTURE 1 FROM A




DEPARTURE 1 FROM B
ARRIVAL 4
IDLING

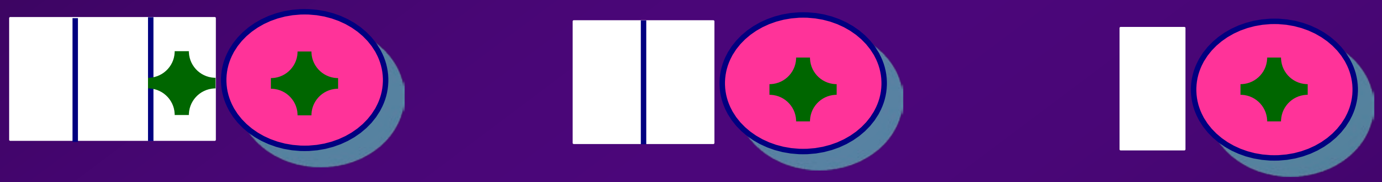
DEPARTURE 2 FROM A



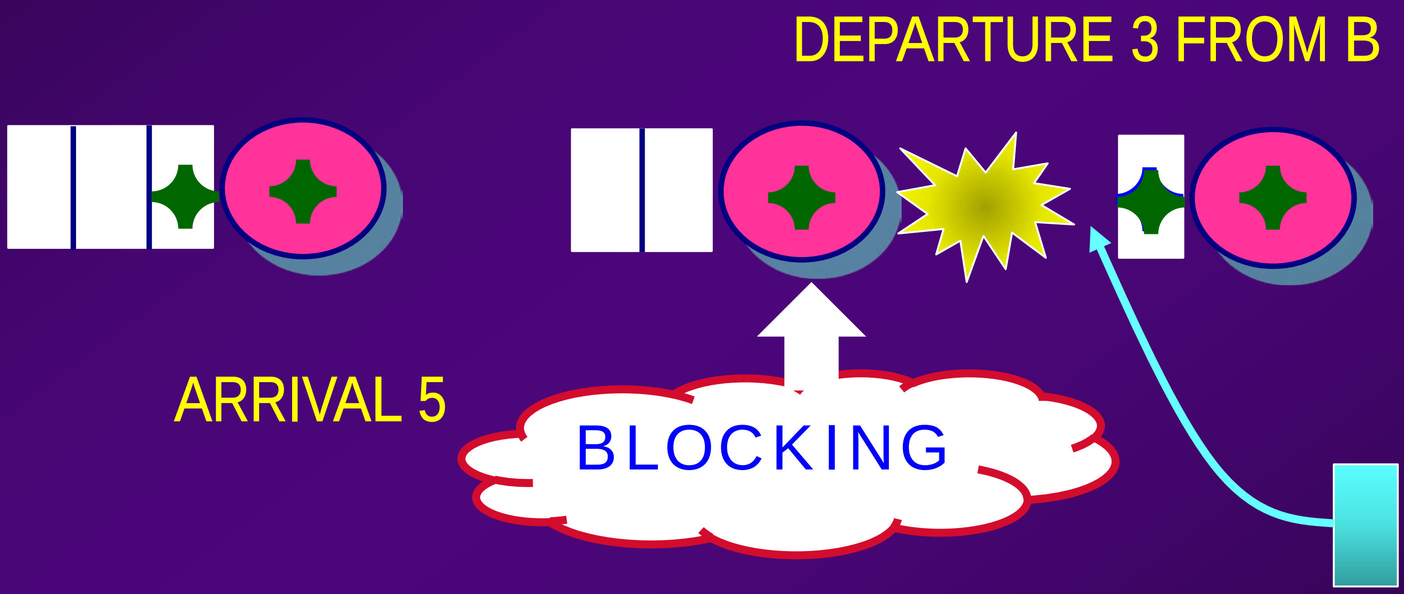
THE PROBLEM: SYSTEM DYNAMICS

DEPARTURE 3 FROM A

DEPARTURE 2 FROM B



THE PROBLEM: SYSTEM DYNAMICS



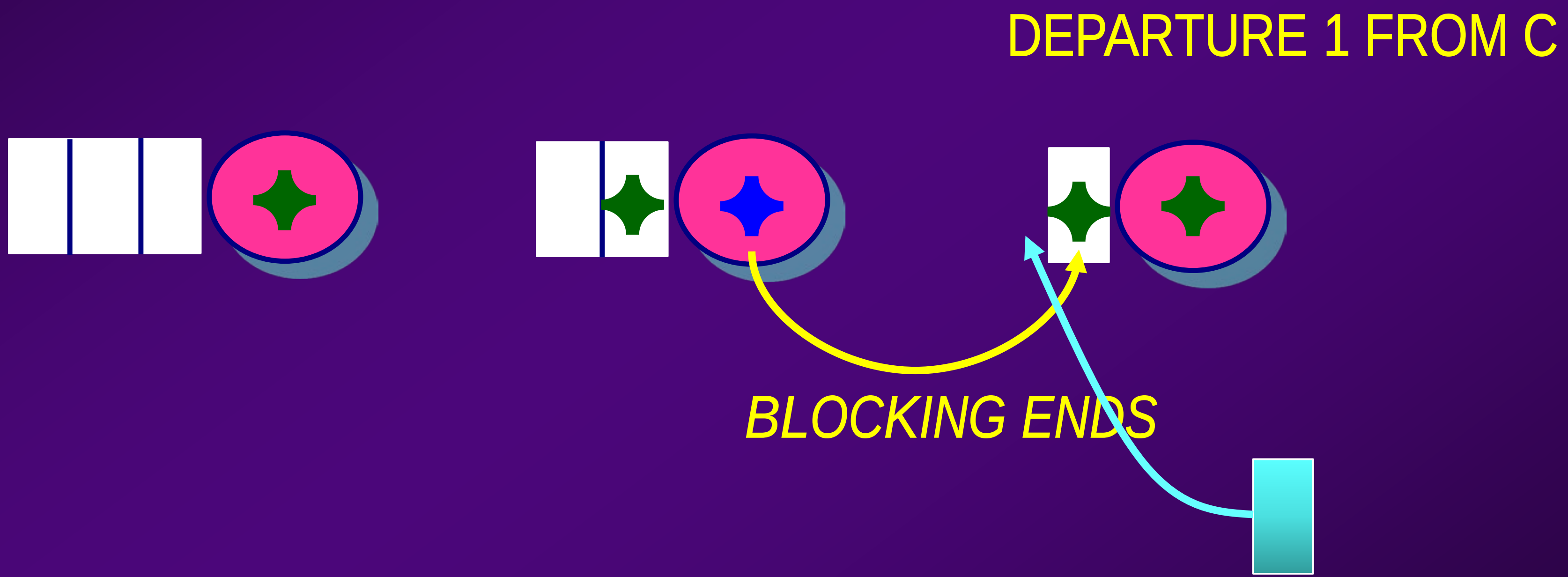
PERTURBATION ANALYSIS

WHAT IF THIS HAD BEEN ADDED?

RECORD BLOCK START TIME: $D_B(3)$

THE PROBLEM: SYSTEM DYNAMICS

DEPARTURE 4 FROM A

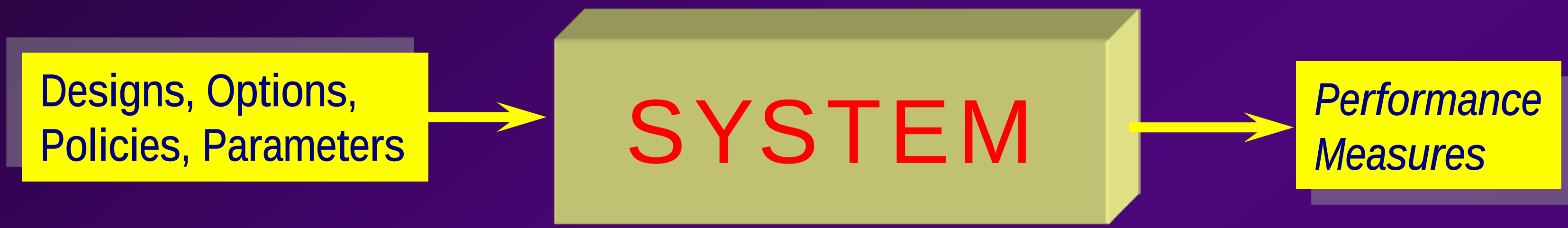


PERTURBATION ANALYSIS

RECORD BLOCK END TIME: $D_c(1)$

PART SYSTEM DELAY WOULD HAVE BEEN REDUCED BY: $D_c(1) - D_B(3)$

LEARNING BY TRIAL AND ERROR



CONVENTIONAL TRIAL-AND-ERROR ANALYSIS

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

Slow and painful…

N "What-If" questions $\Rightarrow$ N+1 trials !

LEARNING WITH *PERTURBATION ANALYSIS*

Designs, Options, Policies, Parameters → SYSTEM → *Performance Measures*

PERTURBATION ANALYSIS

WHAT IF…

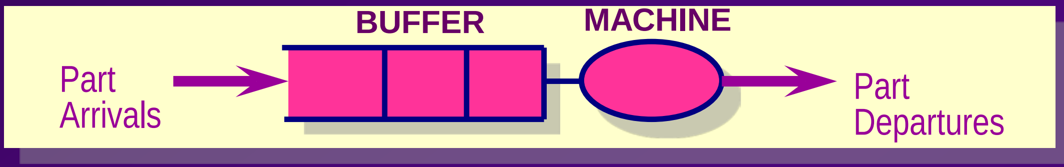
- Parameter $p_1 = a$ were replaced by $p_1 = b$
- Design option 1 were replaced by *option 2*

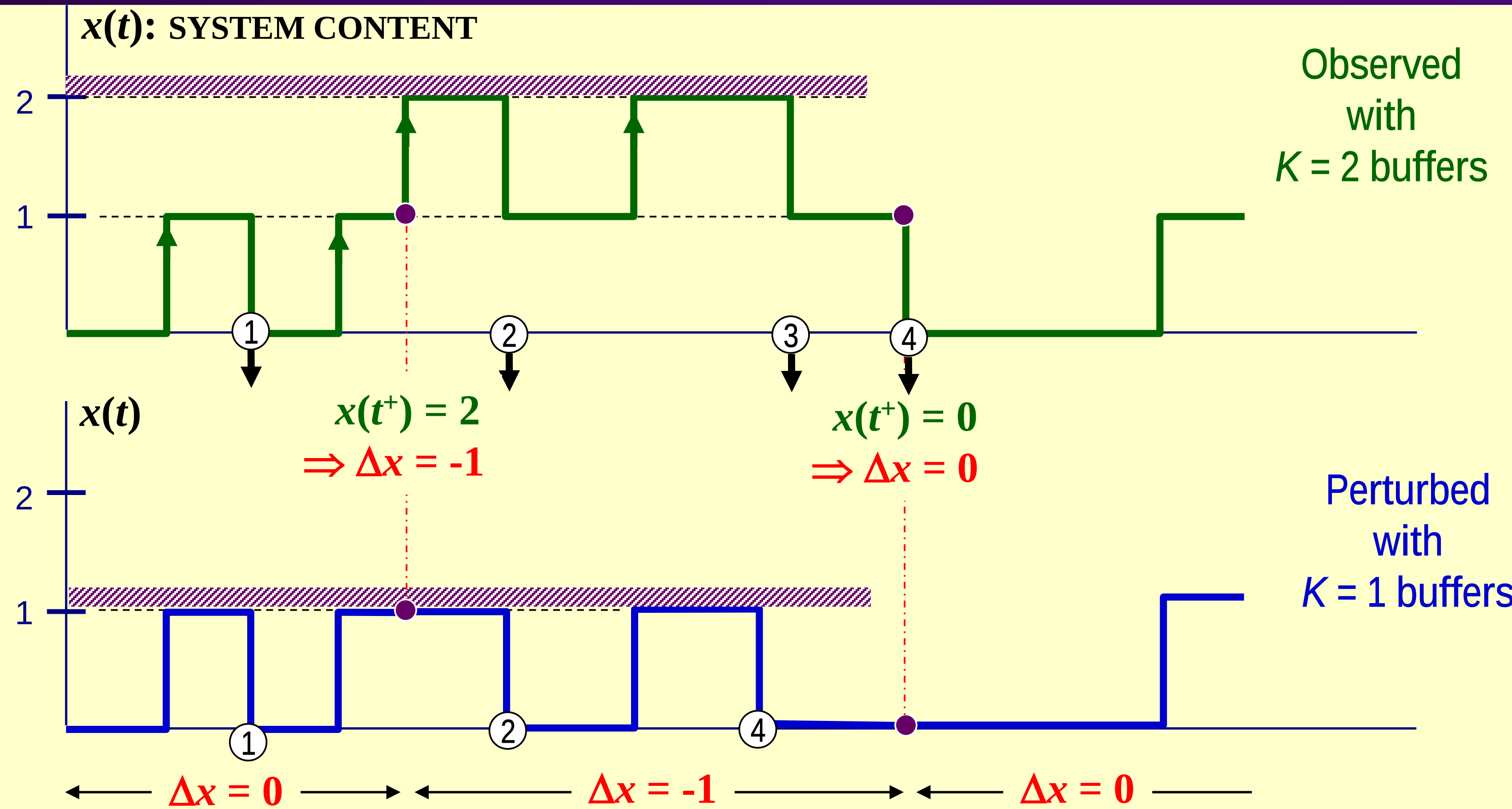
:

Perform… …s
under all wh… …tions

ANSWERS TO MULTIPLE "WHAT IF" Q… …TIONS
AUTOMATICALLY PROVIDED FROM A SIN… TRIAL

LEARNING THROUGH *PERTURBATION ANALYSIS*





[THOUGHT EXPERIMENT]

EVENT-DRIVEN SYSTEMS

The mathematical tools we need to play these "WHAT-IF" games are not given by the usual differential equations and calculus…

These EVENT-DRIVEN SYSTEMS require a new set of models and methodologies

DIFFERENTIAL EQUATIONS v MAX-PLUS CALCULUS

TIME-DRIVEN SYSTEMS

[time: independent variable]

$$\frac{dx}{dt} = f(x,u,t)$$

$$x(t+1) = Ax(t) + Bu(t)$$

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2}\right) - mu^4$$

EVENT-DRIVEN SYSTEMS

[time: state variable]

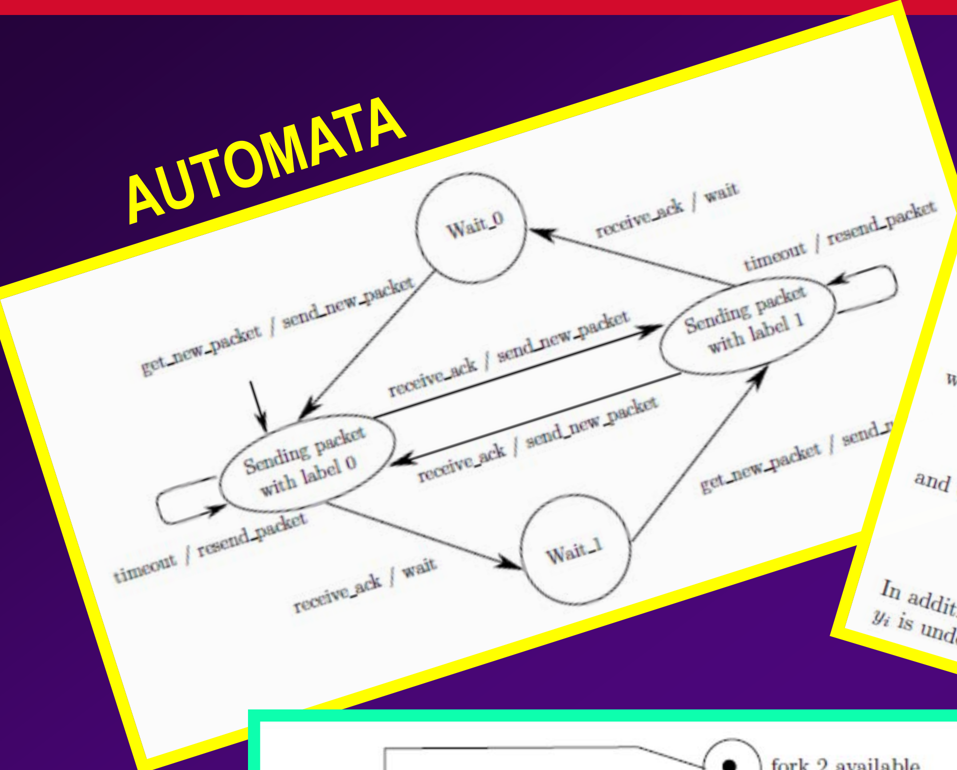
$$x(k+1) = \max_{e \in E_k}\{x(k) + u_e(k)\}$$

ALL CONDITIONS NEEDED TO TRIGGER $(k+1)$th EVENT TIME

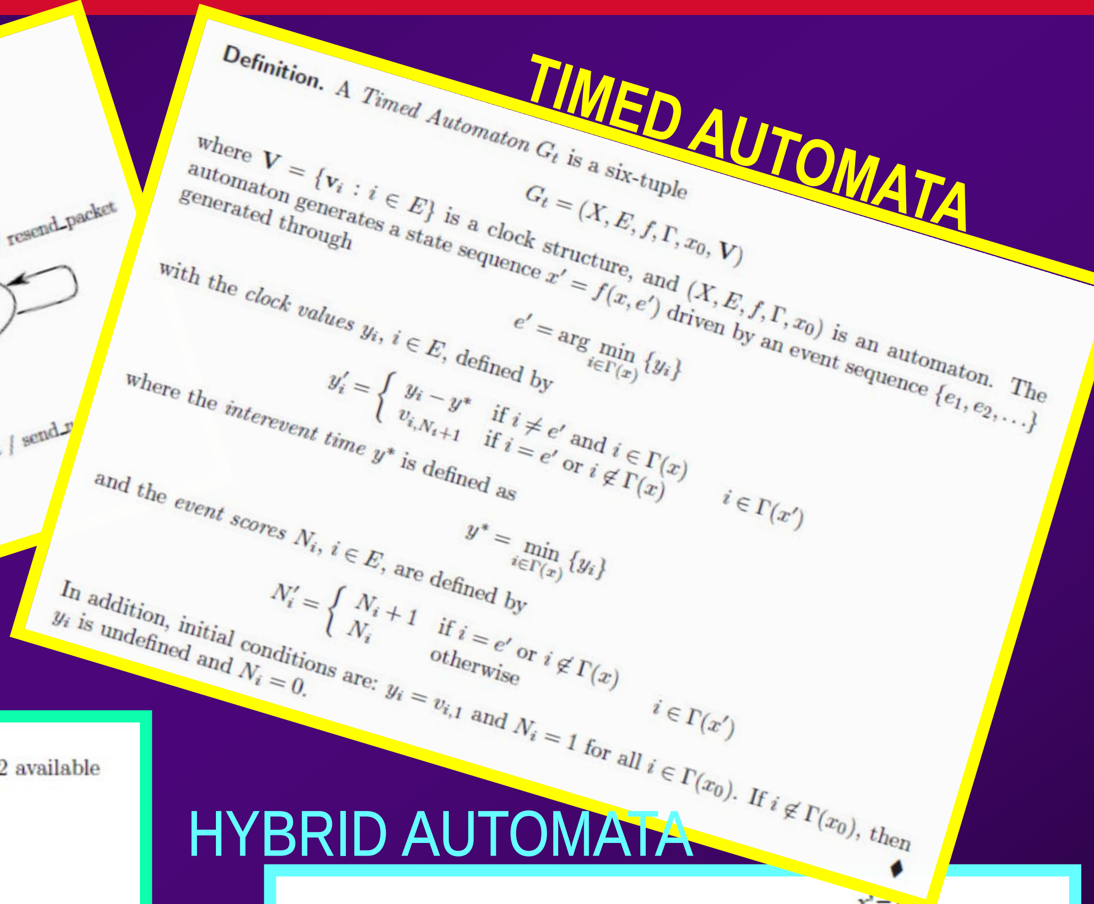
TIME MOVES FORWARD

DISCRETE EVENT DYNAMIC SYSTEMS (DEDS)

AUTOMATA



TIMED AUTOMATA



Definition. A *Timed Automaton* G_t is a six-tuple

$$G_t = (X, E, f, \Gamma, x_0, \mathbf{V})$$

where $\mathbf{V} = \{\mathbf{v}_i : i \in E\}$ is a clock structure, and (X, E, f, Γ, x_0) is an automaton. The automaton generates a state sequence $x' = f(x, e')$ driven by an event sequence $\{e_1, e_2, \ldots\}$ generated through

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

with the *clock values* y_i, $i \in E$, defined by

$$y_i' = \begin{cases} y_i - y^* & \text{if } i \neq e' \text{ and } i \in \Gamma(x) \\ v_{i,N_i+1} & \text{if } i = e' \text{ or } i \notin \Gamma(x) \end{cases} \qquad i \in \Gamma(x')$$

where the *interevent time* y^* is defined as

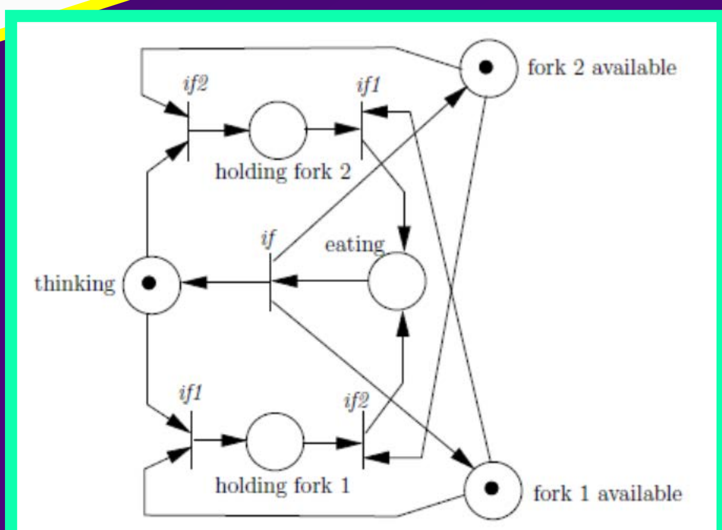
$$y^* = \min_{i \in \Gamma(x)} \{y_i\}$$

and the *event scores* N_i, $i \in E$, are defined by

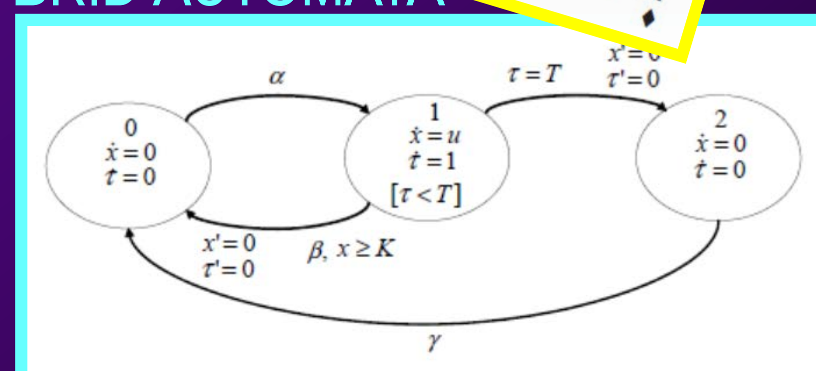
$$N_i' = \begin{cases} N_i + 1 & \text{if } i = e' \text{ or } i \notin \Gamma(x) \\ N_i & \text{otherwise} \end{cases} \qquad i \in \Gamma(x')$$

In addition, initial conditions are: $y_i = v_{i,1}$ and $N_i = 1$ for all $i \in \Gamma(x_0)$. If $i \notin \Gamma(x_0)$, then y_i is undefined and $N_i = 0$. ♦

PETRI NETS

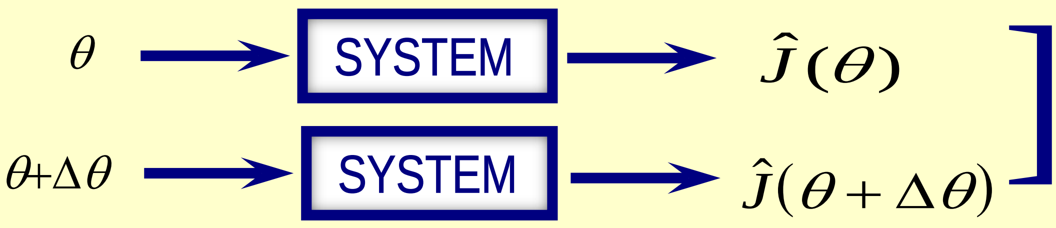


HYBRID AUTOMATA



DERIVATIVE ESTIMATION: INFINITESIMAL PERTURBATION ANALYSIS (IPA)

"Brute Force" Derivative Estimation:

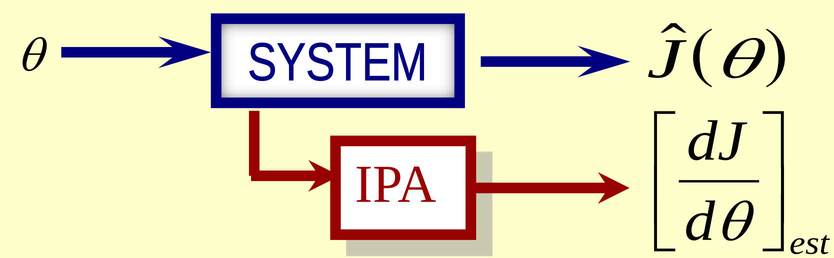


$$\Rightarrow \left[\frac{dJ}{d\theta}\right]_{est} = \frac{\hat{J}(\theta+\Delta\theta)-\hat{J}(\theta)}{\Delta\theta}$$

DRAWBACKS:

- **Intrusive:** actively introduce perturbation $\Delta\theta$
- **Computational cost:** 2 observation processes [$(N+1)$ for N-dim θ]
- **Inherently inaccurate:** $\Delta\theta$ large $\Rightarrow$ poor derivative approx. $\Delta\theta$ small $\Rightarrow$ numerical instability

Infinitesimal Perturbation Analysis (IPA):



SAMPLE BIBLIOGRAPHY

Ho, Y.C., M.A. Eyler, and D.T. Chien, "A Gradient Technique for General Buffer Storage Design in a Serial Production Line," *Intl. Journal of Production Research*, Vol. 17, pp. 557-580, 1979.

Ho, Y.C., and C.G. Cassandras, "A New Approach to the Analysis of Discrete Event Dynamic Systems," *Automatica*, Vol. 19, No. 2, pp. 149-167, 1983.

Ho, Y.C., X. Cao, and C.G. Cassandras, "Infinitesimal and Finite Perturbation Analysis for Queueing Networks," *Automatica*, Vol. 19, pp. 439-445, 1983.

Cao, X., "Convergence of Parameter Sensitivity Estimates in a Stochastic Experiment," *IEEE Transactions on Automatic Control*, Vol. 30, pp. 834-843, 1985.

Cassandras, C.G., Wardi, Y., Panayotou, C.G., and Yao, C., "Perturbation Analysis and Optimization of Stochastic Hybrid Systems", *European Journal of Control*, Vol. 16, No. 6, pp. 642-664, 2010

Ramadge, P.J., and W.M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 81--98,1989.

David, R., and H. Alla, Petri Nets & Grafcet: Tools for Modelling Discrete Event Systems, Prentice-Hall, New York, 1992.

Moody, J.O., and P.J. Antsaklis, Supervisory Control of Discrete Event Systems Using Petri Nets, Kluwer Academic Publishers, 1998.

SELECTED BOOKS

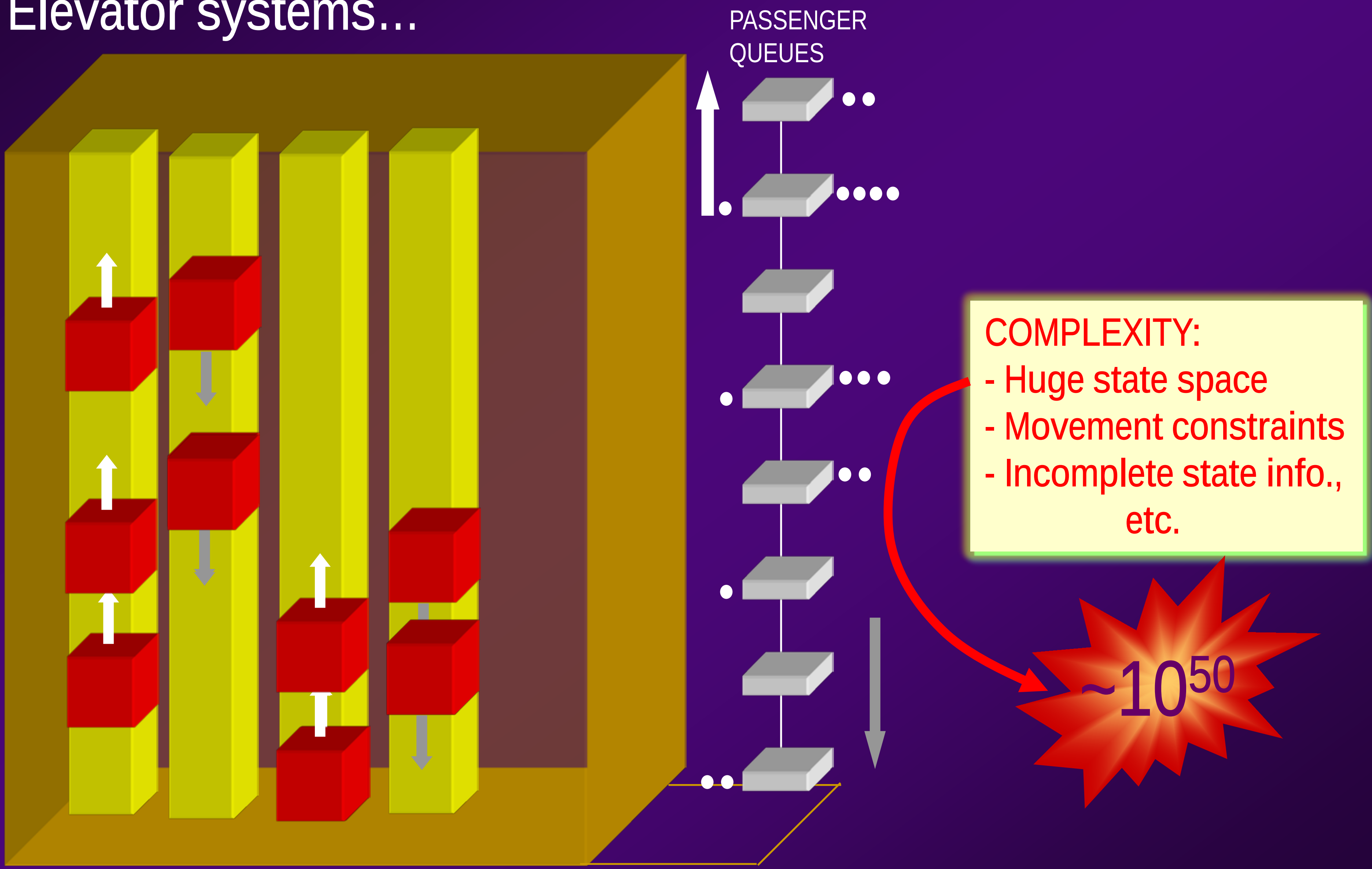
- Ho, Y.C., and X. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic Publisher, 1991.
- Glasserman, P., *Gradient Estimation via Perturbation Analysis*, Kluwer Academic Publishers, Boston, 1991
- Cassandras, C.G., *Discrete Event Systems: Modeling and Performance Analysis*, Irwin Publ. 1993.
- Cao, X., *Realization Probabilities: The Dynamics of Queueing Systems*, Springer-Verlag, 1994.
- Glasserman, P., and D.D. Yao, *Monotone Structure in Discrete Event Systems*, Wiley, 1994.
- Fu, M.C., and J.Q. Hu, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, Kluwer Academic Publ., 1997.
- Cao, X., *Stochastic Learning and Optimization – A Sensitivity-Based Approach*, Springer, 2007.
- Cassandras, C.G, and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd Edition, Springer, 2008.

AN APPLICATION:

ELEVATOR DISPATCHING

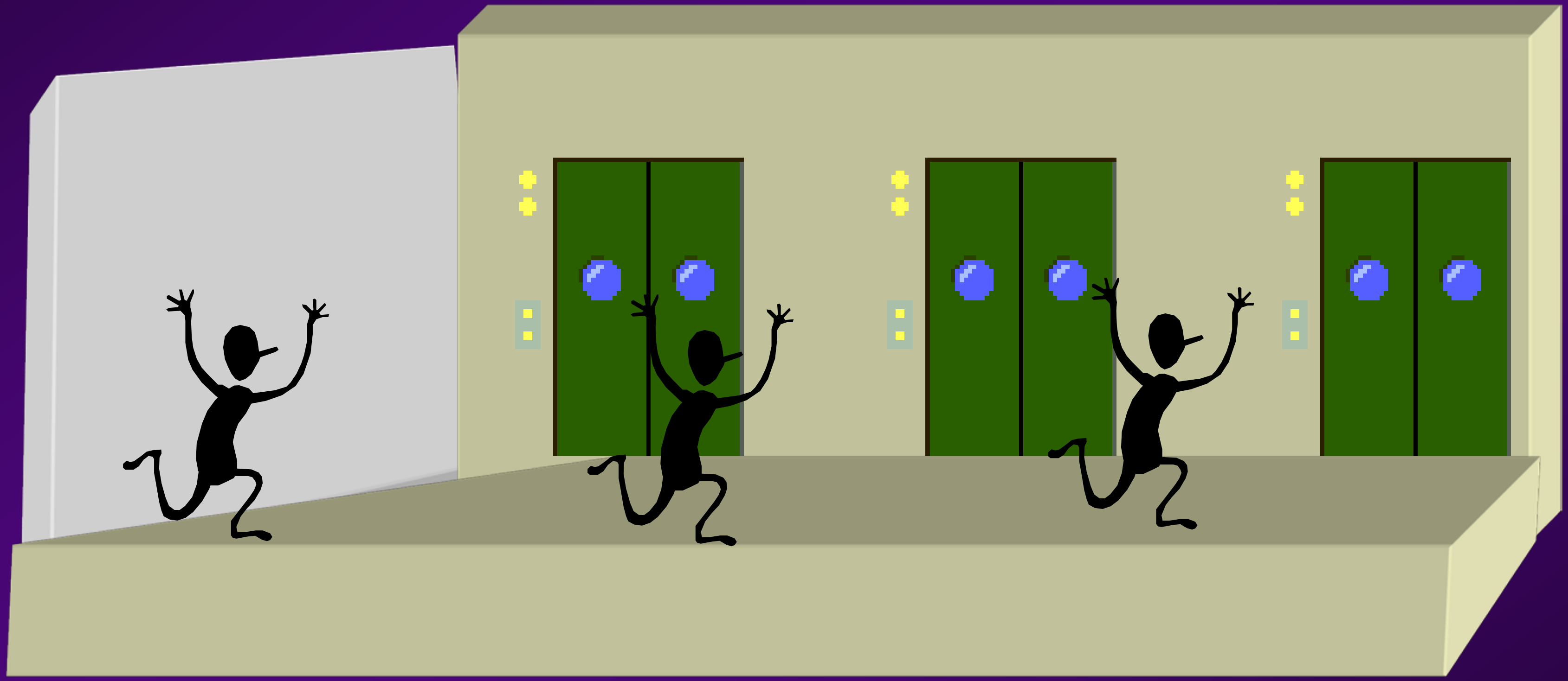
ELEVATOR DISPATCHING

Elevator systems…



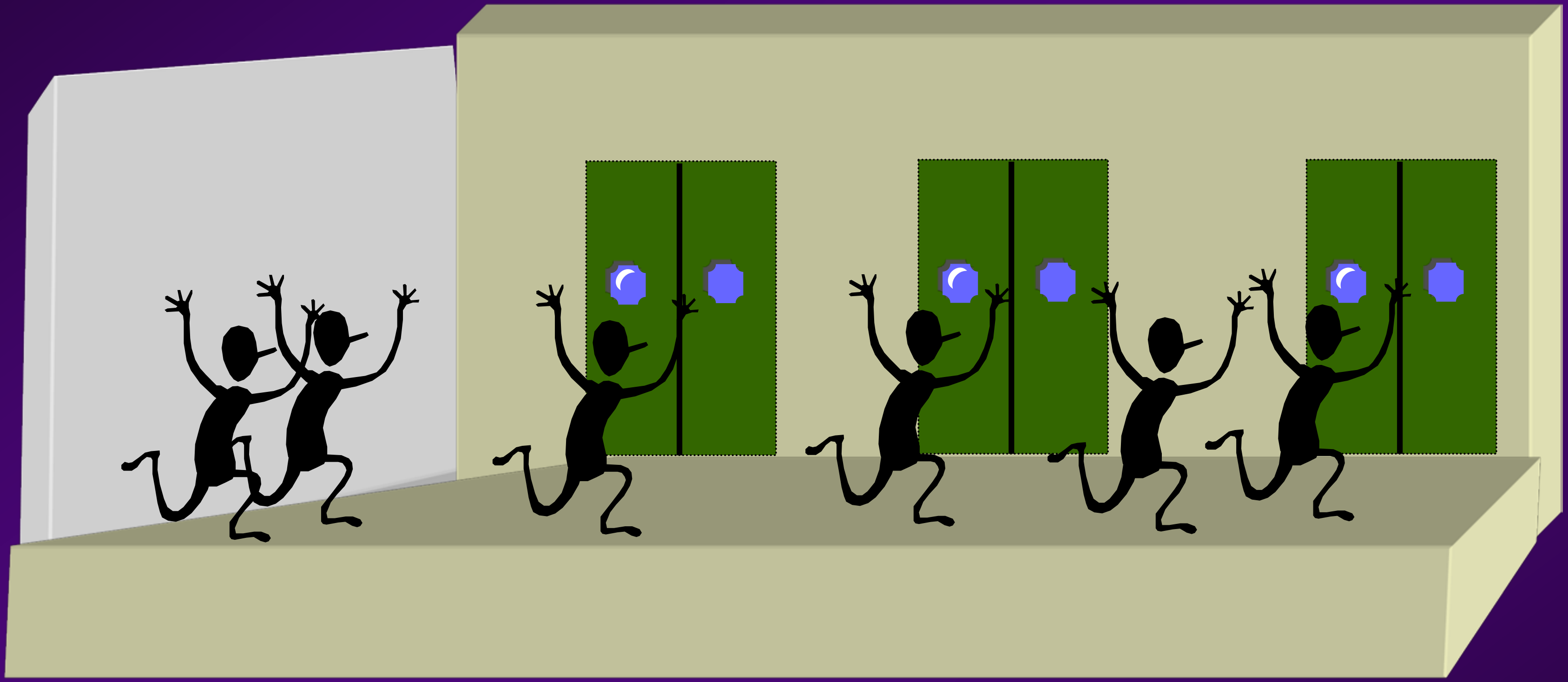
HOW *NOT* TO CONTROL

3 elevators available at lobby…



Each person takes one and goes

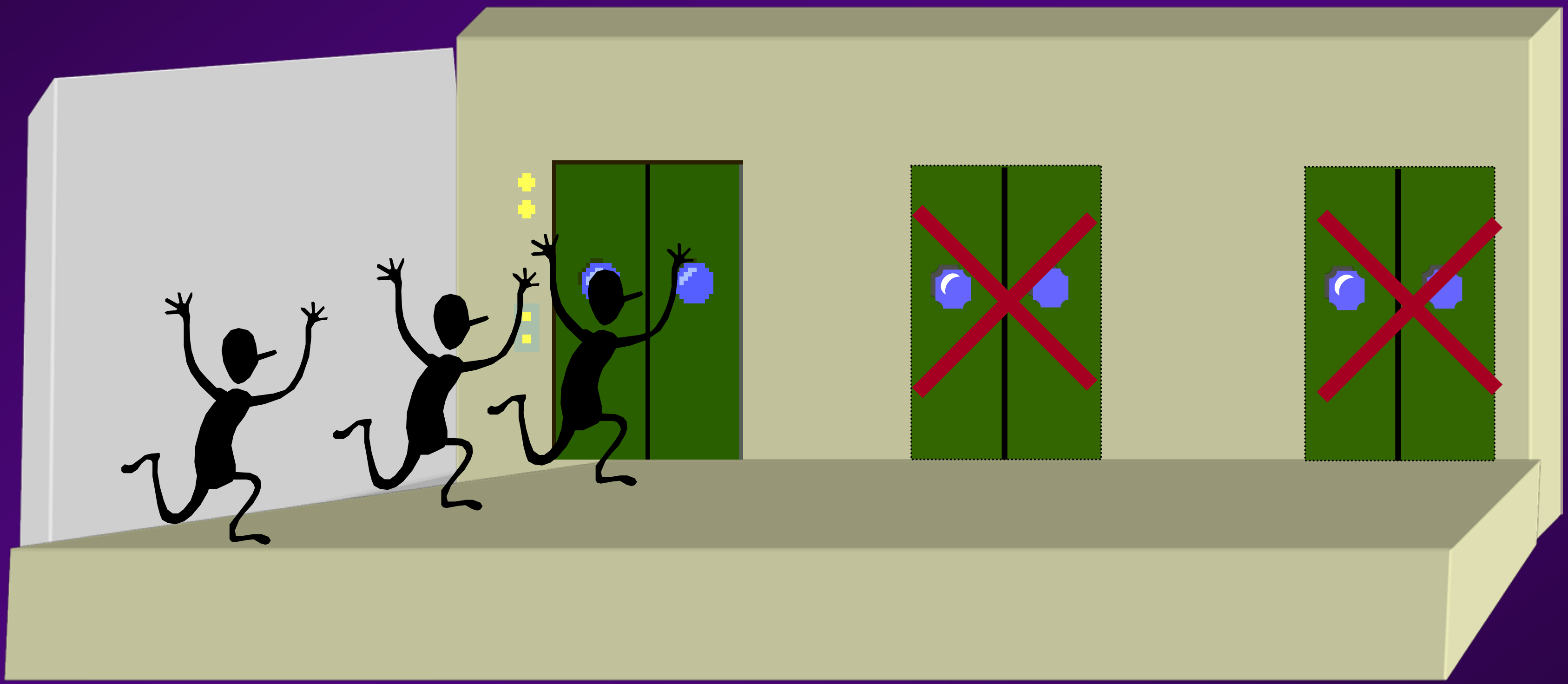
HOW *NOT* TO CONTROL



Long waiting results...

A BETTER WAY TO CONTROL

Force only 1 of the 3 elevators to be available



ELEVATOR DISPATCHING

CONTROLLER (Threshold-based):

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

number of passengers inside car ≥ THRESHOLD

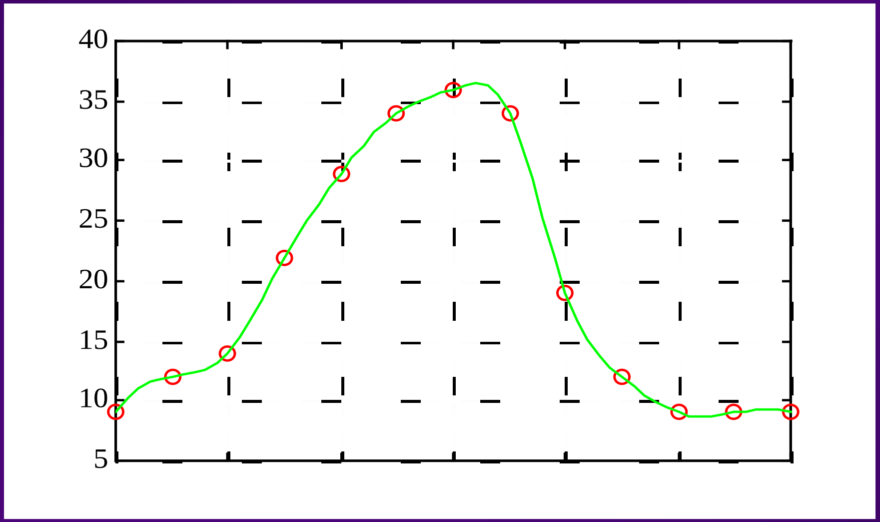
THRESHOLD depends on
- *passenger arrival rate*
- *car service rate*

THIS IS IN FACT OPTIMAL!

Pepyne and Cassandras, *IEEE Trans. on Control Systems Tech.*, 1998.

ELEVATOR DISPATCHING

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



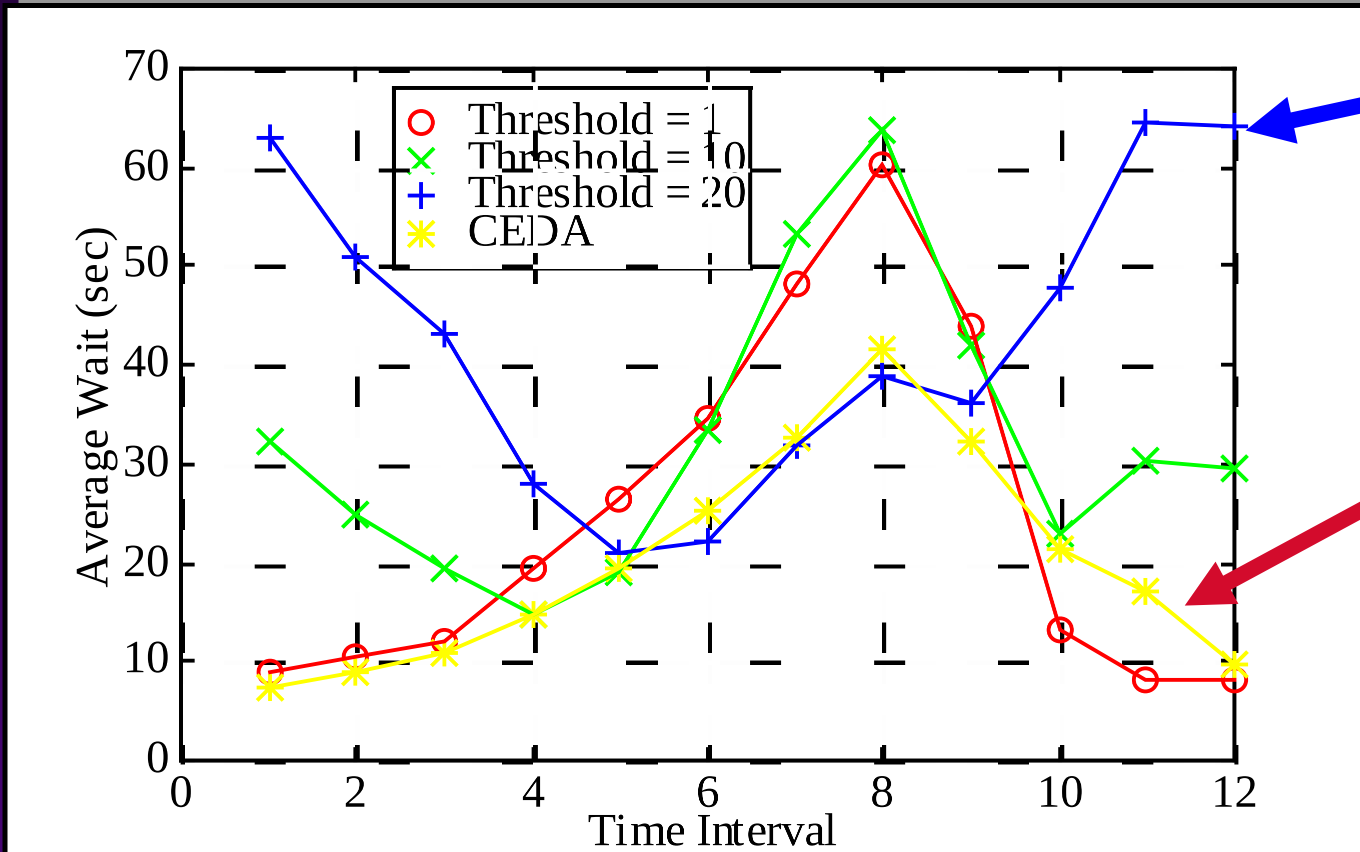
PROBLEM:

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

PERTURBATION ANALYSIS APPROACH:

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

ELEVATOR DISPATCHING



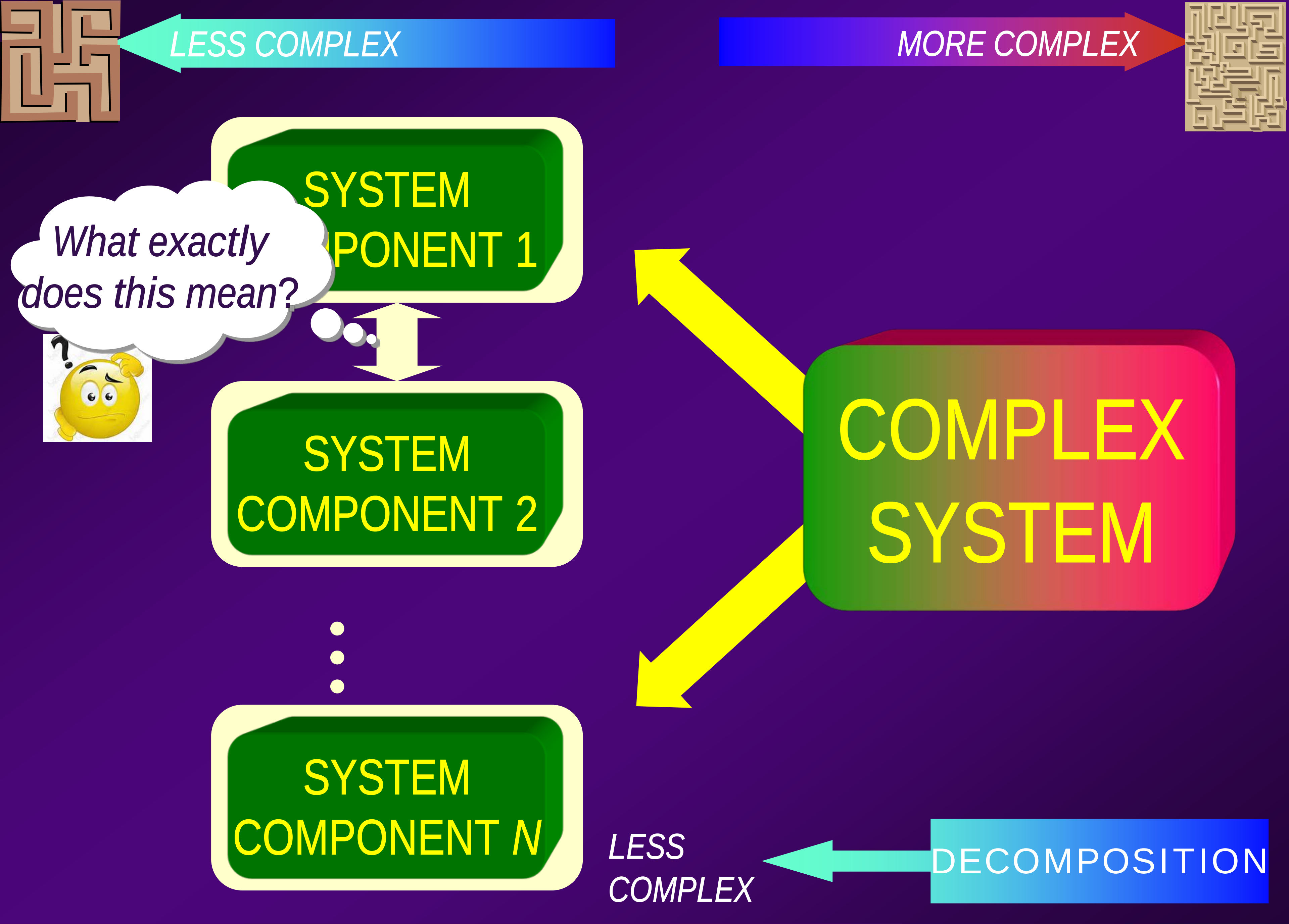
Uncontrolled

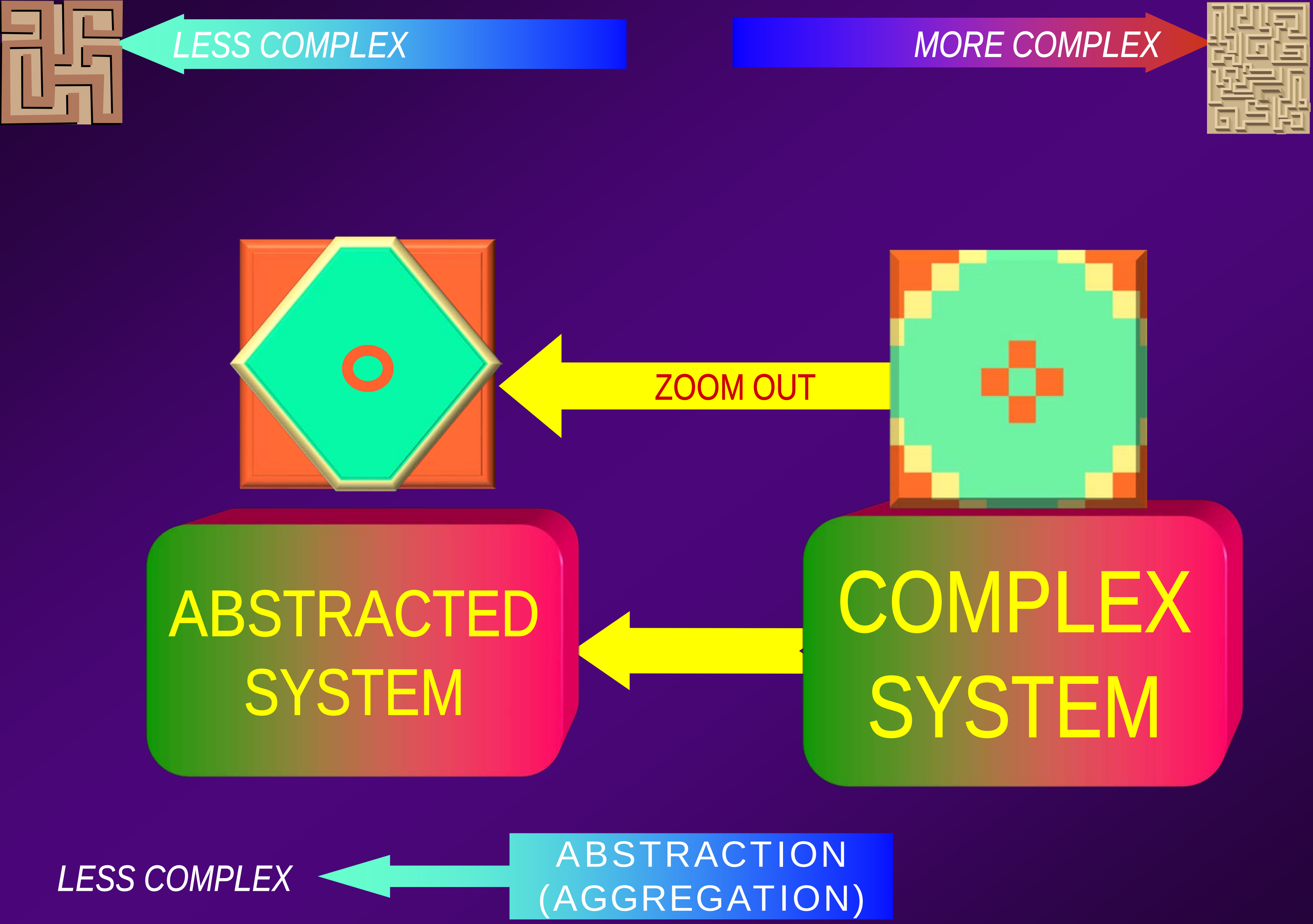
CEDA:
Concurrent
Estimation
Dispatching
Algorithm

How fast did the CEDA *learn* optimal thresholds? ➡ Approximately 5 real days

Artificial Intelligence (AI) methods : over 1 year…

DECOMPOSITION AND ABSTRACTION

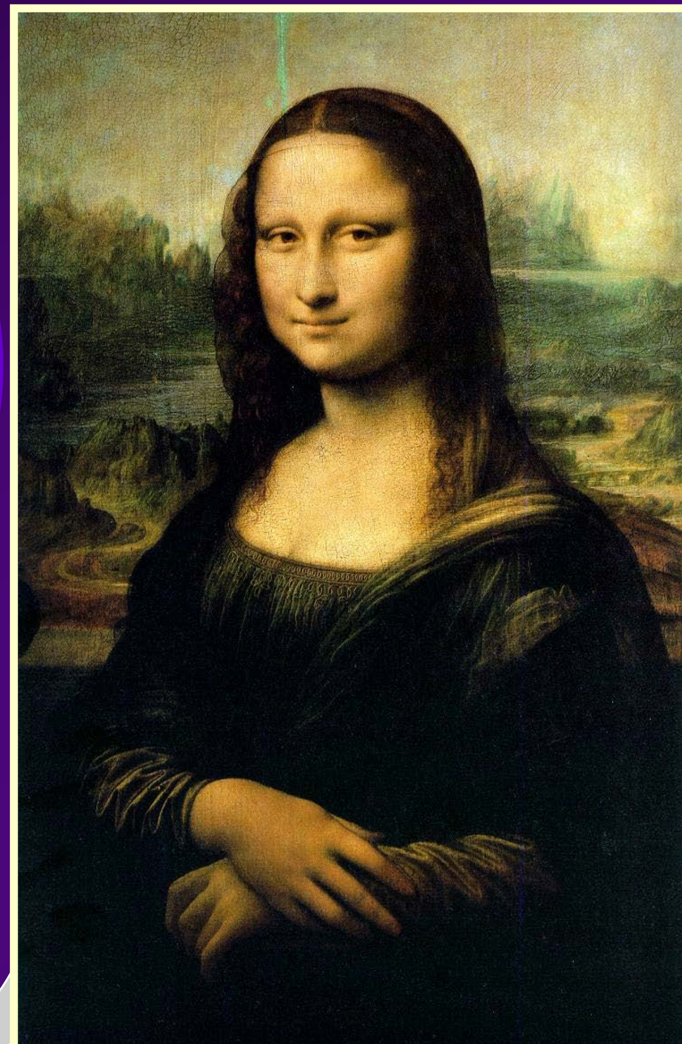

LESS COMPLEX
MORE COMPLEX
SYSTEM
PONENT 1
What exactly does this mean?
SYSTEM COMPONENT 2
SYSTEM COMPONENT N
COMPLEX SYSTEM
LESS COMPLEX
DECOMPOSITION


LESS COMPLEX
MORE COMPLEX
ZOOM OUT
ABSTRACTED SYSTEM
COMPLEX SYSTEM
LESS COMPLEX
ABSTRACTION (AGGREGATION)

WHAT IS THE RIGHT ABSTRACTION LEVEL ?

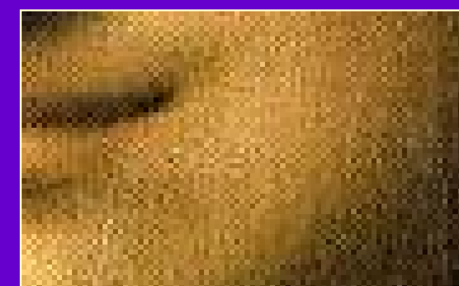
TOO FAR…

model not detailed enough



JUST RIGHT…

good model



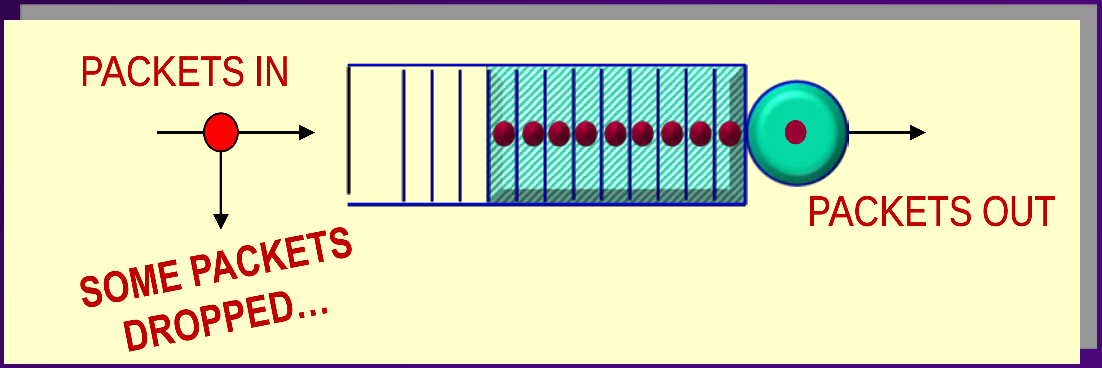
TOO CLOSE…

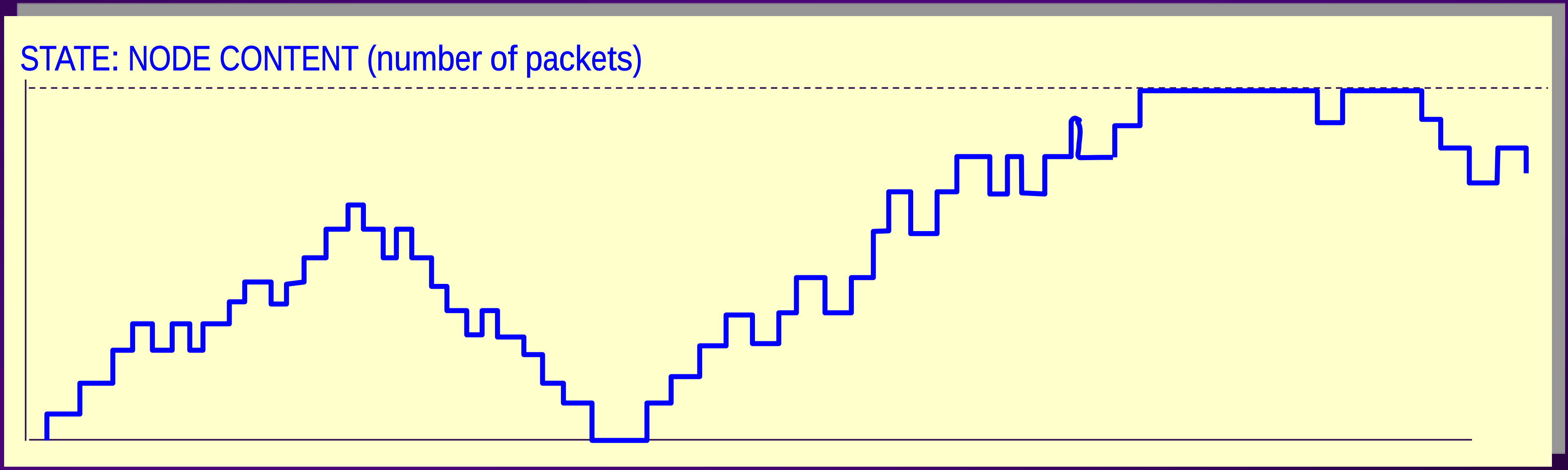
too much undesirable detail

CREDIT: W.B. Gong

ABSTRACTION

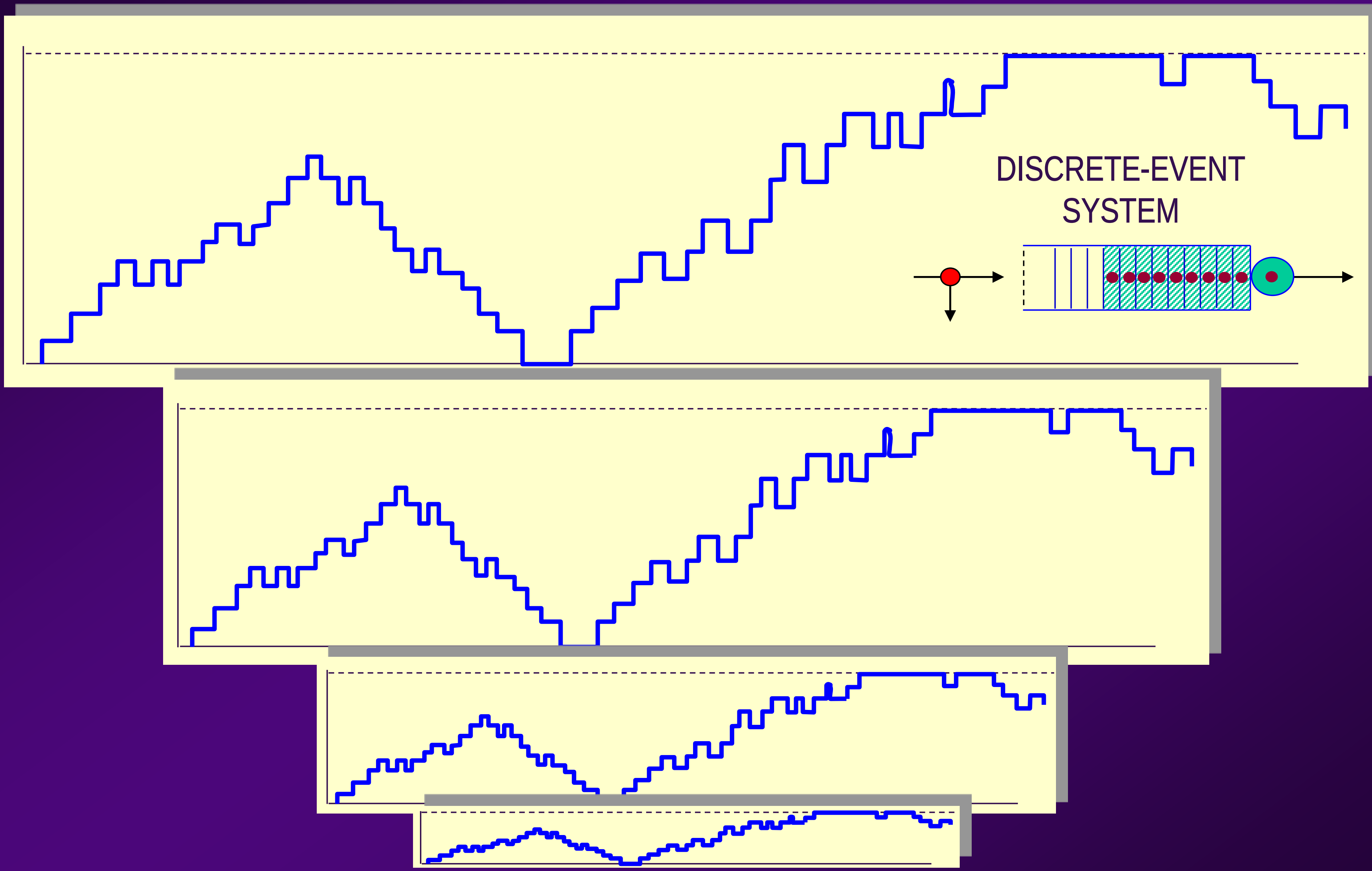
A SECOND IN THE LIFE OF AN INTERNET NODE…



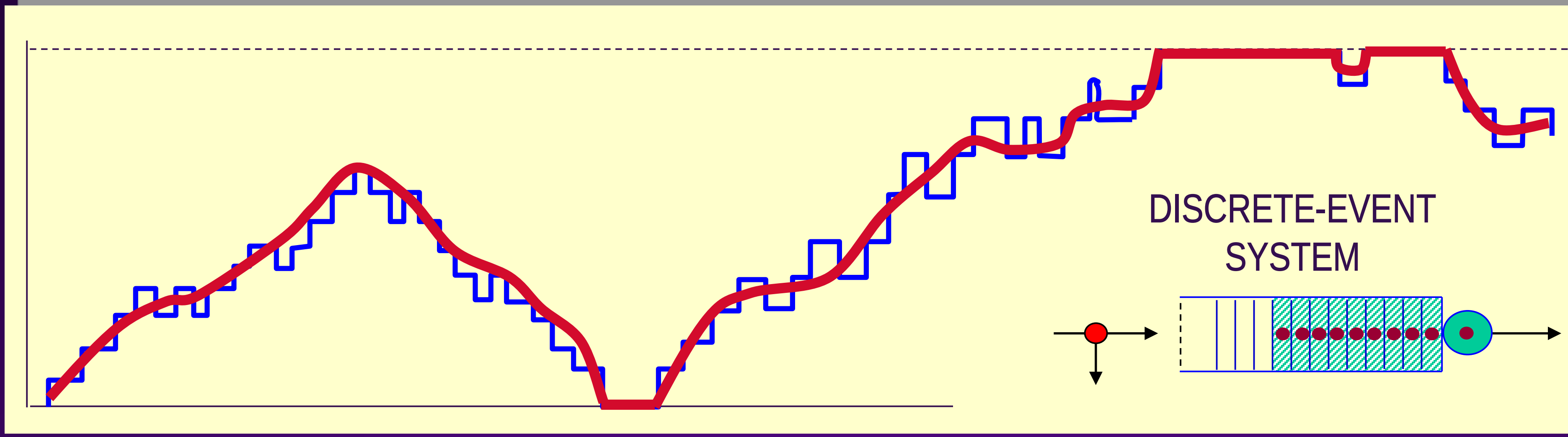


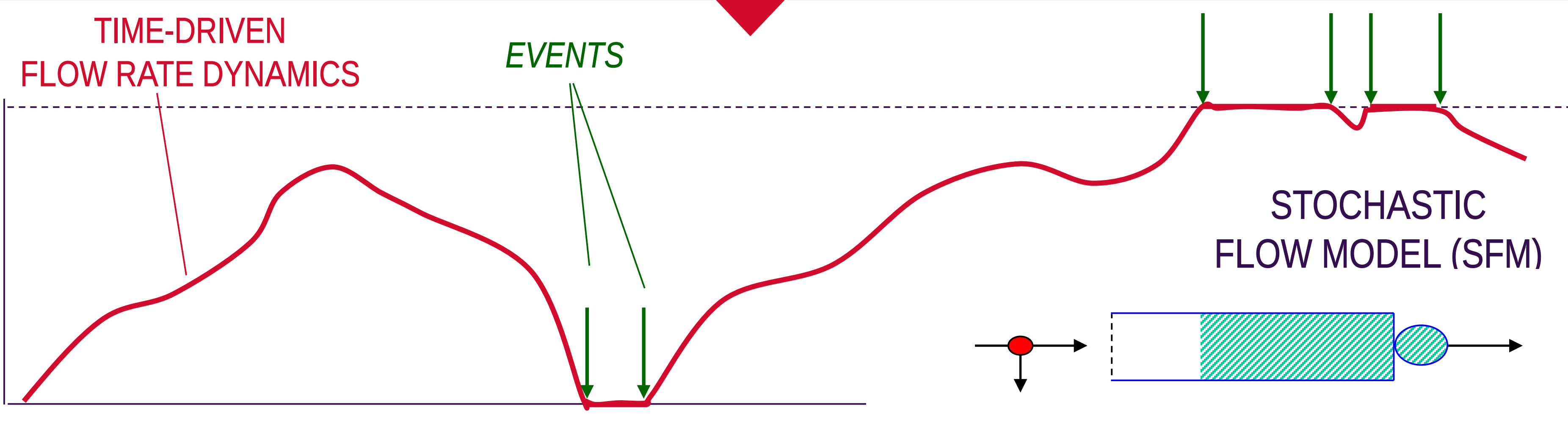
… a pure DISCRETE EVENT SYSTEM

ABSTRACTION OF A DISCRETE-EVENT SYSTEM



ABSTRACTION OF A DISCRETE-EVENT SYSTEM





WHY SFM?

- "Lower resolution" model of "real" system intended to capture *just enough* info. on system dynamics

- Aggregates many events into simple continuous dynamics, preserves only events that cause drastic change

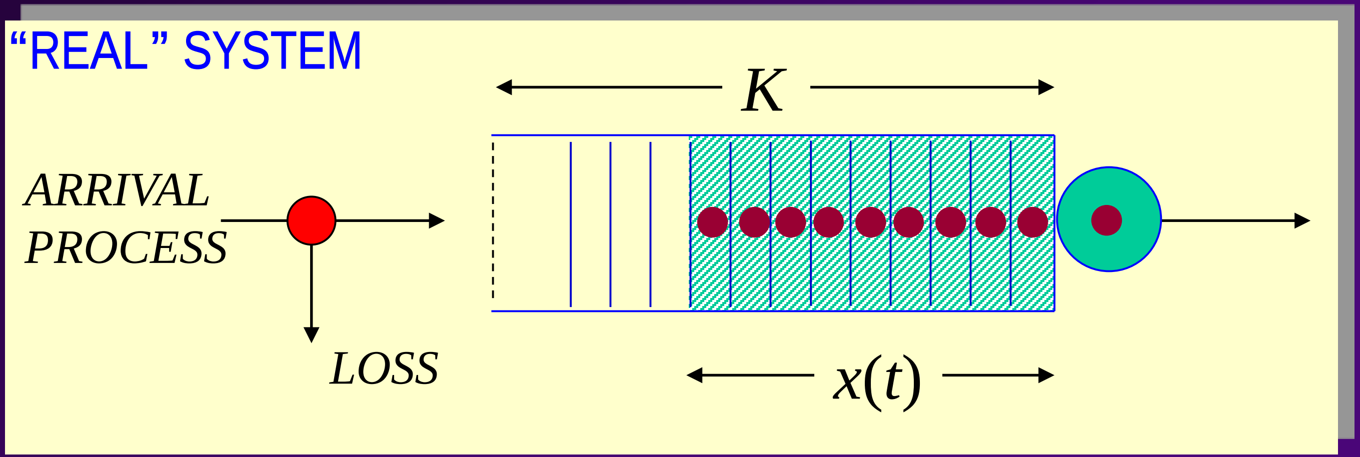
  $\Rightarrow$ computationally efficient

  (e.g., *orders of magnitude faster simulation*)

- If the RIGHT QUESTIONS are asked,

  the loss of detailed information becomes insignificant…

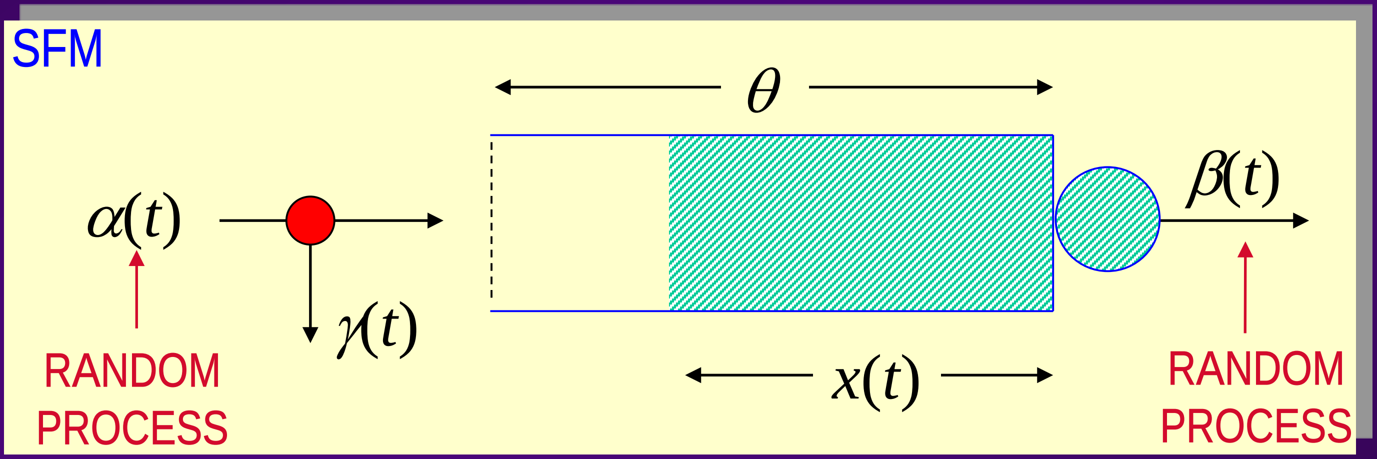
AN OPTIMIZATION PROBLEM



$L(K)$: Loss Rate

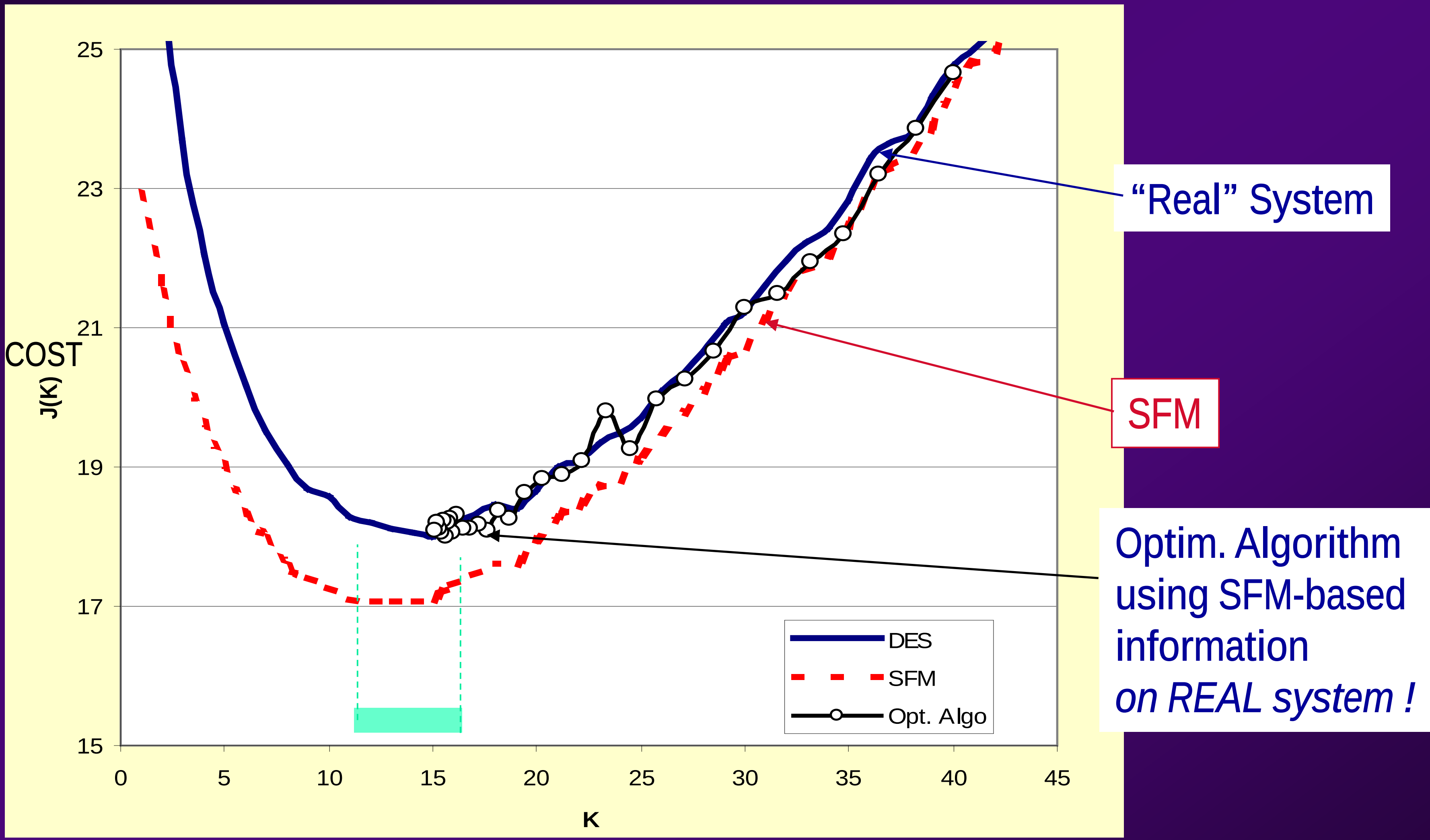
$Q(K)$: Mean Content

PROBLEM: Determine K to minimize $[Q(K) + R \cdot L(K)]$



SURROGATE PROBLEM: Determine θ to minimize $[Q^{SFM}(\theta) + R \cdot L^{SFM}(\theta)]$

AN OPTIMIZATION PROBLEM



Cassandras, Wardi, Melamed, Sun, and Panayiotou, *IEEE Trans. on Automatic Control*, 2002.

THE "RIGHT QUESTION"…

Can the ABSTRACTION model be used to *predict* the real system's behavior?

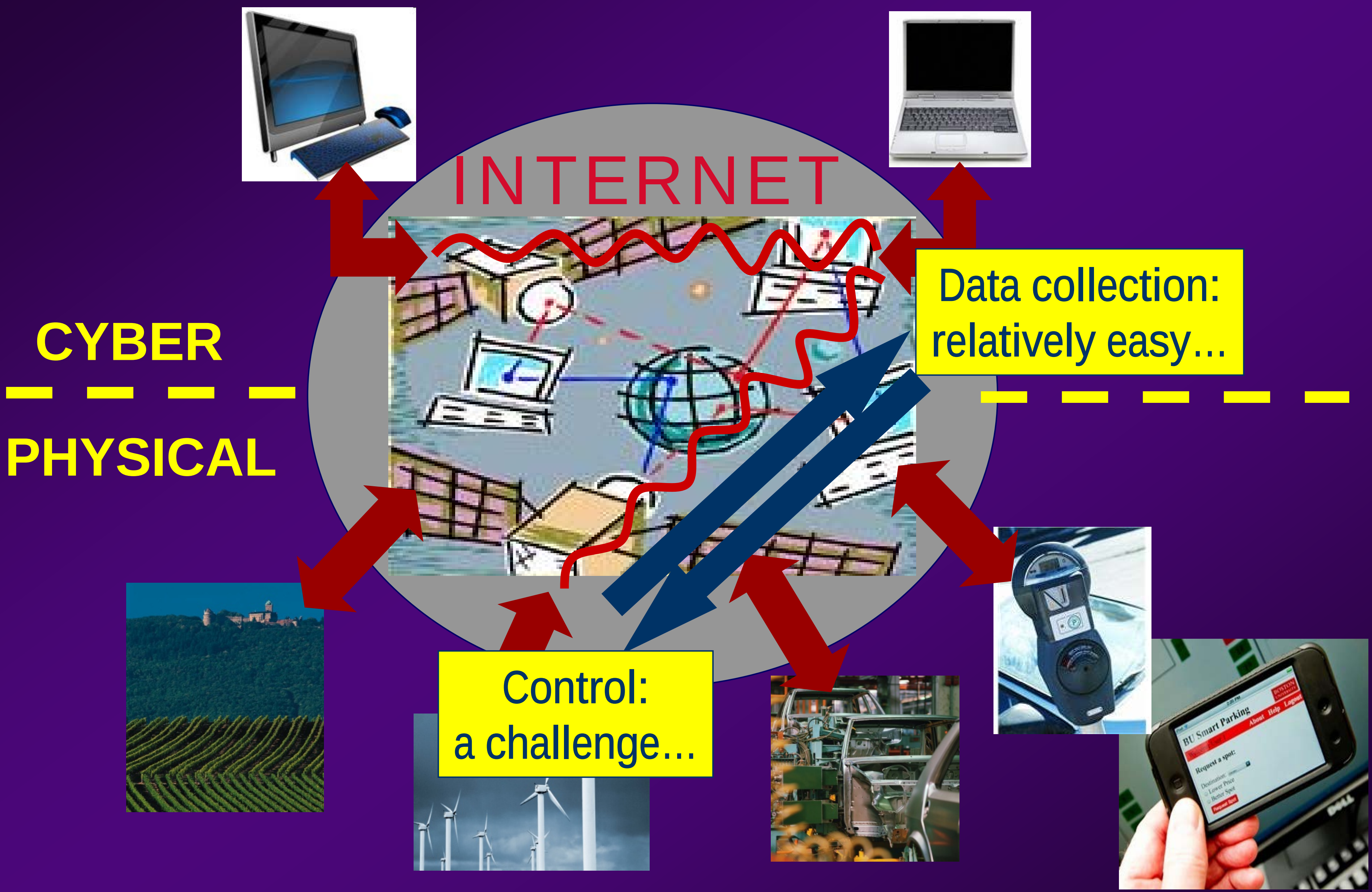
Maybe, but that's too much to hope for.
➡ BAD question…

Can the ABSTRACTION model be used to *control or optimize* the real system's behavior ?

Often yes, and sometimes this can be proved.
➡ GOOD question…

DECOMPOSITION

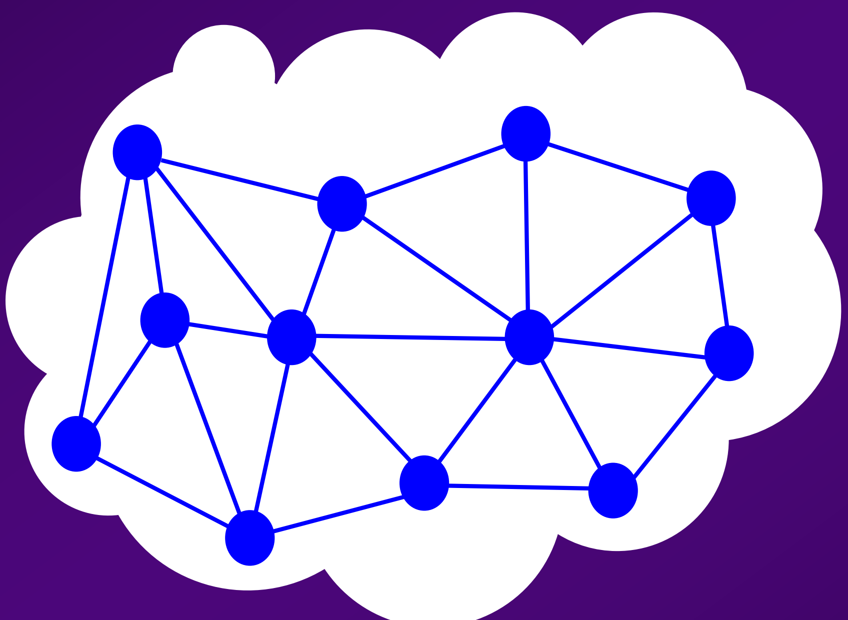
CYBER-PHYSICAL SYSTEMS



DISTRIBUTED COOPERATIVE CONTROL AND OPTIMIZATION

N system components (processors, agents, vehicles, sensors), one common objective:

$$\min_{s_1,\ldots,s_N} H(s_1,\ldots,s_N)$$
$$s.t. \text{ constraints on each } s_i$$



$$\min_{s_1} H(s_1,\ldots,s_N)$$
$$s.t. \text{ constraints on } s_1$$

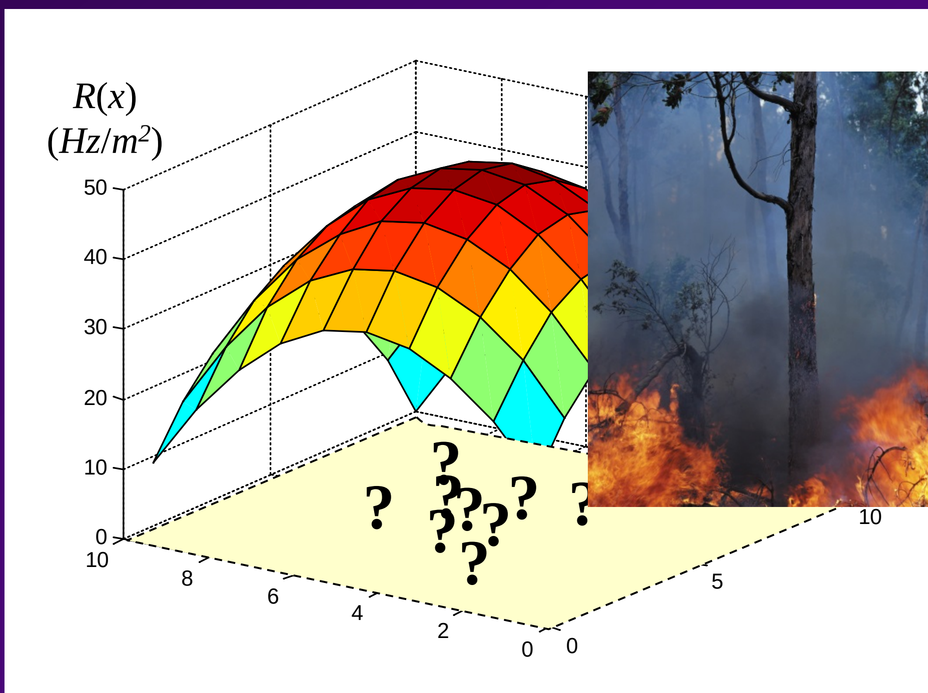
$\vdots$

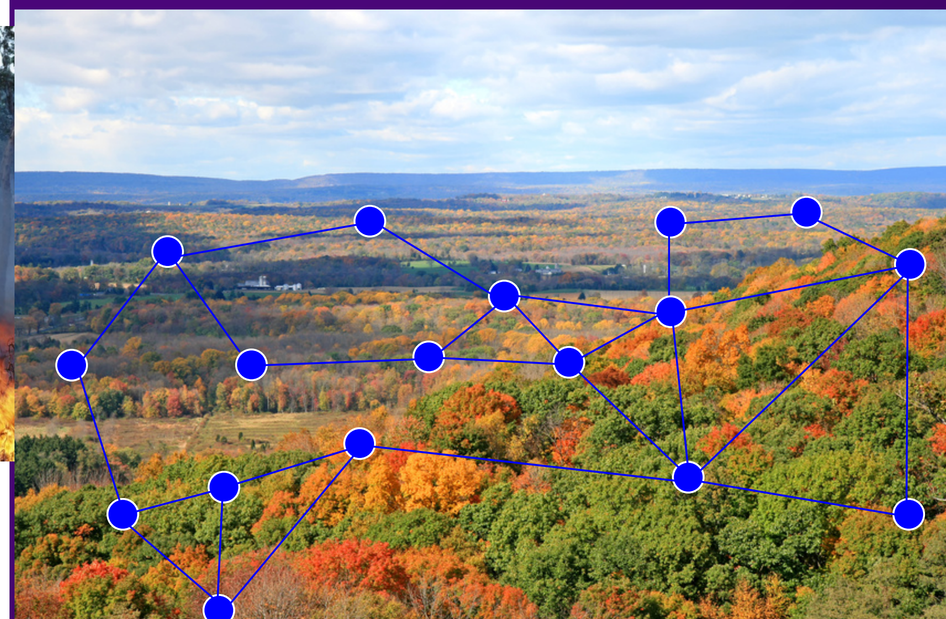
$$\min_{s_N} H(s_1,\ldots,s_N)$$
$$s.t. \text{ constraints on } s_N$$

MOTIVATIONAL PROBLEM: *COVERAGE CONTROL*

Deploy a SENSOR NETWORK to maximize "event" detection probability

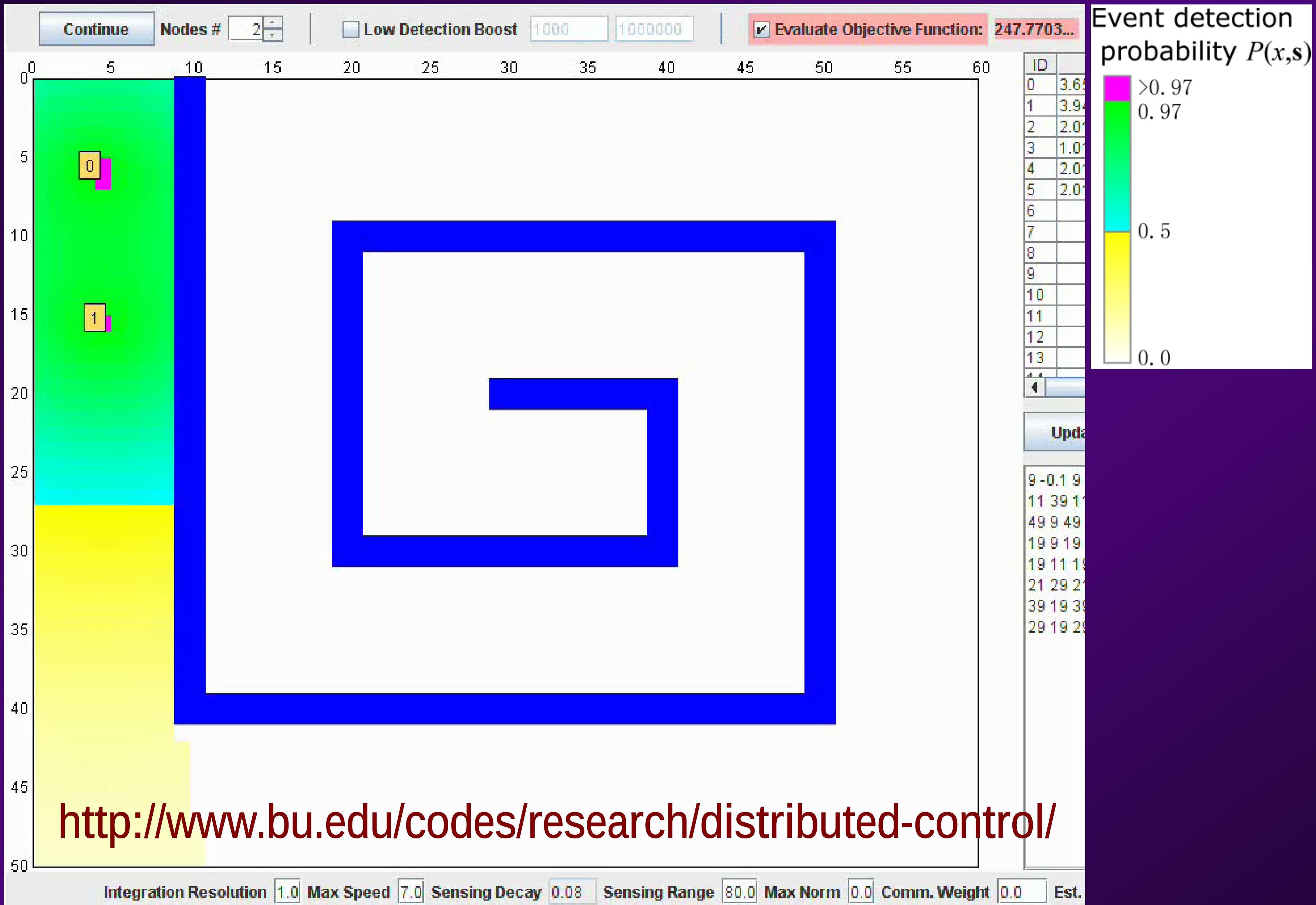
- – unknown event locations
- – event sources may be mobile
- – sensors may be mobile





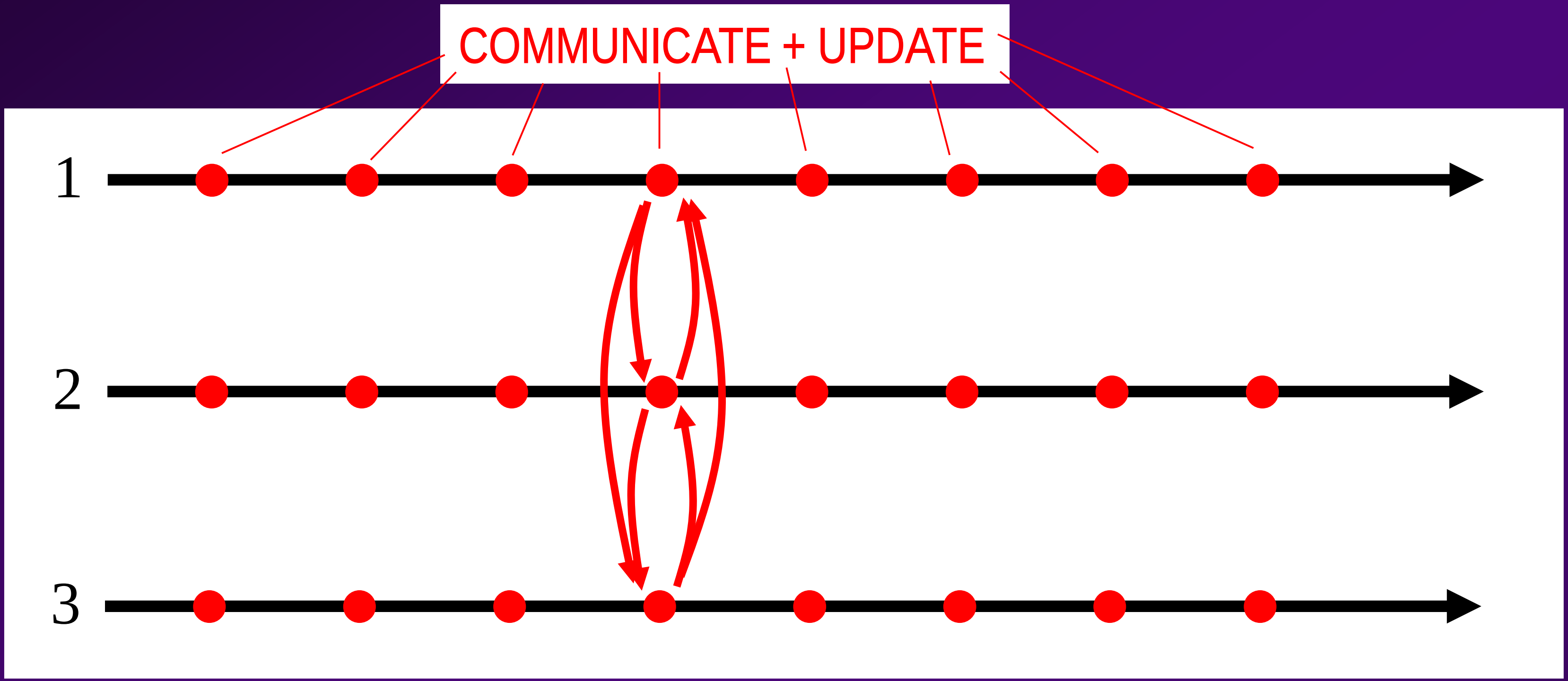
Perceived event density (data sources) over given region (mission space)

OPTIMAL COVERAGE IN A MAZE



http://www.bu.edu/codes/research/distributed-control/

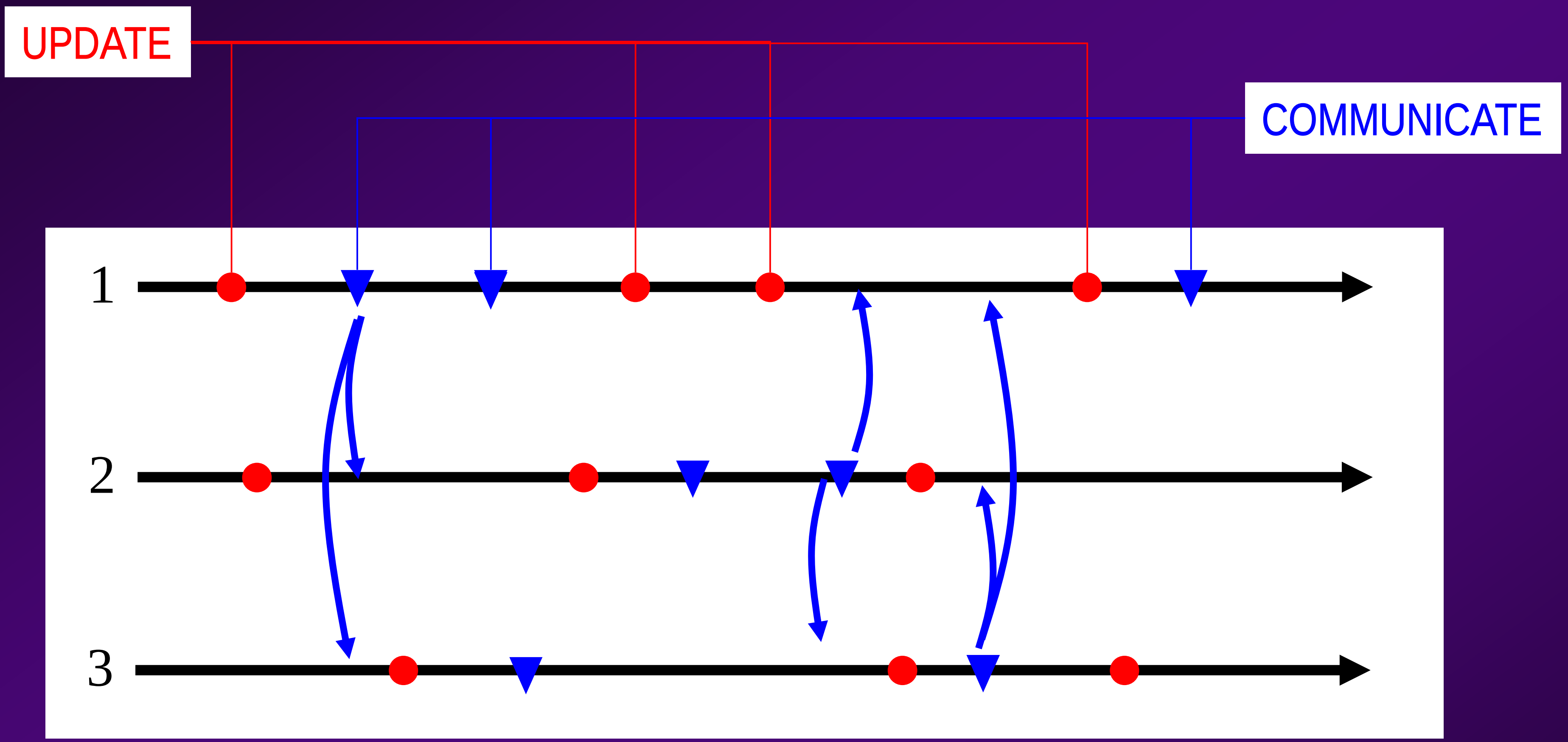
SYNCHRONIZED (TIME-DRIVEN) COOPERATION



Drawbacks:

- Excessive communication (critical in wireless settings!)
- Faster nodes have to wait for slower ones
- Clock synchronization infeasible
- Bandwidth limitations
- Security risks

ASYNCHRONOUS (EVENT-DRIVEN) COOPERATION



- UPDATE at i : locally determined, arbitrary (possibly periodic)
- COMMUNICATE from i : only when absolutely necessary

EVENT-DRIVEN COMMUNICATION

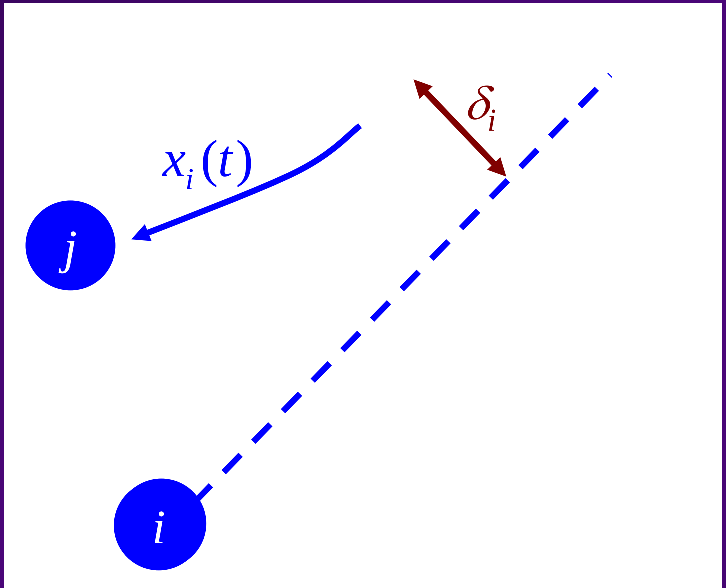
When should a network node communicate with others?

What is the minimum amount of communication required to guarantee a network objective is met?

Communication is expensive, insecure, and kills our precious batteries…

WHEN SHOULD A NODE COMMUNICATE?

Node ***i*** communicates its state to node j only when it detects that its *true state* $x_i(t)$ deviates from ***j'*** *estimate of it* $x_i^j(t)$ so that $g\left(x_i(t), x_i^j(t)\right) \geq \delta_i$ for a given g and δ_i

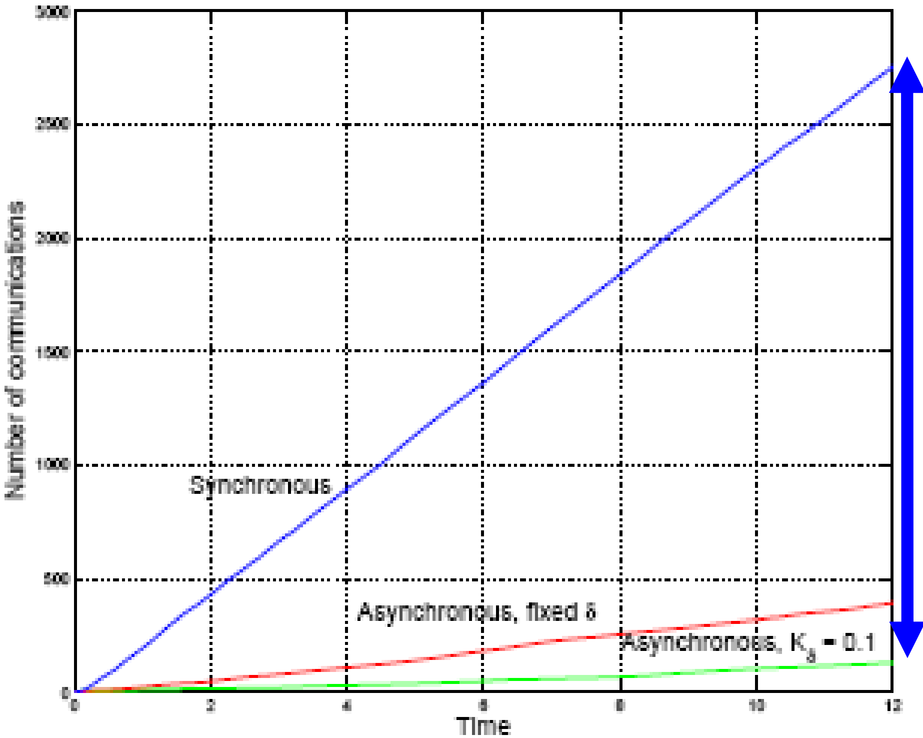


$\Rightarrow$ *Event-Driven* Communication and Control

Theorem formally proving optimality guaranteed under this limited communication scheme (even with delays…)

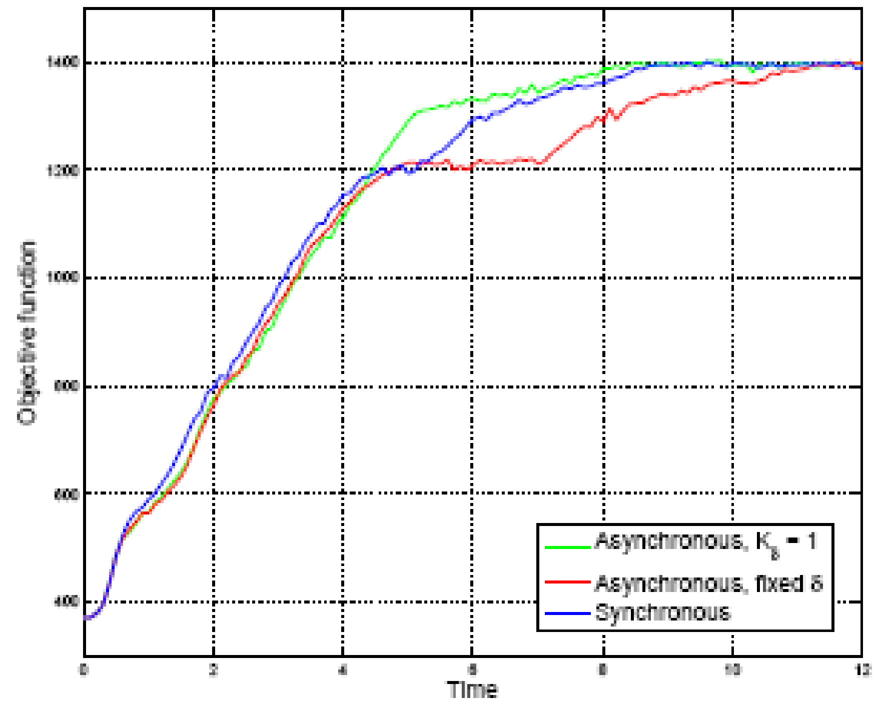
Zhong and Cassandras, *IEEE Trans. on Automatic Control*, 2010

Energy savings + Extended lifetime



SYNCHRONOUS v ASYNCHRONOUS:

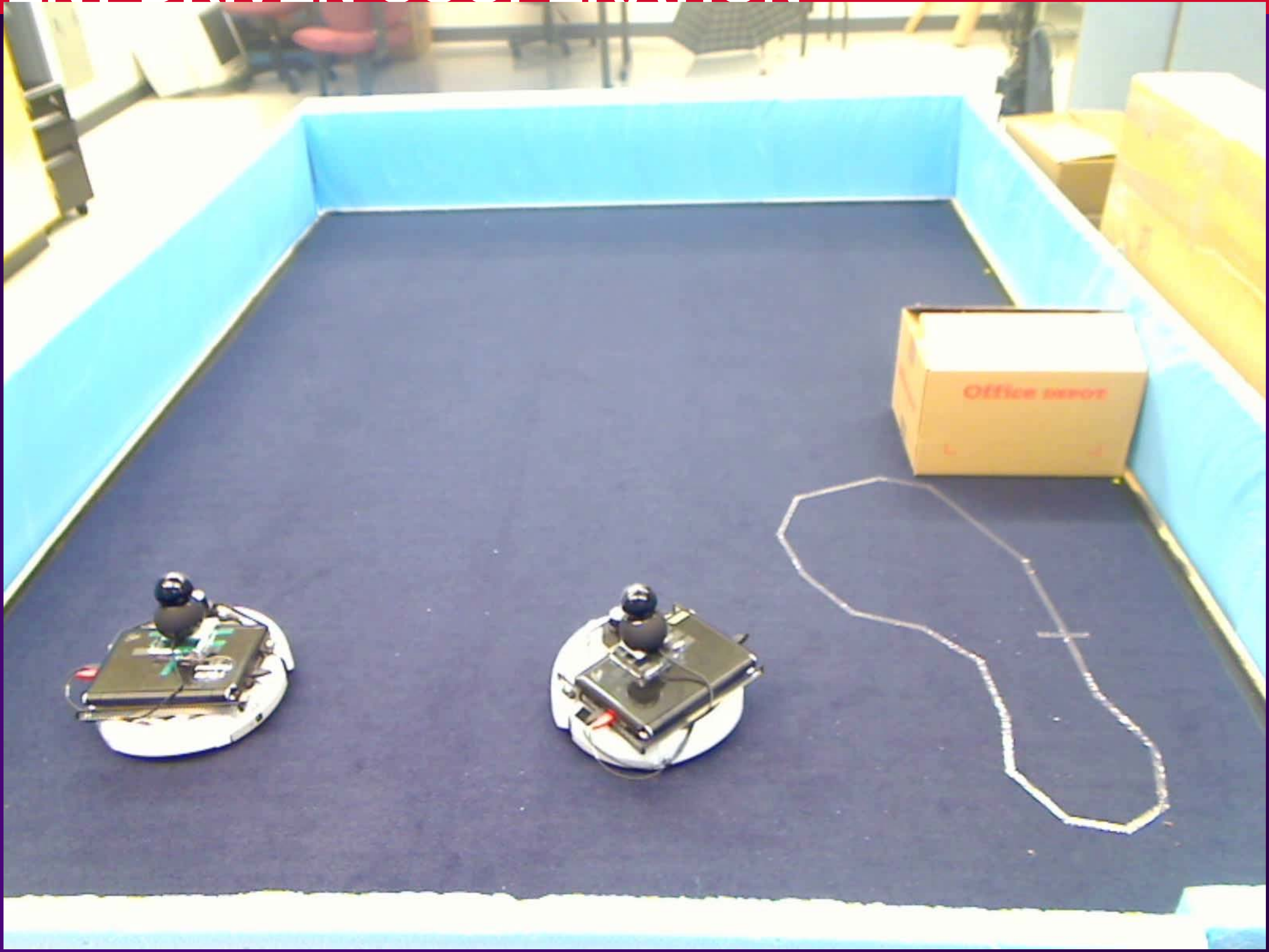
No. of communication events
for a deployment problem *with obstacles*



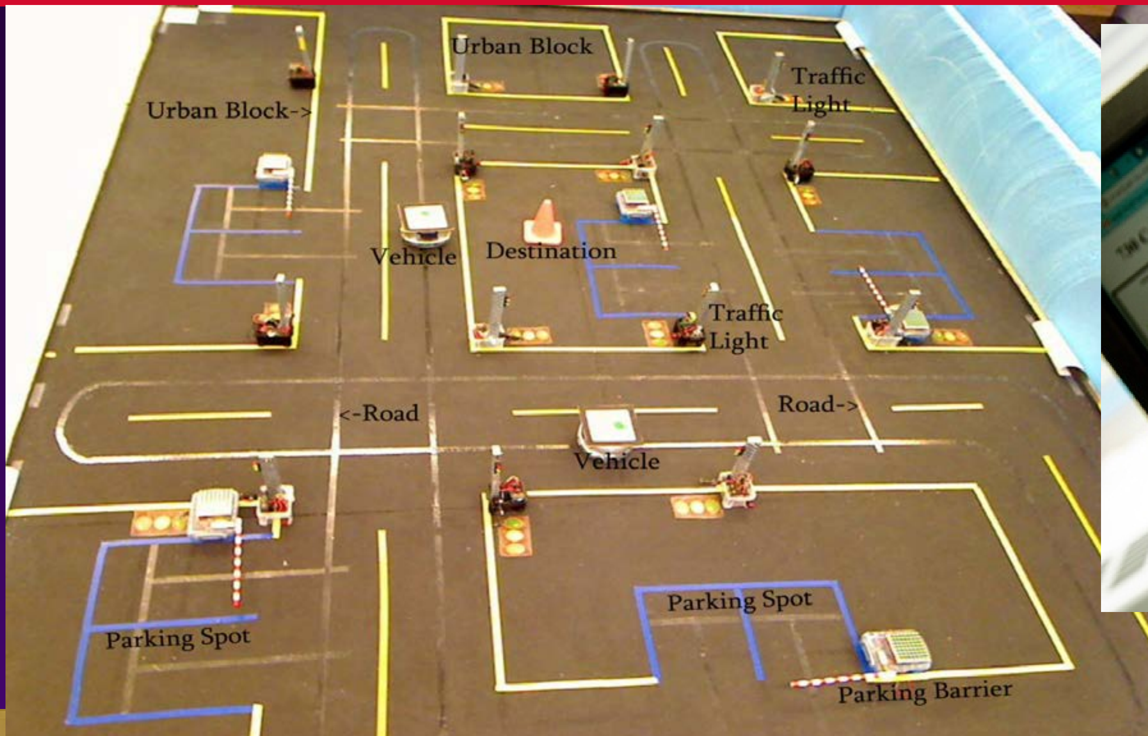
SYNCHRONOUS v ASYNCHRONOUS:

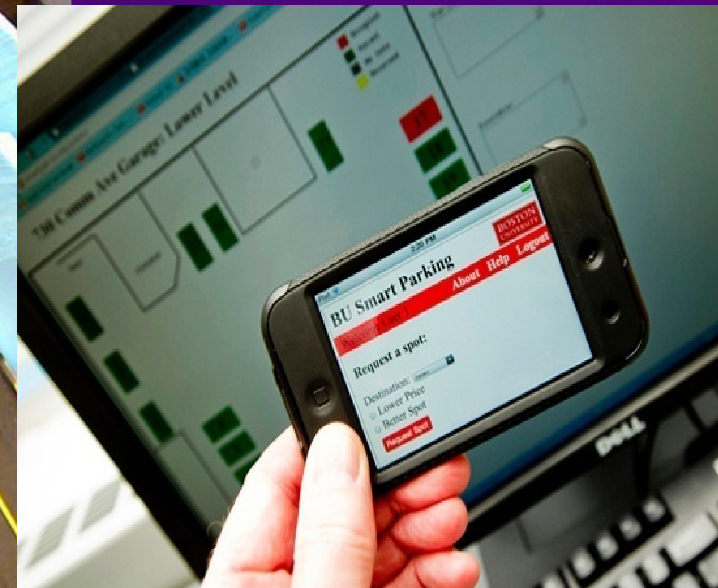
Achieving optimality
in a problem *with obstacles*

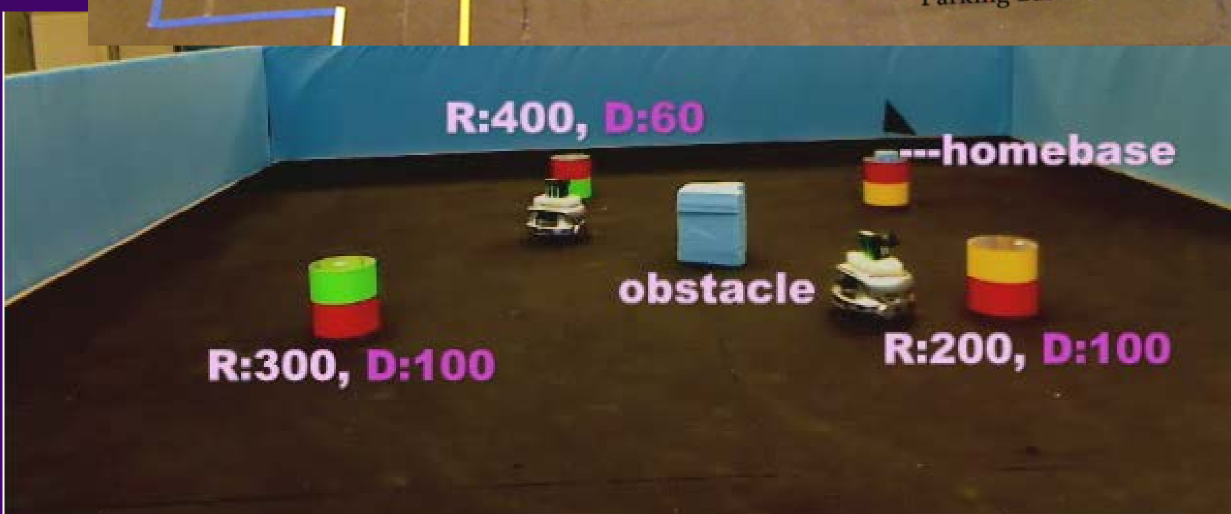
DEMO: COVERAGE + EVENT DETECTION WITH EVENT-DRIVEN COOPERATION



CODES LAB TEST BEDS



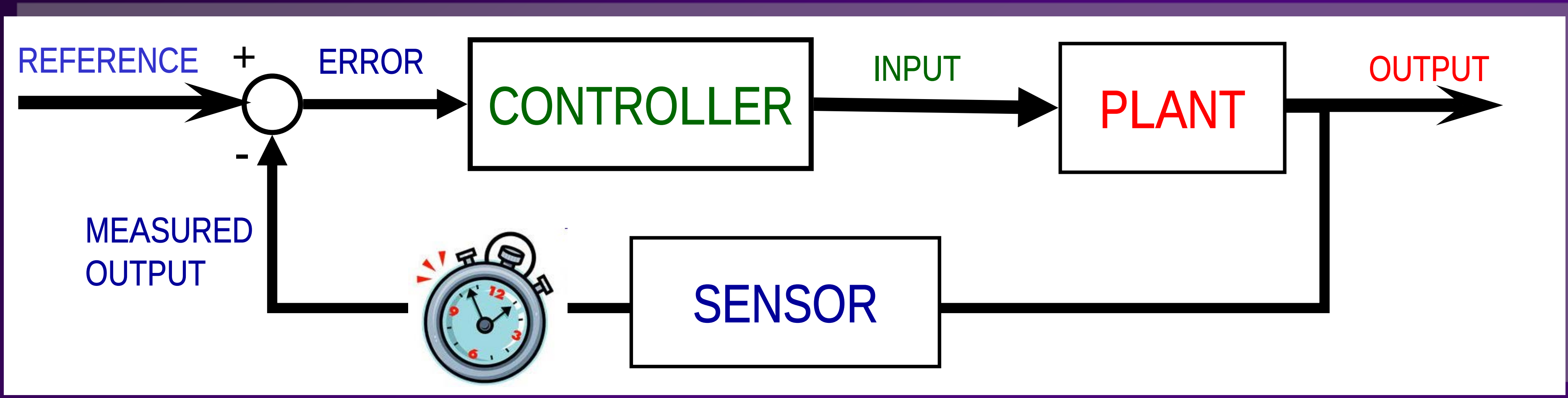




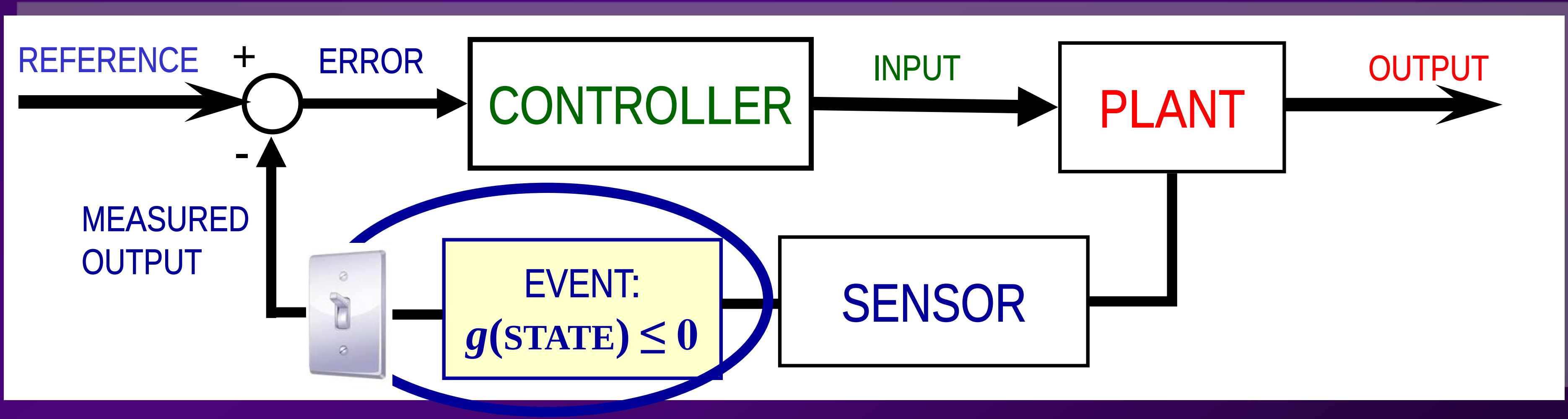

WHAT LIES AHEAD…

"SMART CITIES" AS CYBER-PHYSICAL SYSTEMS

TIME-DRIVEN v EVENT-DRIVEN CONTROL



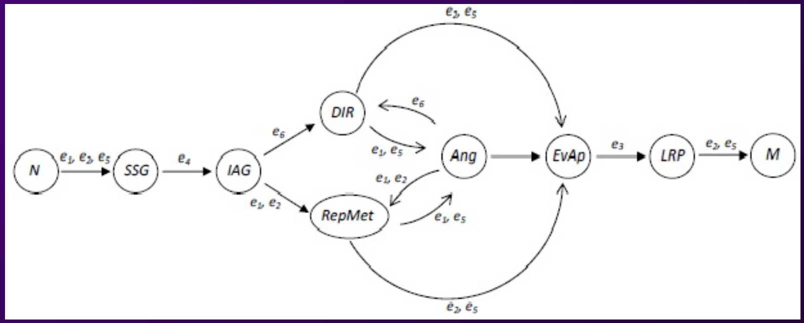
EVENT-DRIVEN CONTROL: Act *only when needed* (or on TIMEOUT) - not based on a clock



COMPLEXITY MADE SIMPLE *AT A SMALL PRICE*

- ...in SMART CITIES ... Smart Parking, Traffic Light Control
  Street Bump
- ...in SENSOR NETWORKS ... abstracting battery models for optimal power management
- ...in MULTI-AGENT SYSTEMS ... UAVs, Robotics
- ...in CANCER TREATMENT ?????

Cancer as a "disease of stages"
i.e., a Discrete Event System!



ACKNOWLEDGEMENTS

Thanks to former (27) and current PhD students whose contributions are reflected in this talk...

UMass/Amherst (1988-1999):
S. Strickland, J.B Suk, J. Lewis, M. Kallmes,
B. Mohanty, P. Sparaggis, J. Pan, G. Bao, A. Gandhi,
V. Julka, D. Pepyne, C. Panayiotou

BU (2000-2013):
K. Gokbayrak, G. Sun, H. Yu, W. Li, X. Wu, L. Miao,
K. Wesselowski, S. Zhuang, J. Mao, N. Xu, M. Zhong,
Y. Chen, A. Kebarighotbi, Y. Geng, T. Wang

X. Lin, Y. Khazaeni, X. Sun, S. Pourazarm,
J. Lima Fleck

+ Colleagues and friends...

Y.C. (Larry) Ho

Wei-Bo Gong
Liyi Dai
Xiren Cao
Pirooz Vakili

Yorai Wardi
Felisa Vazquez-Abad
Stephane Lafortune
John Lygeros
Robert Gao
Q.C. Zhao

Yannis Paschalidis
David Castanon
Michael Caramanis
John Baillieul
Calin Belta
Azer Bestavros
Janusz Konrad

+ Sponsors... ***NSF, AFOSR, ONR, ARO, AFRL, NRL, DARPA, NASA, DOE, EPRI, UT, ALCOA, NOKIA, GE, RTZ, Honeywell, MathWorks***