EVENT-DRIVEN CONTROL AND OPTIMIZATION: WHERE LESS IS OFTEN MORE…

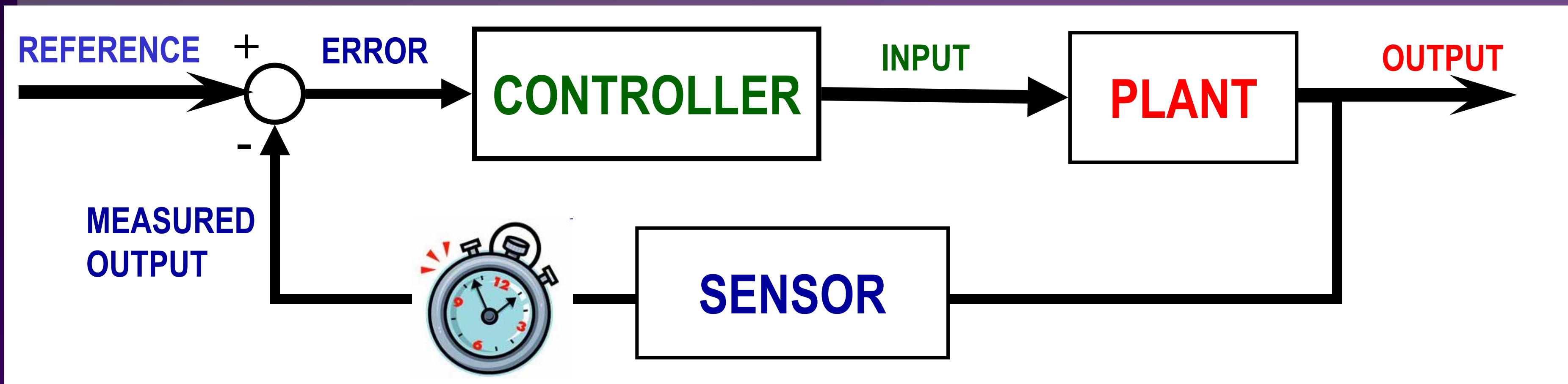
C. G. Cassandras

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

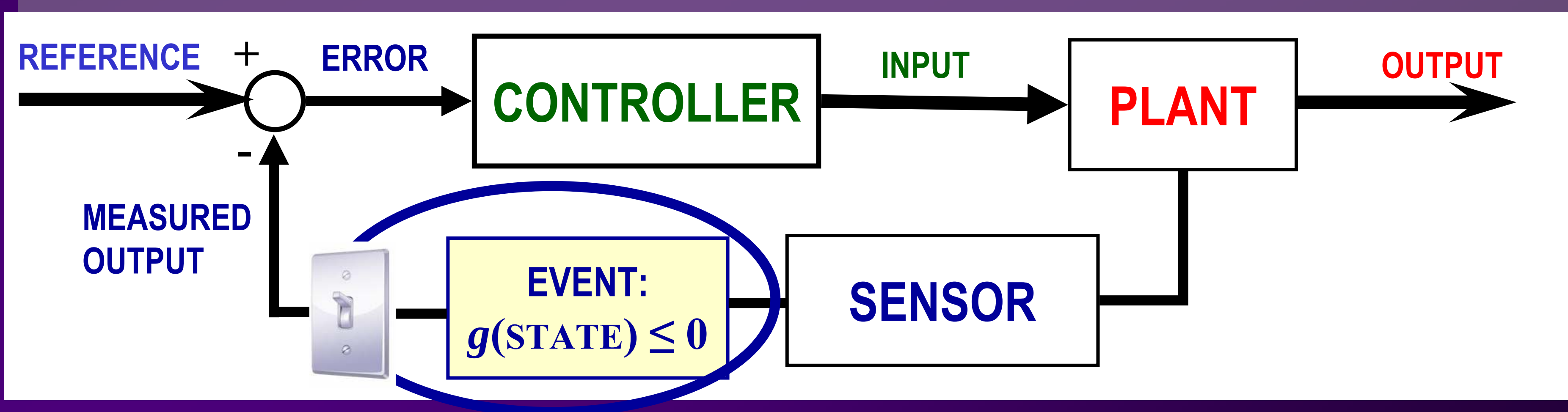
OUTLINE

- Reasons for **EVENT-DRIVEN** Control and Optimization
- **EVENT-DRIVEN** Control in Distributed Systems
- **EVENT-DRIVEN** Control in Managing Uncertainty
- **EVENT-DRIVEN** Sensitivity Analysis

TIME-DRIVEN v EVENT-DRIVEN CONTROL



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



REASONS FOR *EVENT-DRIVEN* MODELS, CONTROL, OPTIMIZATION

- Many systems are naturally **Discrete Event Systems (DES)** (e.g., Internet)
  → *all* state transitions are event-driven

- Most of the rest are **Hybrid Systems (HS)**
  → *some* state transitions are event-driven

- Many systems are **distributed**
  → components interact asynchronously (through events)

- Time-driven sampling inherently inefficient ("open loop" sampling)

REASONS FOR *EVENT-DRIVEN* MODELS, CONTROL, OPTIMIZATION

- Many systems are **stochastic**
  → actions needed in response to random events

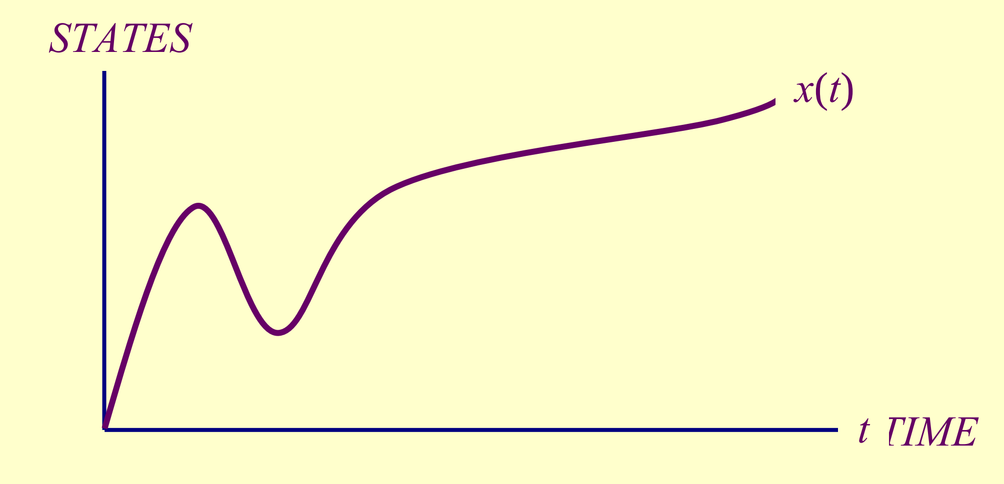
- Event-driven methods provide significant advantages in **computation** and **estimation** quality

- System performance is often **more sensitive to event-driven** components than to time-driven components

- Many systems are **wirelessly networked** → energy constrained
  → time-driven communication consumes significant energy
  UNNECESSARILY!

TIME-DRIVEN v EVENT-DRIVEN SYSTEMS

TIME-DRIVEN SYSTEM



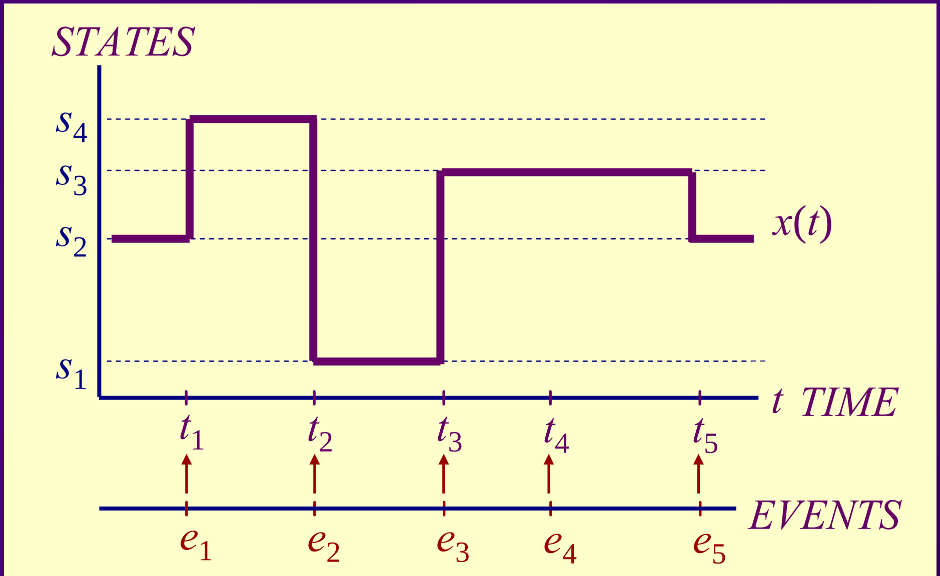
STATE SPACE:

$$X = \Re$$

DYNAMICS:

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

EVENT-DRIVEN SYSTEM



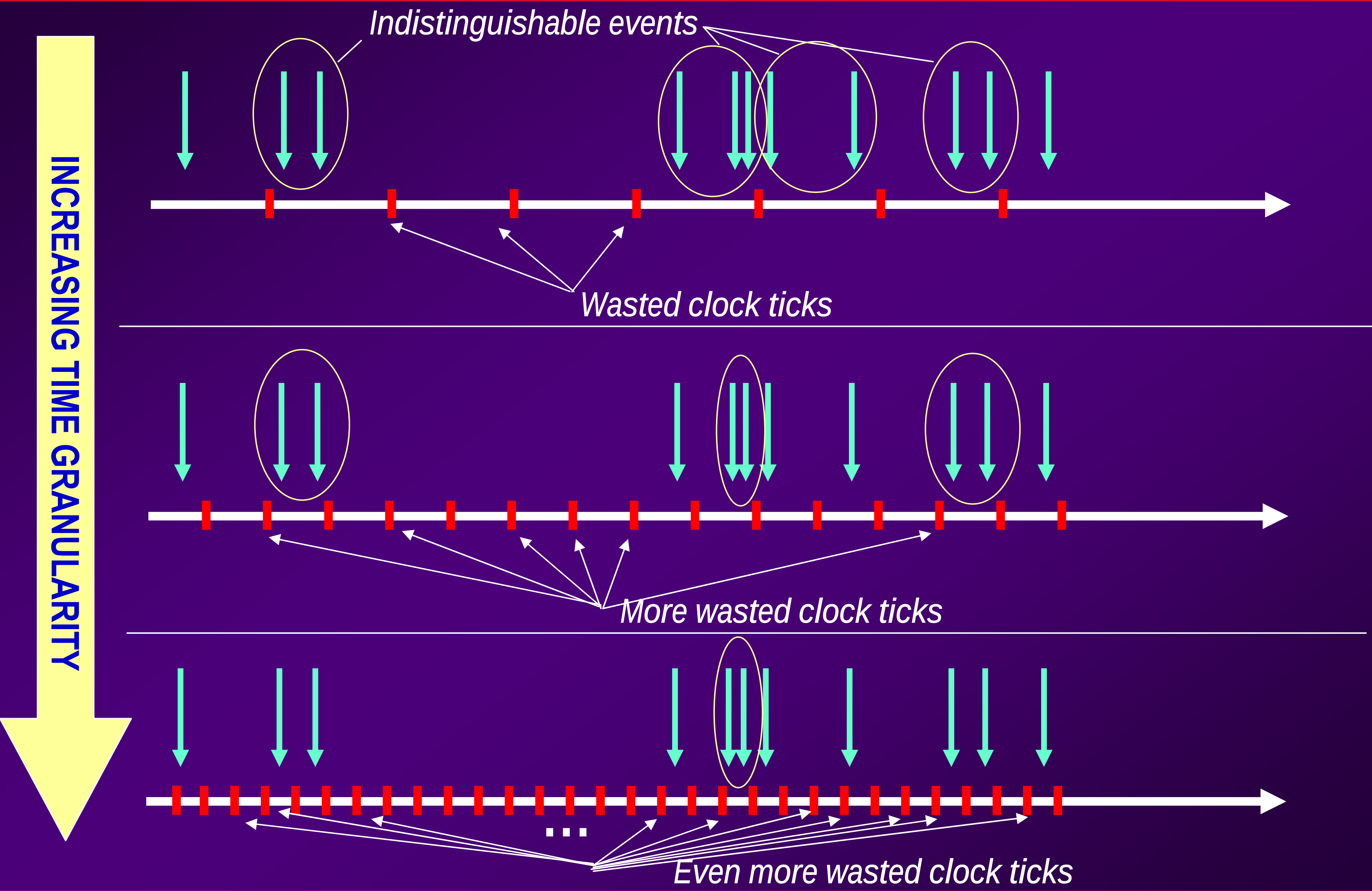
STATE SPACE:

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

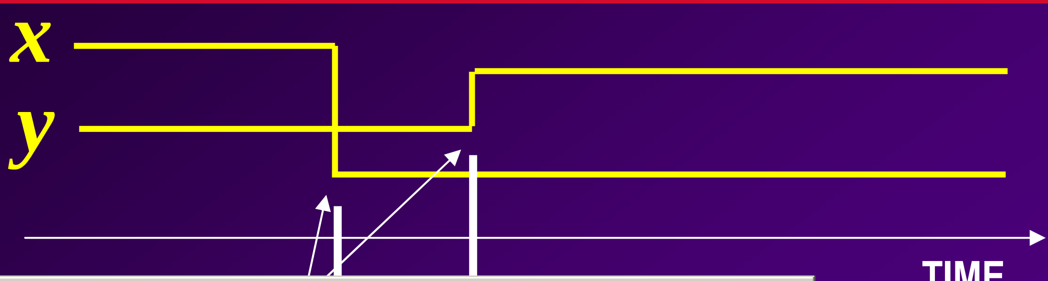
DYNAMICS:

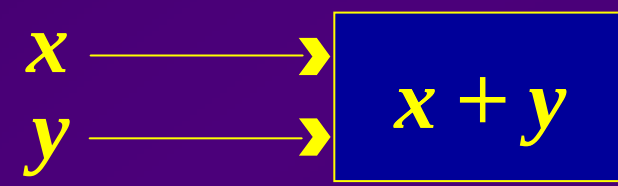
$$x' = f(x,e)$$

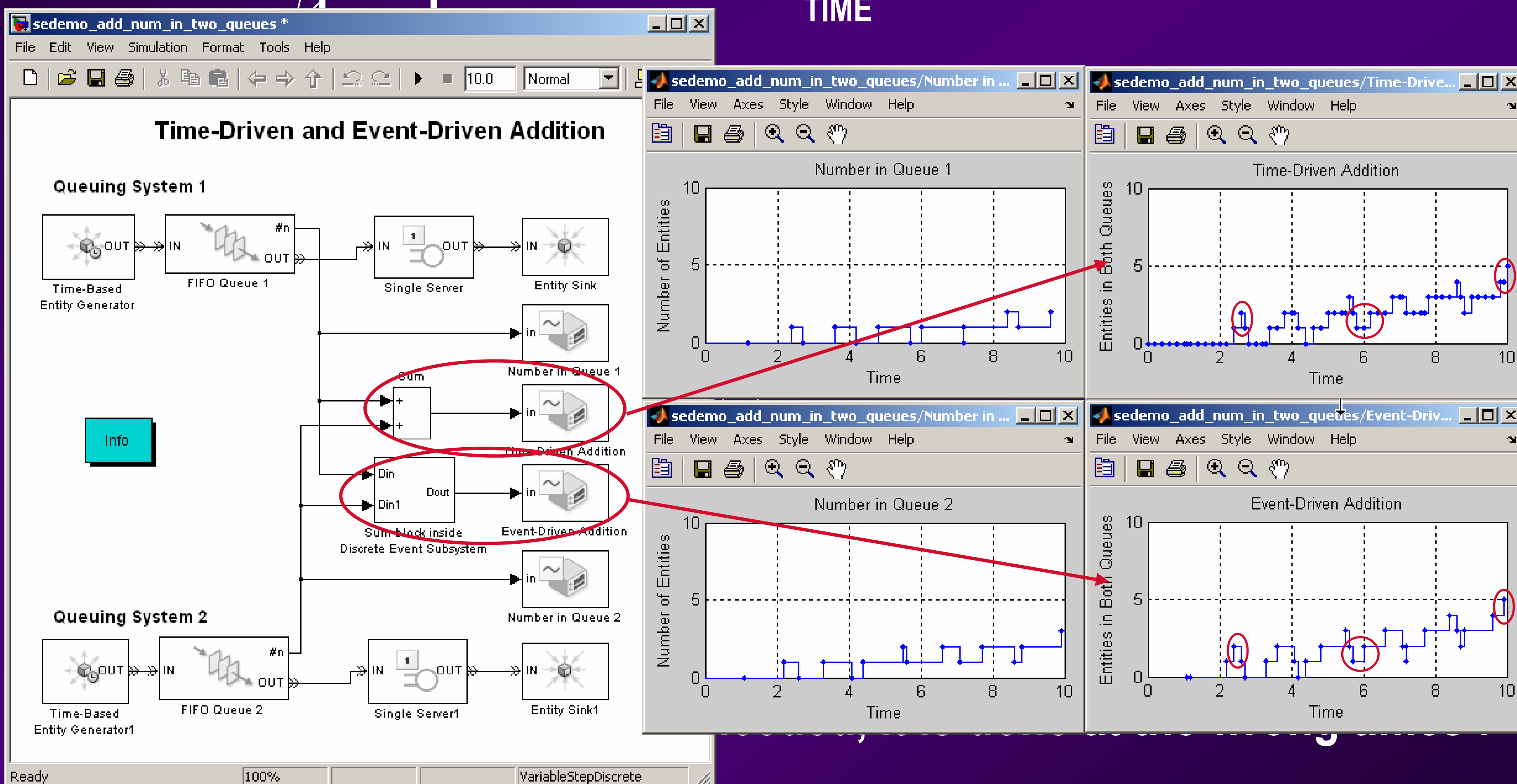
SYNCHRONOUS v ASYNCHRONOUS BEHAVIOR



SYNCHRONOUS v ASYNCHRONOUS COMPUTATION







SELECTED REFERENCES - EVENT-DRIVEN CONTROL

- Astrom, K.J., and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," *Proc. 41st Conf. Decision and Control*, pp. 2011–2016, 2002.
- T. Shima, S. Rasmussen, and P. Chandler, "UAV Team Decision and Control using Efficient Collaborative Estimation," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 609–619, 2007.

- Heemels, W. P. M. H., J. H. Sandee, and P. P. J. van den Bosch, "Analysis of event-driven controllers for linear systems," *Intl. J. Control*, 81, pp. 571–590, 2008.
- P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, pp. 1680–1685, 2007.
- J. H. Sandee, W. P. M. H. Heemels, S. B. F. Hulsenboom, and P. P. J. van den Bosch, "Analysis and experimental validation of a sensor-based event-driven controller," *Proc. American Control Conf.*, pp. 2867–2874, 2007.
- J. Lunze and D. Lehmann, "A state-feedback approach to event-based control," *Automatica*, 46, pp. 211–215, 2010.

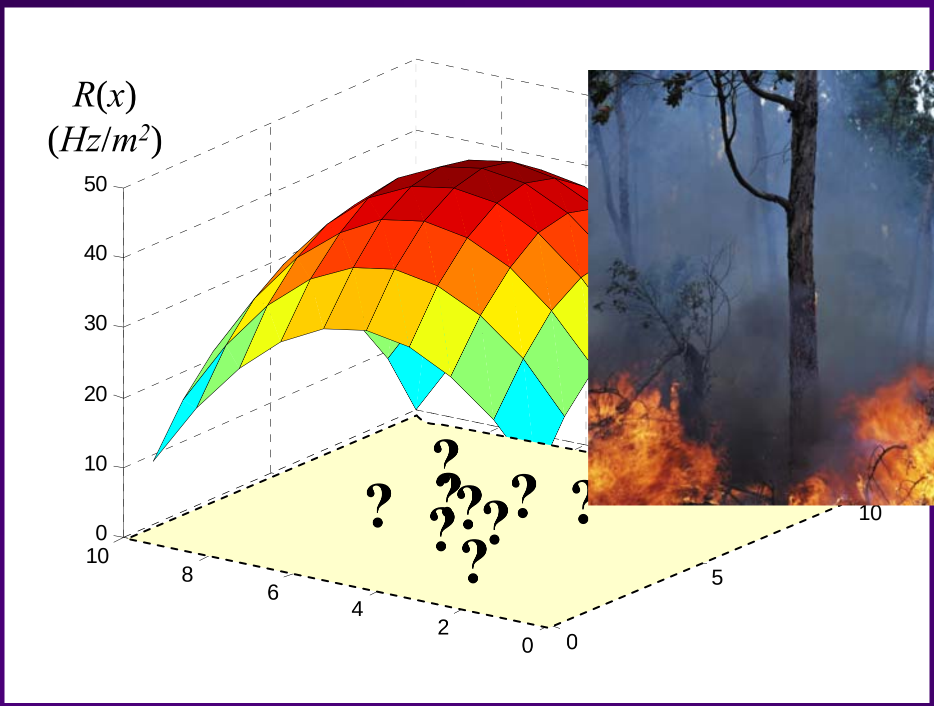
- P. Wan and M. D. Lemmon, "Event triggered distributed optimization in sensor networks," *Proc. of 8th ACM/IEEE Intl. Conf. on Information Processing in Sensor Networks*, 2009.
- Zhong, M., and Cassandras, C.G., "Asynchronous Distributed Optimization with Event-Driven Communication", *IEEE Trans. on Automatic Control*, AC-55, 12, pp. 2735-2750, 2010.

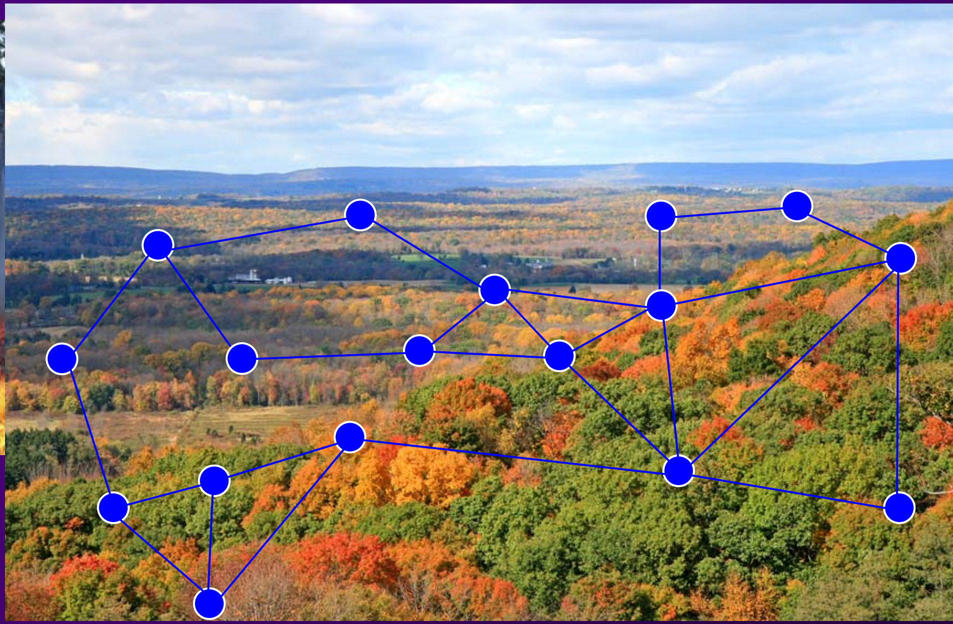
EVENT-DRIVEN CONTROL IN DISTRIBUTED SYSTEMS

MOTIVATIONAL PROBLEM: *COVERAGE CONTROL*

Deploy sensors to maximize "event" detection probability

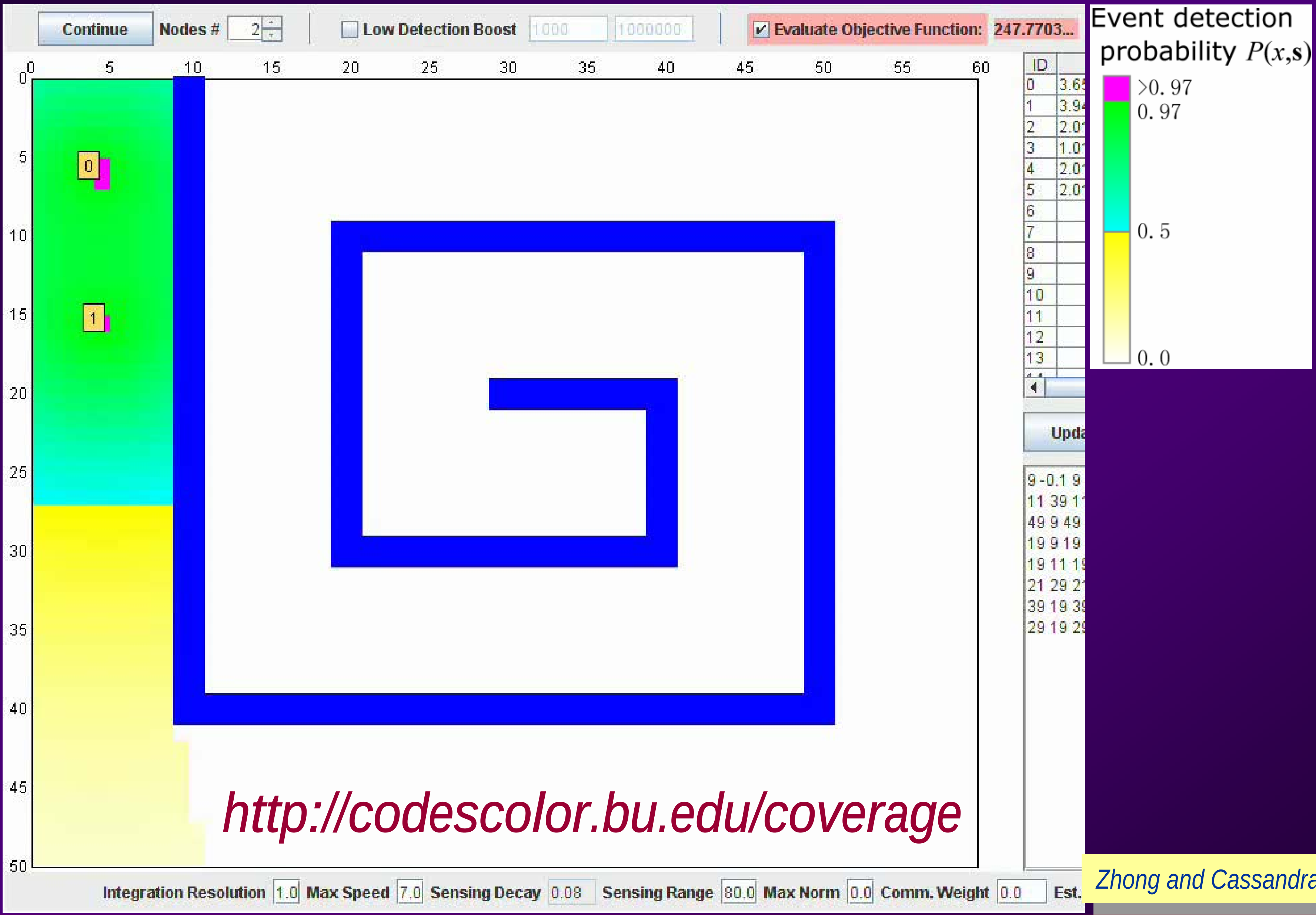
- unknown event locations
- event sources may be mobile
- sensors may be mobile





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

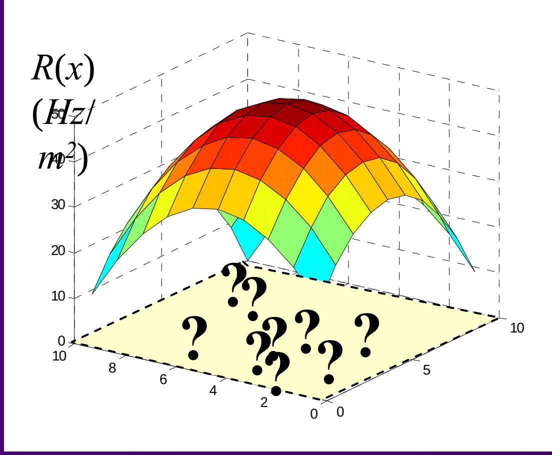
OPTIMAL COVERAGE IN A MAZE



http://codescolor.bu.edu/coverage

Zhong and Cassandras, 2008

COVERAGE: PROBLEM FORMULATION



- N mobile sensors, each located at $s_i \in \mathbf{R}^2$
- Data source at x emits signal with energy E
- Signal observed by sensor node i (at s_i)
- SENSING MODEL:

$$p_i(x, s_i) \equiv P[\text{Detected by } i \mid A(x), s_i]$$

($A(x)$ = data source emits at x)

- Sensing attenuation:

$p_i(x,\ s_i)$ monotonically decreasing in $d_i(x) \equiv \|x - s_i\|$

COVERAGE: PROBLEM FORMULATION

- **Joint detection prob. assuming sensor independence ($\mathbf{s} = [s_1,\ldots,s_N]$: node locations)**

Event sensing probability

$$P(x,\mathbf{s}) = 1 - \prod_{i=1}^{N}\left[1 - p_i(x,s_i)\right]$$

- **OBJECTIVE: Determine locations $\mathbf{s} = [s_1,\ldots,s_N]$ to maximize total *Detection Probability*:**

$$\max_{\mathbf{s}} \int_{\Omega} R(x)P(x,\mathbf{s})dx$$

Perceived event density

DISTRIBUTED COOPERATIVE SCHEME

- **Set**

$$H(s_1,\ldots,s_N)=\int_\Omega R(x)\left\{1-\prod_{i=1}^{N}\left[1-p_i(x)\right]\right\}dx$$

- **Maximize $H(s_1,\ldots,s_N)$ by forcing nodes to move using gradient information:**

$$\frac{\partial H}{\partial s_k}=\int_\Omega R(x)\prod_{i=1,i\neq k}^{N}\left[1-p_i(x)\right]\frac{\partial p_k(x)}{\partial d_k(x)}\frac{s_k-x}{d_k(x)}dx$$

$$s_i^{k+1}=s_i^k+\beta_k\frac{\partial H}{\partial s_i^k}$$

Desired displacement $= V\cdot\Delta t$

Cassandras and Li, 2005
Zhong and Cassandras, 2011

DISTRIBUTED COOPERATIVE SCHEME *CONTINUED*

$$\frac{\partial H}{\partial s_k} = \int_{\Omega} R(x) \prod_{i=1, i\neq k}^{N} [1 - p_i(x)] \frac{\partial p_k(x)}{\partial d_k(x)} \frac{s_k - x}{d_k(x)} dx$$

… has to be autonomously evaluated by each node so as to determine how to move to next position:

$$s_i^{k+1} = s_i^k + \beta_k \frac{\partial H}{\partial s_i^k}$$

- **Use truncated $p_i(x) \Rightarrow \Omega$ replaced by node neighborhood**
- **Discretize $p_i(x)$ using a local grid**

DISTRIBUTED COOPERATIVE OPTIMIZATION

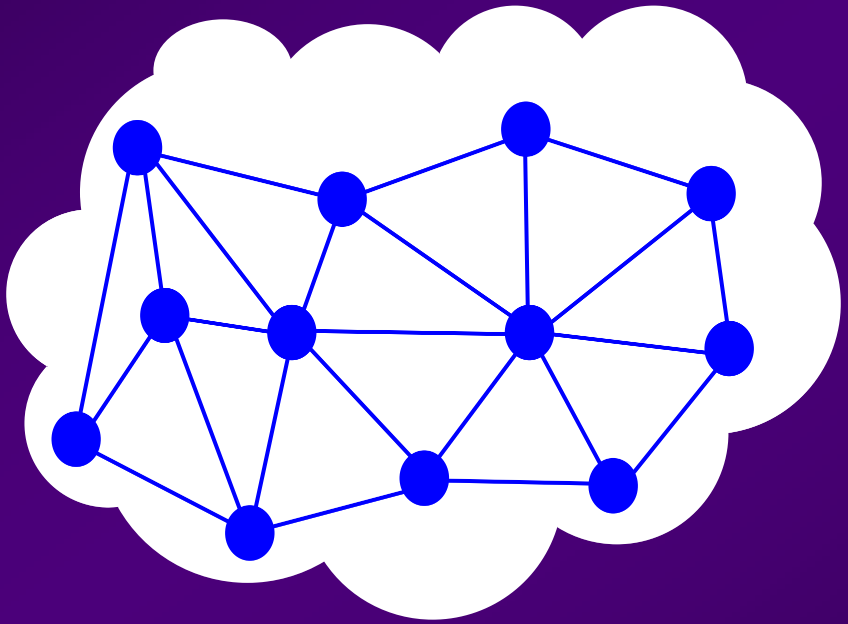
N system components (processors, agents, vehicles, nodes), one common objective:

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

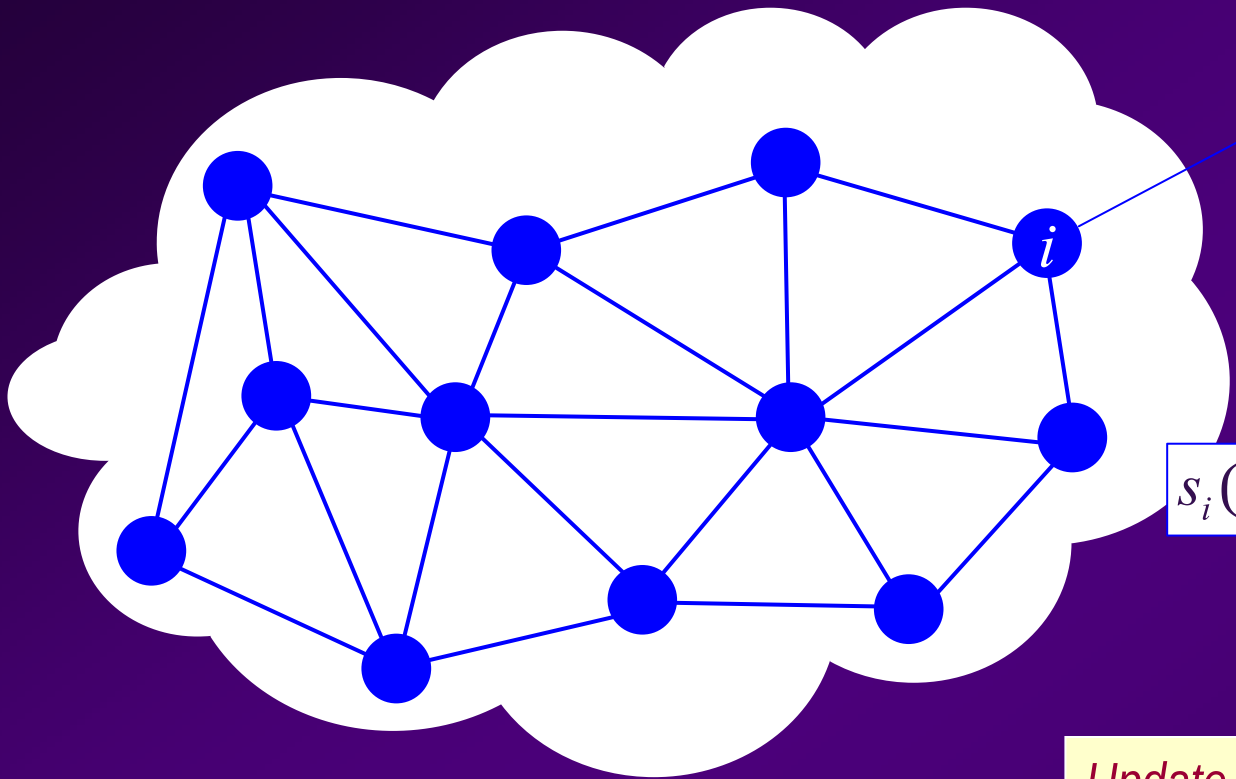
$$\min_{s_1} H(s_1,\ldots,s_N)$$
$$s.t.\ \text{constraints on } s_1$$

⋮

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



DISTRIBUTED COOPERATIVE OPTIMIZATION



Controllable ***state***

$s_i,\ i = 1,\ldots,n_i$

$$s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k))$$

Step Size

Update Direction, usually

$$d_i(\mathbf{s}(k)) = -\nabla_i H(\mathbf{s}(k))$$

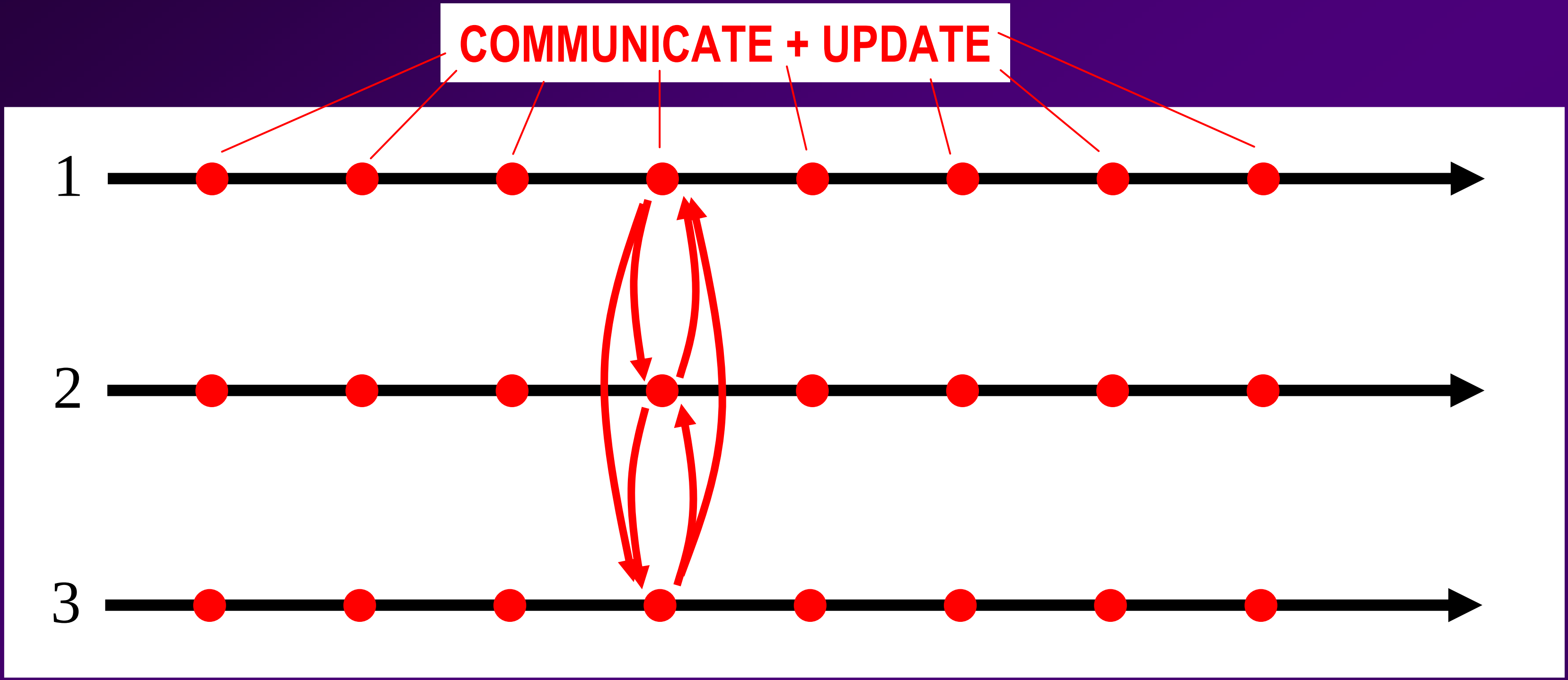
i requires knowledge of all $s_1,\ldots,s_N$

Inter-node communication

$$\min_{s_i} H(s_1,\ldots,s_N)$$

$$s.t.\ \text{constraints on } s_i$$

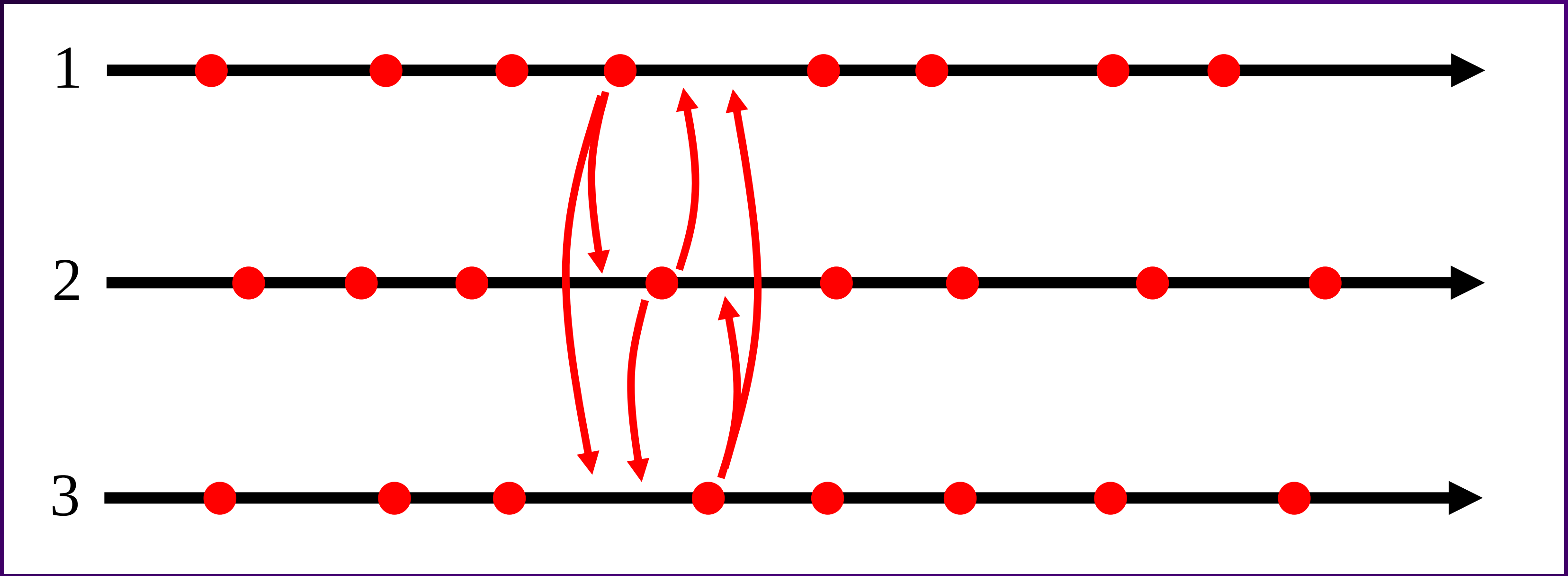
SYNCHRONIZED (TIME-DRIVEN) COOPERATION



Drawbacks:

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

ASYNCHRONOUS COOPERATION



- Nodes not synchronized, delayed information used

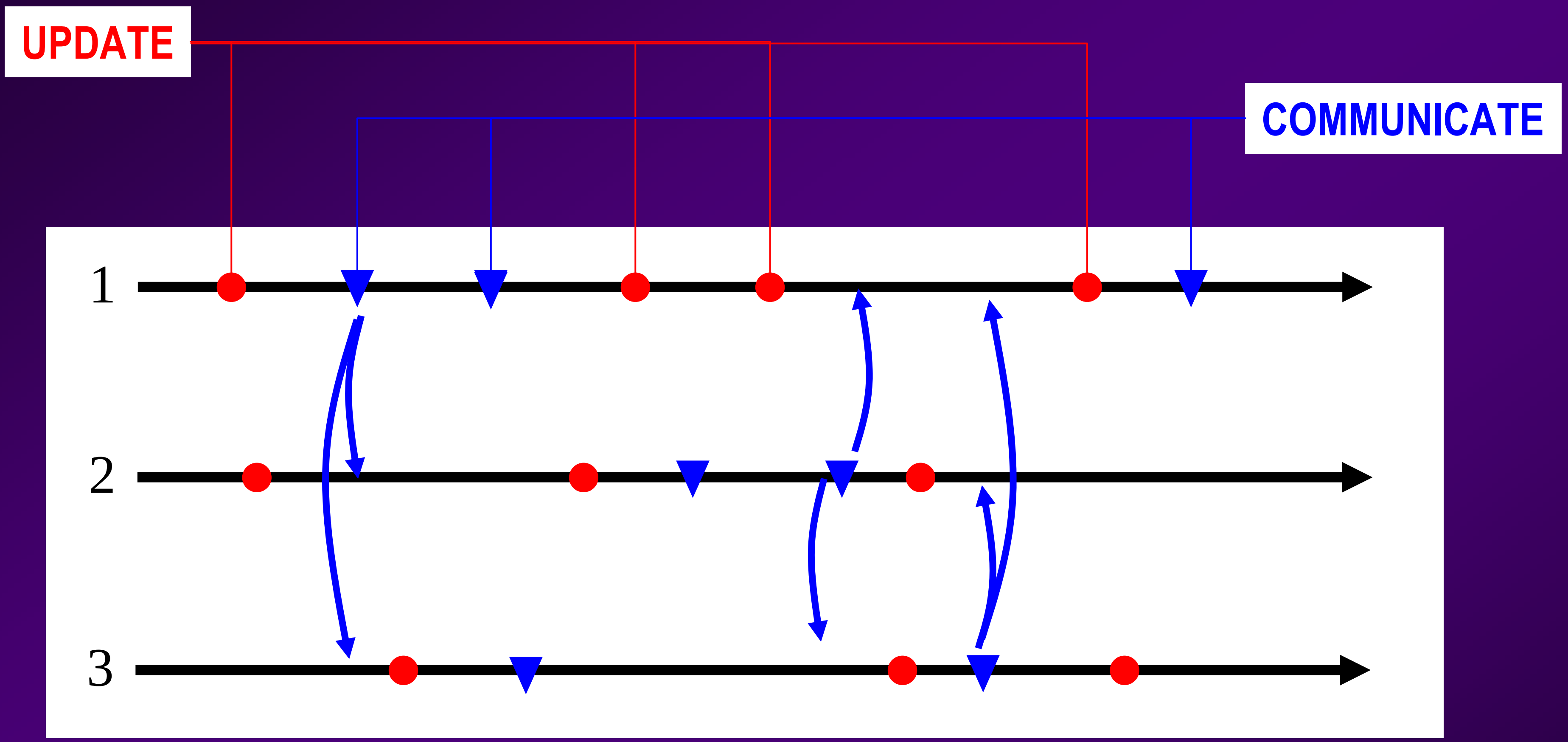
Update frequency for each node is bounded
+
technical conditions

$$\Rightarrow \quad s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k))$$

converges

Bertsekas and Tsitsiklis, 1997

ASYNCHRONOUS (EVENT-DRIVEN) COOPERATION



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

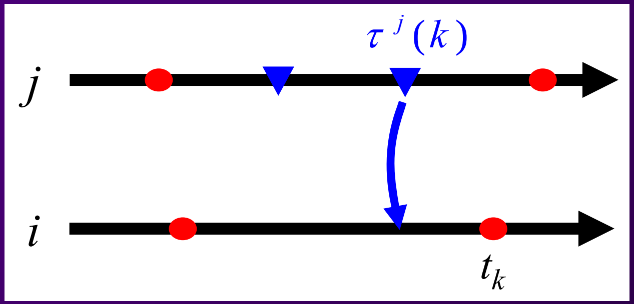
WHEN SHOULD A NODE COMMUNICATE?

$$\left.\begin{array}{ll}\text{Node state at any time } t: & x_i(t) \\ \text{Node state at } t_k: & s_i(k)\end{array}\right\} \Rightarrow s_i(k) = x_i(t_k)$$

AT UPDATE TIME t_k : $s_j^i(k)$: node j **state** estimated by node i

Estimate examples:

➡ $\boxed{s_j^i(k) = x_j(\tau^j(k))}$ **Most recent value**



➡ $\boxed{s_j^i(k) = x_j(\tau^j(k)) + \dfrac{t_k - \tau^j(k)}{\Delta_j} \cdot \alpha_i \cdot d_j\left(x_j(\tau^j(k))\right)}$ **Linear prediction**

WHEN SHOULD A NODE COMMUNICATE?

AT ANY TIME t :

- $x_i^j(t)$: node i state estimated by node j

- If node i knows how j estimates its state, then it can evaluate $x_i^j(t)$

- Node i uses
  - its own **true state**, $x_i(t)$
  - the **estimate that j uses**, $x_i^j(t)$

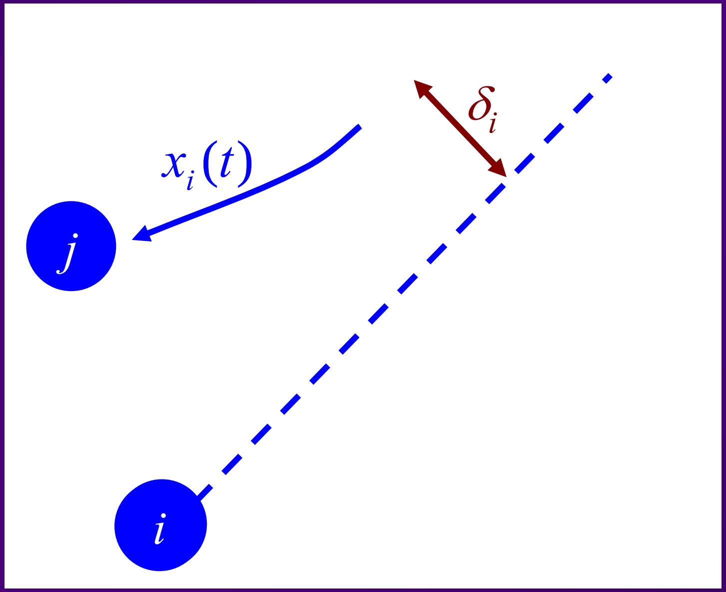
… and evaluates an ERROR FUNCTION $g\left(x_i(t), x_i^j(t)\right)$

Error Function examples: $\left\| x_i(t) - x_i^j(t) \right\|_1, \quad \left\| x_i(t) - x_i^j(t) \right\|_2$

WHEN SHOULD A NODE COMMUNICATE?

Compare ERROR FUNCTION $g\left(x_i(t), x_i^j(t)\right)$ to THRESHOLD δ_i

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 $g\left(x_i(t), x_i^j(t)\right) \geq \delta_i$



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

CONVERGENCE

Asynchronous distributed state update process at each i:

$$s_i(k+1) = s_i(k) + \alpha \cdot d_i(\mathbf{s}^i(k))$$

Estimates of other nodes, evaluated by node i

$$\delta_i(k) = \begin{cases} K_\delta \left\| d_i(\mathbf{s}^i(k)) \right\| & \text{if } k \text{ sends update} \\ \delta_i(k-1) & \text{otherwise} \end{cases}$$

THEOREM: Under certain conditions, there exist positive constants α and K_δ such that

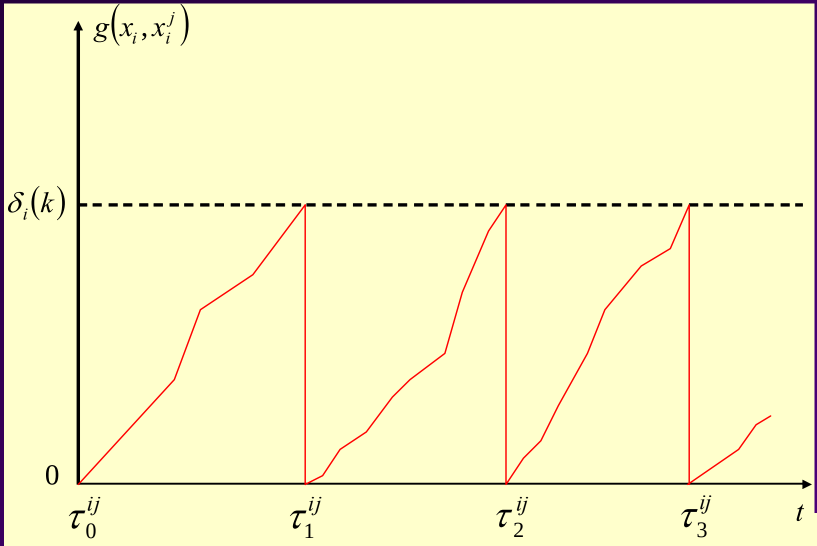
$$\lim_{k \to \infty} \nabla H(\mathbf{s}(k)) = 0$$

Zhong and Cassandras, IEEE TAC, 2010

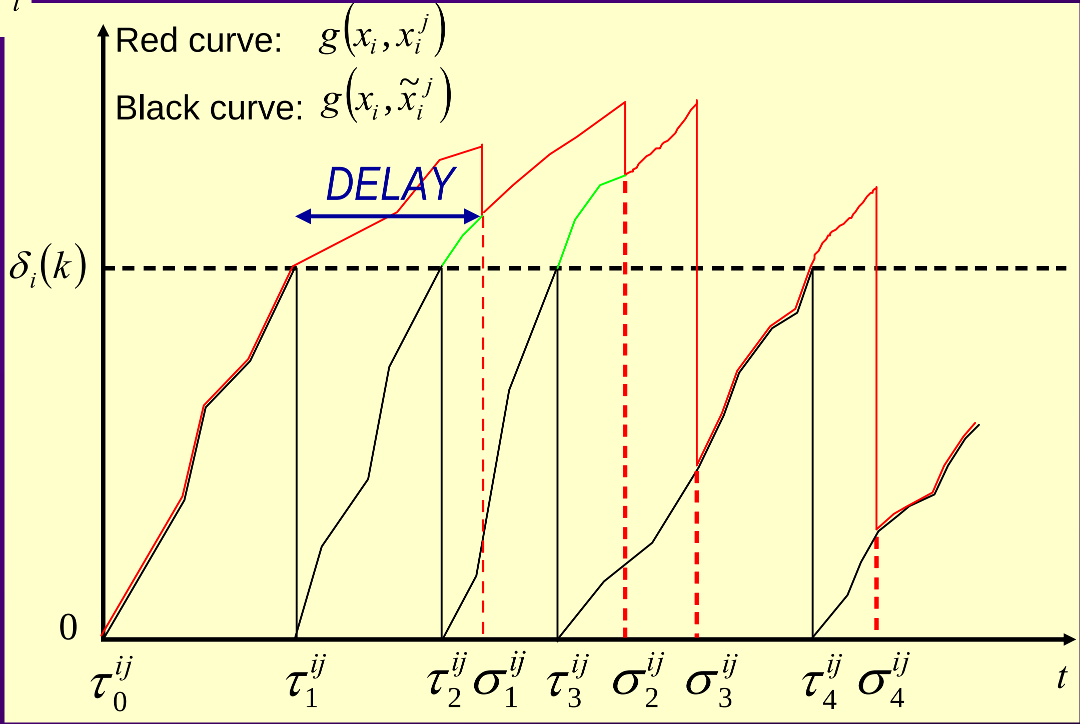
INTERPRETATION:

Event-driven cooperation achievable with minimal communication requirements $\Rightarrow$ energy savings

COONVERGENCE WHEN DELAYS ARE PRESENT



Error function trajectory with NO DELAY



COONVERGENCE WHEN DELAYS ARE PRESENT

Add a boundedness assumption:

ASSUMPTION: There exists a non-negative integer D such that if a message is sent before t_{k-D} from node i to node j, it will be received before t_k.

INTERPRETATION: at most D state update events can occur between a node sending a message and all destination nodes receiving this message.

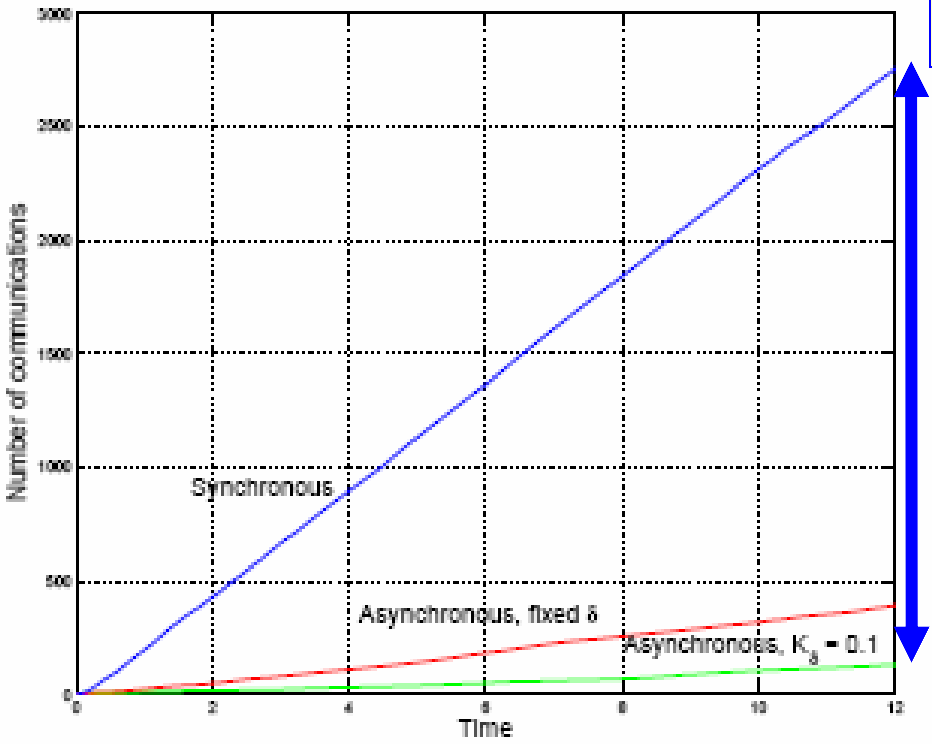
THEOREM: Under certain conditions, there exist positive constants α and K_δ such that

$$\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$$

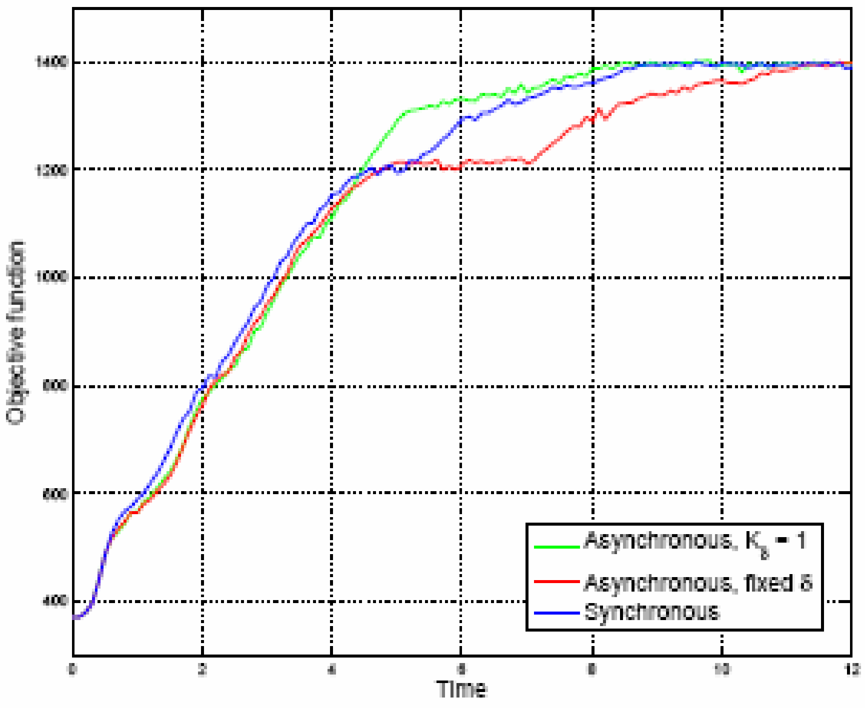
NOTE: The requirements on α and K_δ depend on D and they are tighter.

Zhong and Cassandras, IEEE TAC, 2010

SYNCHRONOUS v ASYNCHRONOUS OPTIMAL COVERAGE PERFORMANCE



Energy savings + Extended lifetime



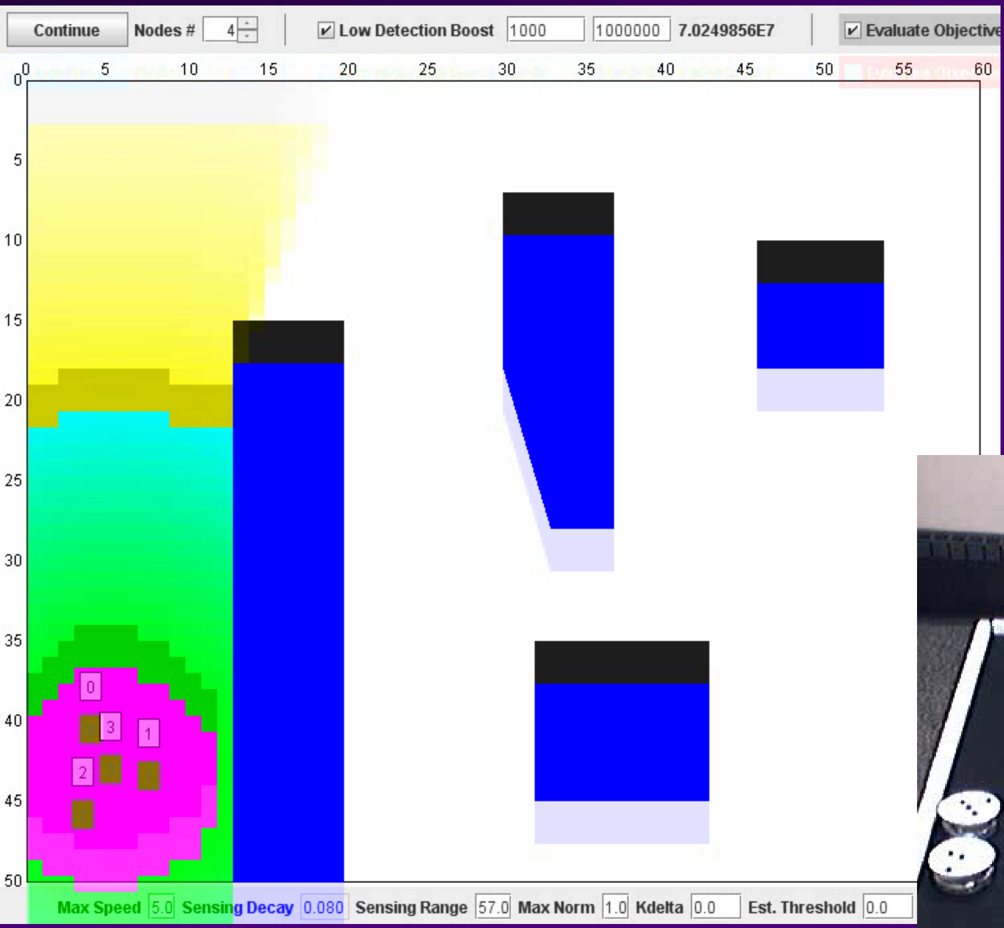
SYNCHRONOUS v ASYNCHRONOUS:

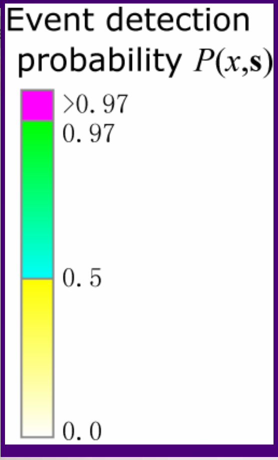
No. of communication events
for a deployment problem *with obstacles*

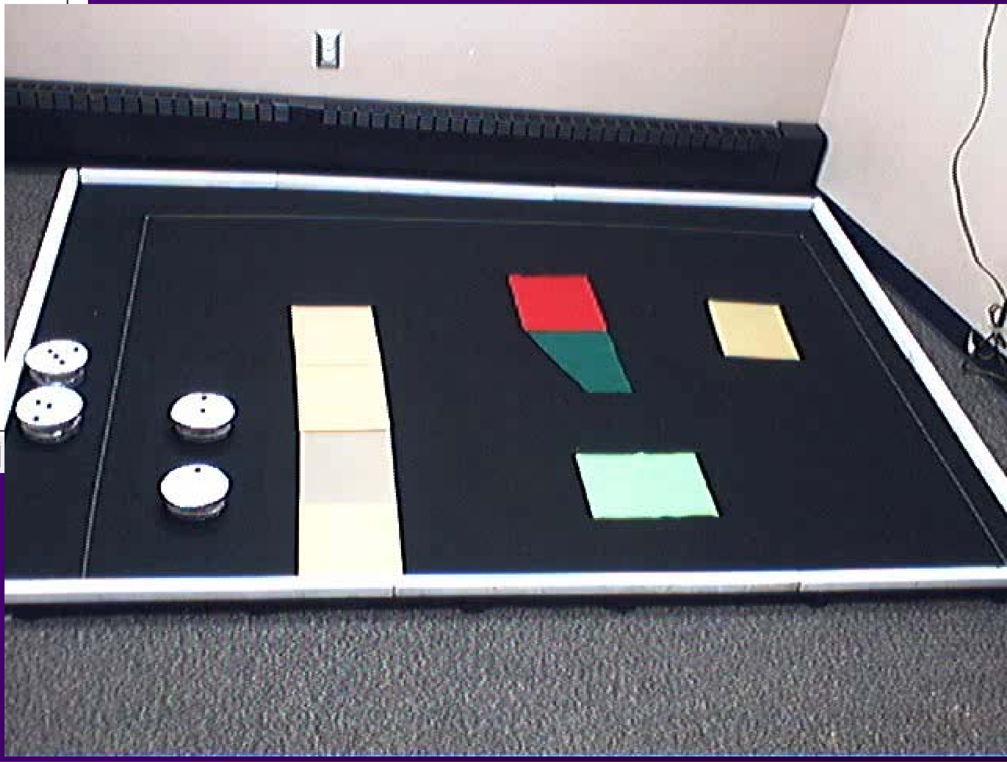
SYNCHRONOUS v ASYNCHRONOUS:

Achieving optimality
in a problem *with obstacles*

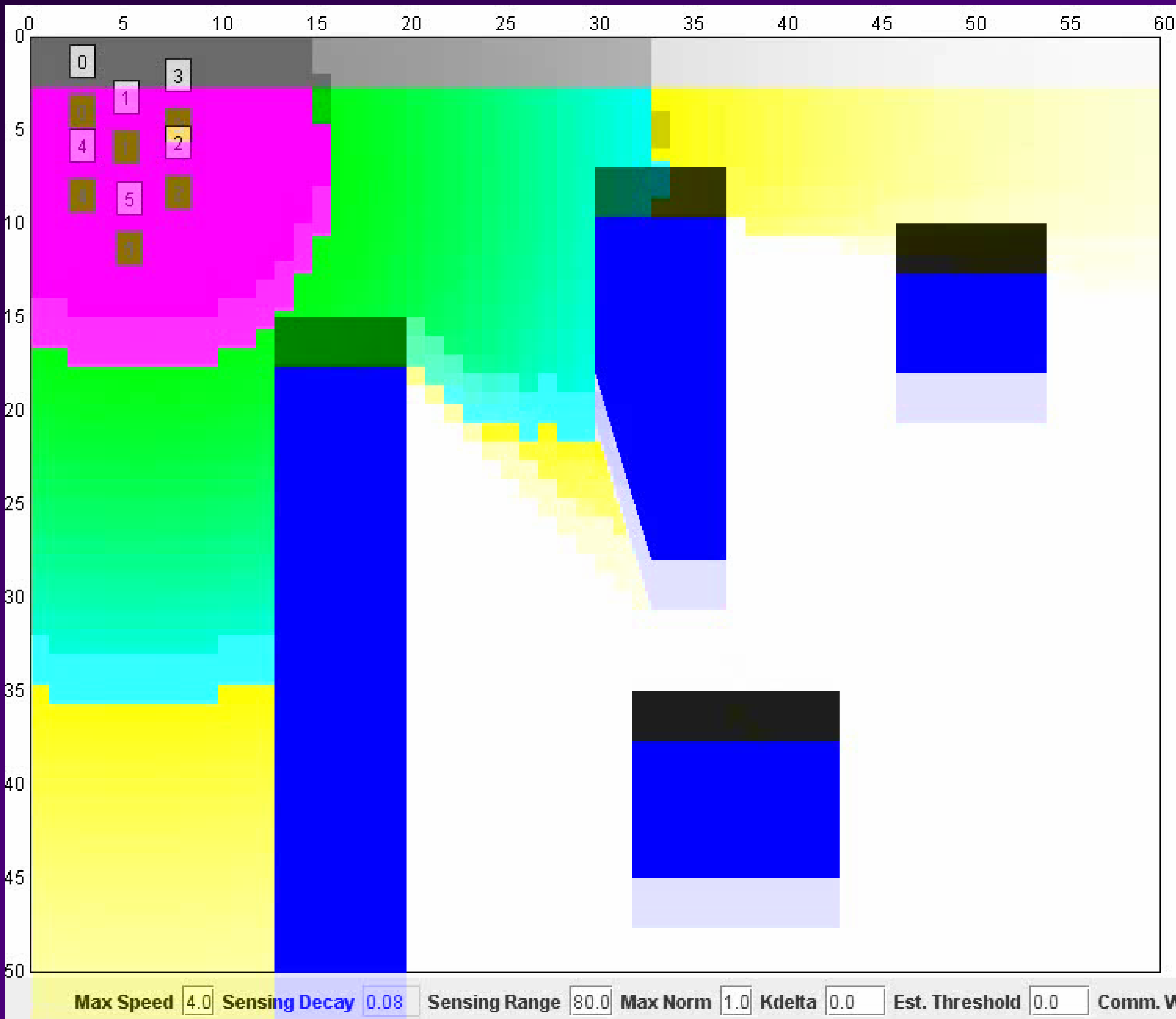
DEMO: OPTIMAL DISTRIBUTED DEPLOYMENT WITH OBSTACLES – *SIMULATED AND REAL*







DEMO: REACTING TO EVENT DETECTION



Important to note:

There is no external control causing this behavior. Algorithm includes tracking functionality automatically

EVENT-DRIVEN CONTROL IN MANAGING UNCERTAINTY

UNCERTAINTY: CONTRAST TWO APPROACHES

ESTIMATE-AND-PLAN vs **HEDGE-AND-REACT**

ESTIMATE-AND-PLAN

- Decisions planned ahead
- Need accurate stochastic models
- Curse of dimensionality

- *Dynamic Programming (DP)*
- *Markov Decision Processes (MDP)*

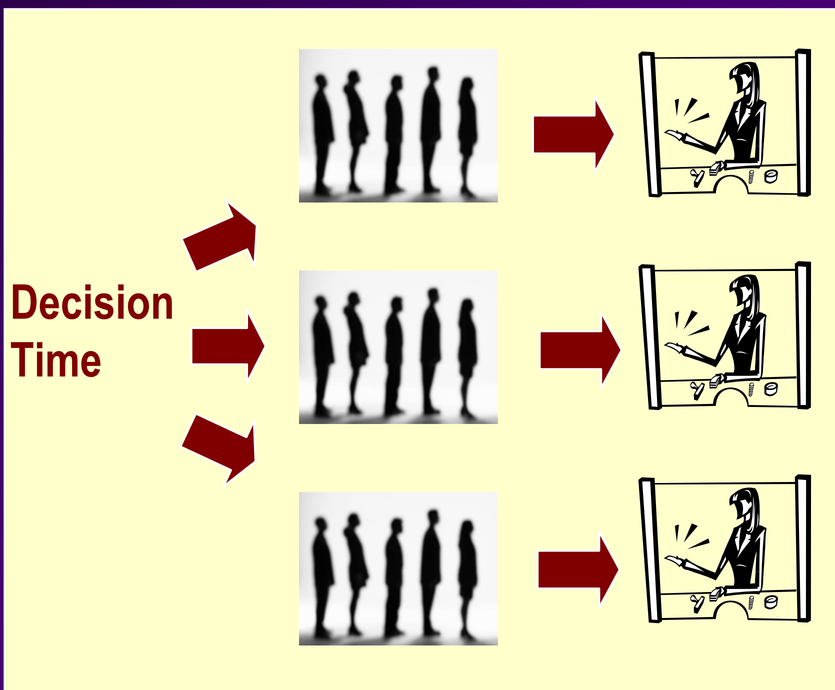
HEDGE-AND-REACT

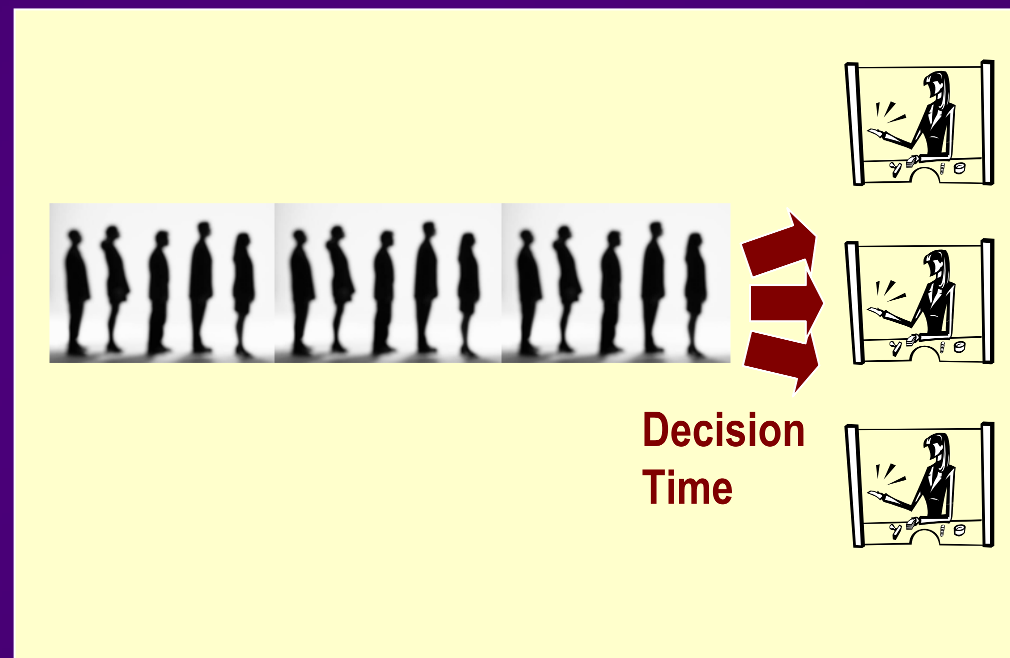
- Delay decisions until last possible instant
- No (detailed) stochastic model
- Simpler opt. problems

- *Receding Horizon Control (RHC)*
- *Model Predictive Control (MPC)*

UNCERTAINTY: CONTRAST TWO APPROACHES

ESTIMATE-AND-PLAN vs **HEDGE-AND-REACT**

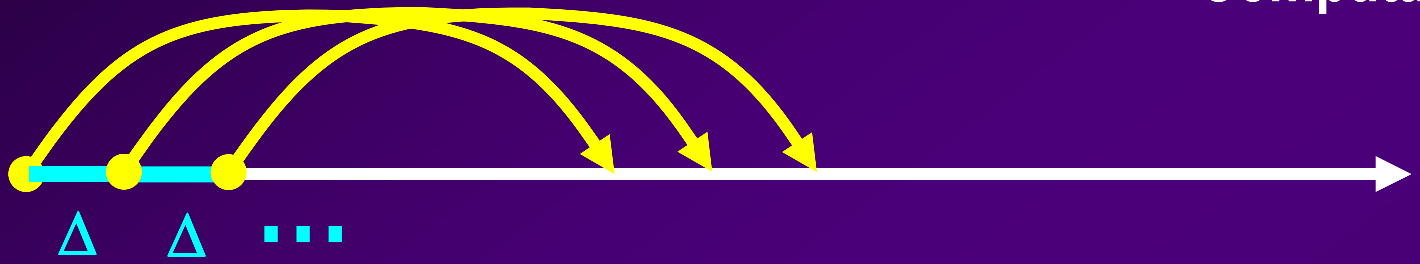




TIME-DRIVEN v EVENT-DRIVEN RHC

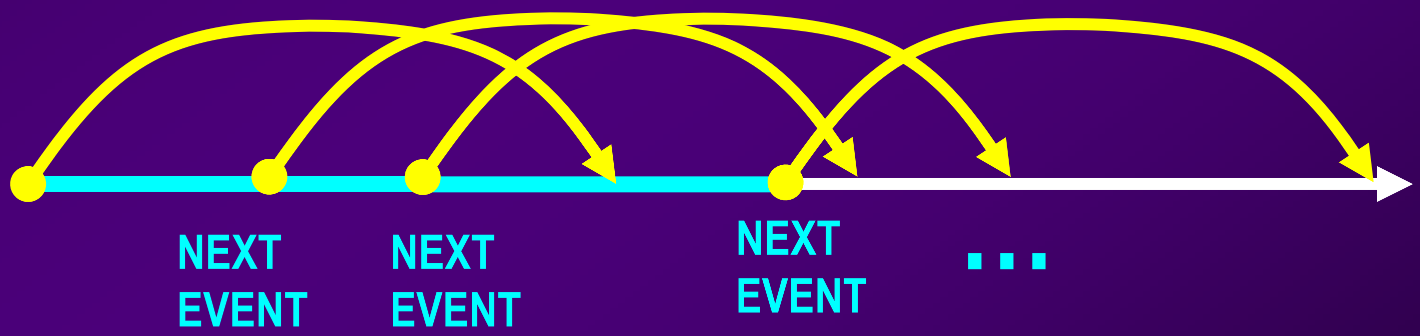
TIME-DRIVEN:

Δ must be small,
Computationally intensive



EVENT-DRIVEN:

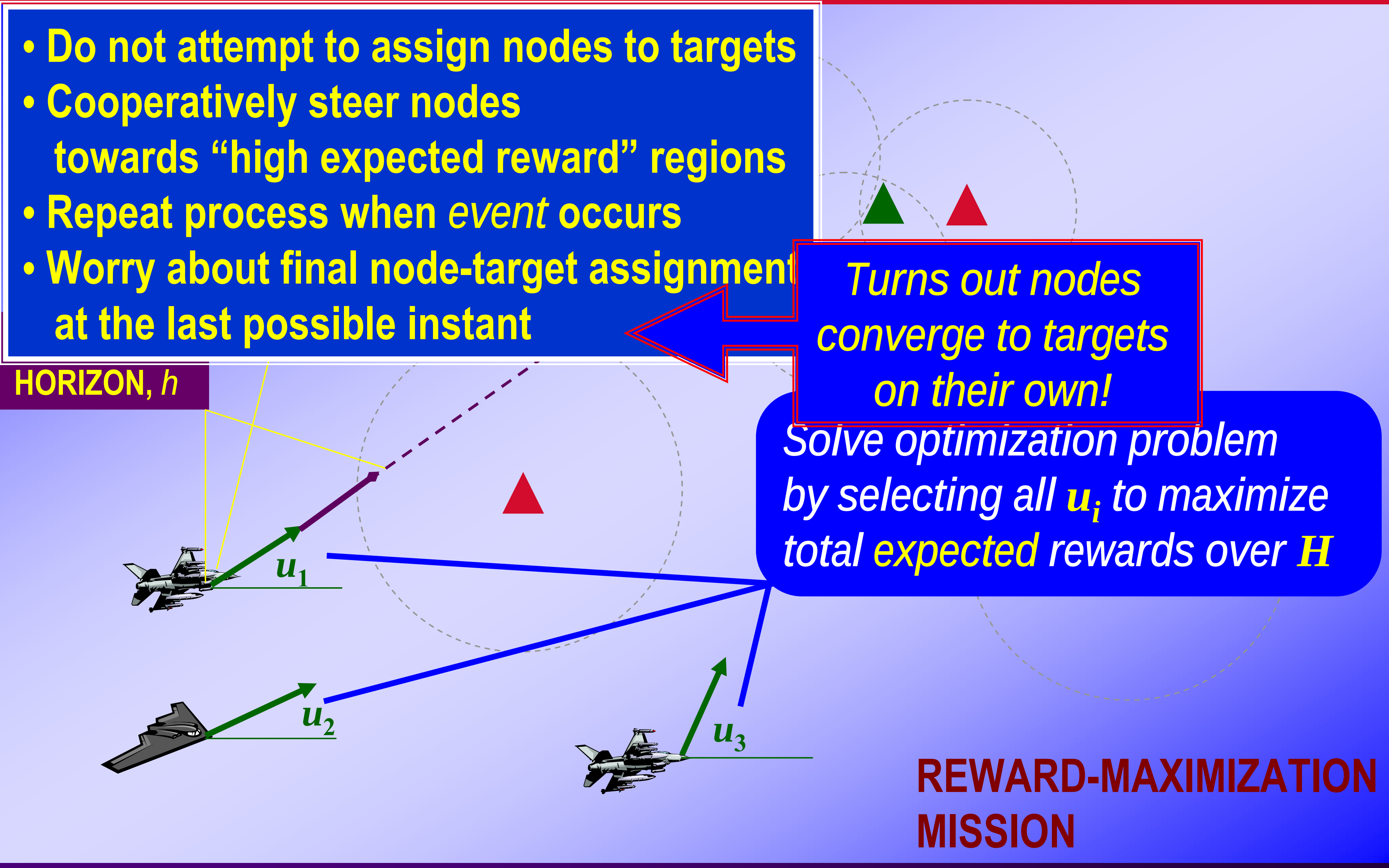
Computational intensity
depends on event frequency



COOPERATIVE RECEDING HORIZON (CRH) CONTROL: *MAIN IDEA*

- Do not attempt to assign nodes to targets
- Cooperatively steer nodes towards "high expected reward" regions
- Repeat process when *event* occurs
- Worry about final node-target assignment at the last possible instant

Turns out nodes converge to targets on their own!



TARGET ASSIGNMENT

MAIN IDEA IN CRH APPROACH:

- **Replace complex** ***Discrete Stochastic Optimization*** **problem by a sequence of simpler** ***Continuous Optimization*** **problems**
- **Solve each new problem whenever a PREDEFINED EVENT occurs (e.g., some node gets to some target)**
- **… or when a RANDOM EVENT (or a TIMEOUT) occurs**

But how do we guarantee that nodes ultimately head for the desired DISCRETE TARGET POINTS?

- **TARGETS:** y_i
- **NODES:** x_j
- **DISTANCE:** d_{ij}

DEFINITION: Node trajectory $\mathbf{x}(t) = [x_1(t), \ldots, x_M(t)]$ **generated by a controller is *stationary*, if there exists some** $t_V < \infty$**, such that** $\|x_j(t_v) - y_i\| \le s_i$ **for some** $i = 1, \ldots, N,\ j = 1, \ldots, M.$

Target Size

QUESTION:

Under what conditions is a CRH-generated trajectory stationary ?

MAIN STABILITY RESULT

Local minima of objective function $J(x)$: $x^l = (x_1^l, \ldots, x_M^l) \in \mathbf{R}^{2M}, \quad l = 1, \ldots, L$

Vector of node positions
at *k*th iteration of CRH controller: $\mathbf{x}_k$

Theorem: Suppose $H_k = \min_{i,j} d_{ij}(t_k)$.
If, for all $l = 1,\ldots,L$, $x_j^l = y_i$ for some $i = 1,\ldots,N$, $j = 1,\ldots,M$,
then $J(\mathbf{x}_k) - J(\mathbf{x}_{k+1}) > b \quad$ ($b > 0$ is a constant).

If all local minima coincide with targets,
the CRH-generated trajectory is stationary

MAIN STABILITY RESULT

QUESTION:

When do all local minima coincide with target points?

1 Node, *N* targets

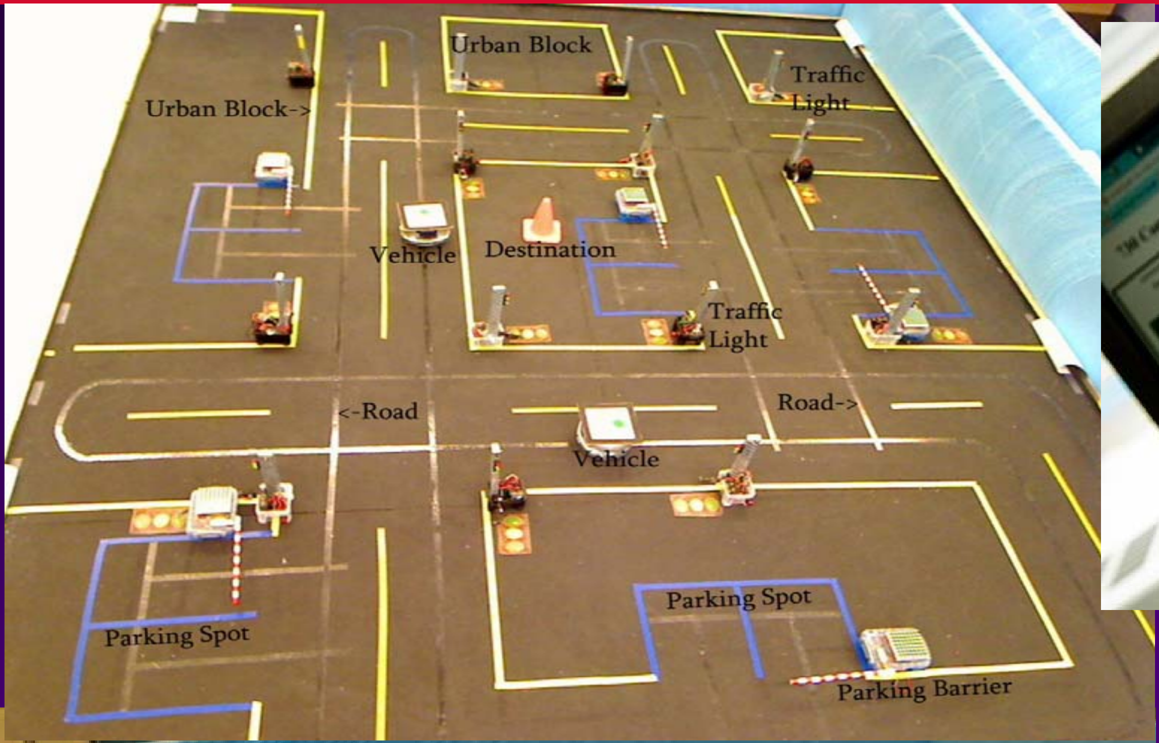
If there exists a y_i s.t. $R_i - \left\| \sum_{j=1, j \neq i}^{N} R_j \frac{y_i - y_j}{\left\| y_i - y_j \right\|} \right\| > 0$

2 Nodes, 1 target

2 Nodes, 2 targets

Li and Cassandras, IEEE TAC, 2006

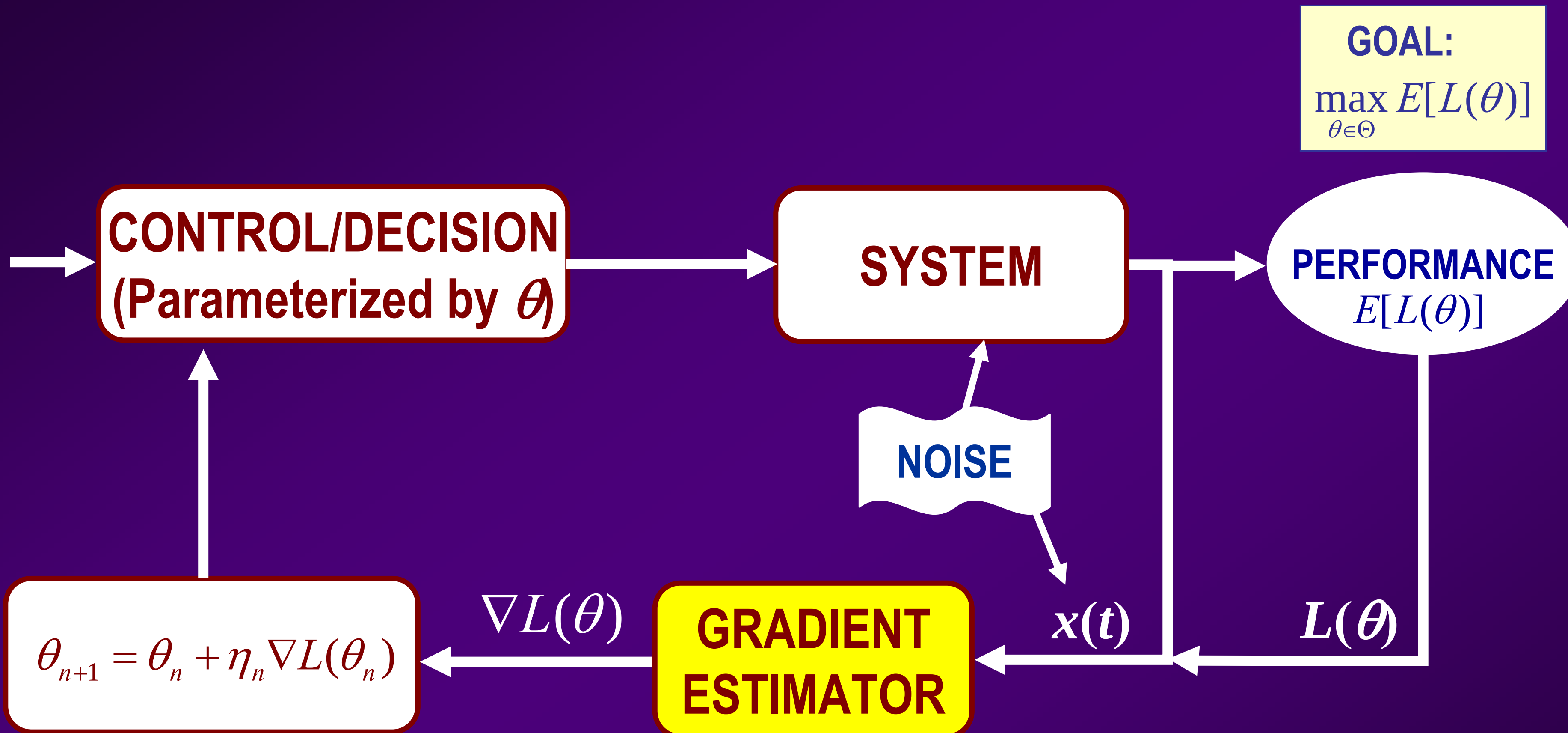
BOSTON UNIVERSITY TEST BEDS



II. 2 Robots, 4 Targets Case

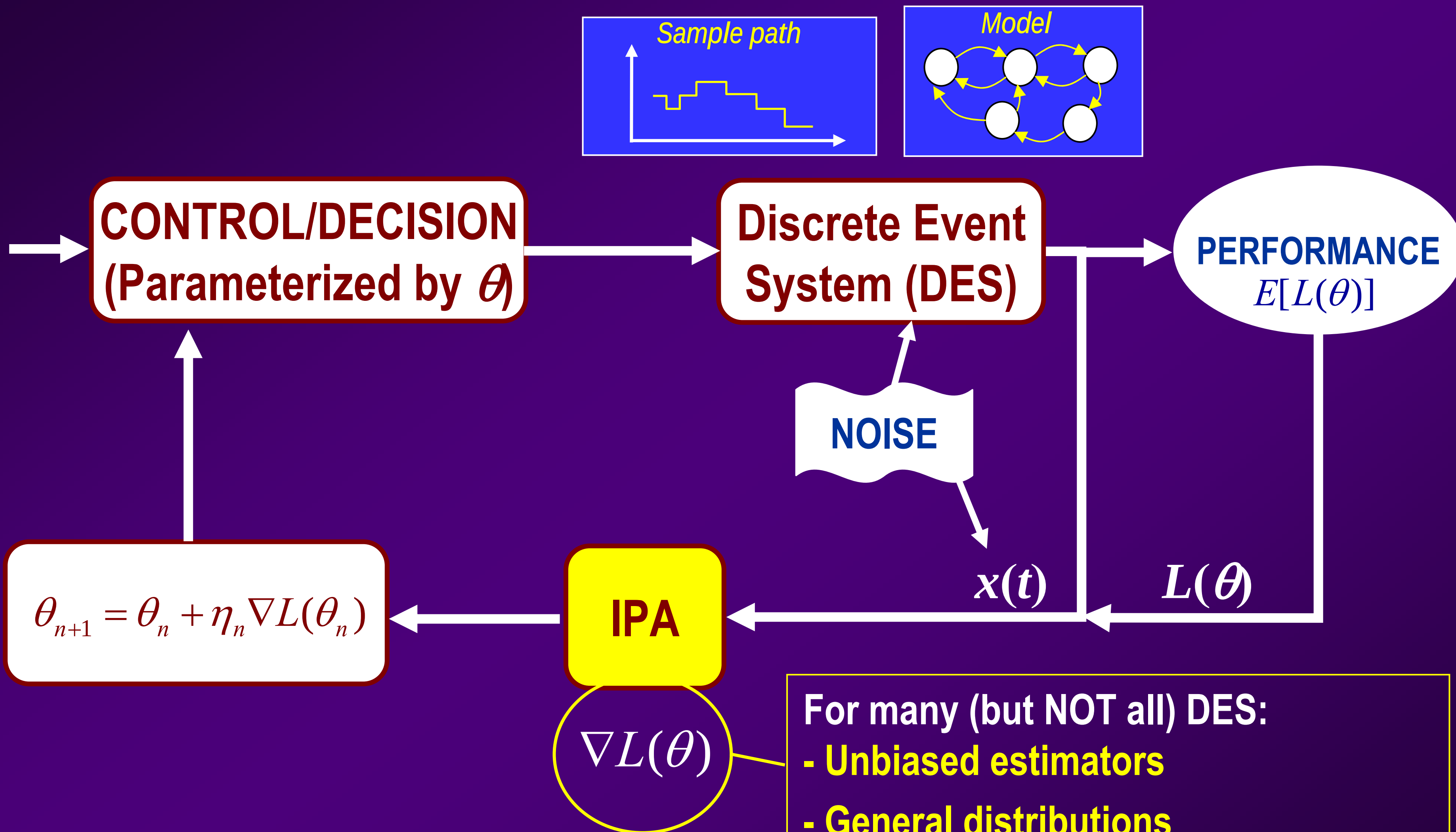
EVENT-DRIVEN SENSITIVITY ANALYSIS

REAL-TIME STOCHASTIC OPTIMIZATION



DIFFICULTIES:
- $E[L(\theta)]$ NOT available in closed form
- $\nabla L(\theta)$ not easy to evaluate
- $\nabla L(\theta)$ may not be a good estimate of $\nabla E[L(\theta)]$

REAL-TIME STOCHASTIC OPTIMIZATION FOR *DES*: INFINITESIMAL PERTURBATION ANALYSIS (IPA)

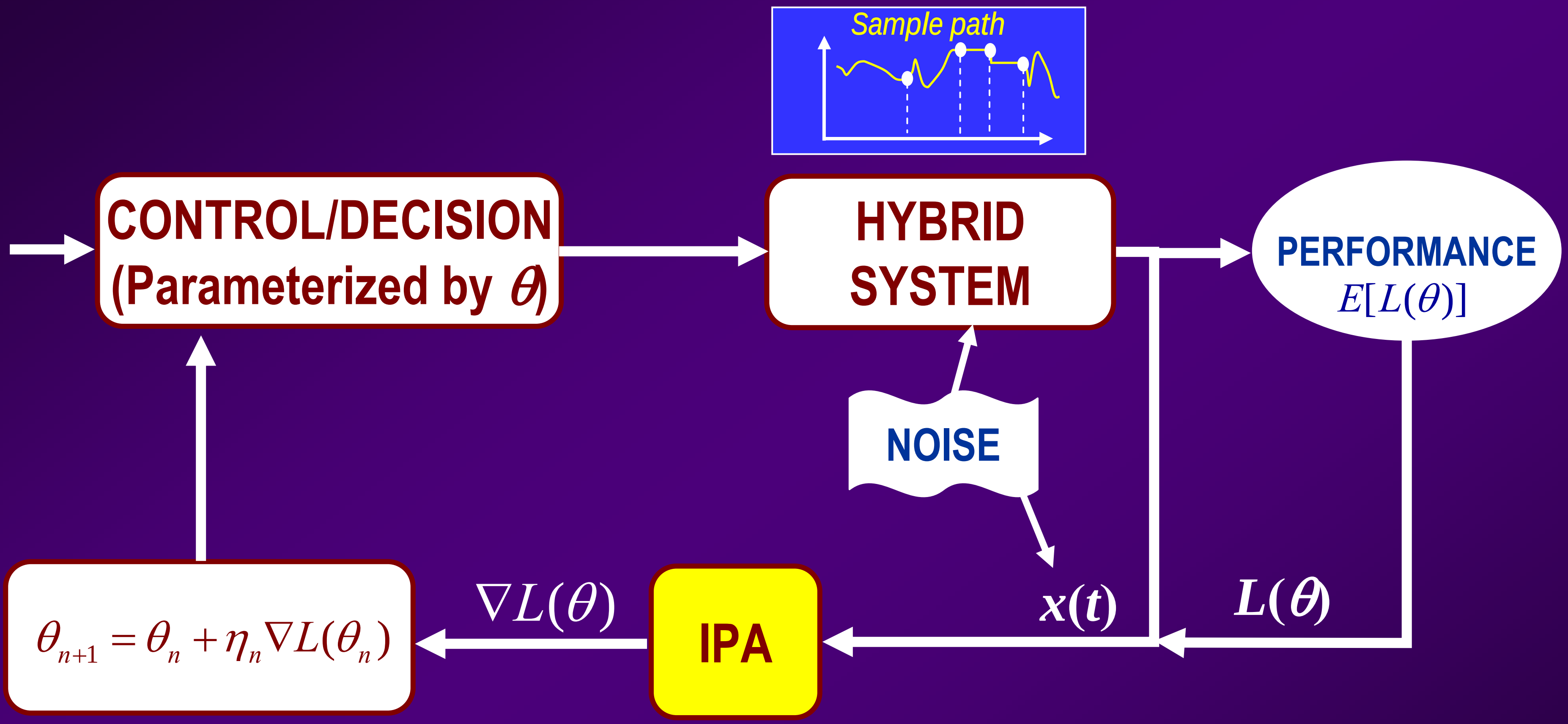


For many (but NOT all) DES:

- Unbiased estimators
- General distributions
- Simple on-line implementation

[*Ho and Cao, 1991*], [*Glasserman, 1991*], [*Cassandras, 1993, 2007*]

REAL-TIME STOCHASTIC OPTIMIZATION: *HYBRID SYSTEMS*



A general framework for an IPA theory in Hybrid Systems?

PERFORMANCE OPTIMIZATION AND IPA

Performance metric (objective function):

$$J\left(\theta;x(\theta,0),T\right)=E\left[L\left(\theta;x(\theta,0),T\right)\right]$$

$$\Downarrow$$

$$L\left(\theta\right)=\sum_{k=0}^{N}\int_{\tau_k}^{\tau_{k+1}}L_k(x,\theta,t)dt$$

IPA goal:

- **Obtain unbiased estimates of** $\dfrac{dJ\left(\theta;x(\theta,0),T\right)}{d\theta}$ **, normally** $\dfrac{dL\left(\theta\right)}{d\theta}$

- **Then:** $\theta_{n+1}=\theta_n+\eta_n\dfrac{dL(\theta_n)}{d\theta}$

NOTATION: $x'\left(t\right)=\dfrac{\partial x\left(\theta,t\right)}{\partial\theta},\quad \tau_k'=\dfrac{\partial\tau_k\left(\theta\right)}{\partial\theta}$

HYBRID AUTOMATA

$$G_h = (Q, X, E, U, f, \phi, Inv, guard, \rho, q_0, \mathbf{x}_0)$$

Q: set of discrete states (modes)

X: set of continuous states (normally R^n)

E: set of events

U: set of admissible controls

f : vector field, $f : Q \times X \times U \to X$

$\boldsymbol{\phi}$: discrete state transition function, $\phi : Q \times X \times E \to Q$

Inv: set defining an invariant condition (domain), $Inv \subseteq Q \times X$

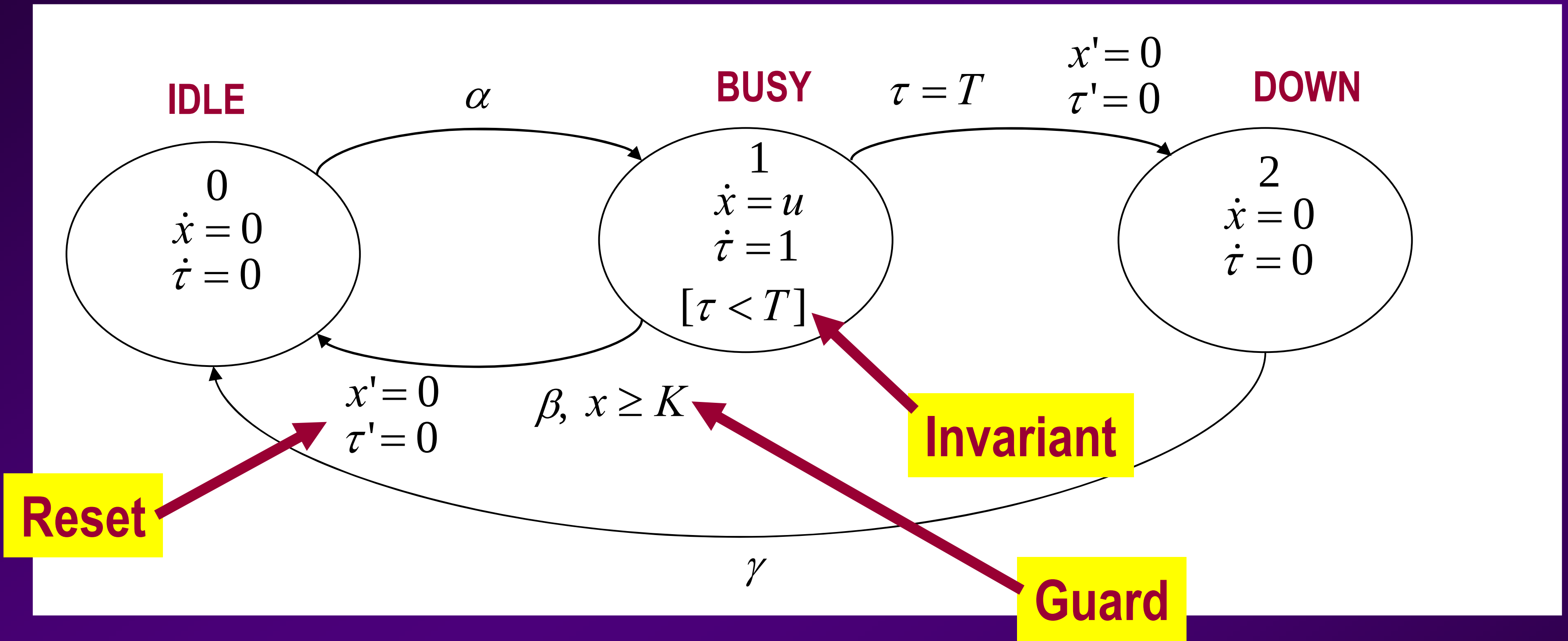
guard: set defining a guard condition, $guard \subseteq Q \times Q \times X$

$\boldsymbol{\rho}$: reset function, $\rho : Q \times Q \times X \times E \to X$

$\mathbf{q_0}$: initial discrete state

$\mathbf{x_0}$: initial continuous state

Unreliable machine with timeouts



$x(t)$: physical state of part in machine

$\tau(t)$: clock

α : START, β : STOP, γ : REPAIR

THE IPA CALCULUS

IPA: *THREE FUNDAMENTAL EQUATIONS*

System dynamics over $(\tau_k(\theta),\ \tau_{k+1}(\theta)]$: $\quad \dot{x} = f_k(x,\theta,t)$

NOTATION: $\quad x'(t) = \dfrac{\partial x(\theta,t)}{\partial \theta}, \quad \tau'_k = \dfrac{\partial \tau_k(\theta)}{\partial \theta}$

---

1. **Continuity at events:** $x(\tau_k^+) = x(\tau_k^-)$

Take $d/d\theta$:

$$x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)]\tau'_k$$

If no continuity, use reset condition $\Rightarrow$

$$x'(\tau_k^+) = \frac{d\rho(q,q',x,\upsilon,\delta)}{d\theta}$$

IPA: *THREE FUNDAMENTAL EQUATIONS*

2. Take $d/d\theta$ of system dynamics $\dot{x} = f_k(x,\theta,t)$ over $(\tau_k(\theta),\ \tau_{k+1}(\theta)]$:

$$\frac{dx'(t)}{dt} = \frac{\partial f_k(t)}{\partial x} x'(t) + \frac{\partial f_k(t)}{\partial \theta}$$

Solve $\frac{dx'(t)}{dt} = \frac{\partial f_k(t)}{\partial x} x'(t) + \frac{\partial f_k(t)}{\partial \theta}$ **over** $(\tau_k(\theta),\ \tau_{k+1}(\theta)]$:

$$x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} \left[\int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{v} \frac{\partial f_k(u)}{\partial x} du} dv + x'(\tau_k^+) \right]$$

initial condition from 1 above

NOTE: If there are no events (pure time-driven system), IPA reduces to this equation

IPA: *THREE FUNDAMENTAL EQUATIONS*

3. Get τ_k' depending on the event type:

- **Exogenous** event: By definition, $\boxed{\tau_k' = 0}$

- **Endogenous** event: occurs when $g_k(x(\theta,\tau_k),\theta) = 0$

$$\boxed{\tau_k' = -\left[\frac{\partial g}{\partial x} f_k(\tau_k^-)\right]^{-1}\left(\frac{\partial g}{\partial \theta} + \frac{\partial g}{\partial x} x'(\tau_k^-)\right)}$$

- **Induced** events:

$$\boxed{\tau_k' = -\left[\frac{\partial y_k(\tau_k)}{\partial t}\right]^{-1} y_k'(\tau_k^+)}$$

IPA: *THREE FUNDAMENTAL EQUATIONS*

Ignoring resets and induced events:

1. $x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)] \cdot \tau'_k$

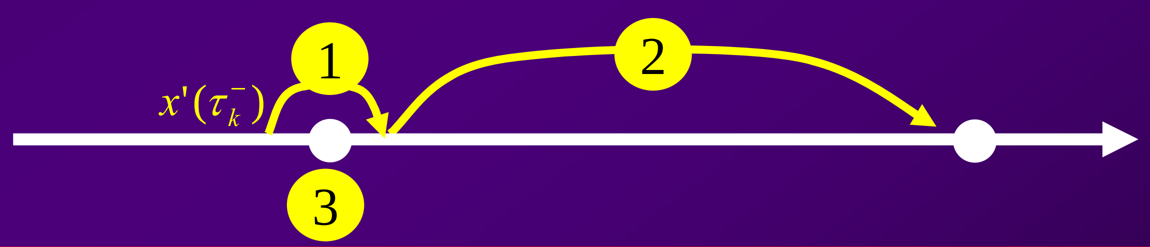
2. $$x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} \left[\int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{v} \frac{\partial f_k(u)}{\partial x} du} dv + x'(\tau_k^+) \right]$$

3. $\tau'_k = 0$ **or** $\tau'_k = -\left[\frac{\partial g}{\partial x} f_k(\tau_k^-) \right]^{-1} \left(\frac{\partial g}{\partial \theta} + \frac{\partial g}{\partial x} x'(\tau_k^-) \right)$

Recall:

$$x'(t) = \frac{\partial x(\theta, t)}{\partial \theta}$$

$$\tau'_k = \frac{\partial \tau_k(\theta)}{\partial \theta}$$



Cassandras et al, Europ. J. Control, 2010

IPA PROPERTIES

Back to performance metric: $L(\theta)=\sum_{k=0}^{N}\int_{\tau_k}^{\tau_{k+1}} L_k(x,\theta,t)dt$

NOTATION: $L'_k(x,\theta,t)=\dfrac{\partial L_k(x,\theta,t)}{\partial\theta}$

Then:

$$\boxed{\frac{dL(\theta)}{d\theta}=\sum_{k=0}^{N}\Bigg[\underbrace{\tau'_{k+1}\cdot L_k(\tau_{k+1})-\tau'_k\cdot L_k(\tau_k)}_{\text{What happens at event times}}+\underbrace{\int_{\tau_k}^{\tau_{k+1}} L'_k(x,\theta,t)dt}_{\text{What happens between event times}}\Bigg]}$$

IPA PROPERTIES

THEOREM 1: If either 1,2 holds, then $dL(\theta)/d\theta$ depends only on information available at event times τ_k:

1. $L(x,\theta,t)$ is independent of t over $[\tau_k(\theta),\ \tau_{k+1}(\theta)]$ for all k

2. $L(x,\theta,t)$ is only a function of x and for all t over $[\tau_k(\theta),\ \tau_{k+1}(\theta)]$:

$$\frac{d}{dt}\frac{\partial L_k}{\partial x} = \frac{d}{dt}\frac{\partial f_k}{\partial x} = \frac{d}{dt}\frac{\partial f_k}{\partial \theta} = 0$$

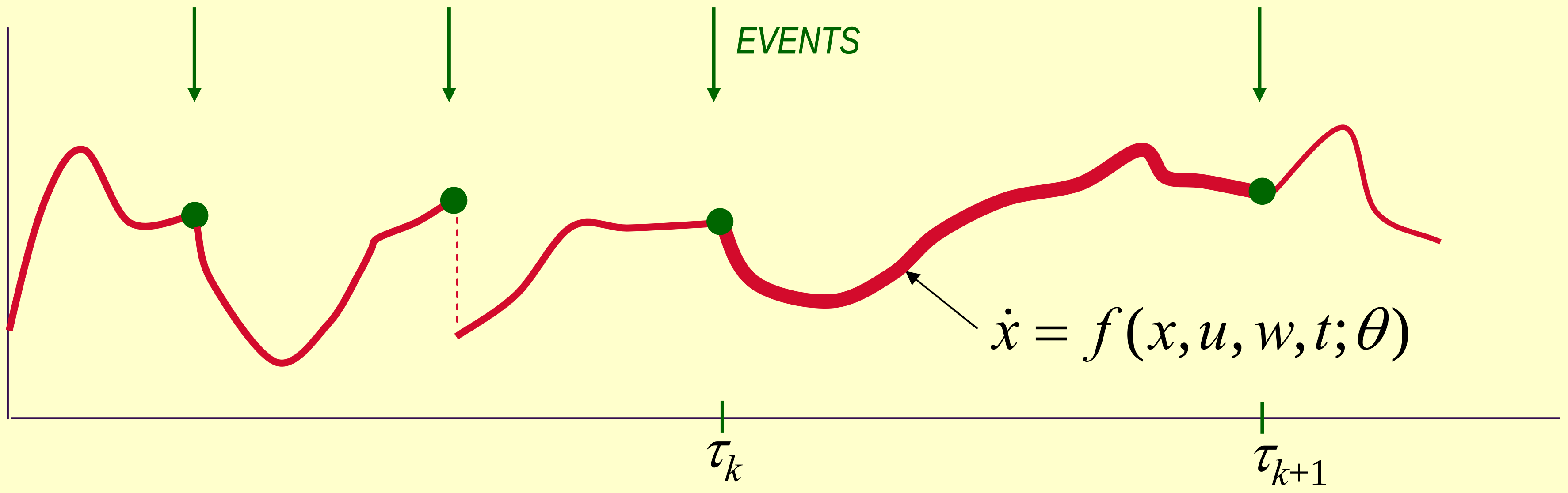
[Yao and Cassandras, 2010]

$$\frac{dL(\theta)}{d\theta} = \sum_{k=0}^{N}\left[\tau'_{k+1}\cdot L_k(\tau_{k+1}) - \tau'_k \cdot L_k(\tau_k) + \cancel{\int_{\tau_k}^{\tau_{k+1}} L'_k(x,\theta,t)dt}\right]$$

IMPLICATION: - Performance sensitivities can be obtained from information limited to event times, which is easily observed

- *No need to track system in between events !*

IPA PROPERTIES



Evaluating $x(t;\theta)$ requires full knowledge of w and f values (obvious)

However, $\dfrac{dx(t;\theta)}{d\theta}$ may be *independent* of w and f values (*NOT* obvious)

It often depends only on: - event times τ_k

- possibly $f(\tau_{k+1}^{-})$

IPA PROPERTIES

In many cases:

- *No need for a detailed model* (captured by f_k) to describe state behavior in between events

- This explains why *simple abstractions of a complex stochastic system* can be adequate to perform sensitivity analysis and optimization, as long as event times are accurately observed and local system behavior at these event times can also be measured.

- This is true in *abstractions of DES as HS* since:
  Common performance metrics (e.g., workload) satisfy THEOREM 1

SOLVING PROBLEMS WITH LINEAR TIME-DRIVEN DYNAMICS

$$\min_{u_1,\ldots,u_K} \sum_{k=0}^{K} \left[\int_{\tau_k}^{\tau_{k+1}} L_k(x(t),u(t))dt + \psi_i(\tau_k) \right]$$

s.t.

$$\dot{x} = a_k x + b_k u_k, \quad t \in (\tau_k, \tau_{k+1}]$$

Common to parameterize controls using basis functions $\beta_l(t)$:

$$u_k(t) = \sum_{l=0}^{L} \theta_{k,l} \beta_l(t)$$

SOLVING PROBLEMS WITH LINEAR TIME-DRIVEN DYNAMICS

Recall IPA equations:

1. $x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)] \cdot \tau'_k$

2. $x'(t) = e^{\int_{\tau_k}^{t} \frac{\partial f_k(u)}{\partial x} du} \left[\int_{\tau_k}^{t} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_k}^{v} \frac{\partial f_k(u)}{\partial x} du} dv + x'(\tau_k^+) \right]$

3. $\tau'_k = 0$ **or** $\tau'_k = -\left[\frac{\partial g}{\partial x} f_k(\tau_k^-) \right]^{-1} \left(\frac{\partial g}{\partial \theta} + \frac{\partial g}{\partial x} x'(\tau_k^-) \right)$

When endogenous event $[g(x(\theta, \tau_k), \theta) = 0]$ **occurs at** τ_k **:**

$$1 \Rightarrow \frac{\partial x(\tau_k^+)}{\partial \theta_{k,l}} = \frac{\partial x(\tau_k^-)}{\partial \theta_{k,l}} + \left[(a_{k-1} - a_k) x(\tau_k) + b_{k-1} \sum_{l=0}^{L} \theta_{k-1,l} \beta_l(\tau_k) - b_k \sum_{l=0}^{L} \theta_{k,l} \beta_l(\tau_k) \right] \frac{\partial \tau_k}{\partial \theta_{k,l}}$$

$$2 \Rightarrow \frac{\partial x(t)}{\partial \theta_{k,l}} = e^{a_k(t-\tau_k)} \left[\frac{b_k \beta_l}{a_k} \left[1 - e^{-a_k(t-\tau_k)} \right] + \frac{\partial x(\tau_k^+)}{\partial \theta_{k,l}} \right]$$

CYBER-PHYSICAL SYSTEMS