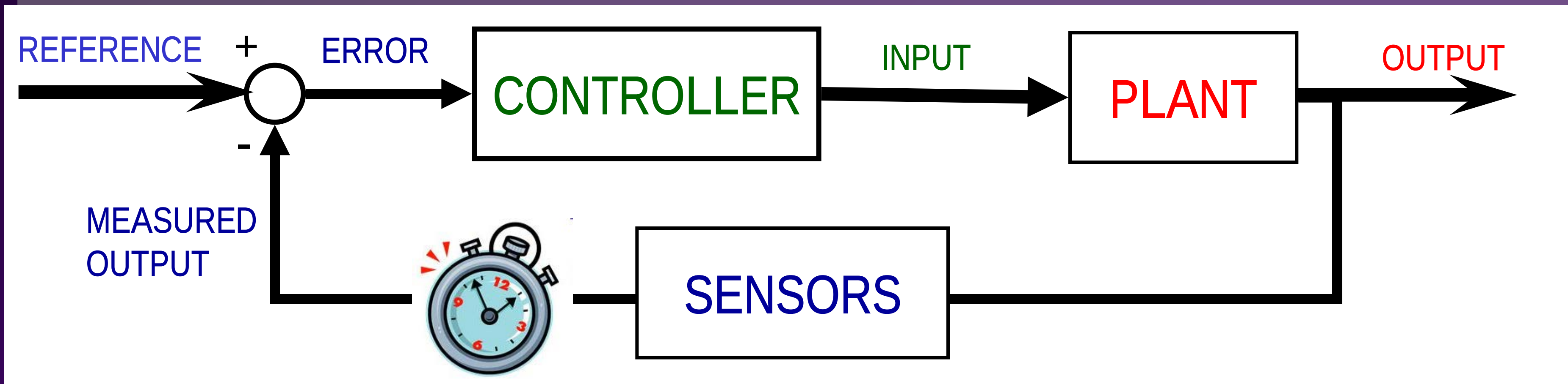
EVENT-DRIVEN AND DATA-DRIVEN CONTROL AND OPTIMIZATION IN CYBER-PHYSICAL SYSTEMS

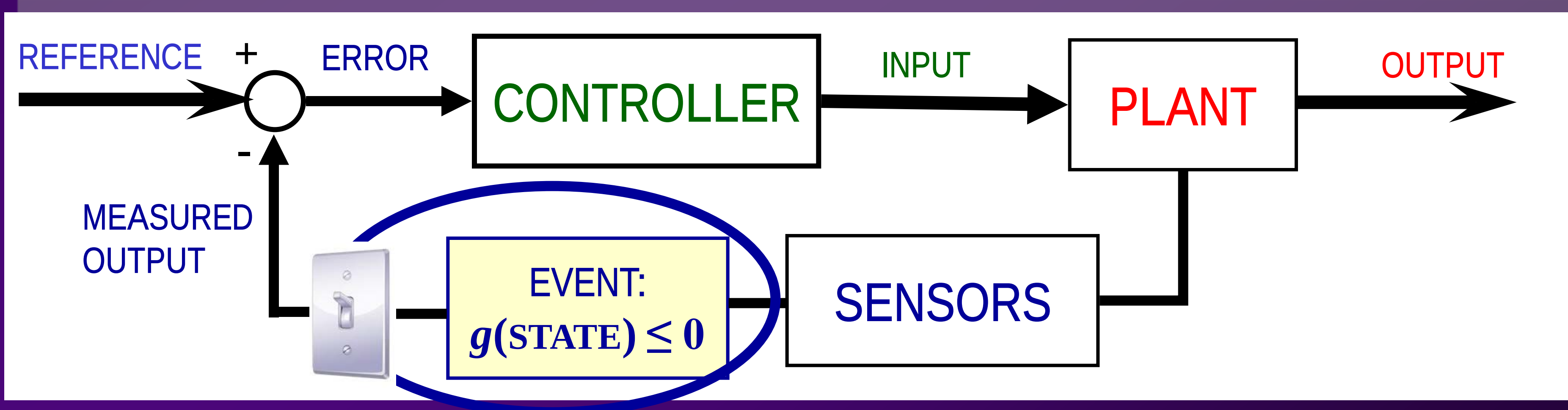
C. G. Cassandras

Division of Systems Engineering
Dept. of Electrical and Computer Engineering
Center for Information and Systems Engineering
Boston University
https://christosgcassandras.org

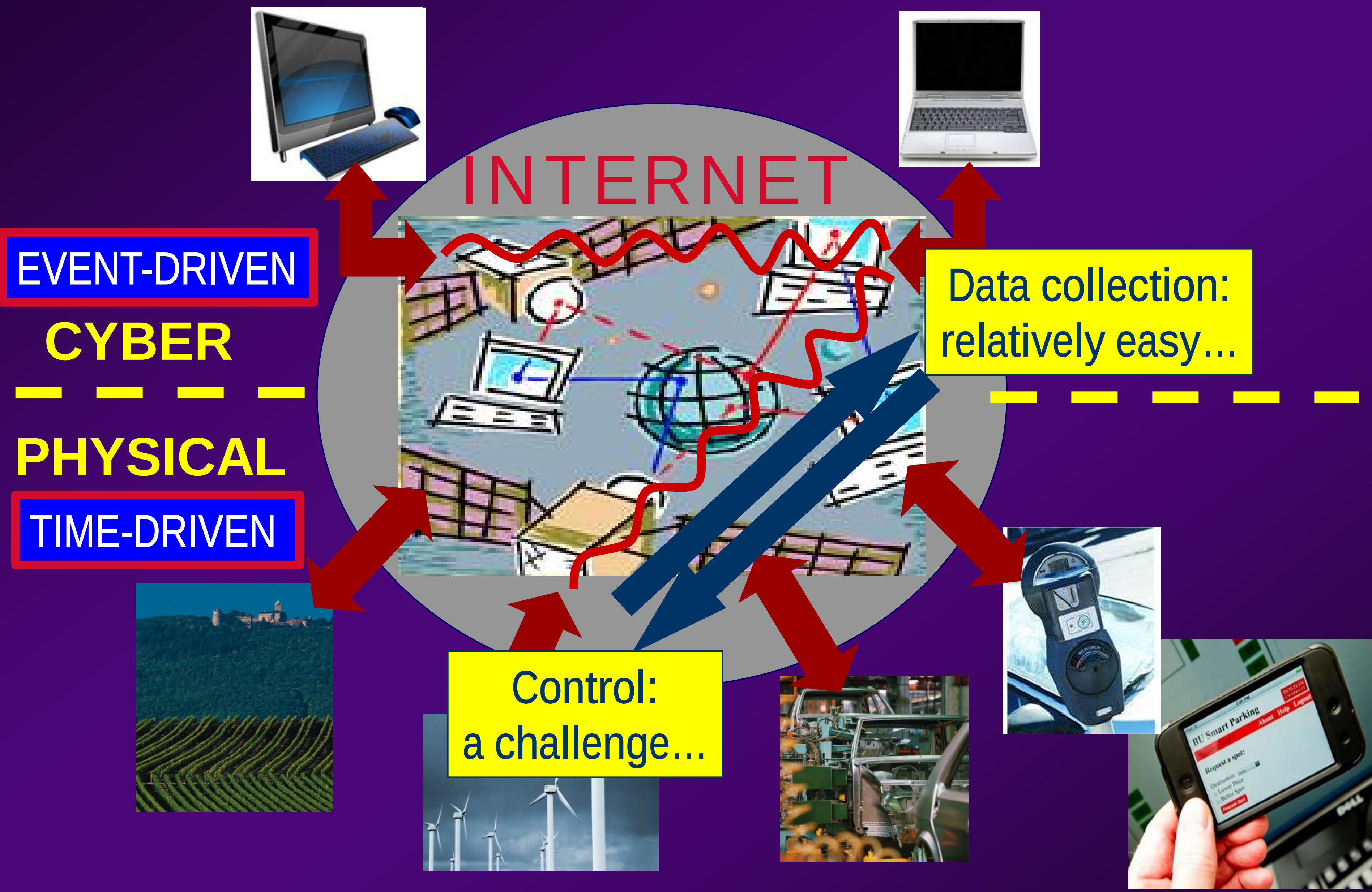
TIME-DRIVEN v EVENT-DRIVEN CONTROL



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



CYBER-PHYSICAL SYSTEMS



OUTLINE

- Why EVENT-DRIVEN Control and Optimization ?
- EVENT-DRIVEN Control in Distributed Multi-Agent Systems
- A General Optimization Framework for Multi-Agent Systems
- EVENT-DRIVEN + DATA-DRIVEN Control and Optimization:

  the IPA (*Infinitesimal Perturbation Analysis*) Calculus

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

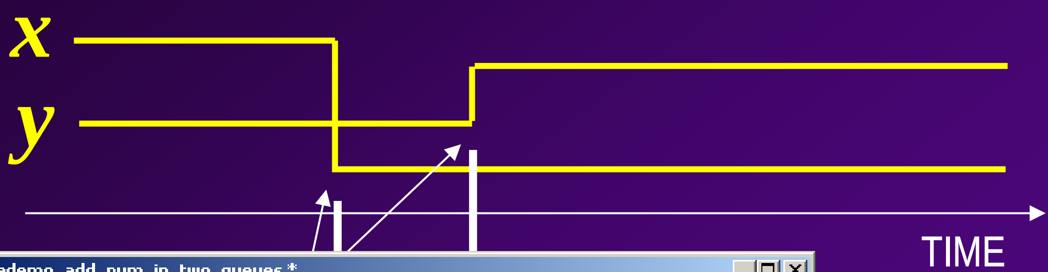
- Many systems are stochastic
  → actions needed in response to random events

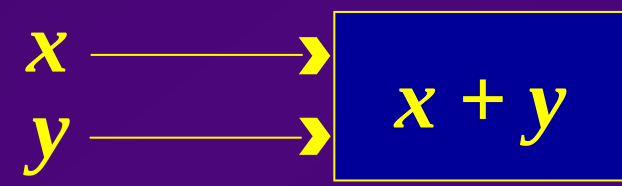
- Event-driven methods provide significant advantages in computation and estimation quality

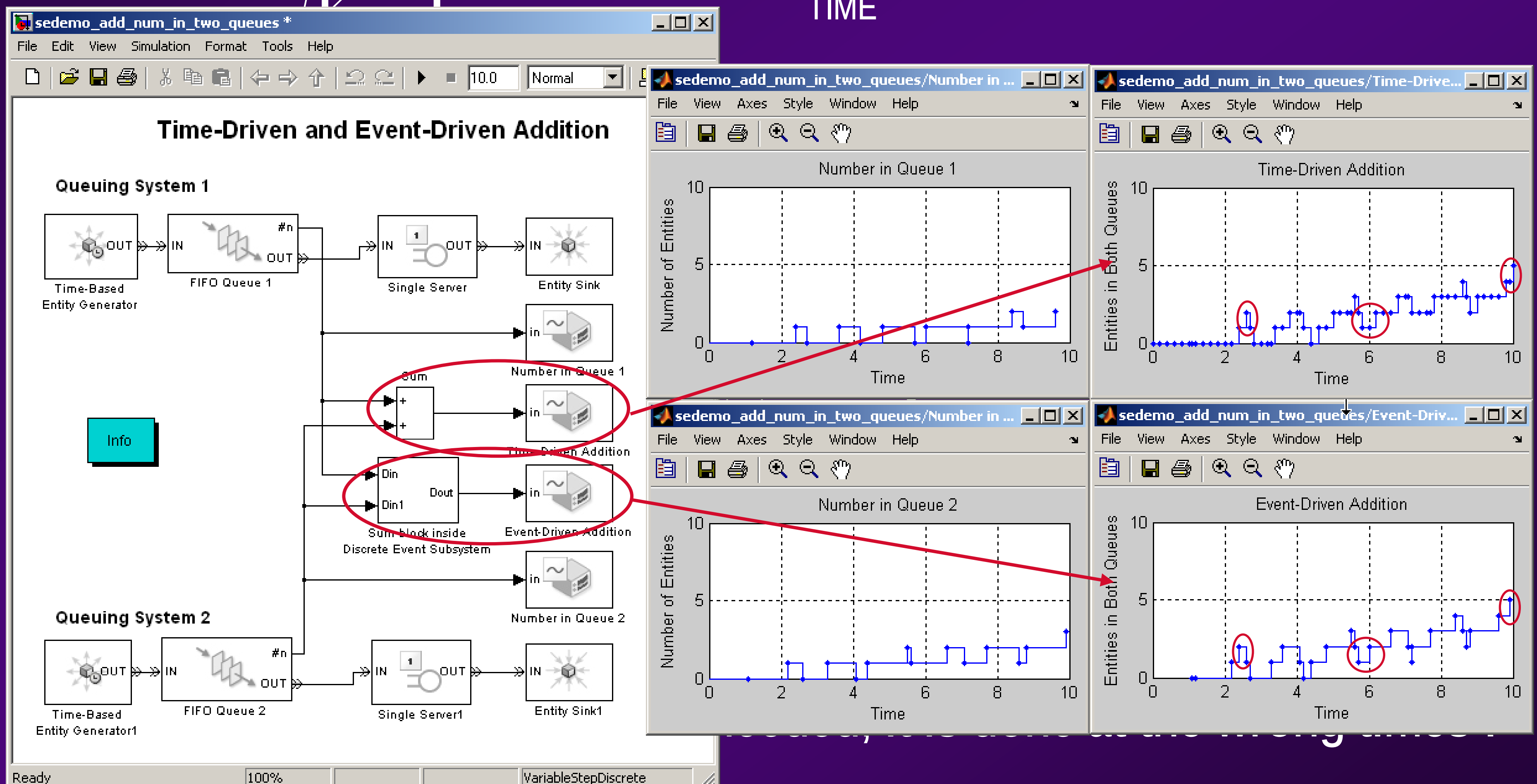
- System performance is often more sensitive to event-driven components than to time-driven components

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

TIME-DRIVEN (SYNCHRONOUS) v EVENT-DRIVEN (ASYNCHRONOUS) COMPUTATION







SELECTED REFERENCES - EVENT-DRIVEN CONTROL, COMMUNICATION, ESTIMATION, OPTIMIZATION

- 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 DISTRIBUTED OPTIMIZATION

DISTRIBUTED COOPERATIVE OPTIMIZATION

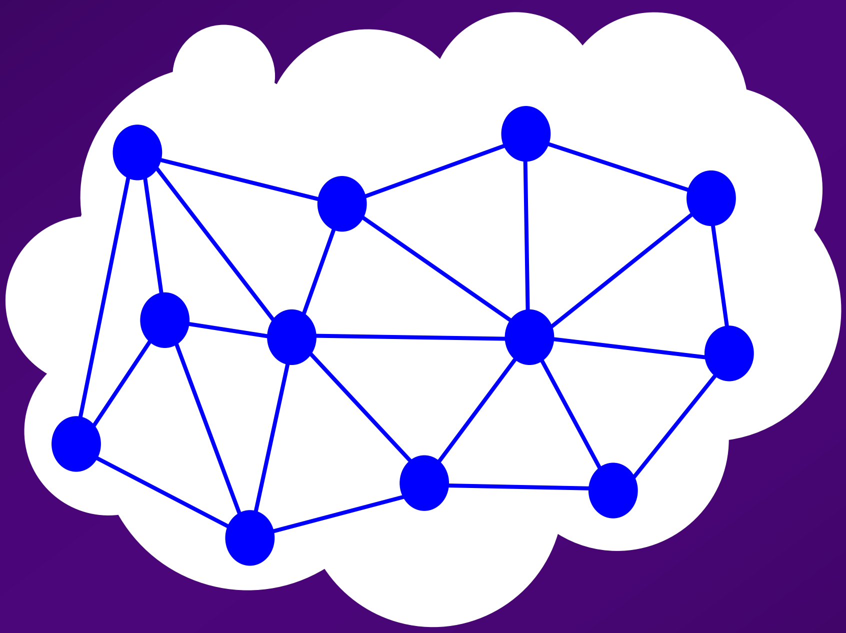
N system components
(processors, agents, vehicles, nodes),
one common objective:

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

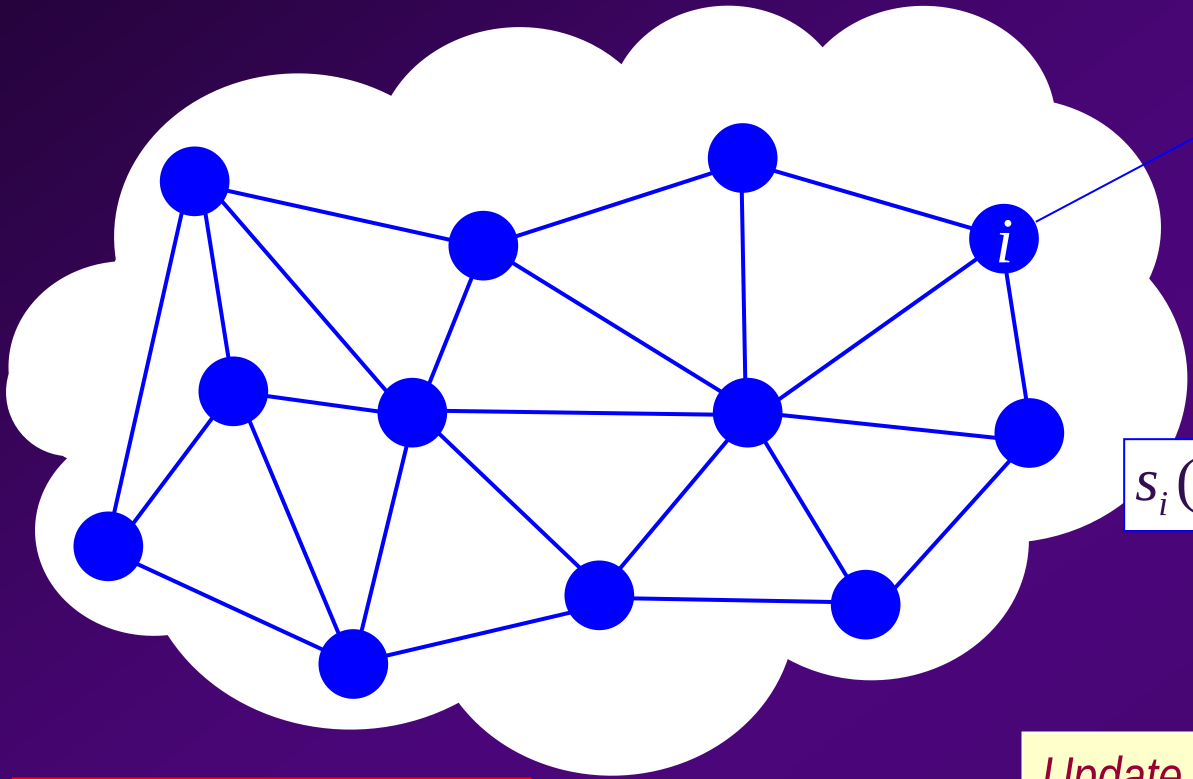
$$\min_{s_1} H(s_1,\ldots,s_N)$$
$$s.t.\ \text{constraints on } s_1$$

⋮

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



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

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

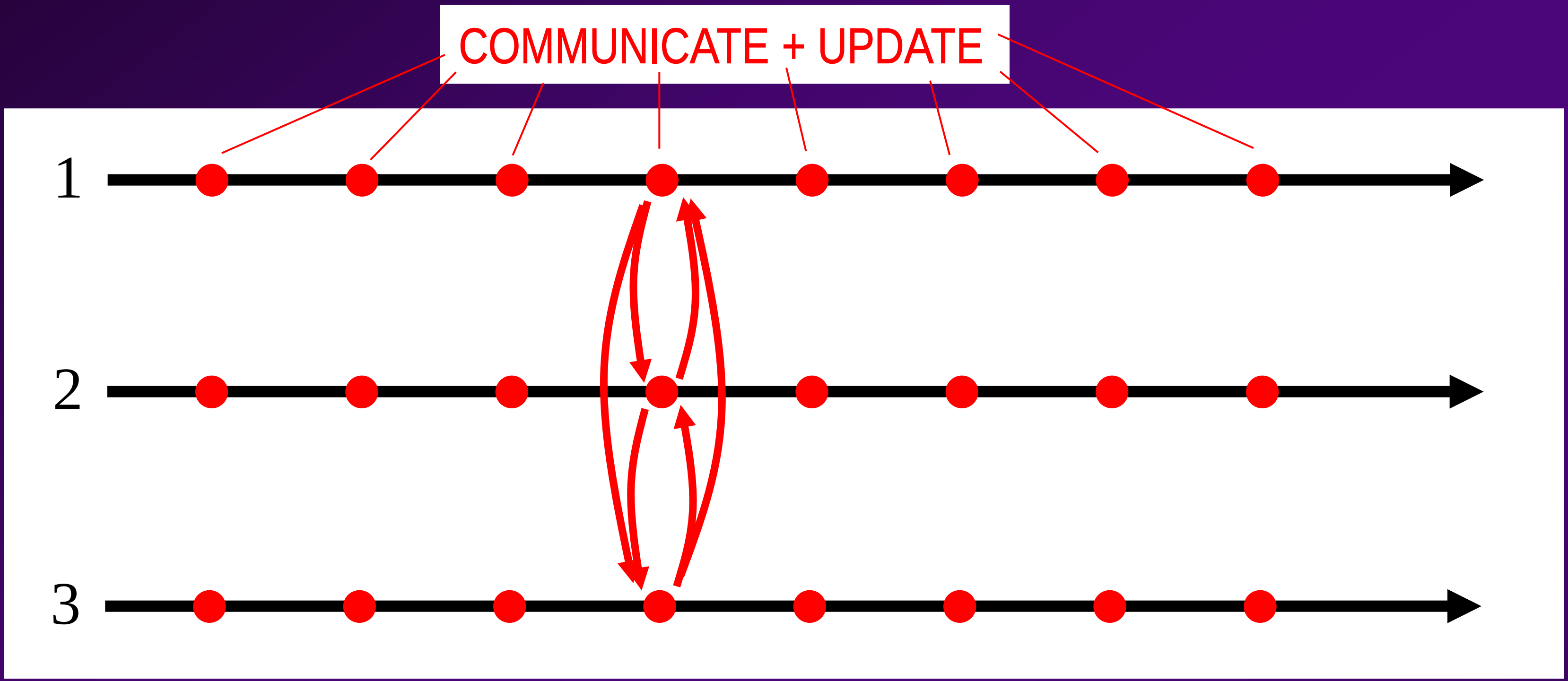
i requires knowledge of all $s_1,\ldots,s_N$

Inter-node communication

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

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

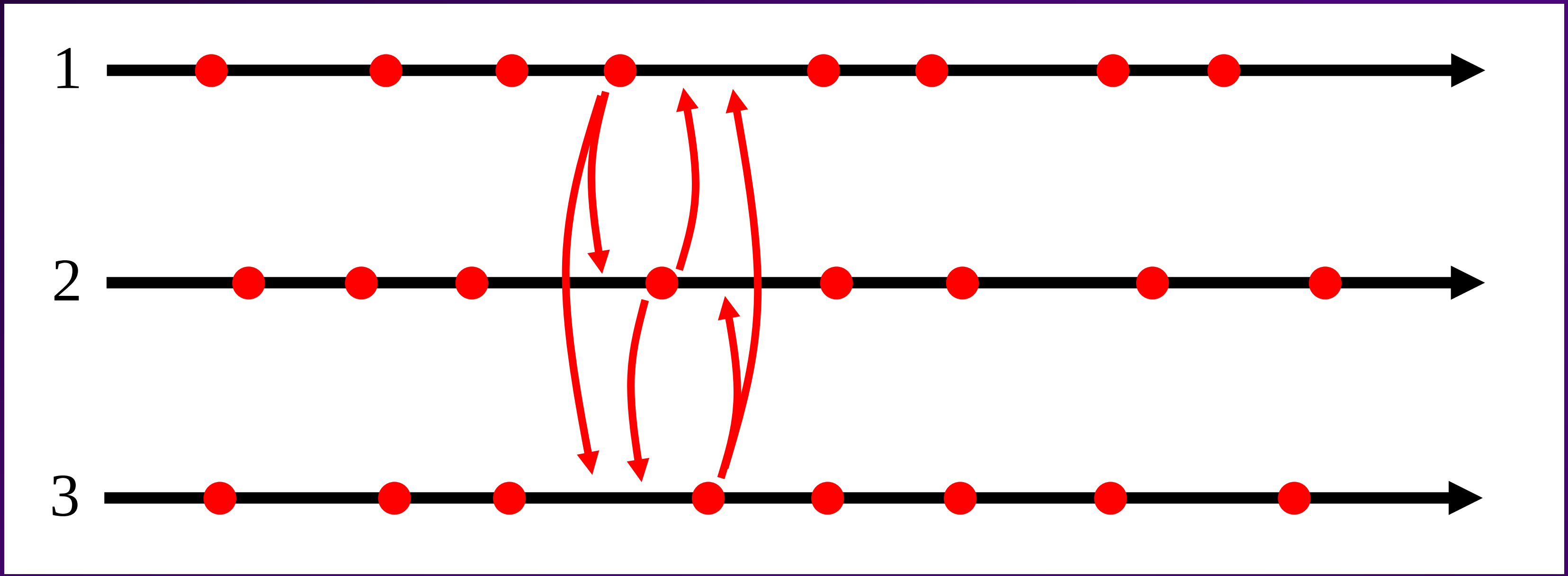
SYNCHRONIZED (TIME-DRIVEN) COOPERATION



Drawbacks:

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

ASYNCHRONOUS COOPERATION

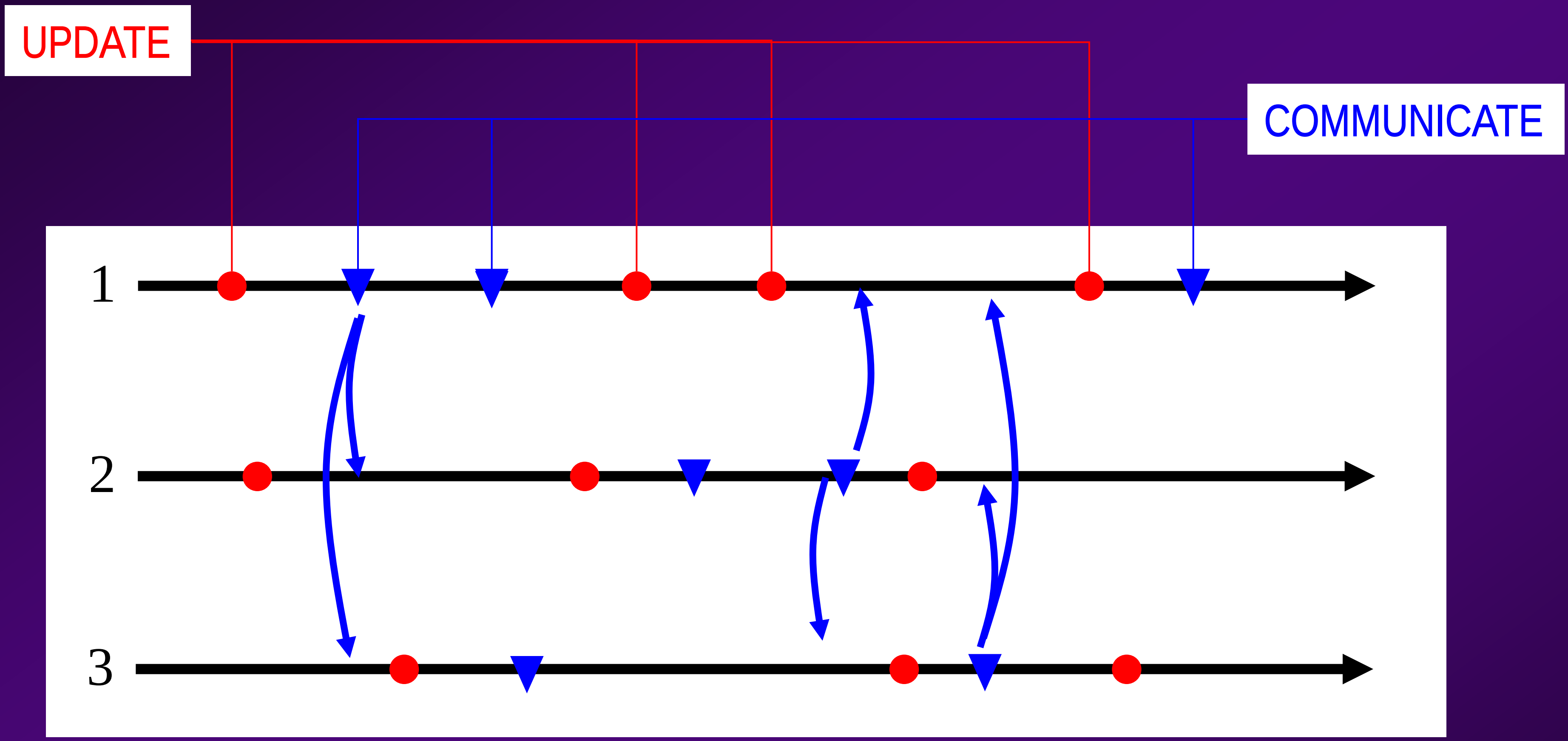


- Nodes not synchronized, delayed information used

$$\left.\begin{array}{l}\text{Update frequency for each node}\\ \text{is bounded}\\ \qquad + \\ \text{technical conditions}\end{array}\right\} \Rightarrow \begin{array}{l} s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k)) \\ \text{converges}\end{array}$$

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?

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, $x_i^j(t)$

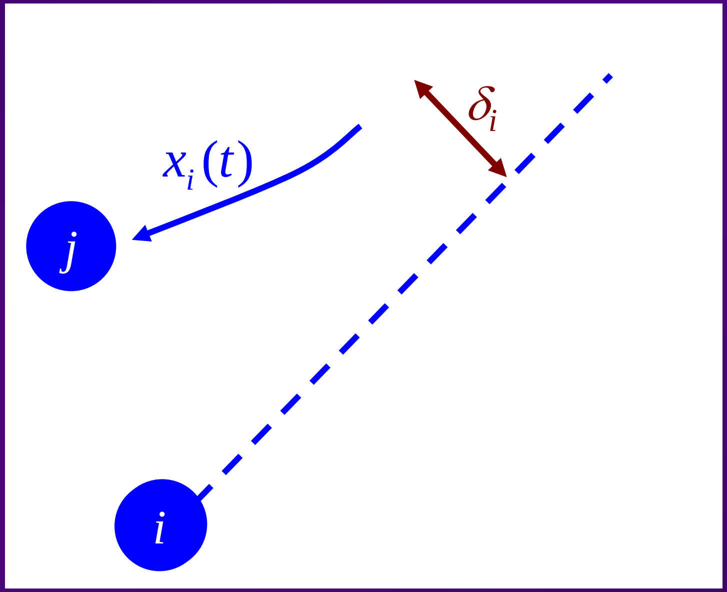
... and evaluates an ERROR FUNCTION $g\left(x_i(t), x_i^j(t)\right)$

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

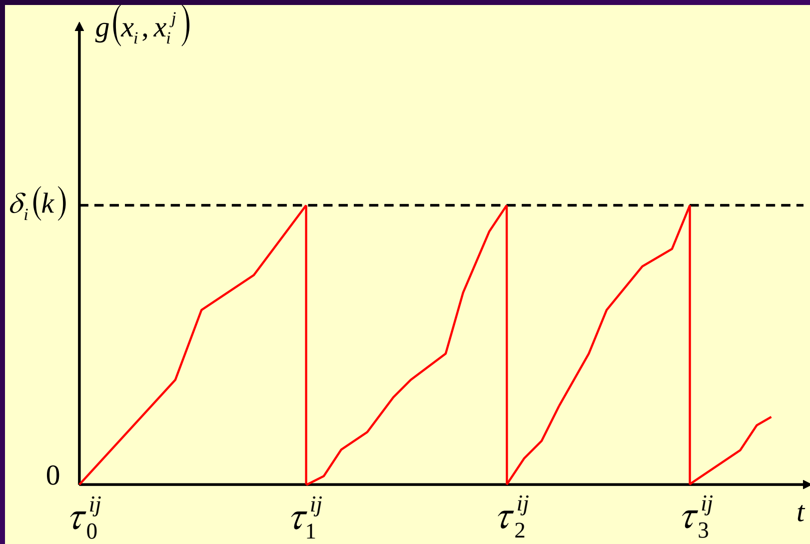
$$\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$$

Zhong and Cassandras, IEEE TAC, 2010

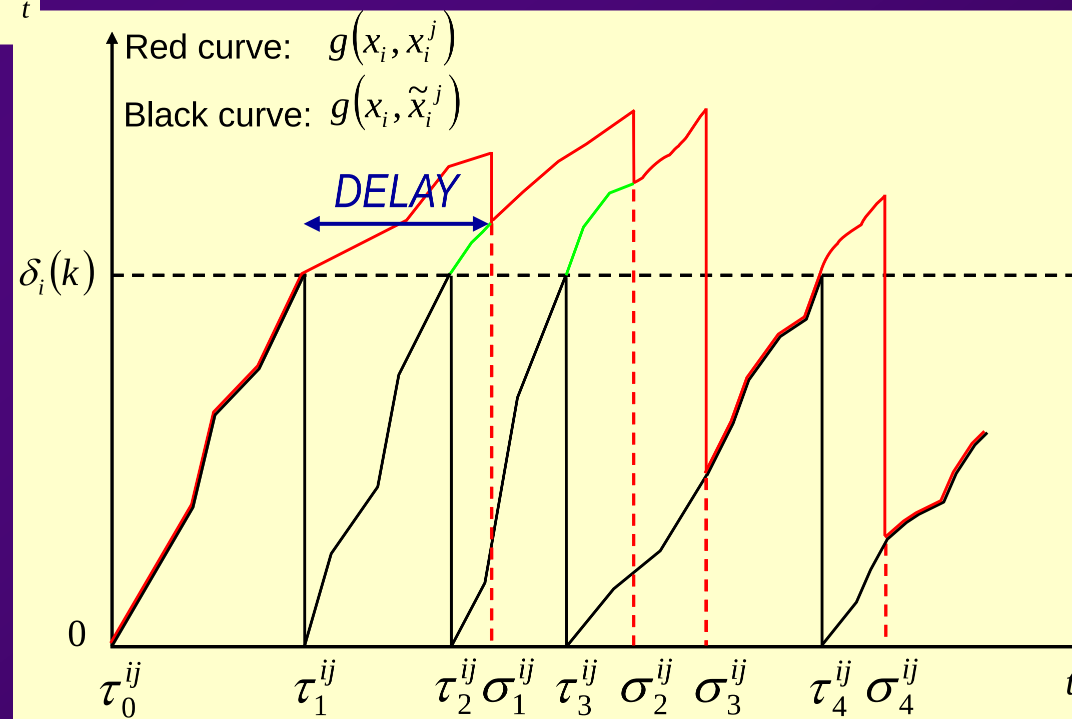
INTERPRETATION:

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 $\boldsymbol{D}$ state update events can occur between a node sending a message and all destination nodes receiving this message.

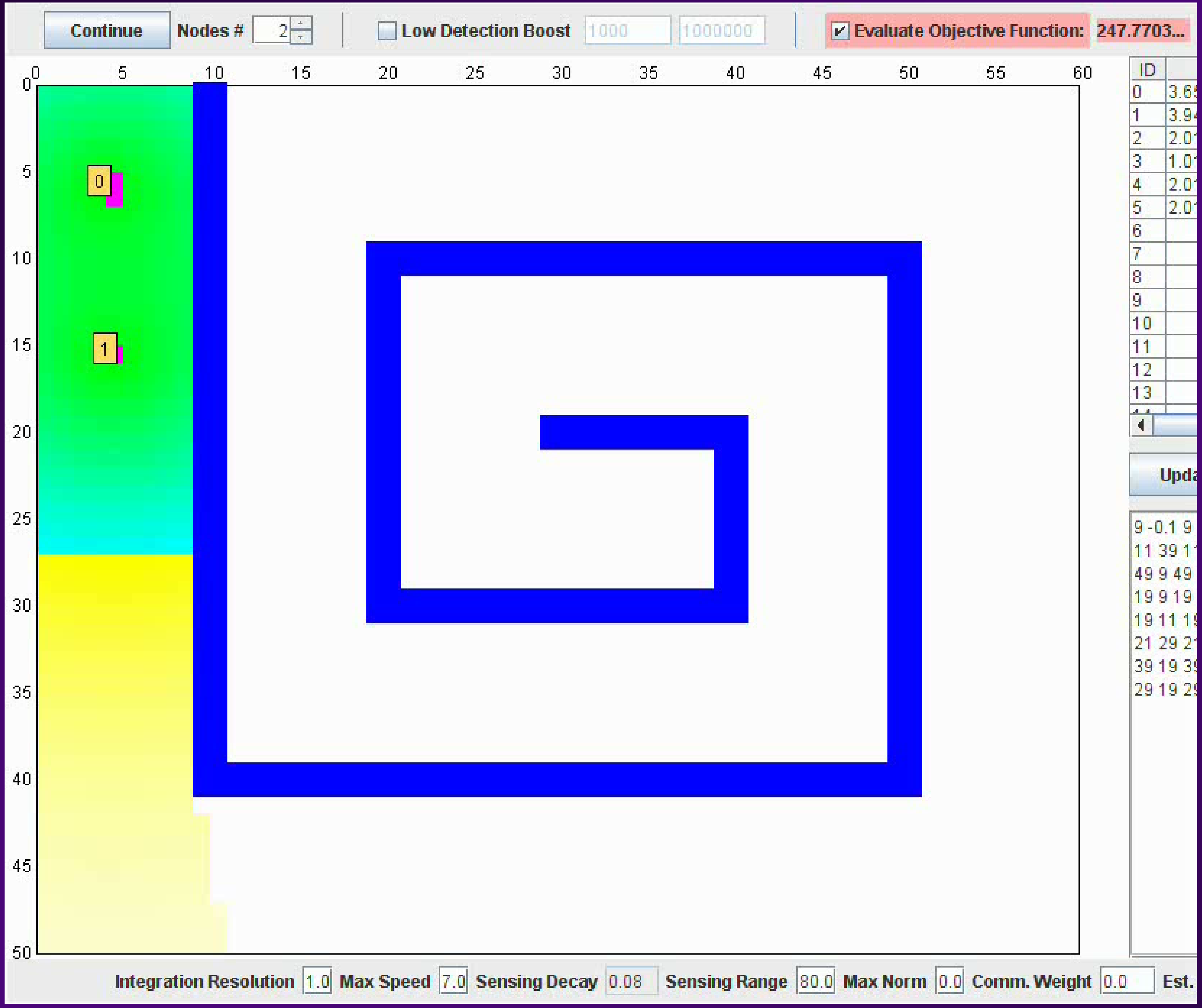
THEOREM: Under certain conditions, there exist positive constants α and K_δ such that

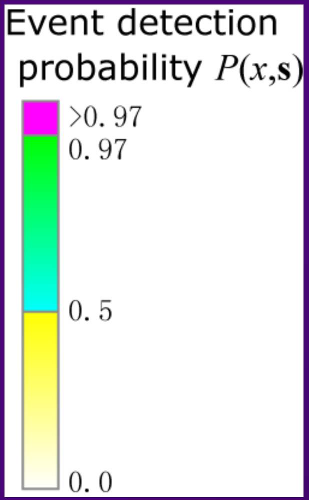
$$\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$$

NOTE: The requirements on α and K_δ depend on $\boldsymbol{D}$ and they are tighter.

Zhong and Cassandras, IEEE TAC, 2010

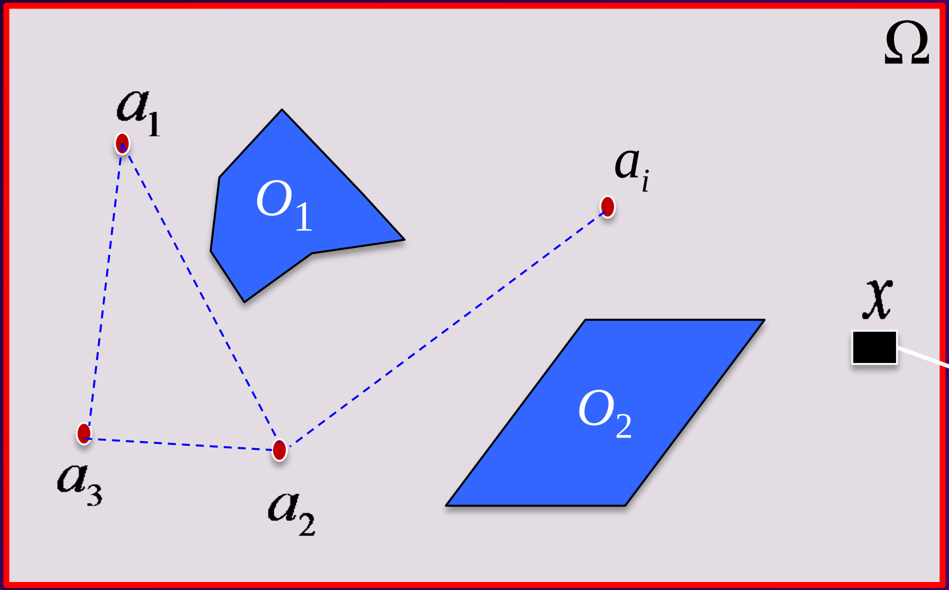
OPTIMAL COVERAGE IN A MAZE





A GENERAL OPTIMIZATION FRAMEWORK FOR MULTI-AGENT SYSTEMS

NETWORKED MULTI-AGENT OPTIMIZATION: PROBLEM 1: *PARAMETRIC OPTIMIZATION*



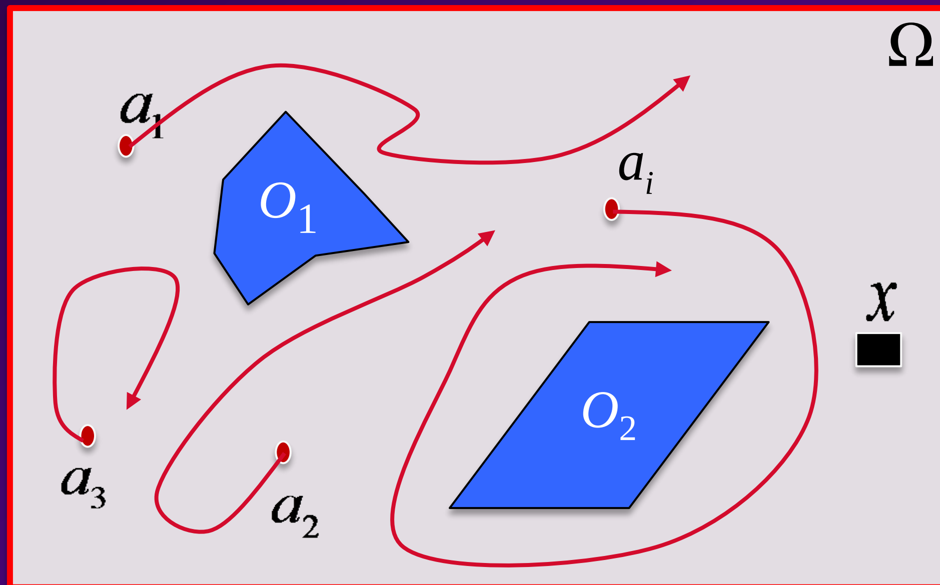
- s_i: agent state, $i = 1,\ldots, N$
  $\mathbf{s}=[s_1, \ldots , s_N]$
- O_j: obstacle (constraint)
- $R(x)$: property of point x
- $P(x, \mathbf{s})$: reward function

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

$$s_i \in F \subseteq \Omega, i = 1,\cdots, N$$

GOAL: Find the best state vector $s=[s_1, \ldots , s_N]$ so that agents achieve a maximal reward from interacting with the mission space

NETWORKED MULTI-AGENT OPTIMIZATION: PROBLEM 2: *DYNAMIC OPTIMIZATION*



$$\max_{\mathbf{u}(t)} J = \int_0^T \int_\Omega P(x, \mathbf{s}(u(t))) R(x) dx dt$$

May also have dyn cs

$s_i(t) \in F \subseteq \Omega, \ i = 1, \cdots, N$ $\quad \dot{s}_i = f_i($

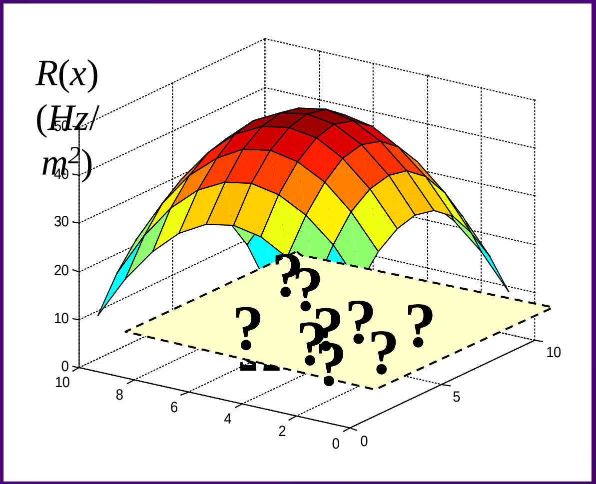
NOTE: Dynamics of both agents and environment are invcorporated !

GOAL: Find the best state trajectories $s_i(t)$, $0 \leq t \leq T$ so that agents achieve a maximal reward from interacting with the mission space

PROBLEMS THAT FIT THIS FRAMEWORK

COVERAGE

Deploy sensors to maximize "event" detection probability - unknown event locations



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

Joint event detection probability:

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

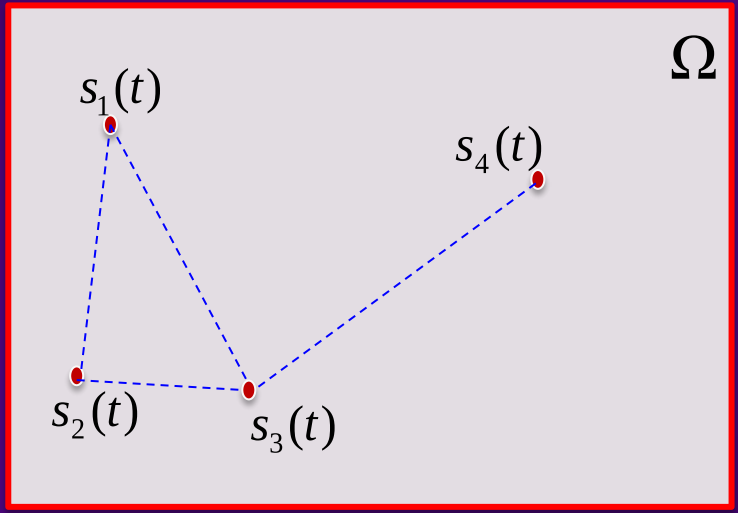
Event sensing probability

Event density: Prior estimate of event occurrence frequency

CONSENSUS

$$\dot{s}_i(t) = \sum_{j \in N_i} s_j(t) - s_i(t)$$

$$s_1 = \cdots = s_N$$



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

$$R(x) = \sum_{i=1}^{N} \mathbf{1}(x - s_i)$$

Only x that matter are agents

$$\max_{\mathbf{s}} H(\mathbf{s}) = -\sum^{N} P(s_i, \mathbf{s})$$

$$P(s_i, s) = \frac{1}{2} \sum_{j \in N_i} p_i(s_j, s_i)$$

$$p_i(s_j, s_i) = \begin{cases} \left\| s_j - s_i \right\|^2 & j \in N_i,\ j > i \\ 0 & \text{otherwise}_i \end{cases}$$

NOTE: Convex optimization, relatively easy problem

PERSISTENT MONITORING

GOAL: Find the best state trajectories $s_i(t)$, $0 \leq t \leq T$ so that agents achieve a maximal reward from interacting with the mission space

ENVIRONMENT MODEL

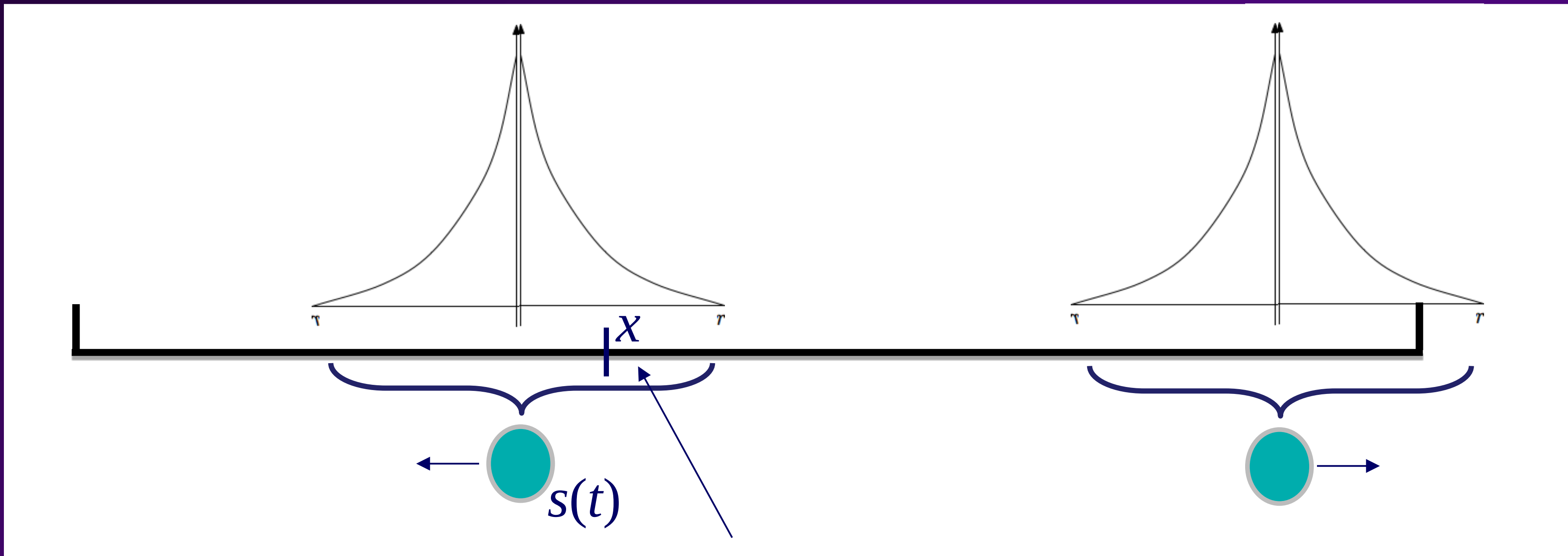
$$\max_{\mathbf{u}(t)} J = \int_0^T \int_\Omega P(x, \mathbf{s}(u(t))) R(x) dx dt$$

SENSING MODEL
(how agents interact with environment)

AGENT MODEL

$$\dot{s}_i = f_i(s_i, u_i, t), \quad i = 1, \cdots, N$$

PERSISTENT MONITORING



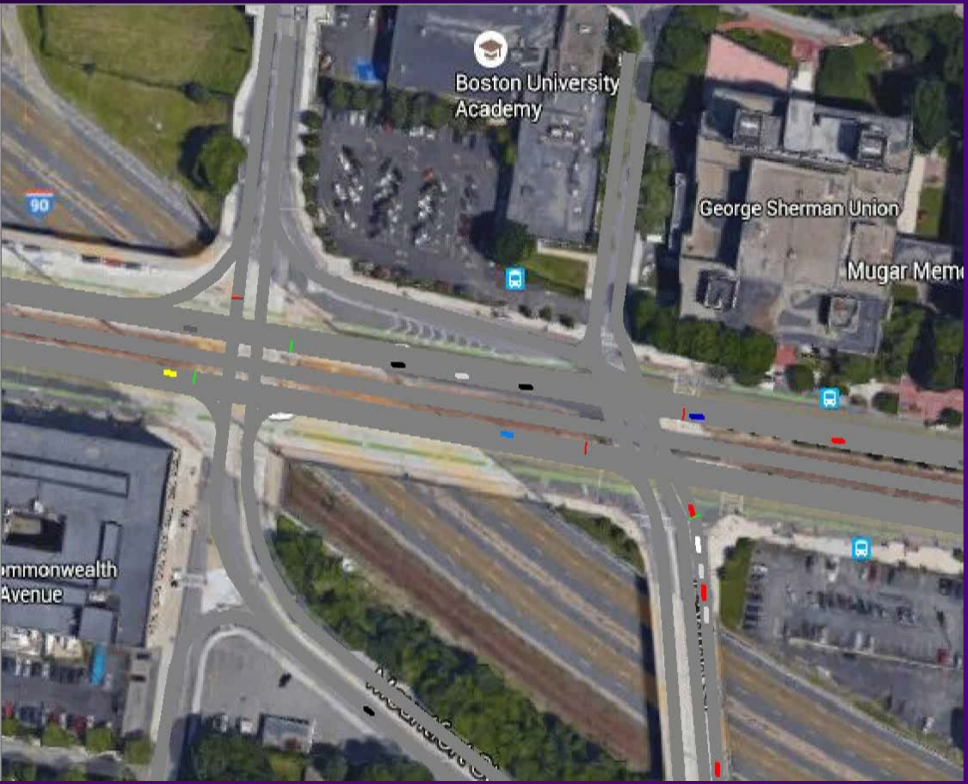
ENVIRONMENT MODEL: Associate to x *Uncertainty Function* $R(x,t)$

$$\dot{R}(x,t) = \begin{cases} 0 & \text{if } R(x,t)=0,\ A(x) < Bp(x,s(t)) \\ A(x) - Bp(x,s(t)) & \text{otherwise} \end{cases}$$

NOTE: Could be stochastic !

THE INTERNET OF CARS…

With traffic lights (non-Cooperative)



No traffic lights: decentralized control of CAVs (Cooperative)

One of the worst-designed double intersections ever…
(BU Bridge – Commonwealth Ave, Boston)

Malikopoulos, Cassandras, Zhang et al, Automatica, 2018

Zhang et al, Proc. of IEEE, 2018

RELATED WORK

COVERAGE AND FORMATION CONTROL

Choset 2001, Leonard and Olshevsky 2013, Tron et al 2014, Egerstedt and Hu 2001

Cortes et al 2004 Zhong and Cassandras 2010, Sun and Cassandras 2016

CONSENSUS

Jadbabaie et al, 2003, Olfati-Saber and Murray, 2004, Ren and Beard 2005

Nedich et al, 2010

SAMPLING AND TRACKING

Leonard and Zhang 2010, Ashley and Andersson 2016

PERSISTENT MONITORING

Smith et al, 2011, Michael et al, 2011, Lan and Schwager, 2014

Cassandras et al, 2013, Yu et al, 2017

NETWORKED MULTI-AGENT OPTIMIZATION– CHALLENGES

1. SCALABILITY
2. AUTONOMY

→ Distributed Algorithms (Decentralization)

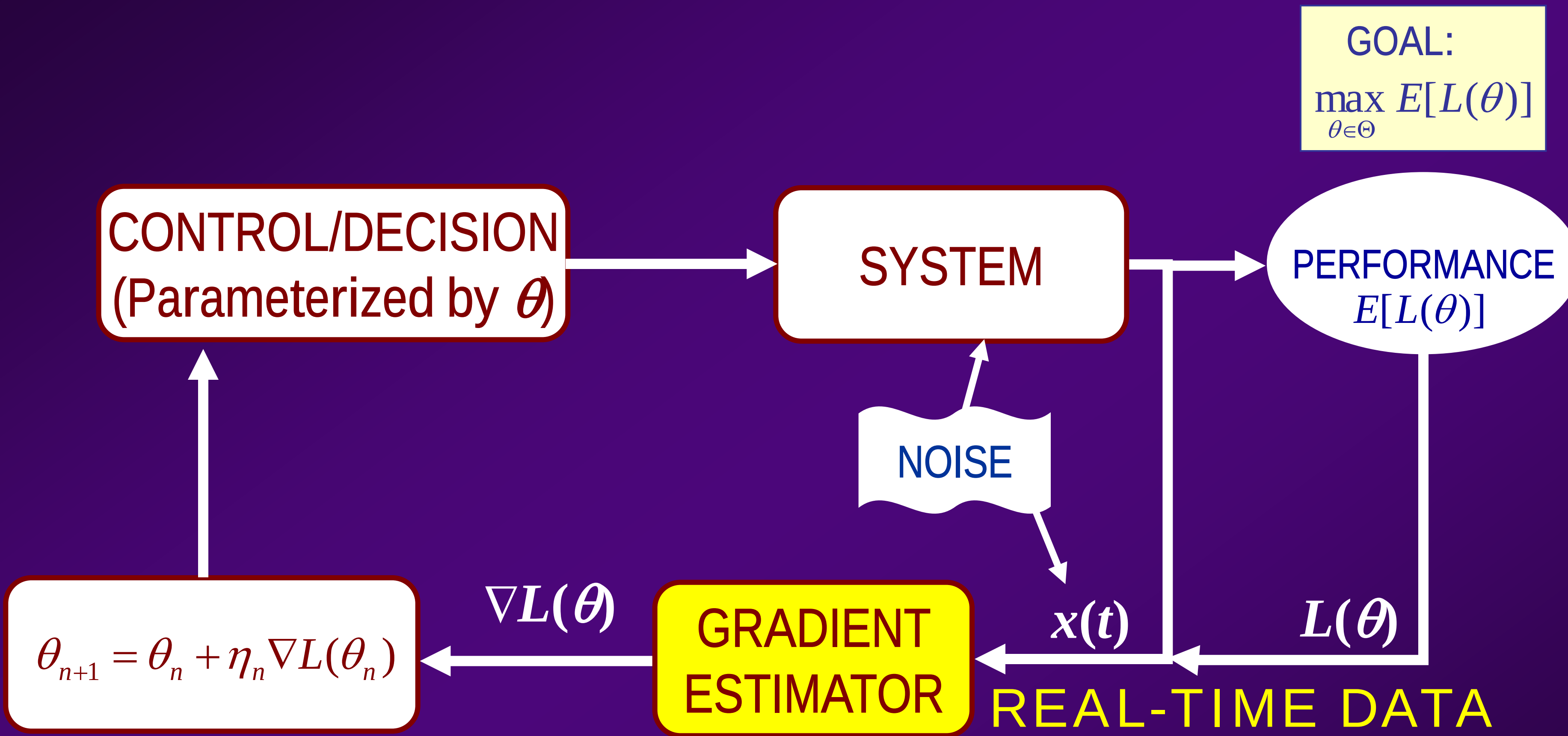
3. COMMUNICATION → Event-driven (asynchronous) Algorithms

4. NON-CONVEXITY → Global optimality, escape local optima

5. EXLOIT DATA → Data-Driven Algorithms

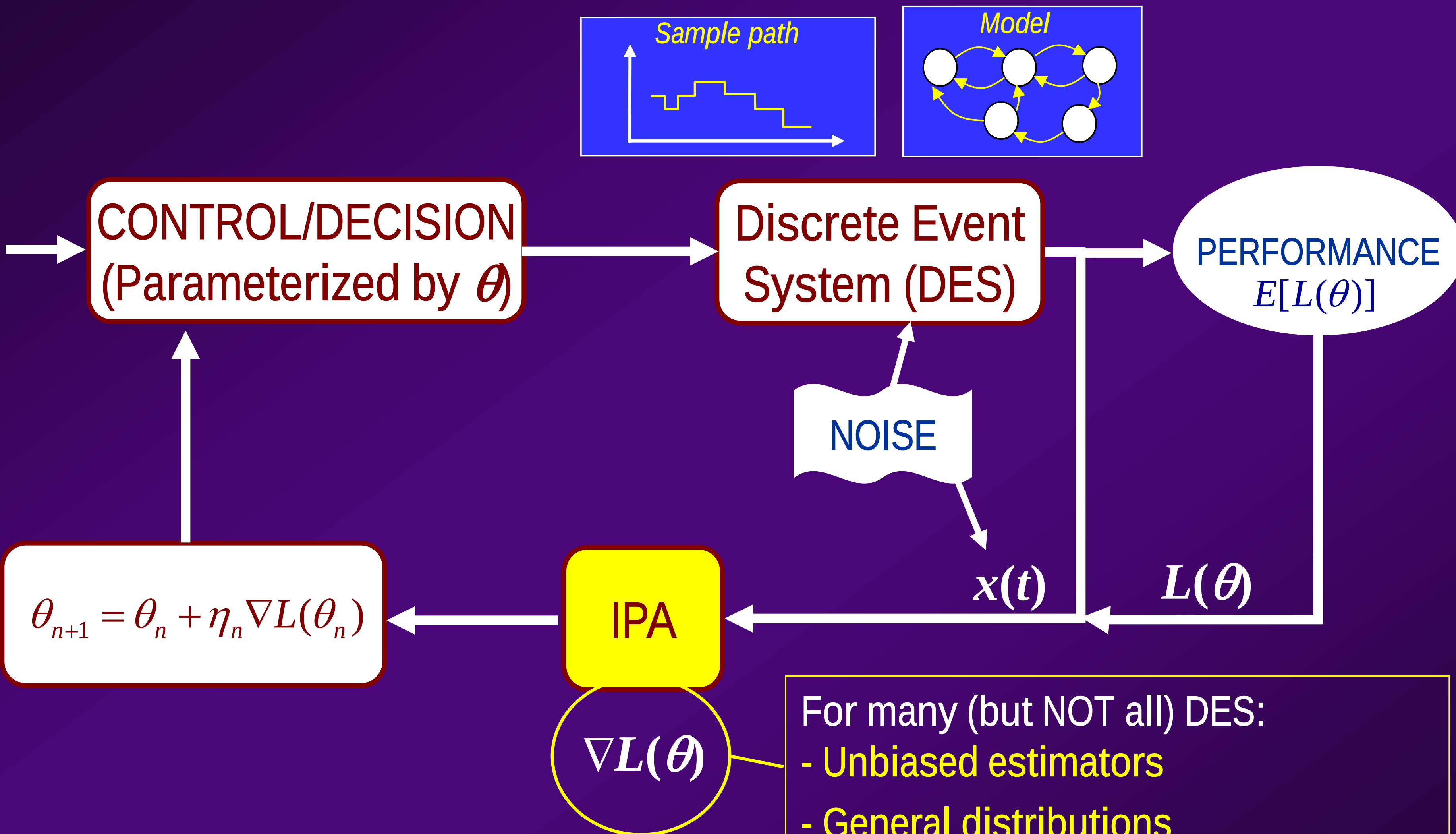
EVENT-DRIVEN
+ DATA-DRIVEN
OPTIMIZATION

DATA-DRIVEN 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)

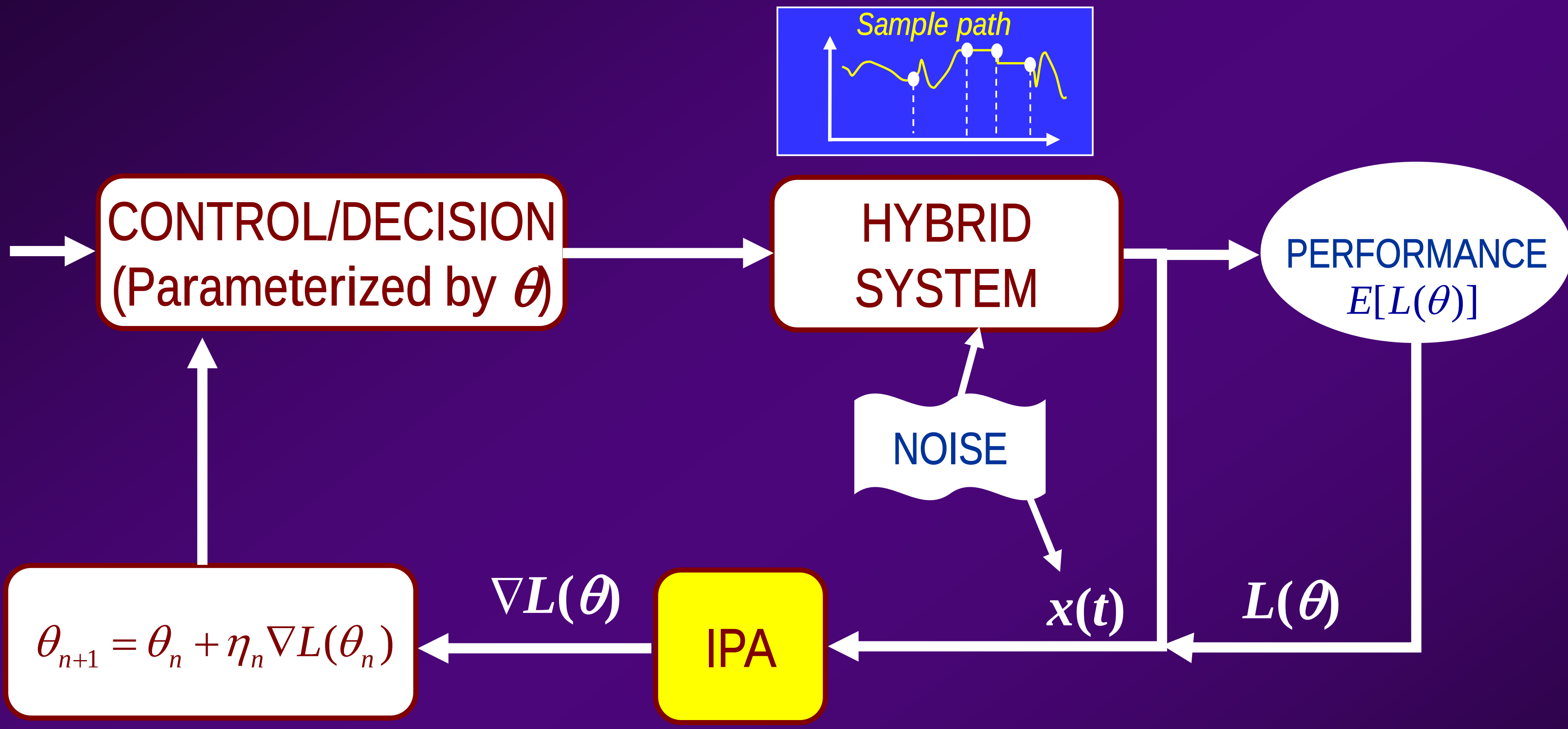


For many (but NOT all) DES:

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

Ho and Cao, 1991; Glasserman, 1991; Cassandras, 1993; Cassandras and Lafortune, 2008

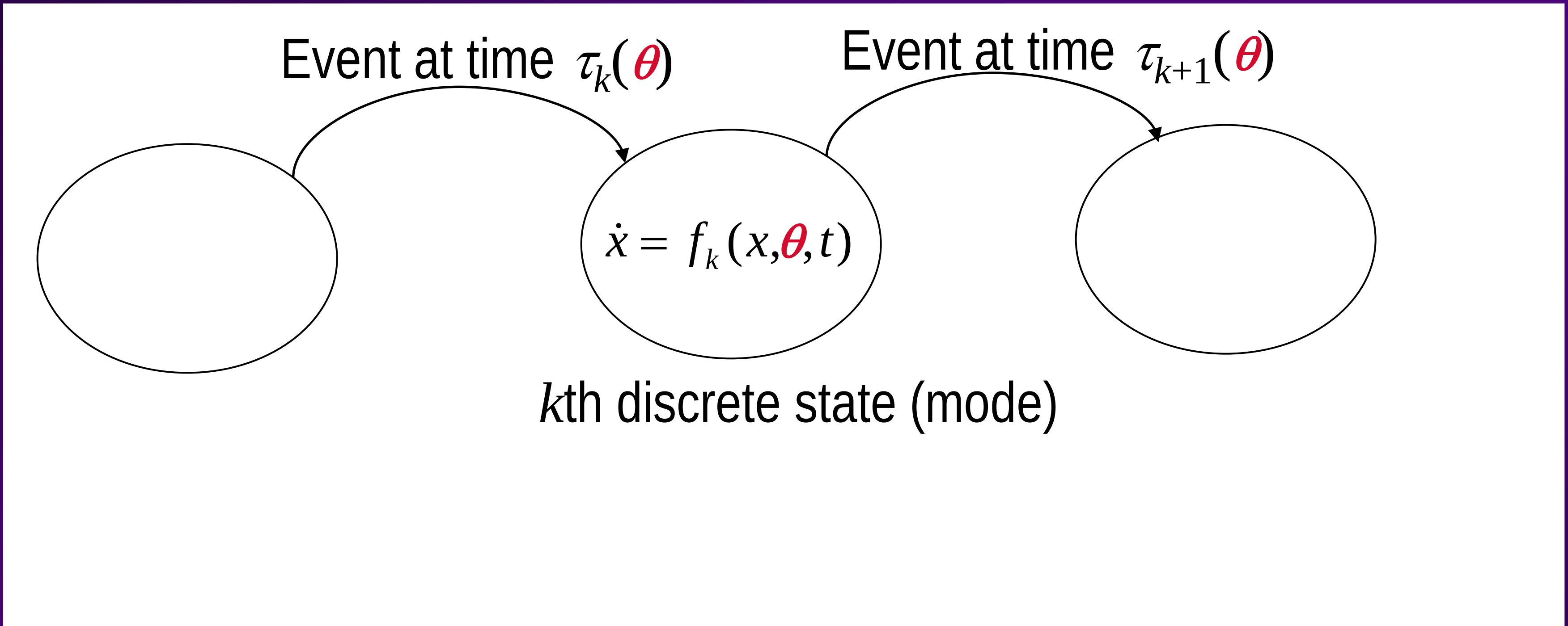
REAL-TIME STOCHASTIC OPTIMIZATION:
HYBRID SYSTEMS, CYBER-PHYSICAL SYSTEMS



A general framework for an IPA theory in Hybrid Systems?

THE IPA CALCULUS

STOCHASTIC HYBRID AUTOMATA



θ : control parameter, $\theta \in \Theta$ (system design parameter, parameter of an input process, or parameter that characterizes a control policy)

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

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

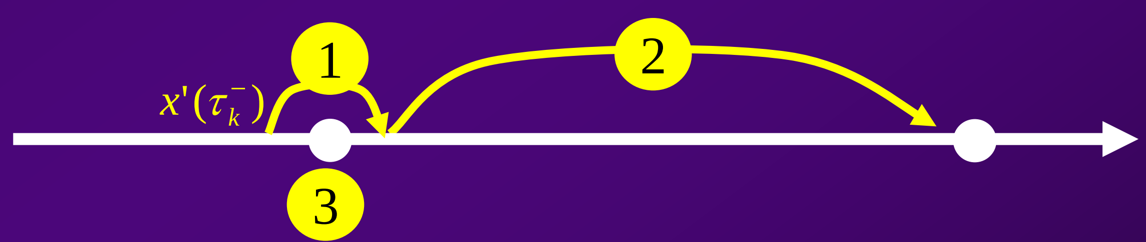
2. $x'(\tau_k) = e^{\int_{\tau_{k-1}}^{\tau_k} \frac{\partial f_k(u)}{\partial x} du} \left[\int_{\tau_{k-1}}^{\tau_k} \frac{\partial f_k(v)}{\partial \theta} e^{-\int_{\tau_{k-1}}^{v} \frac{\partial f_k(u)}{\partial x} du} dv + x'(\tau_{k-1}^+) \right]$

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:

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

IPA PROPERTIES: *ROBUSTNESS*

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 $[\tau_k(\theta),\ \tau_{k+1}(\theta)]$:

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

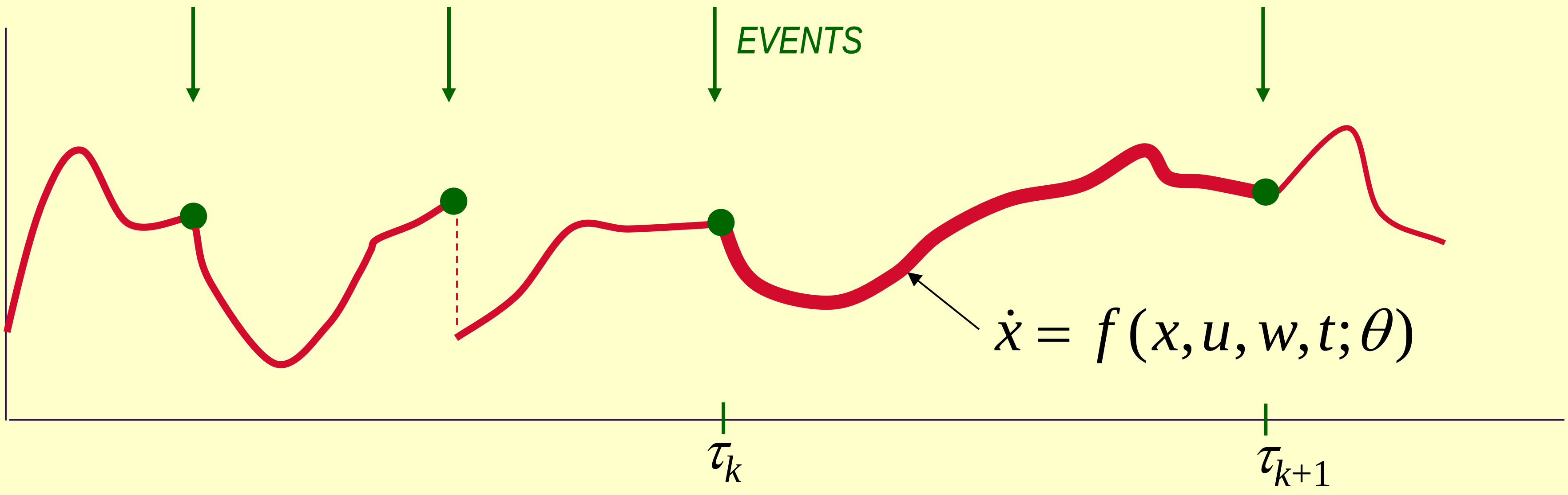
Yao and Cassandras, J. DEDS, 2011

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

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: *SCALABILITY*

IPA estimators are EVENT-DRIVEN
$\Rightarrow$ IPA scales with the EVENT SET, not the STATE SPACE !
$\Rightarrow$ no time discretization needed

As a complex system grows with the addition of more states, the number of EVENTS often remains unchanged or increases at a much lower rate.

EXAMPLE: *A queueing network may become very large, but the basic events used by IPA are still "arrival" and "departure" at different nodes*.

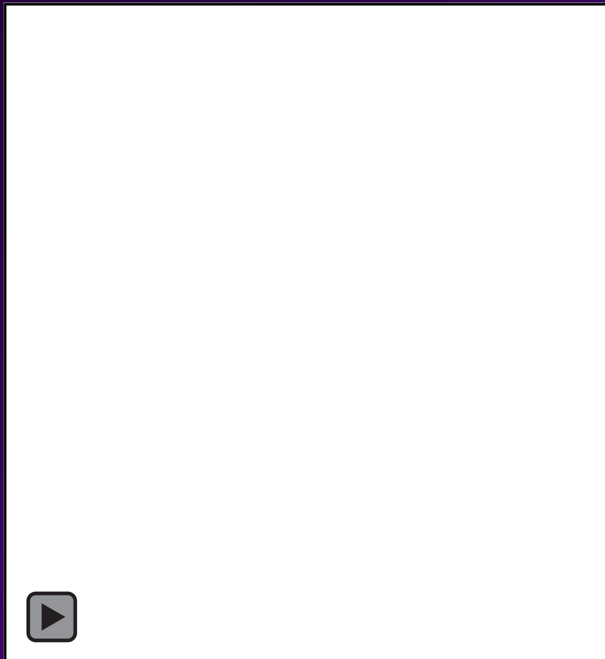
DECENTRALIZING CAN BE HARD

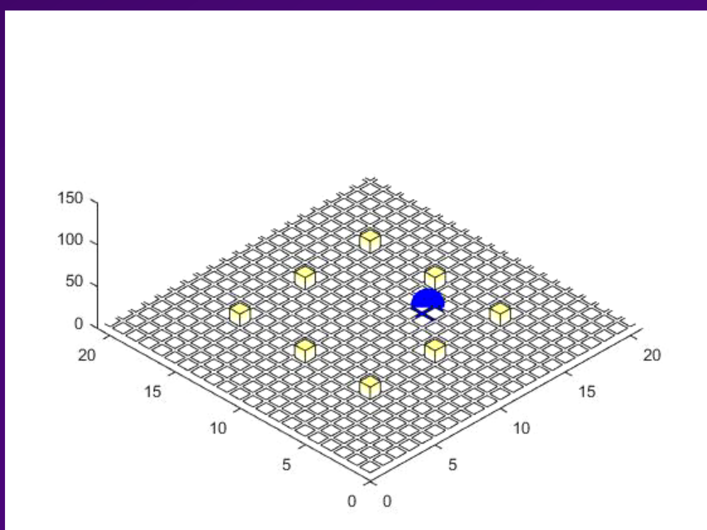
DECENTRALIZED SOLUTION = CENTRALIZED SOLUTION

(AGENTS ACTING USING ONLY LOCAL INFO.)

(no performance loss due to decentralization)

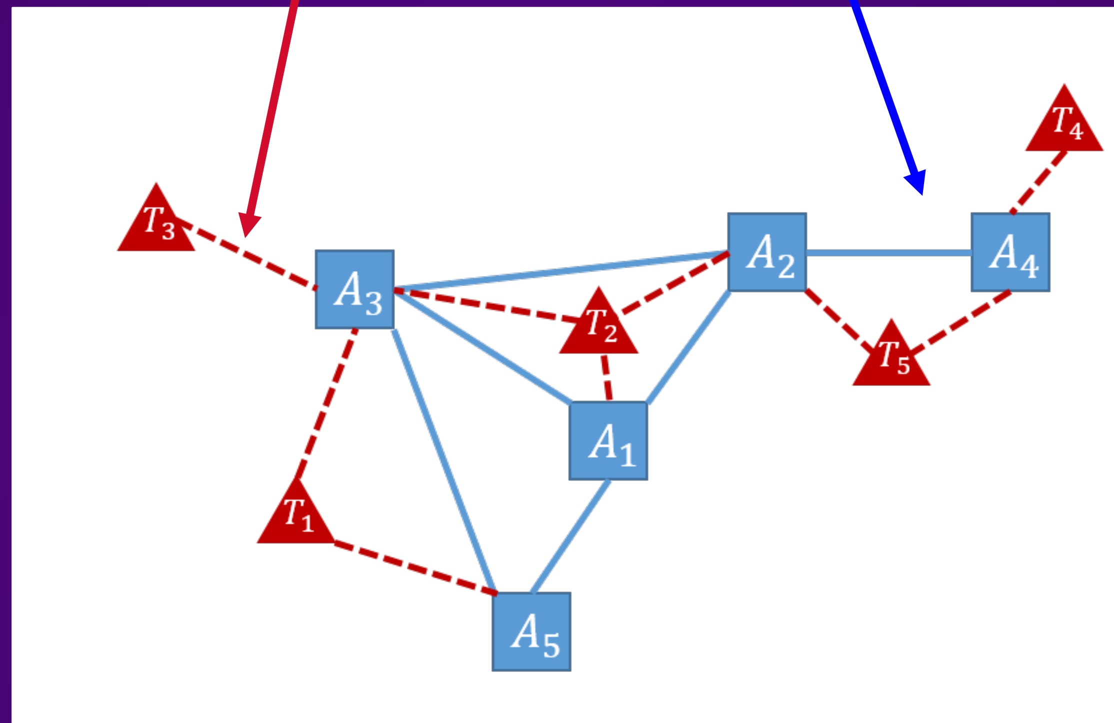
PERSISTENT MONITORING WITH KNOWN TARGETS





Agent-Target Interaction Network (time-varying)

Agent Network (time-varying)



Hard to decentralize in the presence of time-varying agent-environment interactions

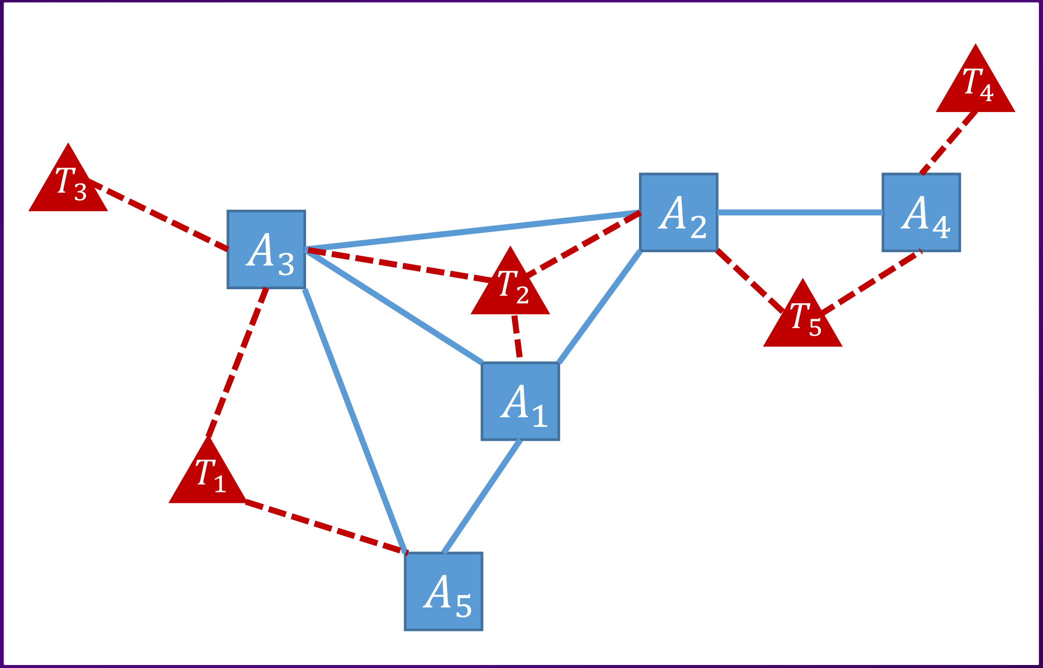
THREE TYPES OF NEIGHBORHOODS

The agent neighborhood of an agent (conventional)

The agent neighborhood of agent j is the set
$\mathcal{A}_j(t) = \{k : \|s_k(t) - s_j(t)\| \leq r_c,\ k \neq j,$
$k = 1, \ldots, N\}.$

The target neighborhood of an agent

The target neighborhood of agent j is the set
$\mathcal{T}_j(t) = \{i : |x_i - s_j(t)| \leq r_j,\ i = 1, \ldots, M\}.$



The agent neighborhood of an target

The agent neighborhood of target i is the set
$\mathcal{B}_i(t) = \{j : |s_j(t) - x_i| \leq r_j,\ j = 1, \ldots, N\}.$

“ALMOST DECENTRALIZATION” RESULT

- Show that optimal trajectories consist of *hybrid dynamics*:
  segments defined by observable *EVENTS*
  e.g., *agent enters target sensing range*, *agent leaves neighborhood*

- Develop *EVENT-DRIVEN* gradient-based algorithms using the *Infinitesimal Pertubation Analysis* (IPA) calculus:
  Each agent evaluates its IPA derivative

- Does an agent’s IPA derivative depend only on LOCAL events?

DECENTRALIZATION ⟺ EVENT OBSERVABILITY

THEOREM: Each agent’s IPA derivative depends only on LOCAL events except for one global event

Zhou et al, IEEE TAC 2018

NETWORKED MULTI-AGENT OPTIMIZATION– CHALLENGES

1. SCALABILITY
2. AUTONOMY

→ Distributed Algorithms

When are these possible?
How to design?

3. COMMUNICATION → Event-driven (asynchronous) Algorithms

How do *Event-Driven* algorithms perform compared to *Time-Driven* ones?

4. NON-CONVEXITY → Global optimality, escape local optima

Can this be done in a distributed manner? Is convergence guaranteed?

5. EXLOIT DATA → Data-Driven Algorithms

Solve stochastic optimization problems robust to modeling assumptions