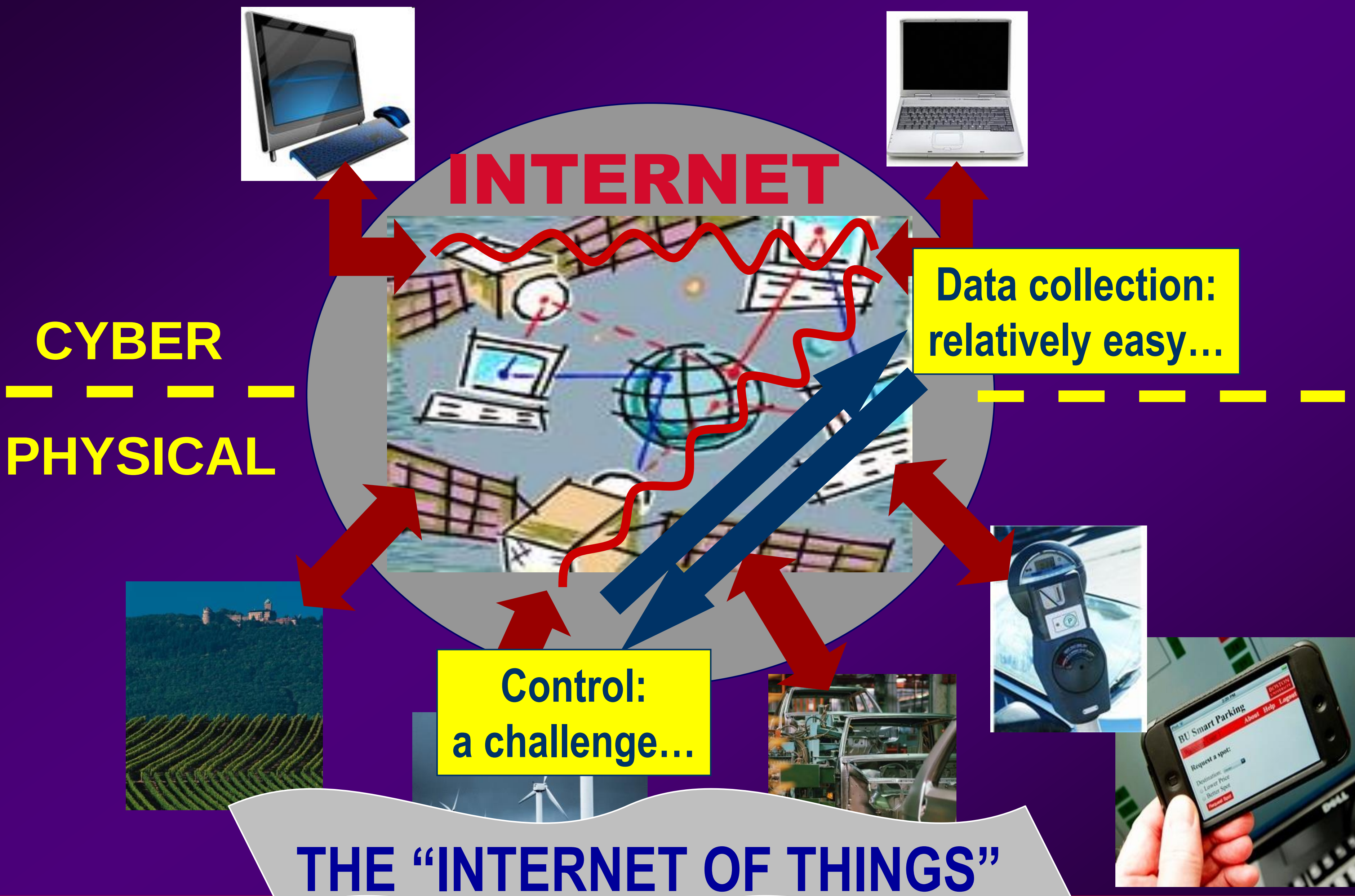
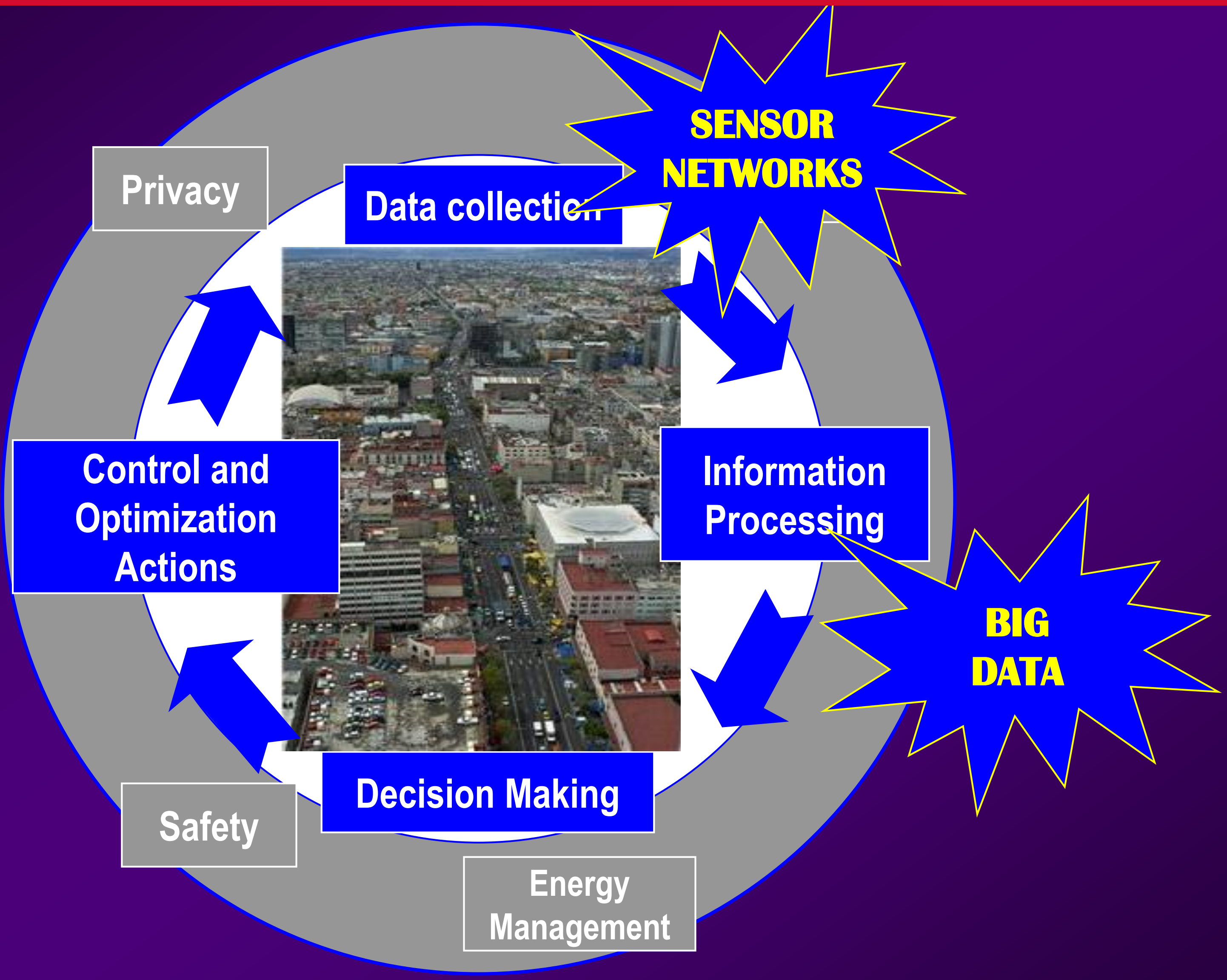
# CYBER-PHYSICAL SYSTEMS: MOTIVATION AND CHALLENGES

**C. G. Cassandras**
**Division of Systems Engineering**
**Center for Information and Systems Engineering**
**Boston University**

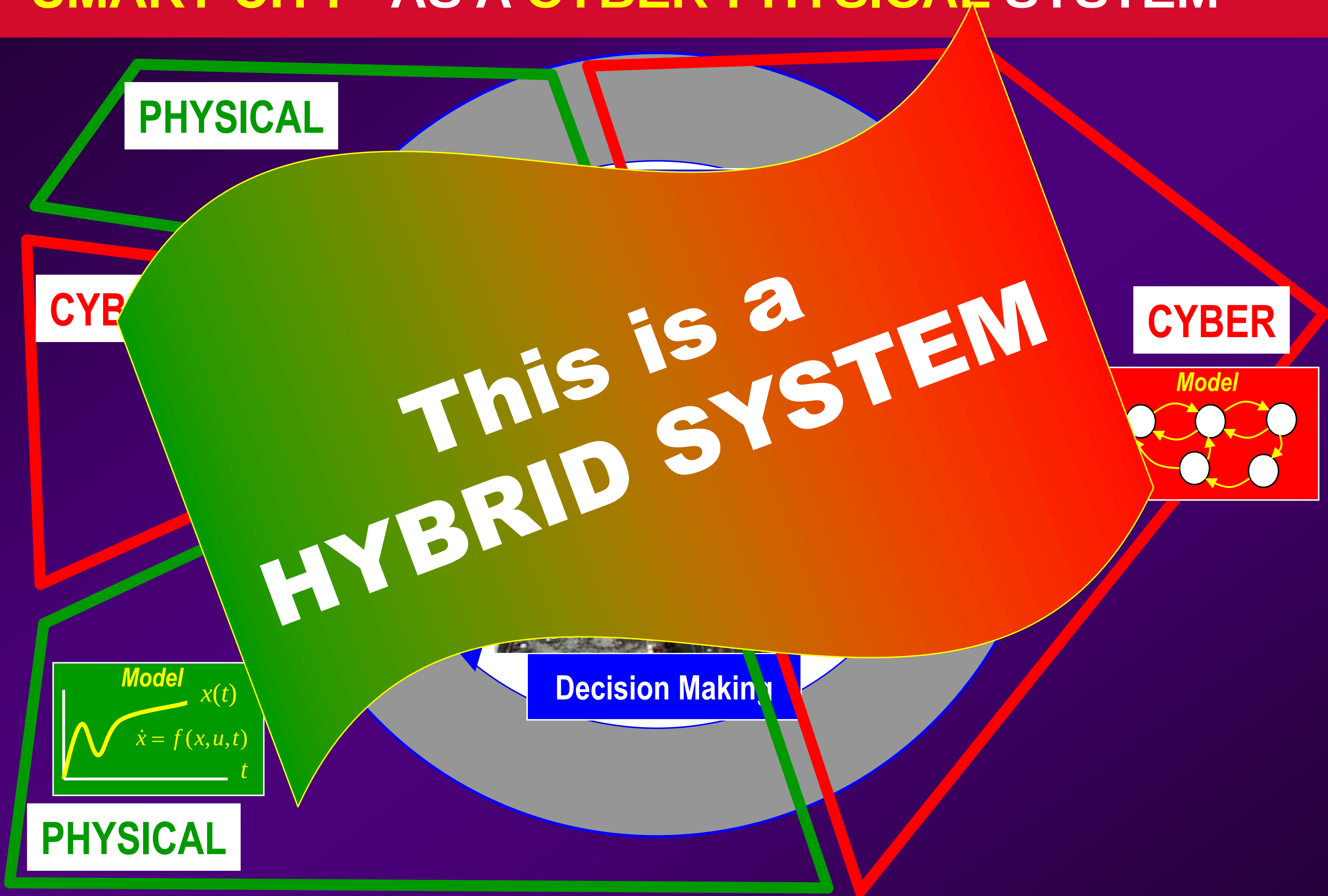
# CYBER-PHYSICAL SYSTEMS



# "SMART CITY" AS A CYBER-PHYSICAL SYSTEM



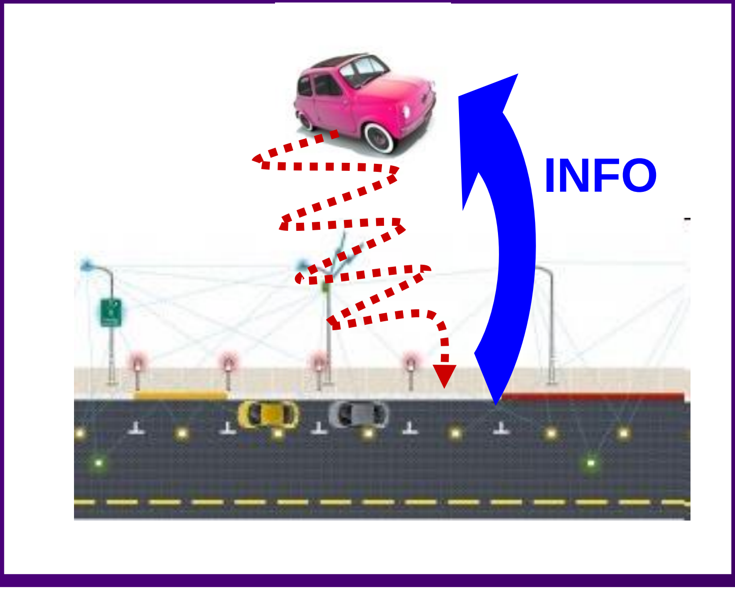
# "SMART CITY" AS A CYBER-PHYSICAL SYSTEM



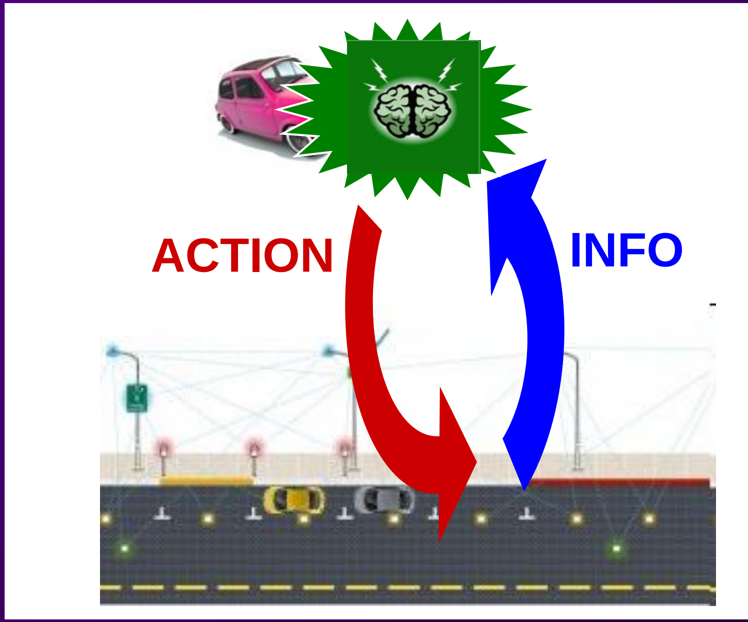
# WHAT IS *REALLY* "SMART" ?

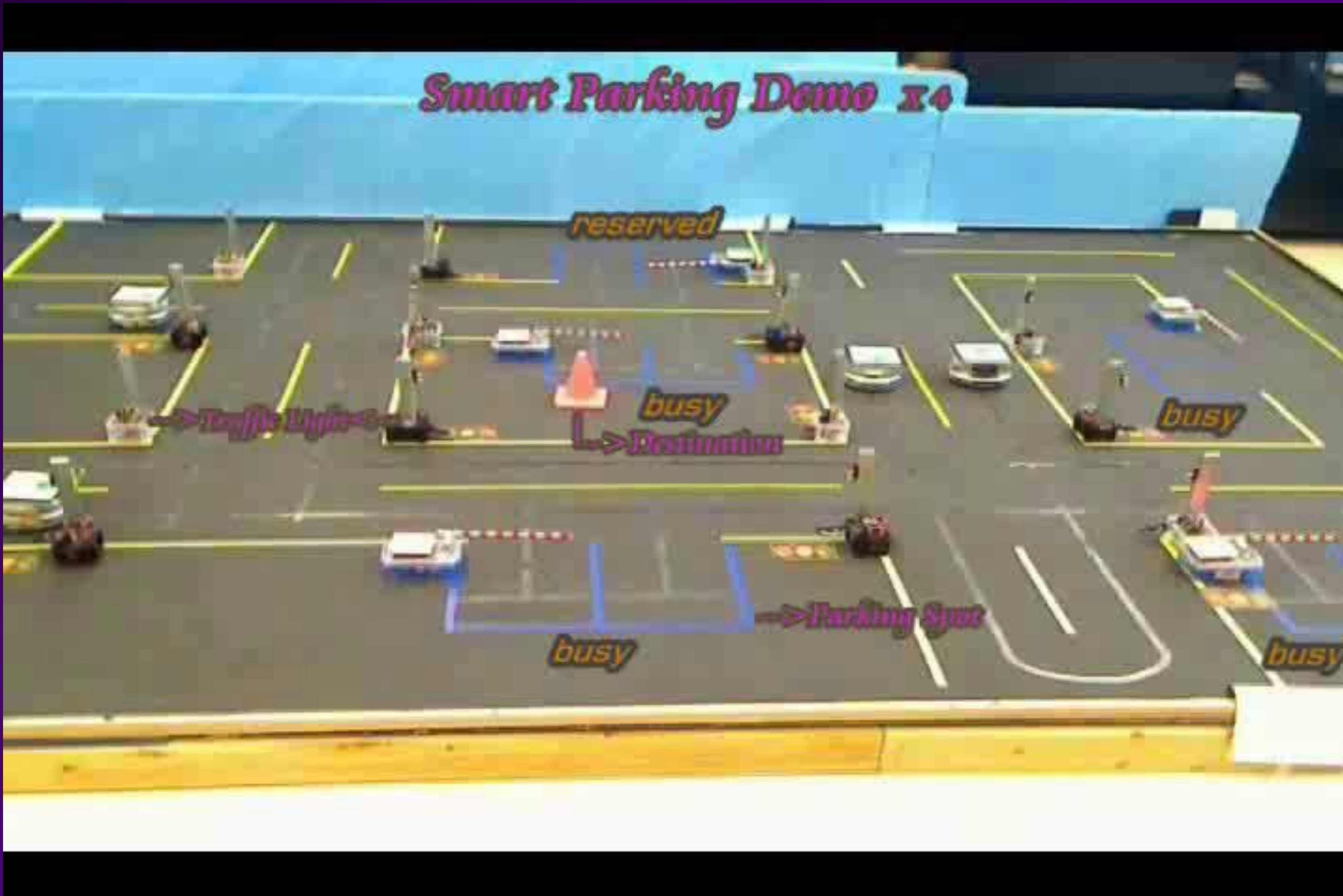
COLLECTING DATA IS NOT "SMART"

- JUST A NECESSARY STEP TO BEING "SMART"




PROCESSING DATA TO MAKE GOOD DECISIONS IS "SMART"




Smart Parking Demo x4
reserved
busy
L->Destination
busy
-->Parking Spot
busy
busy

# MODELING:

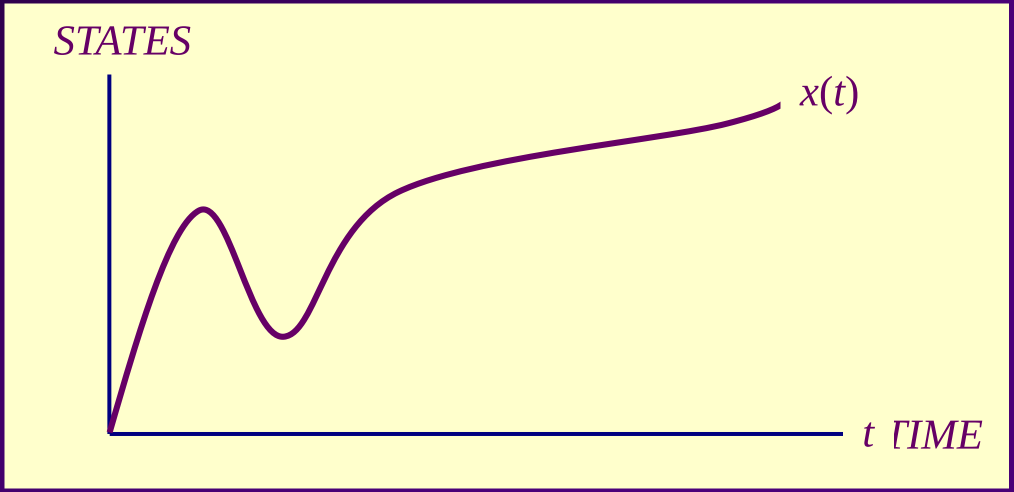
## TIMED-DRIVEN

## vs

## EVENT-DRIVEN

# TIME-DRIVEN v EVENT-DRIVEN SYSTEMS

*TIME*-DRIVEN SYSTEM



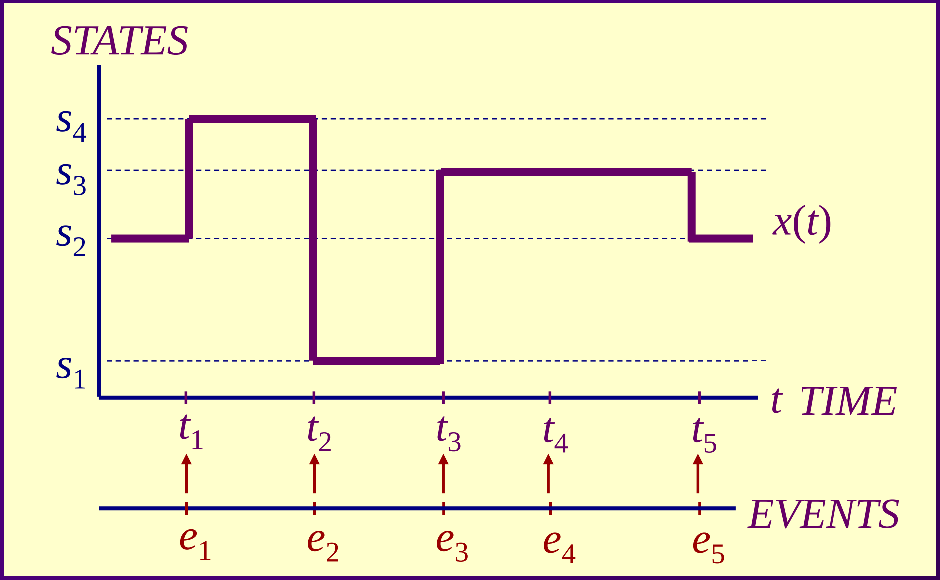

**STATE SPACE:**

$$X = \Re$$

**DYNAMICS:**

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

*EVENT*-DRIVEN SYSTEM



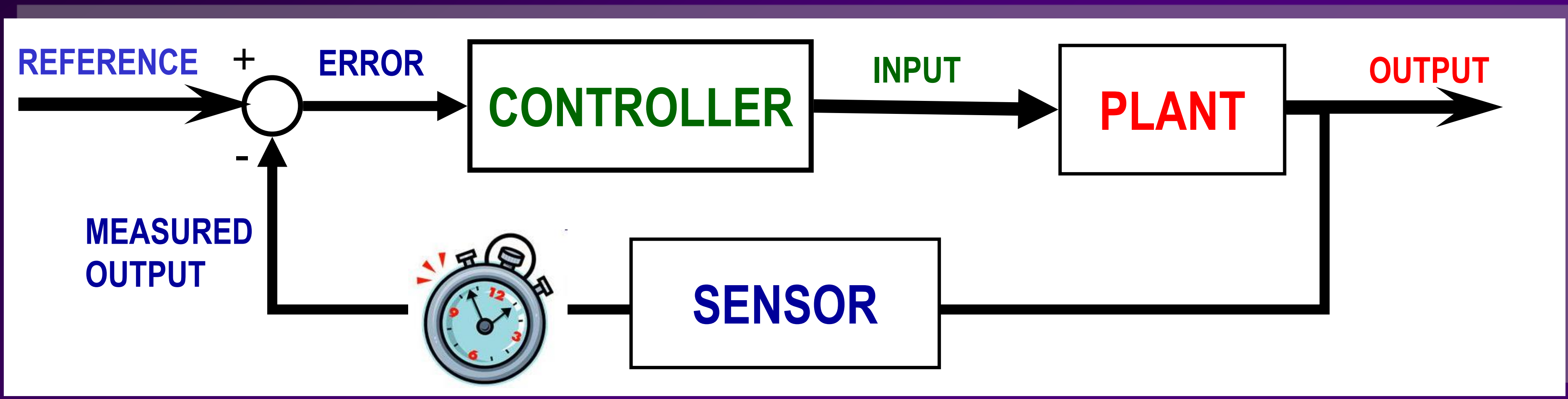

**STATE SPACE:**

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

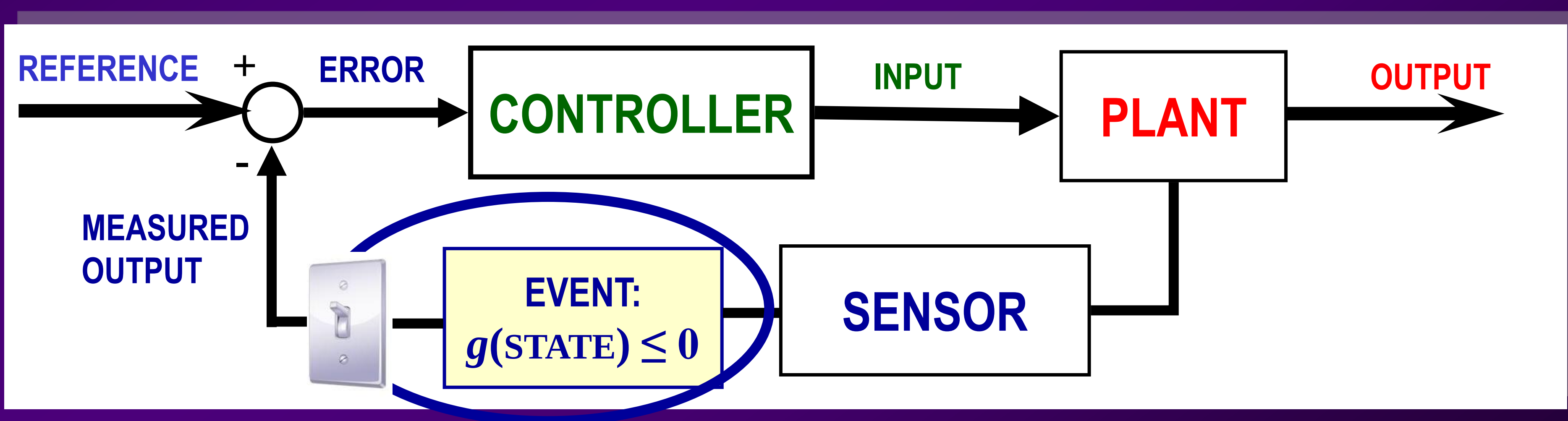
**DYNAMICS:**

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

# TIME-DRIVEN v EVENT-DRIVEN CONTROL




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



# 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.

# 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)

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

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

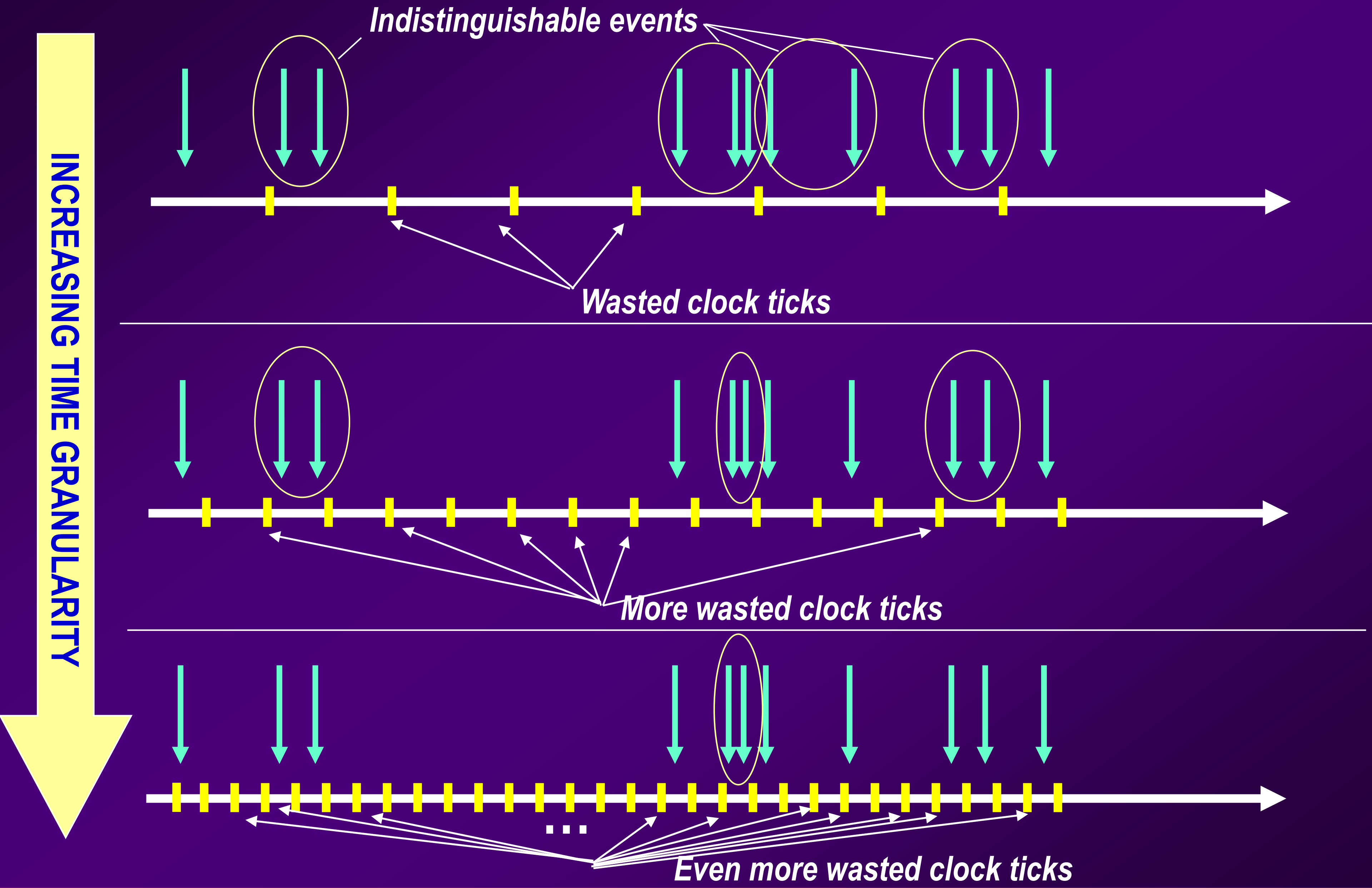
- **Many systems are stochastic**
  **$\longrightarrow$ actions needed in response to random events**

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

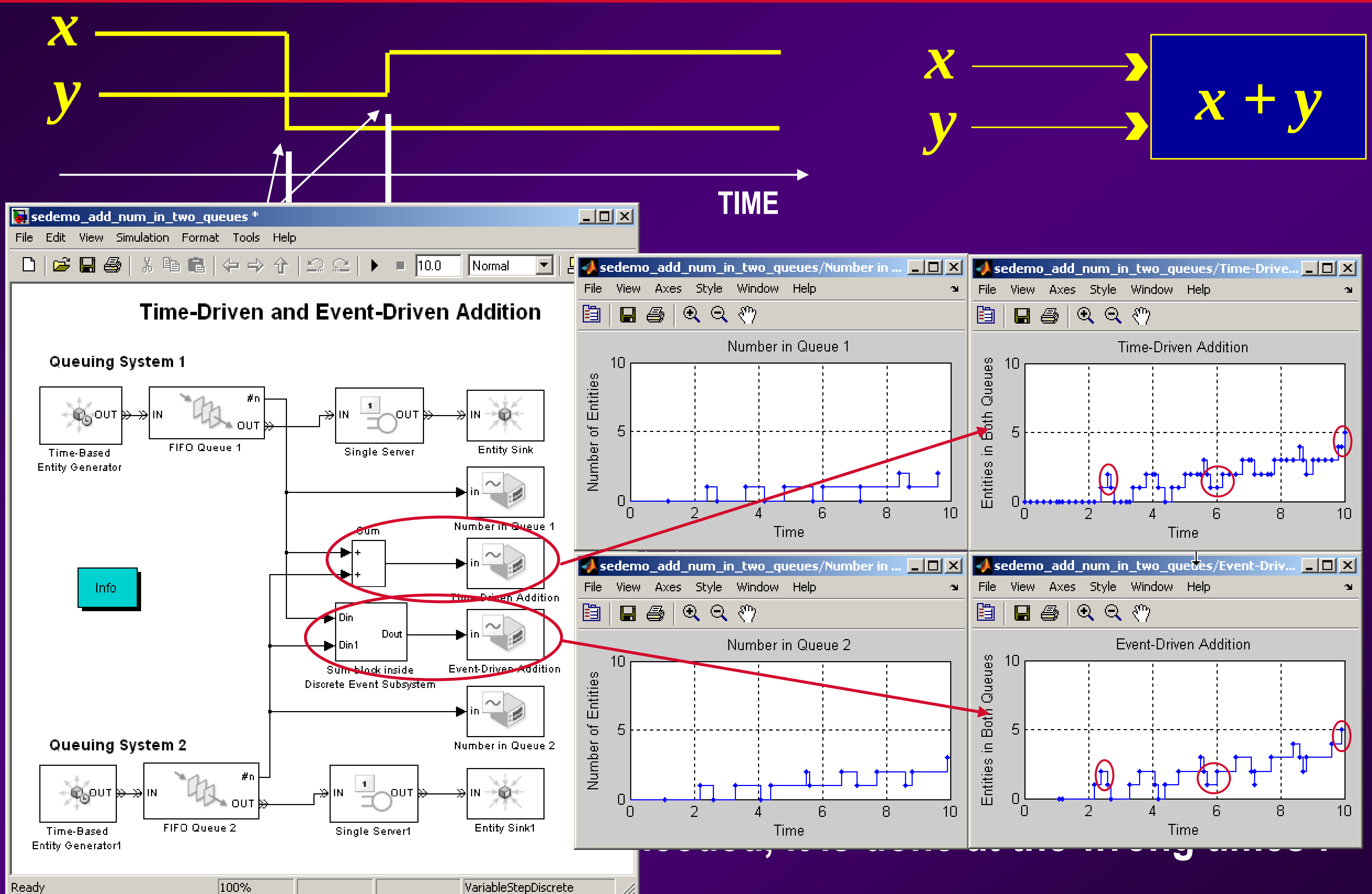
- **Time-driven sampling inherently inefficient ("open loop" sampling)**

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

# SYNCHRONOUS v ASYNCHRONOUS BEHAVIOR



# SYNCHRONOUS v ASYNCHRONOUS COMPUTATION



# MULTI-AGENT NETWORK SYSTEMS

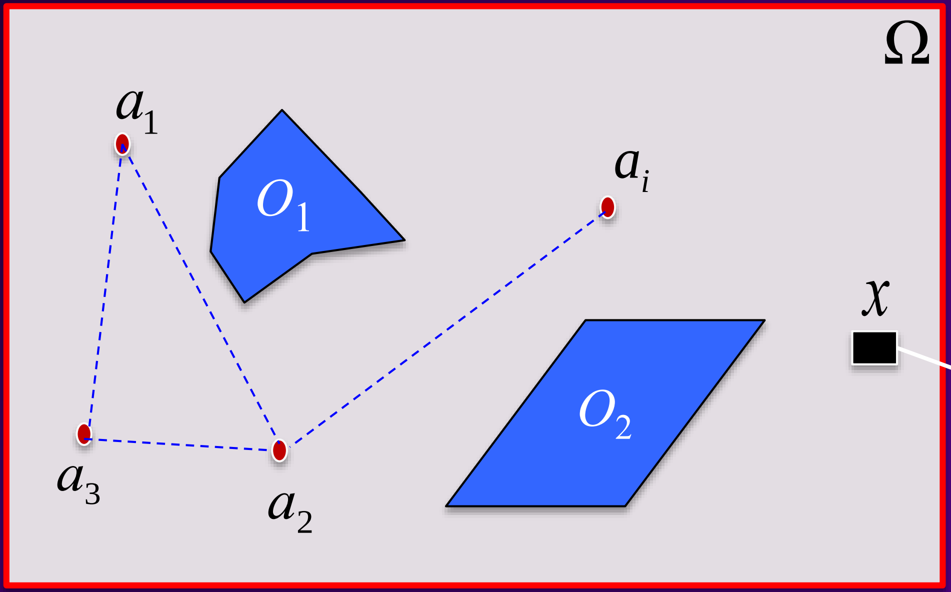
# COOPERATIVE MULTI-AGENT SYSTEMS

The multi-agent system framework consists of a team of autonomous agents cooperating to carry out complex tasks within a given environment.

Applications:

- Monitoring (data sources/targets)
- Search and rescue
- Smart buildings
- Intelligent transportation
- Formation flight of Unmanned Aerial Vehicles

# MULTI-AGENT OPTIMIZATION: PROBLEM 1



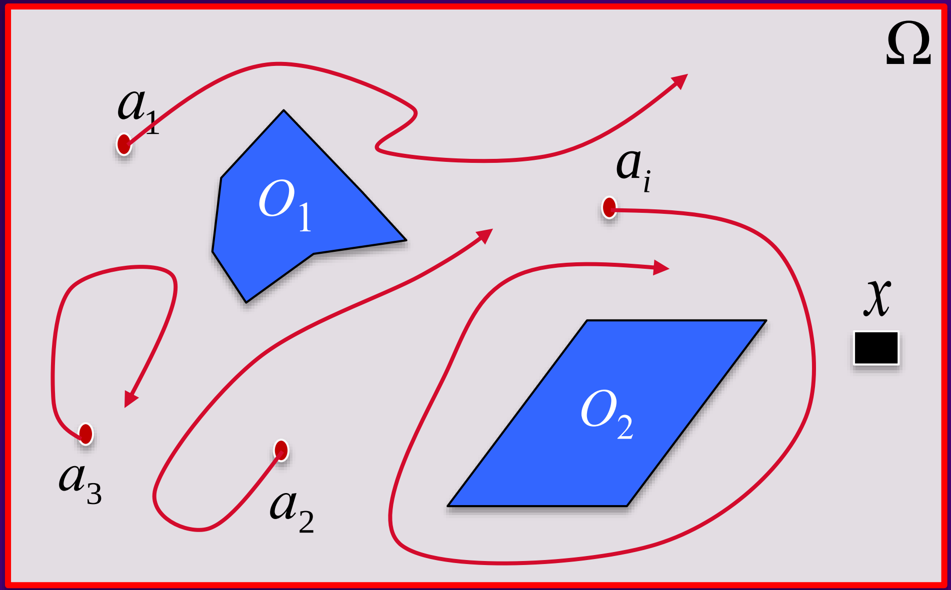

- $s_i$: agent state, $i = 1,\ldots, N$
  $\mathbf{s}=[s_1, \ldots, s_N]$
- $O_j$: obstacle (constraint)
- $R(x)$: property of point $x$
- $P(x, \mathbf{s})$: reward function

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

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

**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

# MULTI-AGENT OPTIMIZATION: PROBLEM 2




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

May also have dynamics

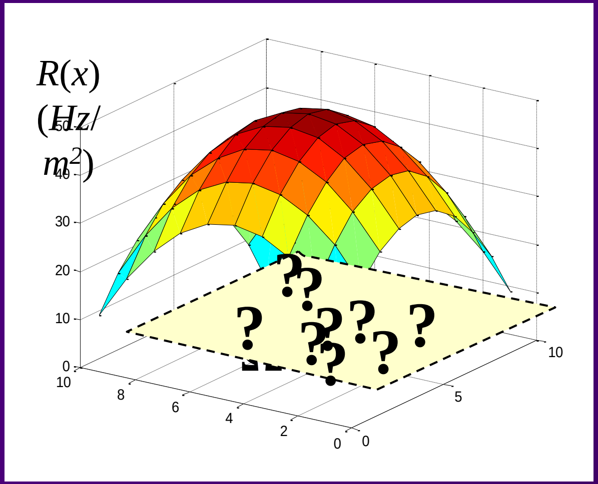
$$s_i(t) \in F \subseteq \Omega, \ i = 1, \cdots, N \qquad \dot{s}_i = f_i(s_i, u_i, t), \ \ i = 1, \cdots, N$$

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

# PROBLEMS THAT FIT THIS FRAMEWORK

# COVERAGE CONTROL: ACTIVE COOPERATION

**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

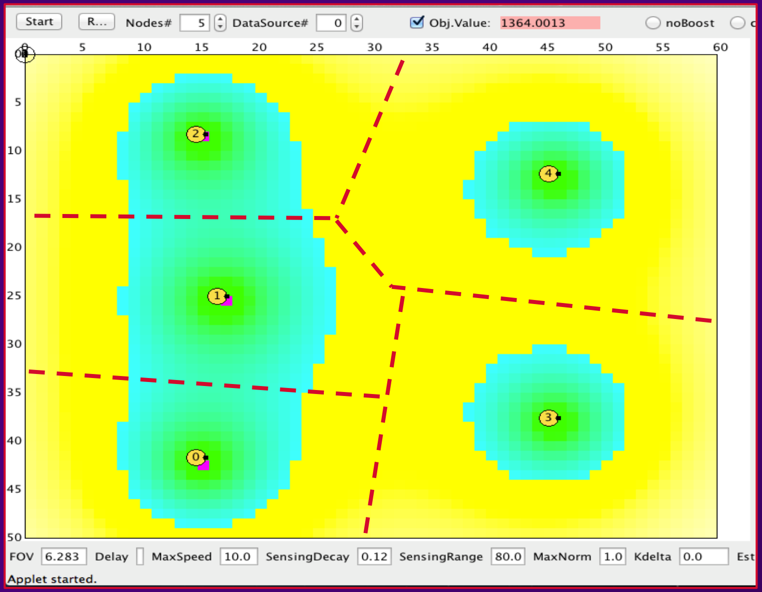
# COVERAGE CONTROL: VORONOI PARTITIONING

$$\max_{\mathbf{s}} H(\mathbf{s}) = \sum_{i=1}^{N} \int_{V_i} f(\|x - s_i\|) R(x) dx$$

$$V_i = \{x \in \Omega : \|x - s_i\| \le \|x - s_j\|, j \ne i\}$$

$f(\|x - s_i\|)$ : sensing quality

$R(x)$ : event occurrence frequency

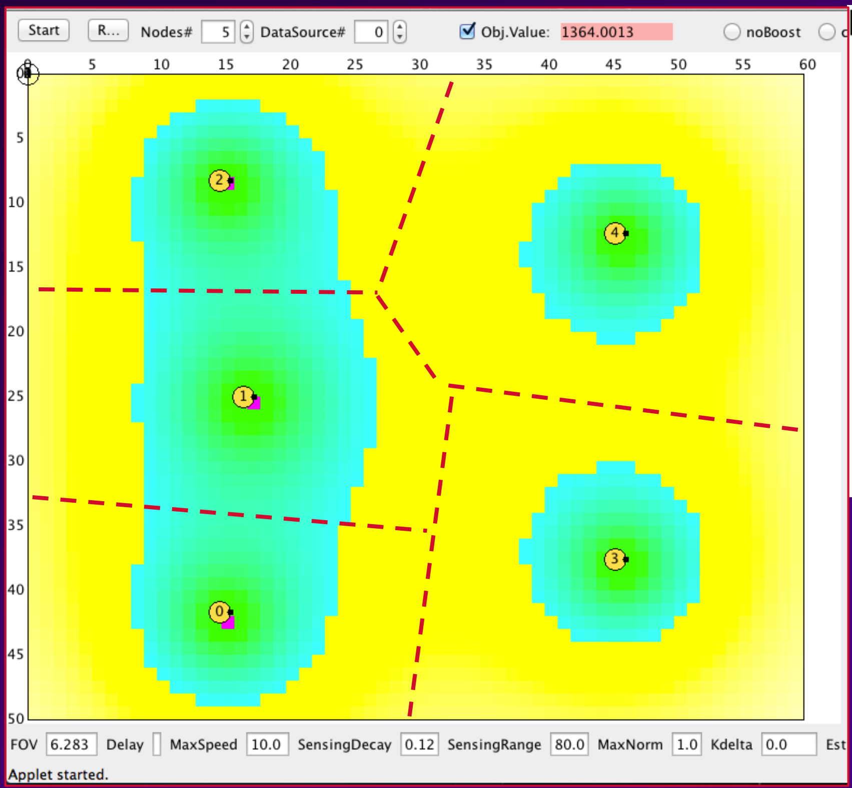


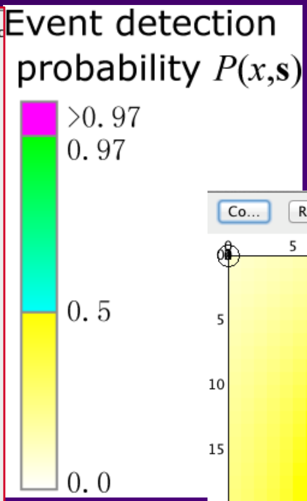

$$\max_{\mathbf{s}} H(\mathbf{s}) = \int_{\Omega} P(x, \mathbf{s}) R(x) dx$$

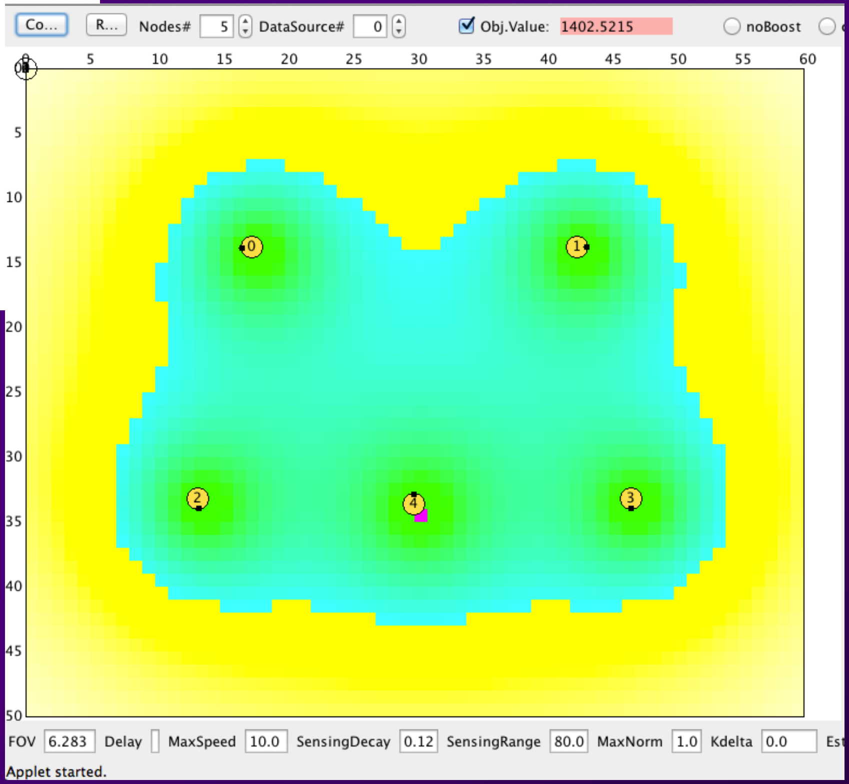
$$P(x, \mathbf{s}) = \sum_{i=1}^{N} p_i(x, s_i)$$

$$p_i(x, s_i) = \begin{cases} f(\|x - s_i\|) & x \in V_i \\ 0 & x \notin V_i \end{cases}$$

# COVERAGE CONTROL:
## ACTIVE COOPERATION vs PARTITIONING




***Voronoi patition*;**

**Optimal obj. function = 1346.5**

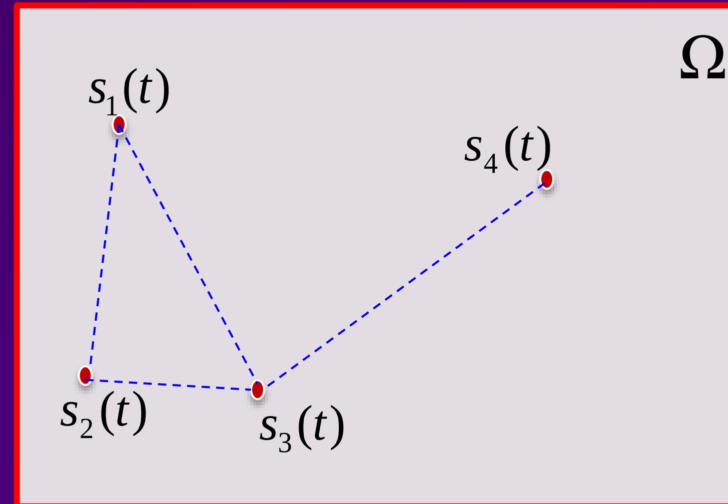
***Gradient-based cooperative algorithm*;**

**Optimal obj. function = 1388.1**

# 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

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

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

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

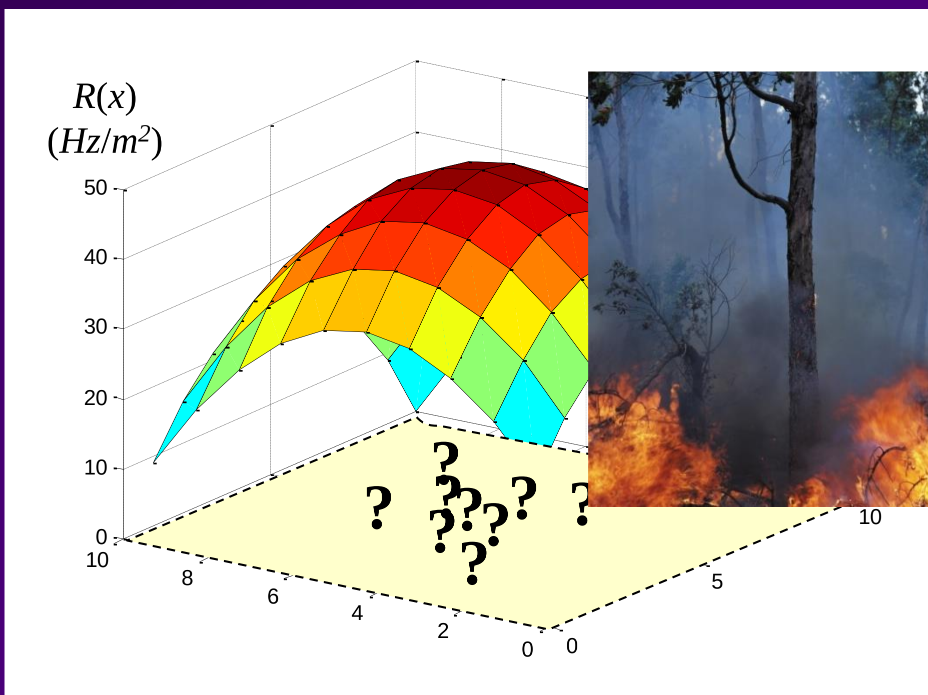
**NOTE: Convex optimization, relatively easy problem**

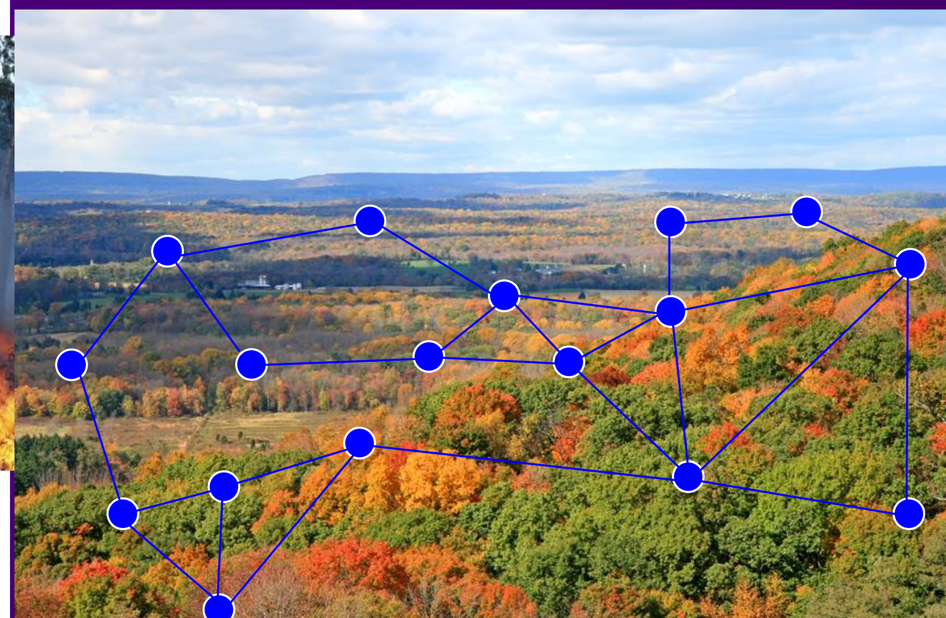
# COVERAGE CONTROL v PERSISTENT MONITORING

**COVERAGE CONTROL:**

**Deploy sensors to maximize "event" detection probability**

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



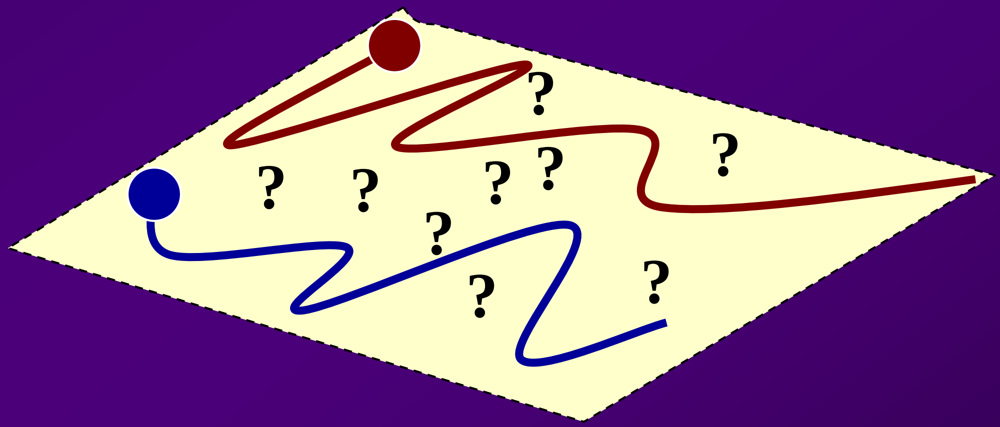



**Perceived event density (data sources) over given region (mission space)**

# COVERAGE CONTROL v PERSISTENT MONITORING

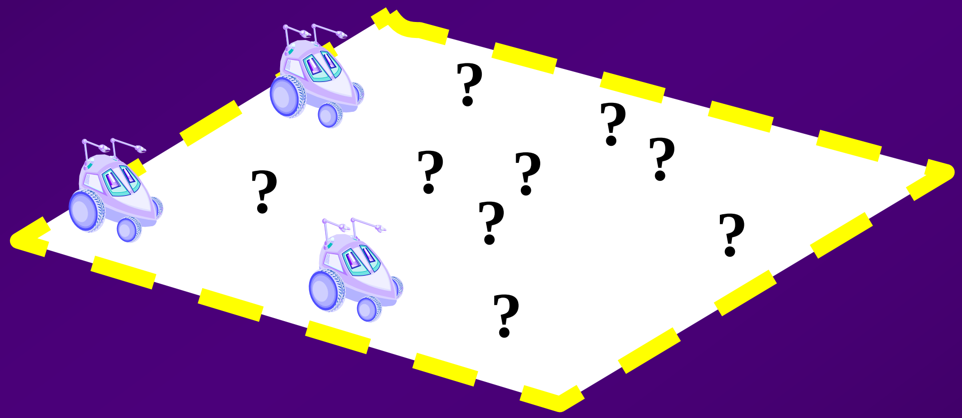
**PERSISTENT MONITORING:**

- environment **cannot be fully covered** by stationary team of agents
- all areas of mission space must be visited infinitely often
- minimize some measure of overall uncertainty

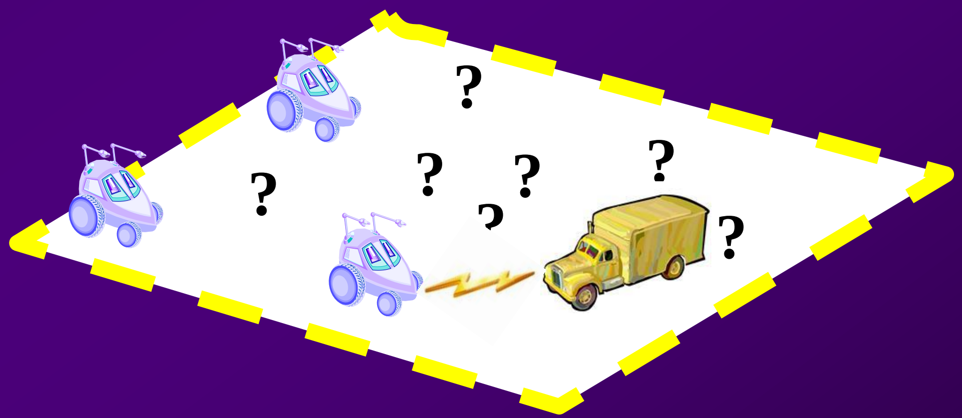


# COVERAGE CONTROL + PERSISTENT MONITORING

1. Seek and detect "**Data Sources**" (or "**Targets**")

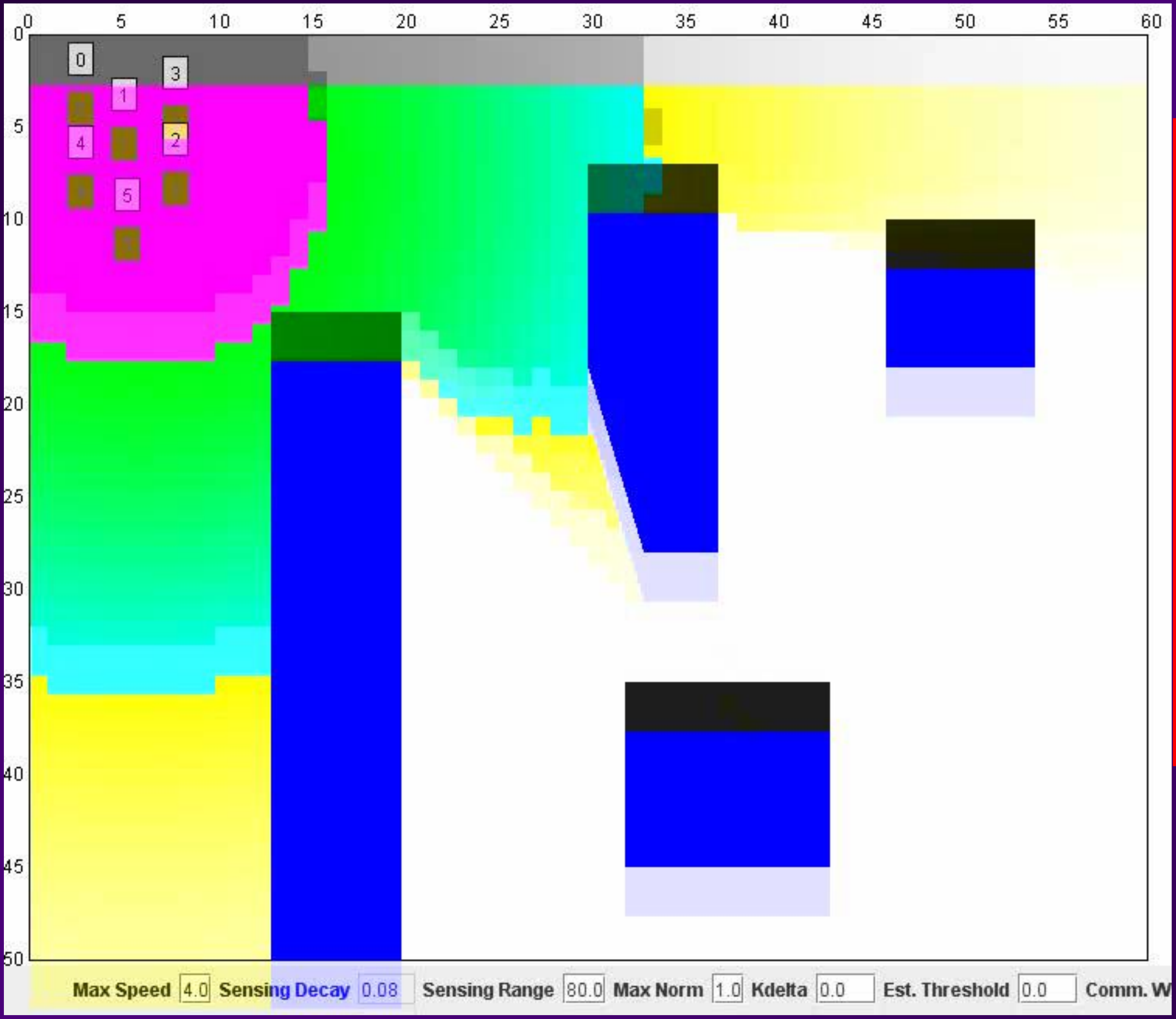



2. Once a Data Source is detected, collect data from it, track it if mobile




3. Continue to seek data sources while collecting data from detected sources

# REACTING TO EVENT DETECTION




**Important to note:**

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

# RELATED WORK

## Coverage control:

- J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” IEEE Trans. on Robotics and Automation, 2004.
- M. Zhong and C. G. Cassandras, “Distributed coverage control and data collection with mobile sensor networks,” IEEE Trans. Autom. Control, 2011.
- W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” IEEE Trans. On Robotics, 2005.

## Persistent monitoring/surveillance:

-I. Rekleitis, V. Lee-Shue, A. New, and H. Choset, “Limited communication, multi-robot team based coverage,” Proc. ICRA’04, 2004.
- S. L. Smith, M. Schwager, and D. Rus, “Persistent monitoring of changing environments using robots with limited range sensing,” IEEE Trans. on Robotics, 2011.
-P. Hokayem, D. Stipanovic, and M. Spong, “On persistent coverage control,” Proc. 46th IEEE Conf. Decision and Control, 2007.
- Y. Elmaliach, N. Agmon, and G. Kaminka, “Multi-robot area patrol under frequency constraints,” Proc. ICRA’07, 2007.
- N. Nigam and I. Kroo, “Persistent surveillance using multiple unmanned air vehicles,” Proc. IEEE Aerospace Conference, 2008.
- Y. Chen, K. Deng, and C. Belta, “Multi-agent persistent monitoring in stochastic environments with temporal logic constraints,” Proc. 51stIEEE Conf. Decision and Control, 2012.
- C. G. Cassandras, X. Lin, and X. C. Ding, “An optimal control approach to the multi-agent persistent monitoring problem,” IEEE Trans. Autom. Control, 2013.

# PERSISTENT MONITORING PROBLEM

**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

Need three elements:

1. ENVIRONMENT MODEL

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

2. SENSING MODEL

(how agents interact with environment)

3. AGENT MODEL

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

# PERSISTENT MONITORING PROBLEM

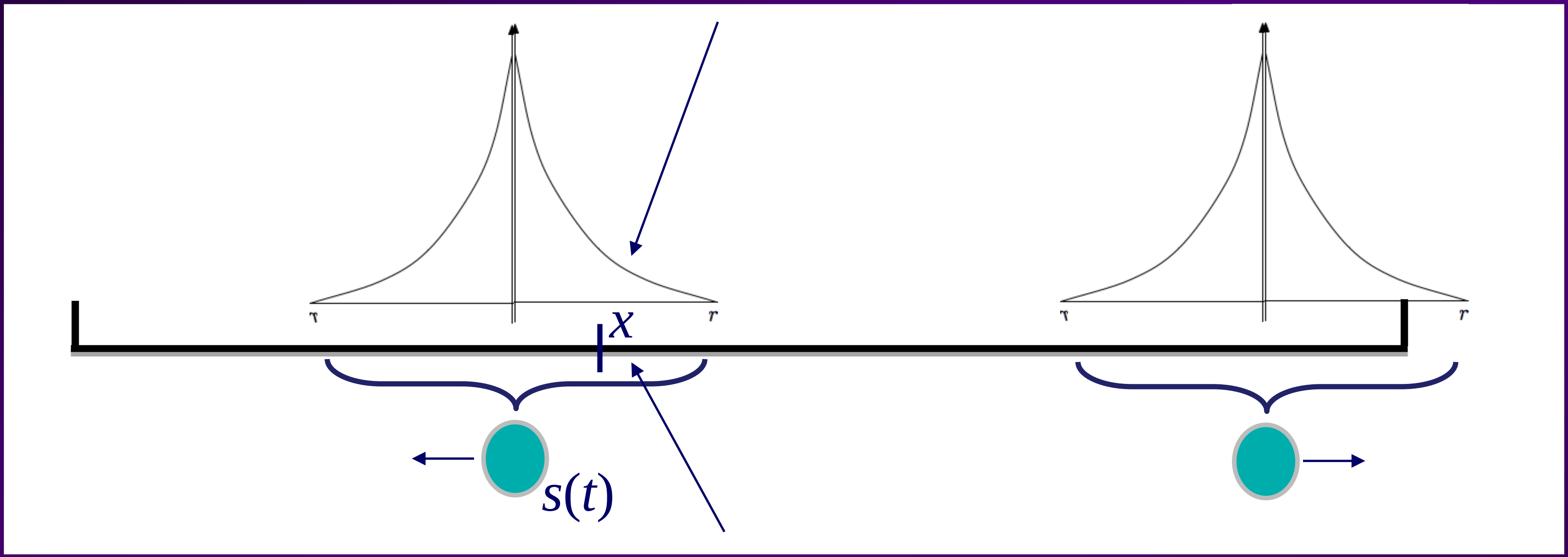
Start with 1-dimensional mission space $\Omega = [0,L]$

**AGENT DYNAMICS:** $\dot{s}_j = u_j, \quad |u_j(t)| \leq 1$

Analysis still holds for: $\dot{s}_j = g_j(s_j) + bu_j, \quad |u_j(t)| \leq 1$

# PERSISTENT MONITORING PROBLEM

**SENSING MODEL:** $p(x,s)$ **Probability agent at** $s$ **senses point** $x$




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

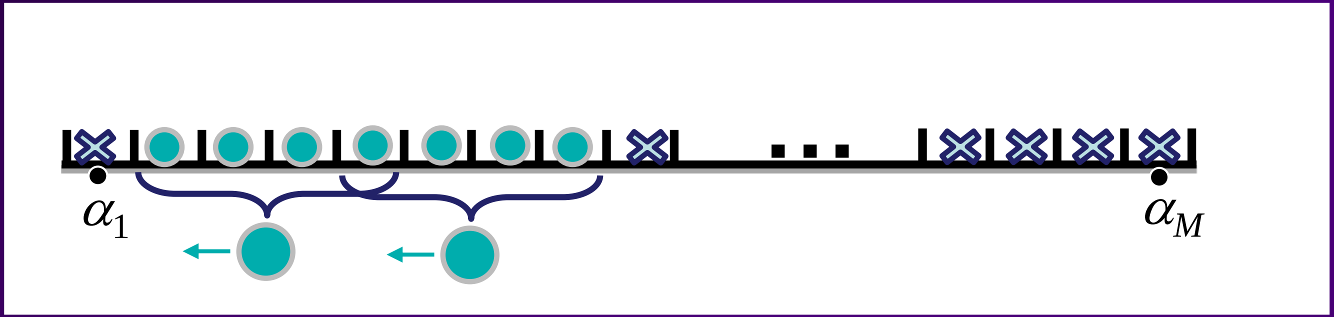
**Use:**

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

**If** $x$ **is a known "target":** $\dot{R}_x(t) = f_x(R,s,t) + noise$

# PERSISTENT MONITORING PROBLEM

Partition mission space $\Omega = [0,L]$ into $M$ intervals:




For each interval $i = 1,\ldots,M$ define ***Uncertainty Function*** $R_i(t)$:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i < BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases}$$

$$P_i(\mathbf{s}) = 1 - \prod_{j=1}^{N} \left[1 - p_i(s_j)\right]$$

$$p_i(s_j) \equiv p_j(\alpha_i, s_j)$$

where $P_i(\mathrm{s})$ = joint prob. $i$ is sensed by agents located at $\mathrm{s} = [s_1,\ldots,s_N]$

# OPTIMAL CONTROL PROBLEM

**Determine $u_1(t),\ldots,u_N(t)$ such that**

$$\min_{u_1,\ldots,u_N} J = \frac{1}{T}\int_0^T \sum_{i=1}^{M} R_i(t)\,dt$$

**Uncertainty measure**

s.t.

$$\dot{s}_j = u_j, \quad |u_j(t)| \le 1, \quad 0 < a \le s_j(t) \le b < L$$

**Agent dynamics**

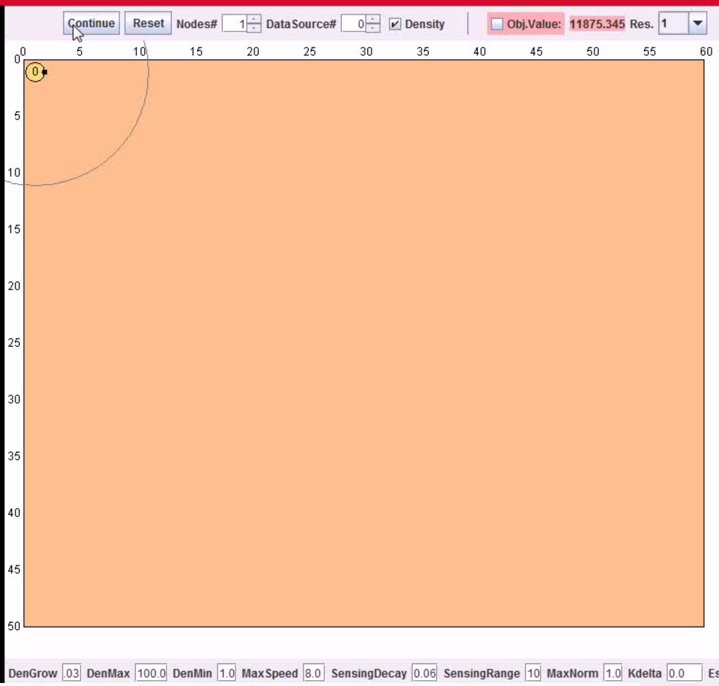
$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i < BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases}$$

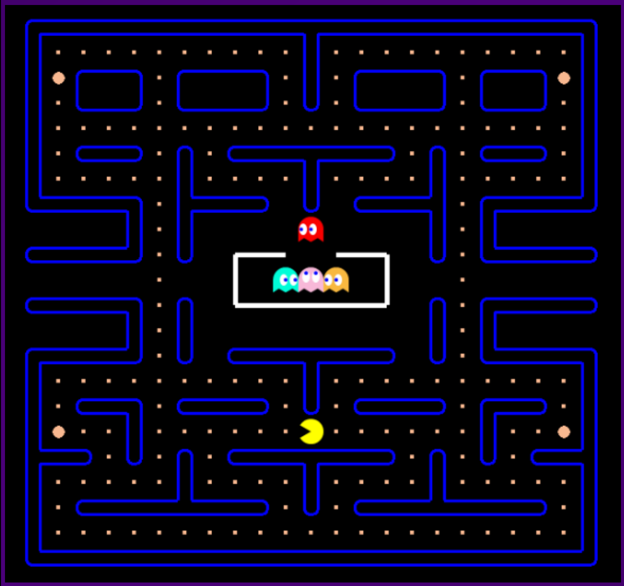
**Uncertainty dynamics**

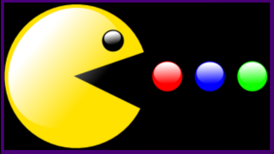
$$p_j(x, s_j) = \begin{cases} 1 - \dfrac{|x - s_j|}{r_j} & \text{if } |x - s_j| \le r_j \\ 0 & \text{if } |x - s_j| > r_j \end{cases}$$

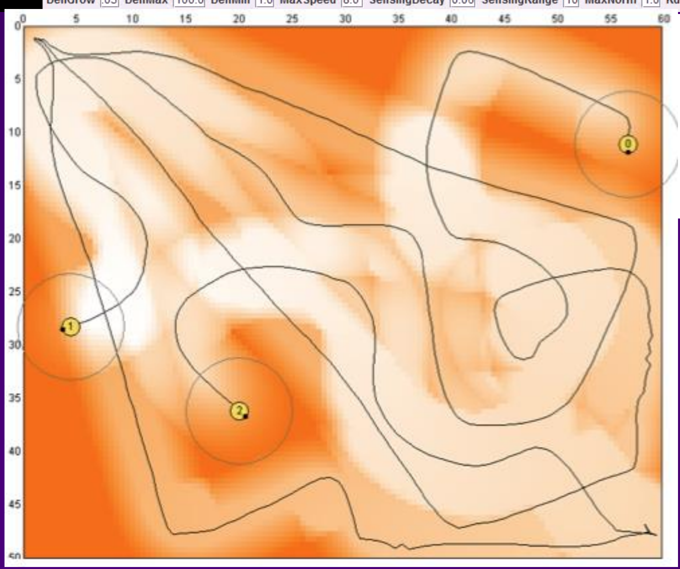
**Sensing model**

# PERSISTENT MONITORING IN 2D MISSION SPACE







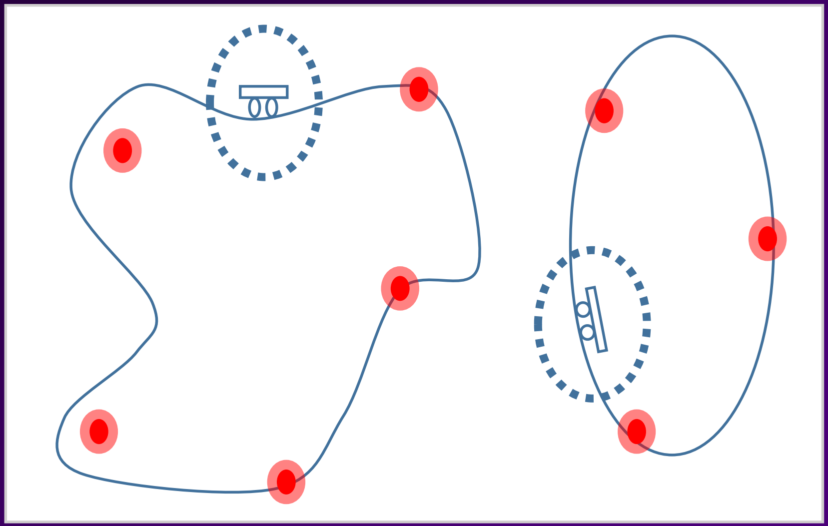
**Dark brown:**
**HIGH uncertainty**

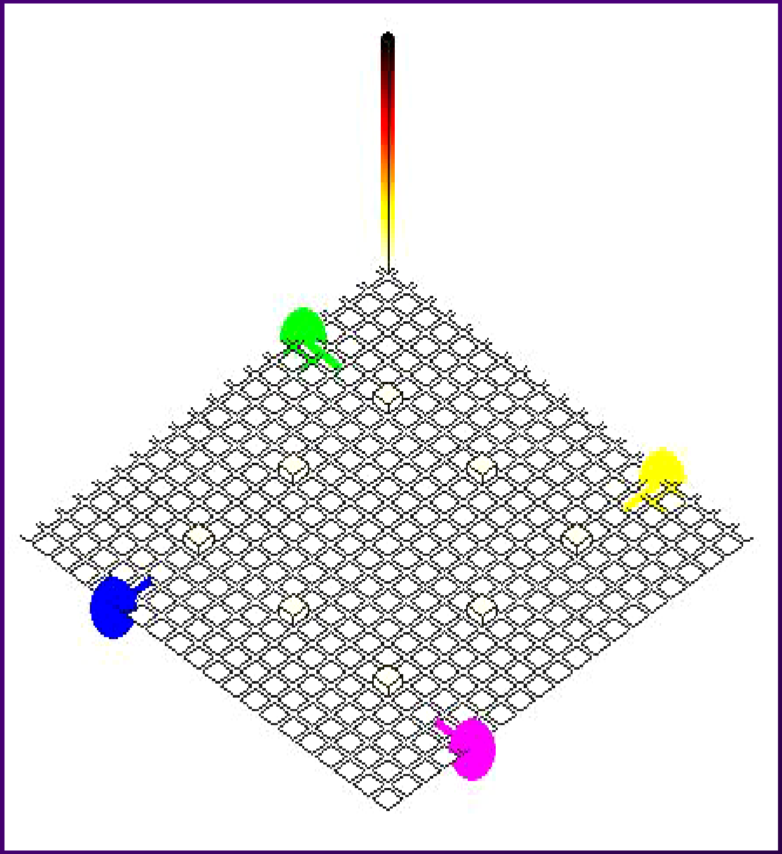
**White:**
**NO uncertainty**

**Agents play a cooperative PACMAN game against "uncertainty" which continuously regenerates…**

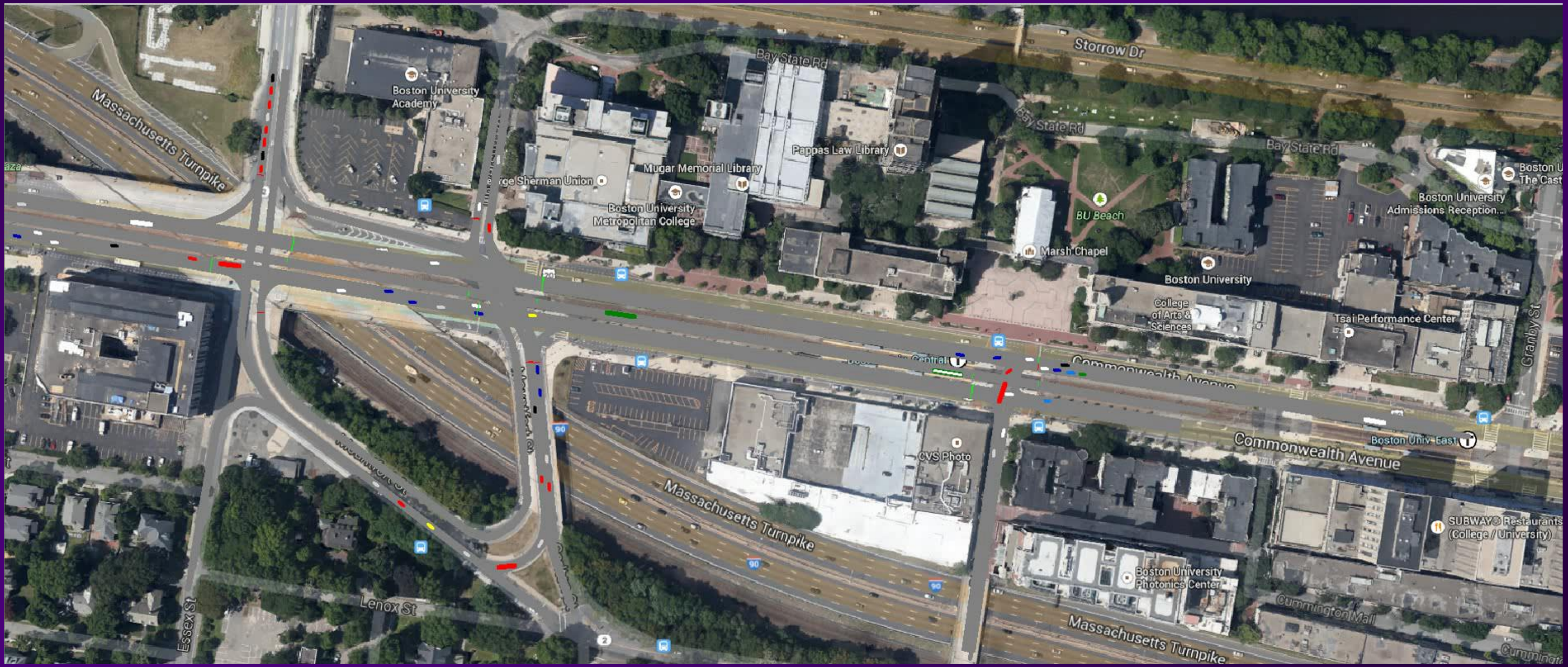
JAVA multi-agent simulator designed to interactively test various controllers. Polygonal obstacles may be added to the environment.
*http://people.bu.edu/cgc/gengyf/density/density.htm*

# PERSISTENT MONITORING WITH KNOWN TARGETS





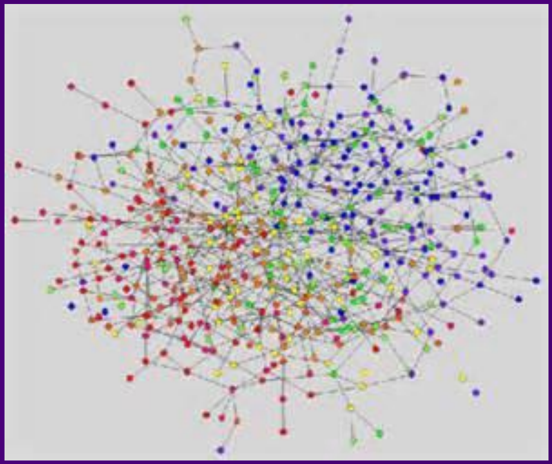
# TRAFFIC NETWORK CONTROL




The BU Bridge mess, Boston, MA (simulation using VISSIM)

# WHY CAN'T WE IMPROVE TRAFFIC…

## … EVEN IF WE KNOW THE ACHIEVABLE OPTIMUM IN A TRAFFIC NETWORK ???



Because:

- **Not enough controls** (traffic lights, tolls, speed fines)

  → No chance to unleash the power of feedback!

- **Not knowing other drivers' behavior** leads to poor decisions (a simple game-theoretic fact)

  → Drivers seek individual (**selfish**) optimum, not system-wide (**social**) optimum } ***PRICE OF ANARCHY* (POA)**

# GAME-CHANGING OPPORTUNITY:
# CONNECTED AUTOMATED VEHICLES (CAVs)

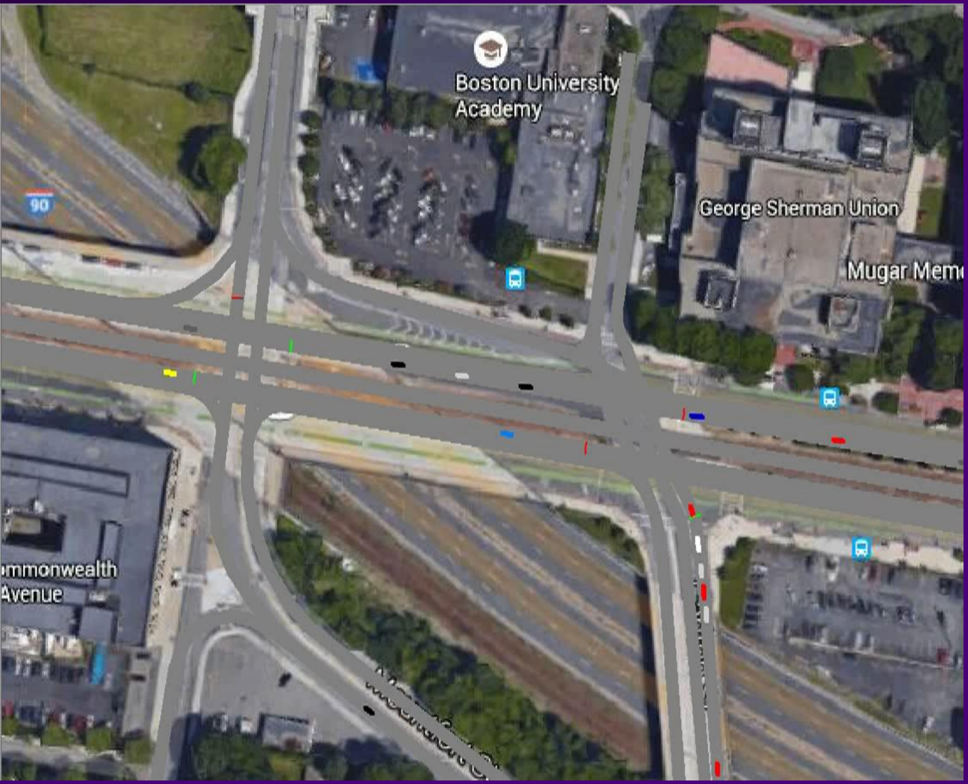
**NO TRAFFIC LIGHTS, NEVER STOP…**

**FROM (SELFISH) “DRIVER OPTIMAL” TO (SOCIAL) “SYSTEM OPTIMAL” TRAFFIC CONTROL**

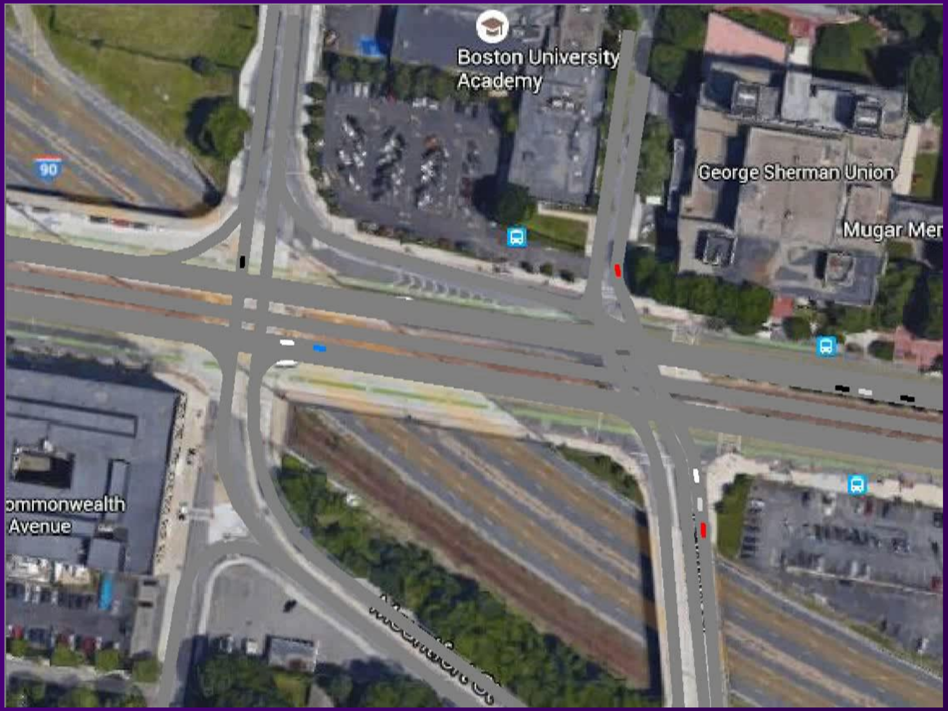
**THE “INTERNET OF CARS”**

# WHO NEEDS TRAFFIC LIGHTS?

With traffic lights




With decentralized control of CAVs




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

# KEY TECHNICAL CHALLENGES

# CONTROL AND OPTIMIZATION – CHALLENGES

1. **SCALABILITY**
2. **DECENTRALIZATION**

→ Distributed Algorithms

3. **COMMUNICATION** → Event-driven (asynchronous) Algorithms

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

5. **EXLOIT DATA** → Data-Driven Algorithms