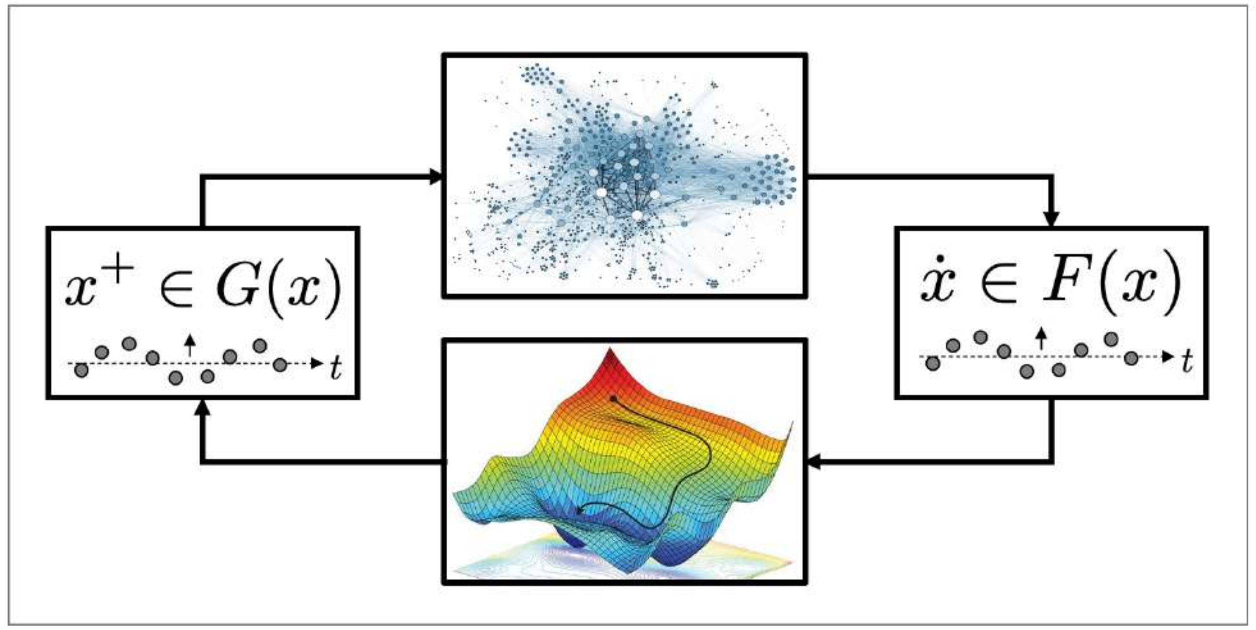
Data-Driven Parameter Estimation with Accelerated Convergence: A Fixed-Time Control Approach



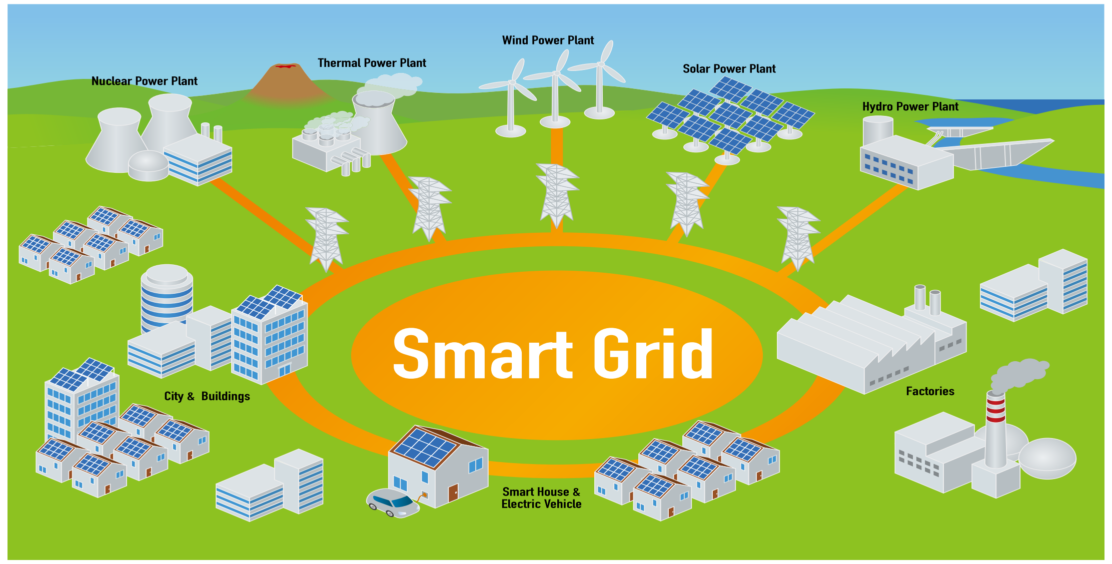
Jorge I. Poveda

Department of Electrical and Computer Engineering

UC San Diego

September 7th, 2022

Autonomous Systems with Data-Assisted Feedback Loops



Autonomous Systems with Data-Assisted Feedback Loops

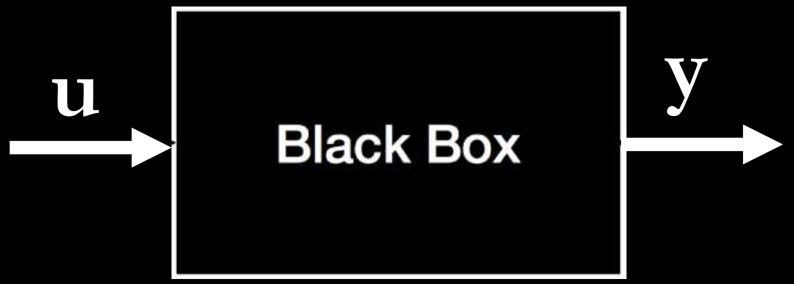
Nonlinear hybrid dynamical systems

$$\dot{\mathbf{x}} \in \mathbf{F}(\mathbf{x}) \qquad \mathbf{x} \in \mathbf{C}$$
$$\mathbf{x}^{+} \in \mathbf{G}(\mathbf{x}) \qquad \mathbf{x} \in \mathbf{D}$$

Large scale systems with limited information

Multiple decision makers with conflicting interests

Mathematical model is too complex or unavailable



Achieving Autonomy is Challenging

US Defense Science Board (DOD), “Counter Autonomy”, Executive Summary, 2020, pp. 1-16.

Autonomous Systems with Data-Assisted Feedback Loops

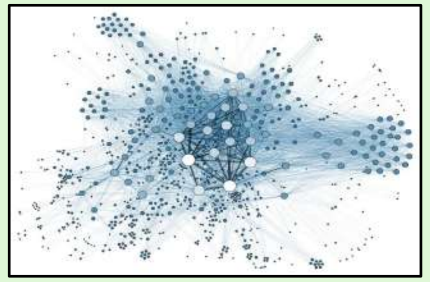
UNKNOWN MODELS

CORRECT DECISIONS UNDER UNCERTAINTY

DYNAMIC/NOISY ENVIRONMENTS

AUTONOMOUS SYSTEMS

DATA SETS & ALGORITHMS



PHYSICAL & DIGITAL COMPONENTS

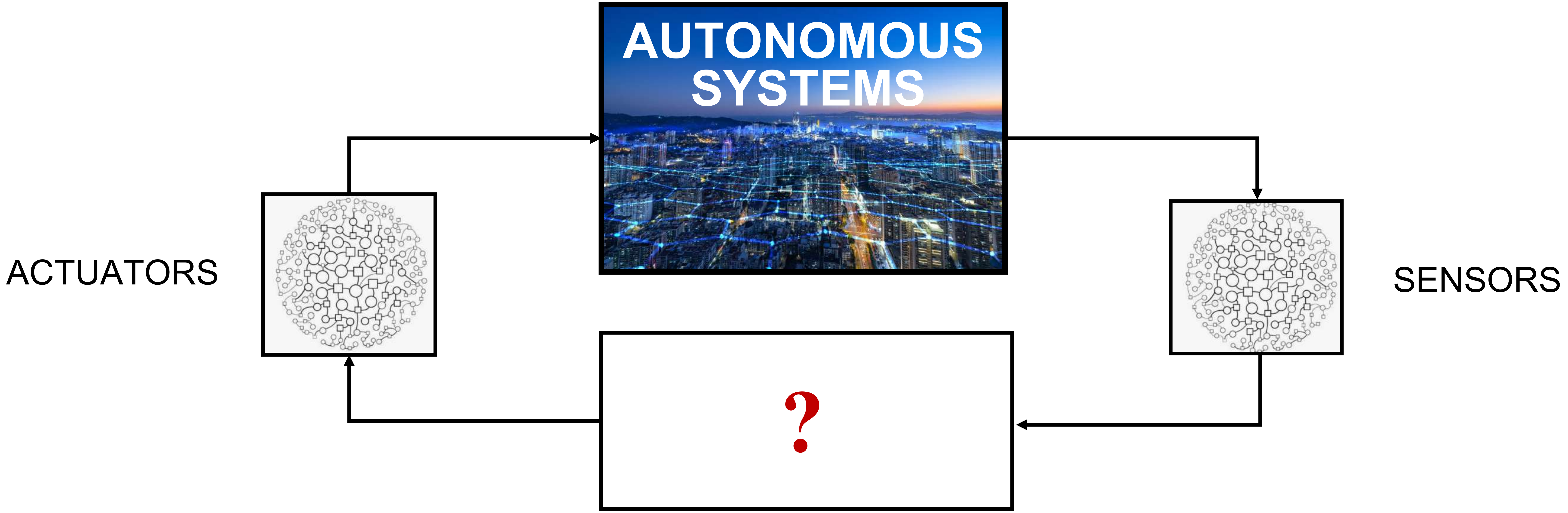
LEARN FROM EXPERIENCE

ADAPTIVE AND SELF-OPTIMIZING

OPTIMAL PERFORMANCE

NO EXTERNAL INPUT FROM HUMANS

Our Overarching Goal:



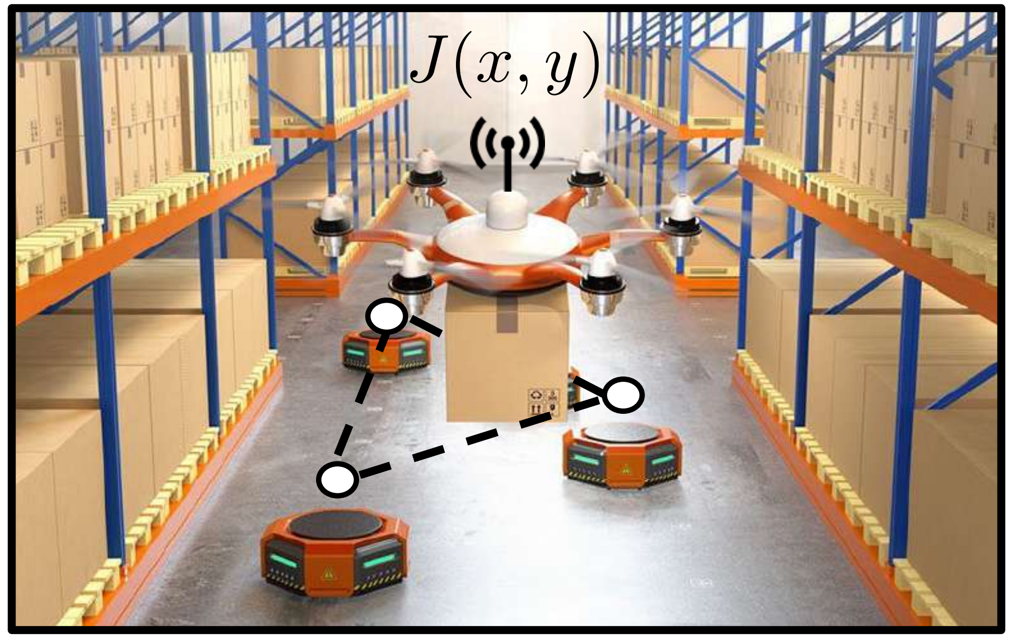
We are looking for principles for the analysis and synthesis of **data-assisted** feedback-based algorithms in Autonomous Systems

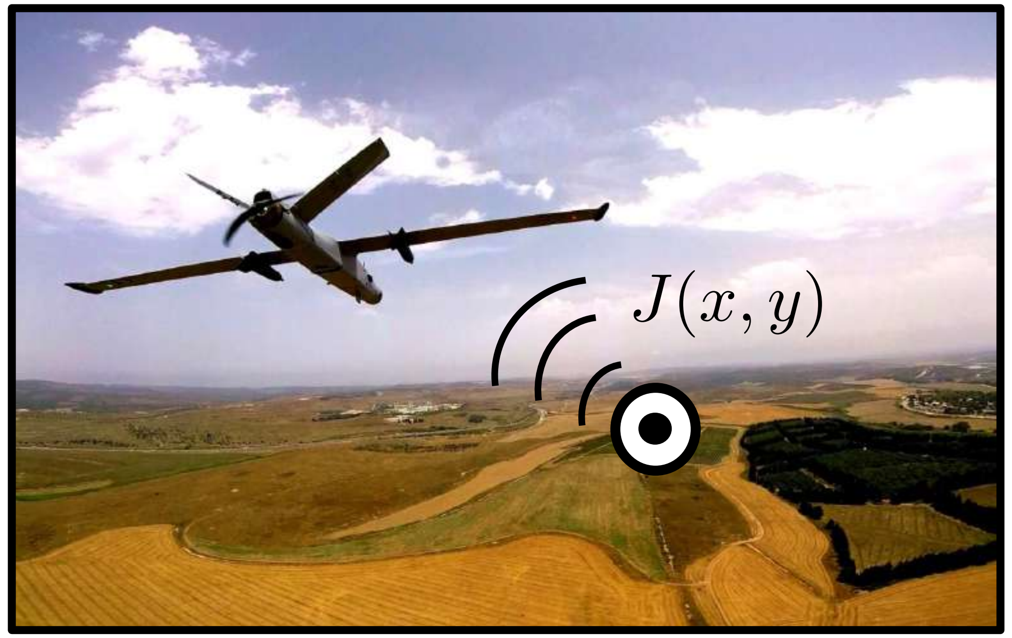
- ➢ Robust Decision Making Under Uncertainty
- ➢ Adaptive and Self-Optimizing Behaviors

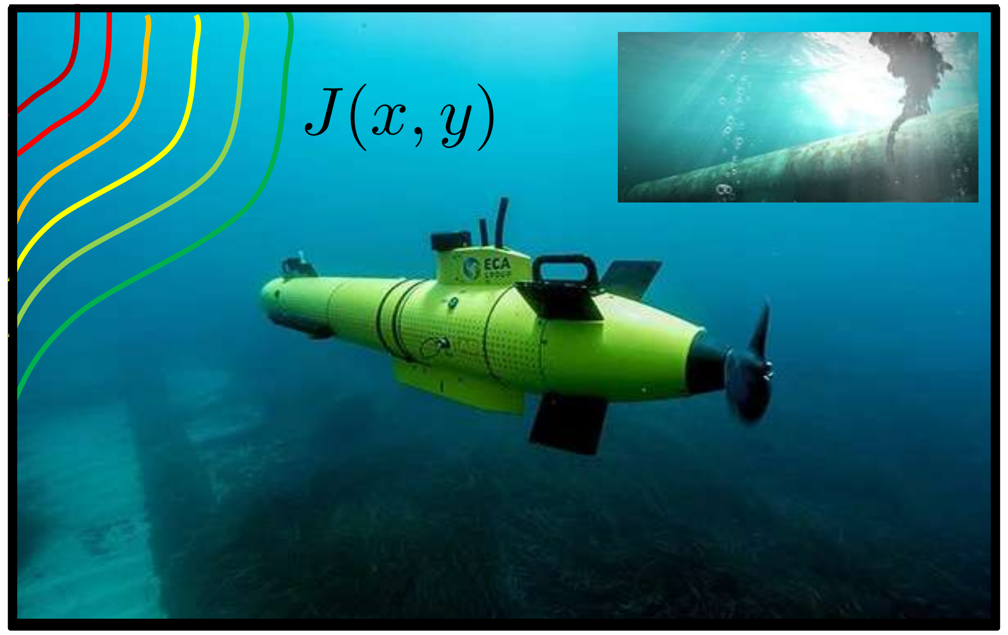
Data-Driven Robust Learning:

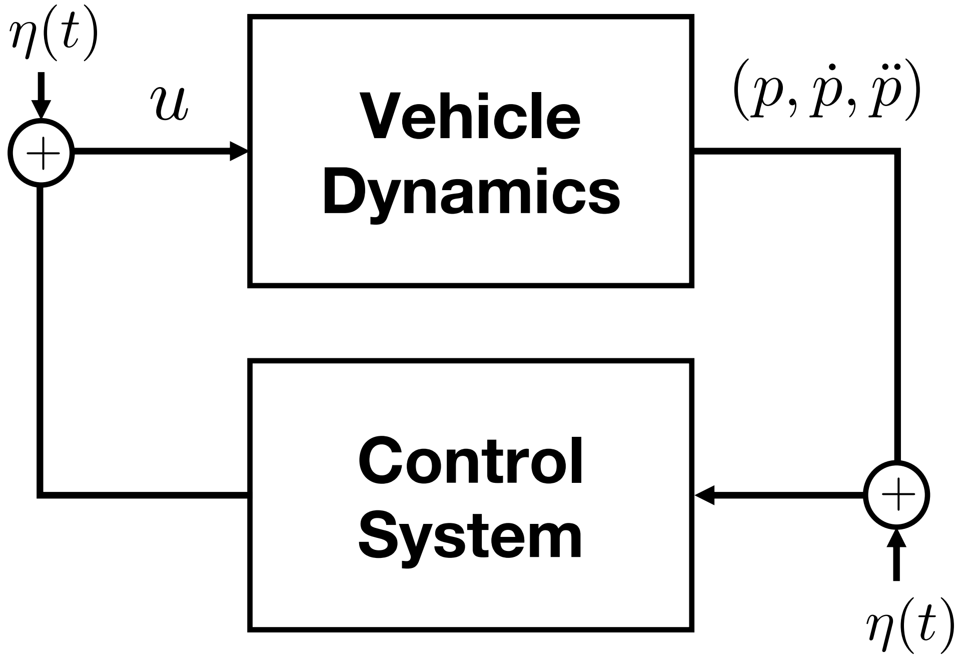
- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

Motivation: Control of autonomous vehicle systems in dynamic, unknown, and contested environments.



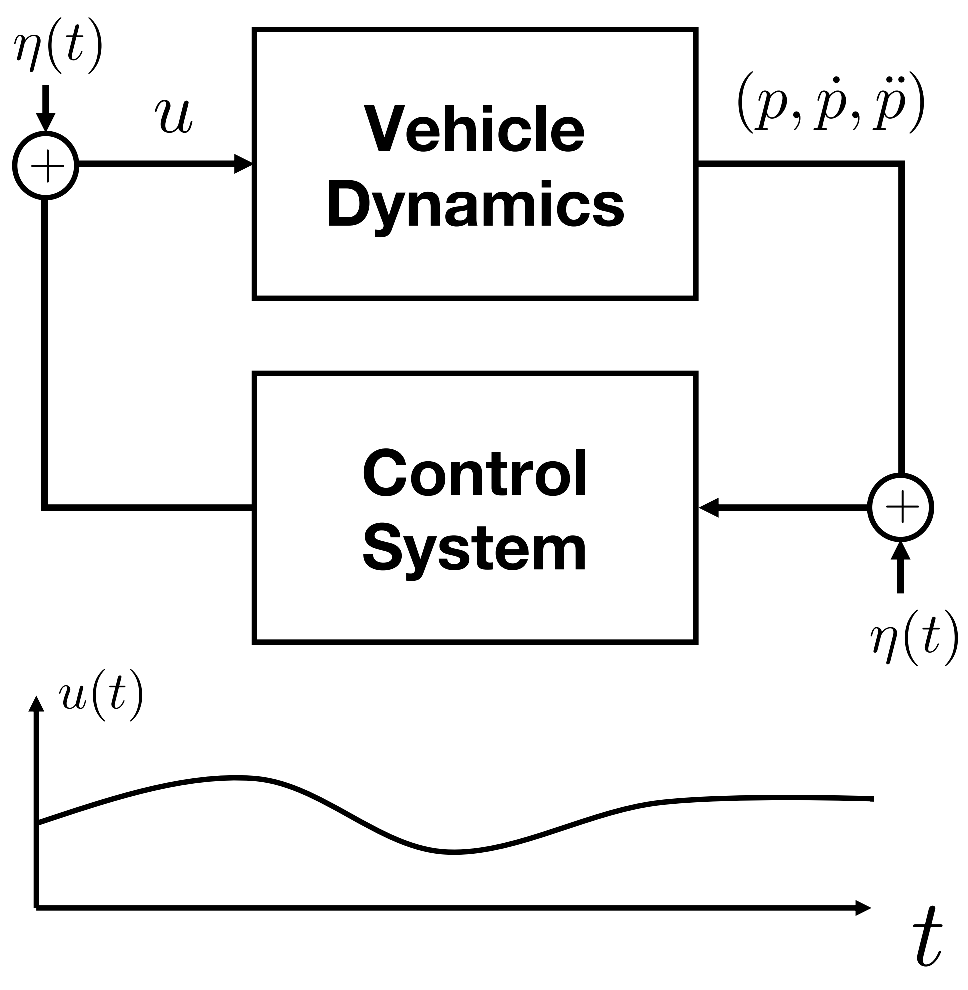


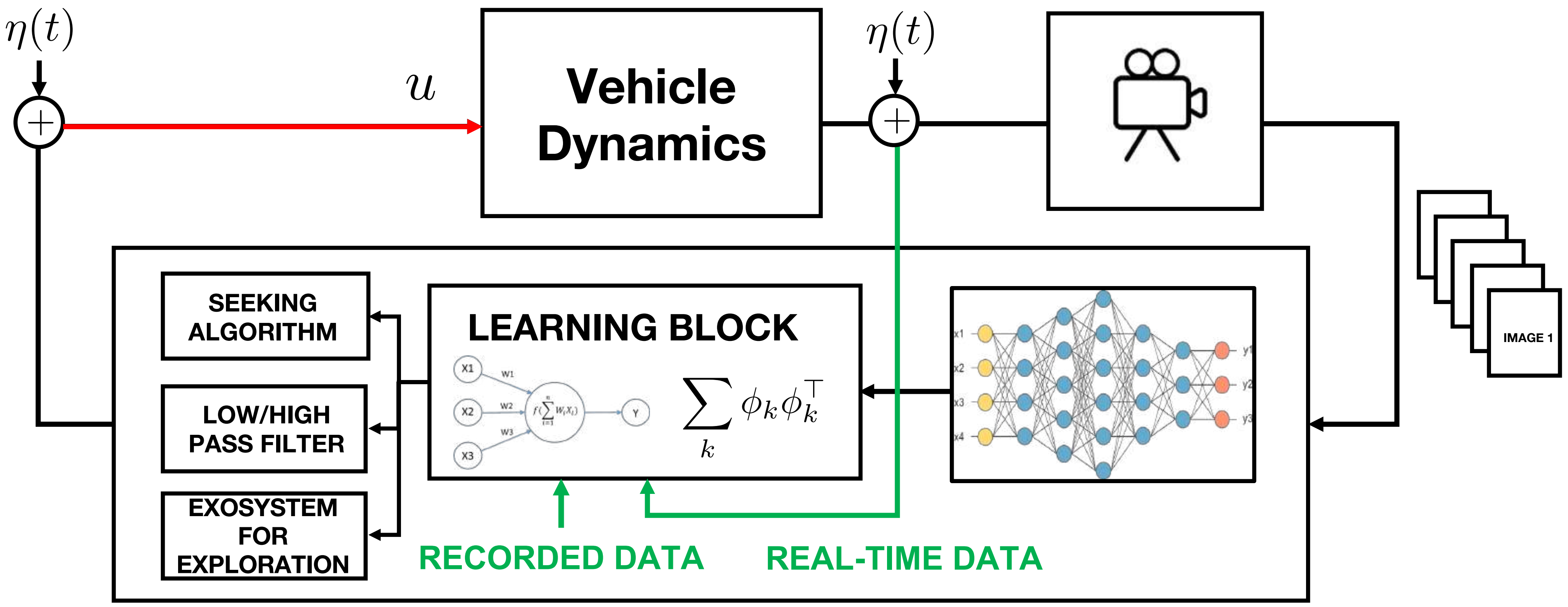


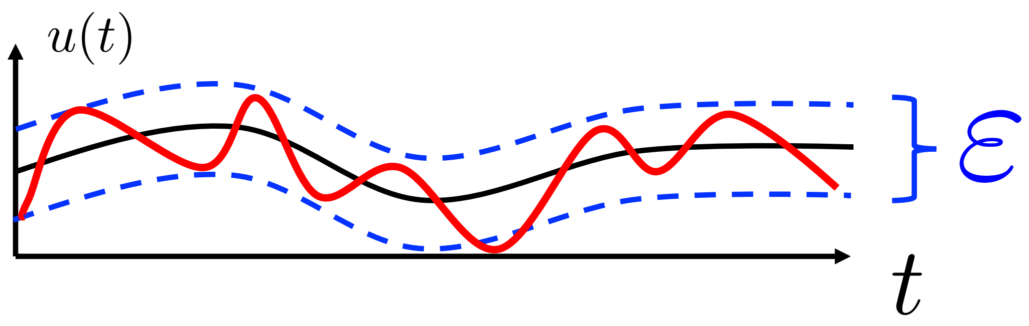


Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations







Feedback Loops Should be Intrinsically Robust:

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

STABLE NOMINAL SYSTEMS

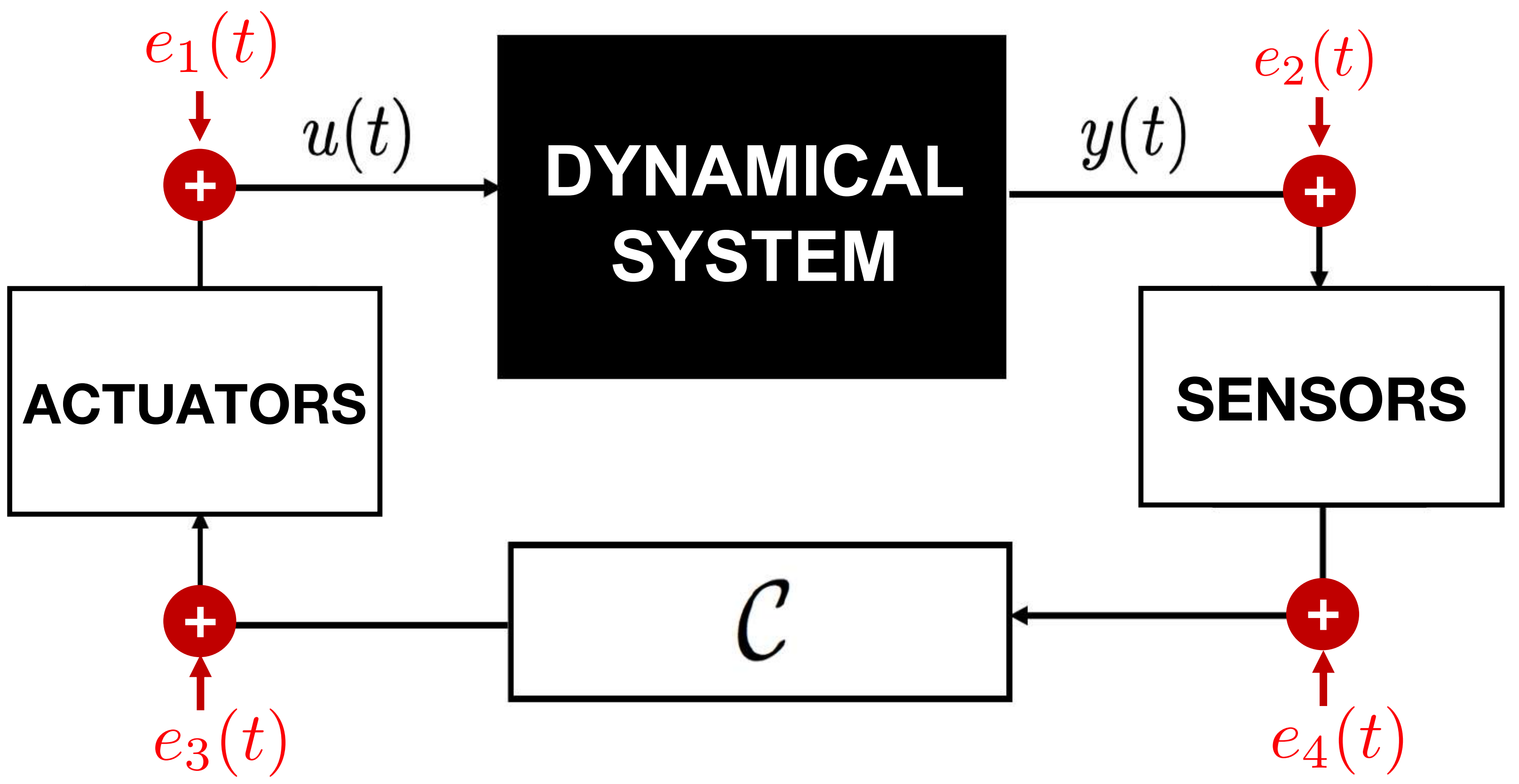
$$\dot{x} = f(x + e) + e \qquad \sup_t |e(t)| \le \varepsilon$$

"PRACTICAL STABILITY"

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

Feedback control systems can be seen as real-time decision-making algorithms operating in dynamic environments under perturbations:



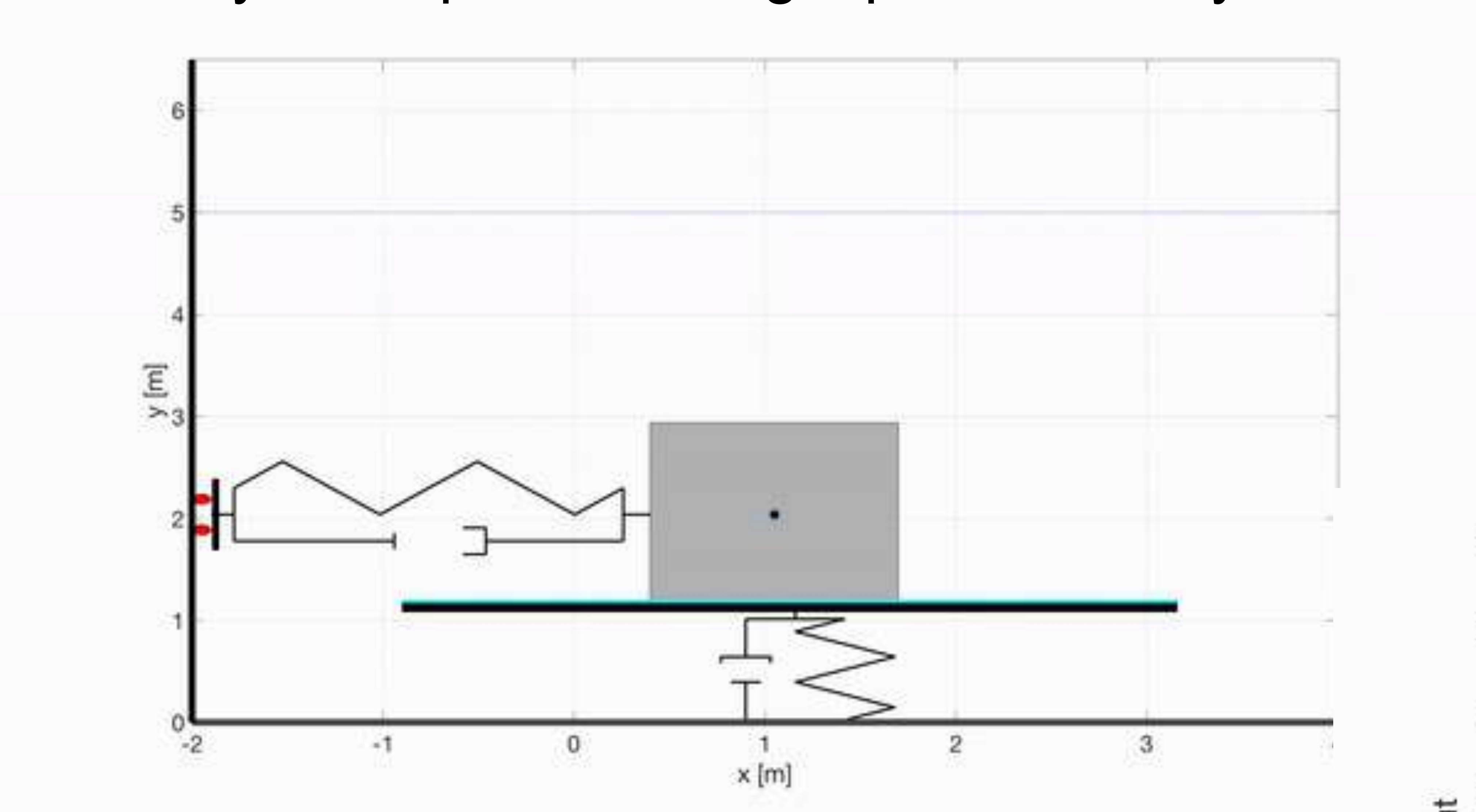
Typical decision-making problems include:

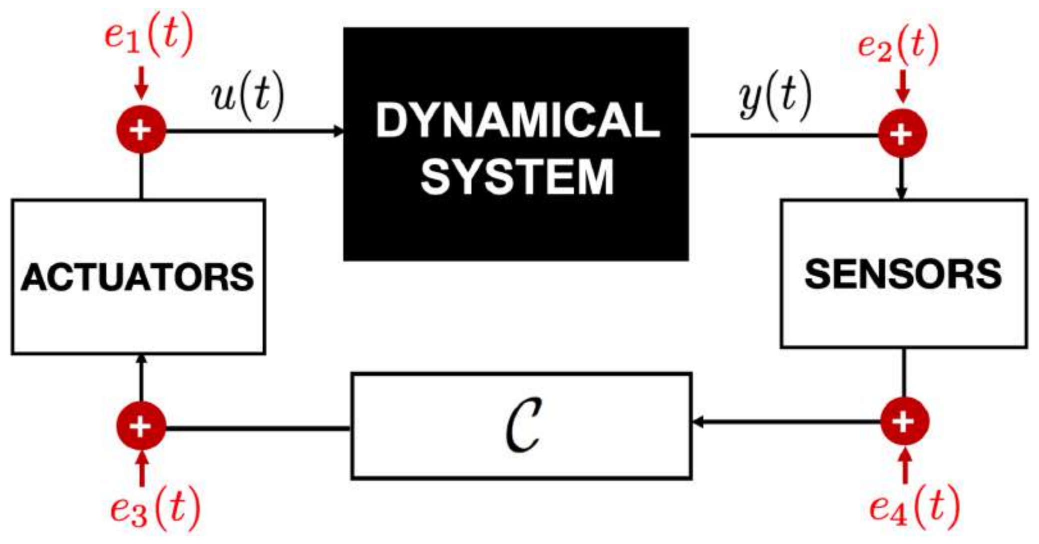
- ➢ Steady-state performance optimization.
- ➢ Optimal regulation.
- ➢ Adaptive control and optimization.
- ➢ Real-time coordination in multi-agent systems.

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

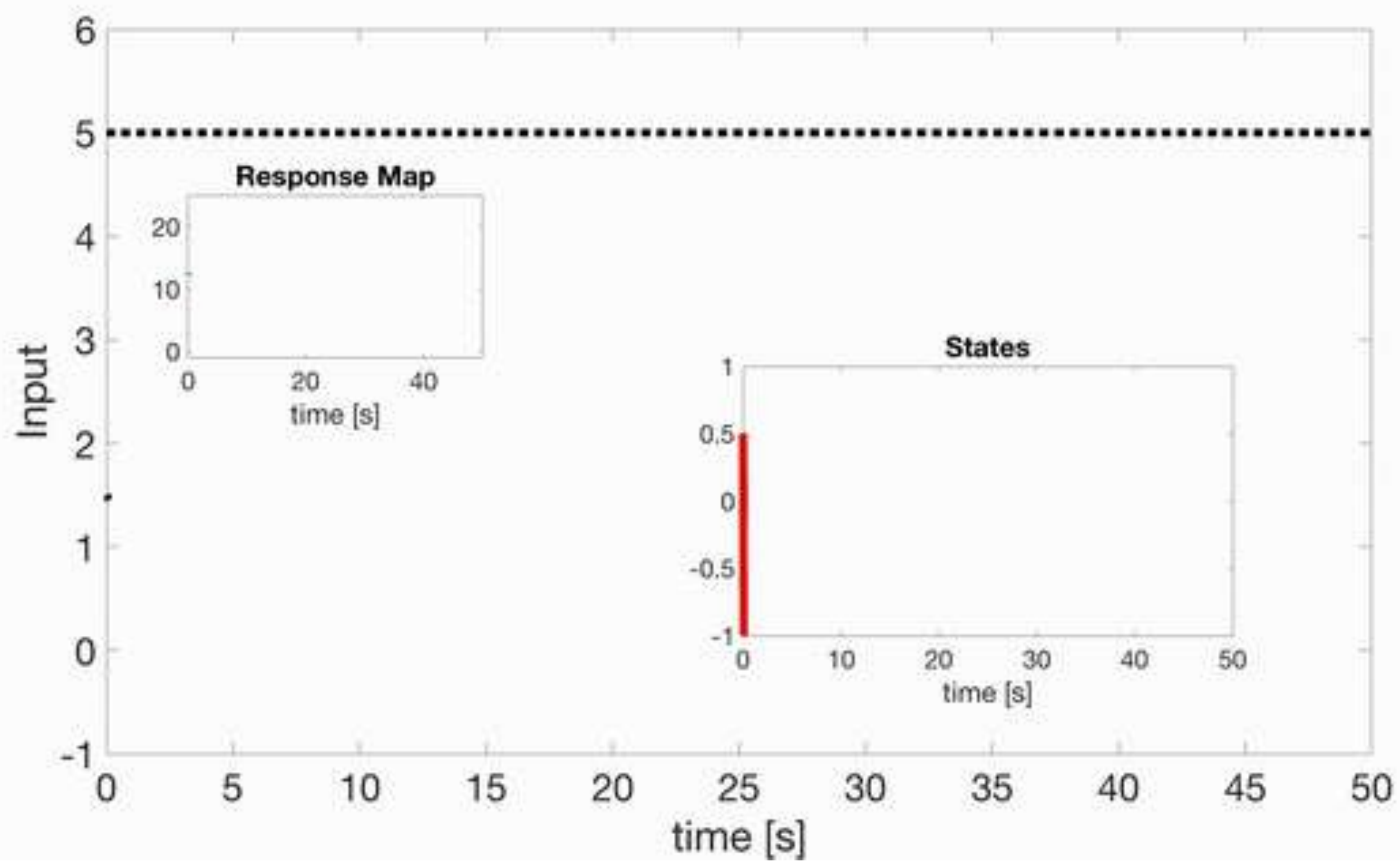
Toy Example: Learning Optimal Steady-State Control for a Mass in a Mobile Platform





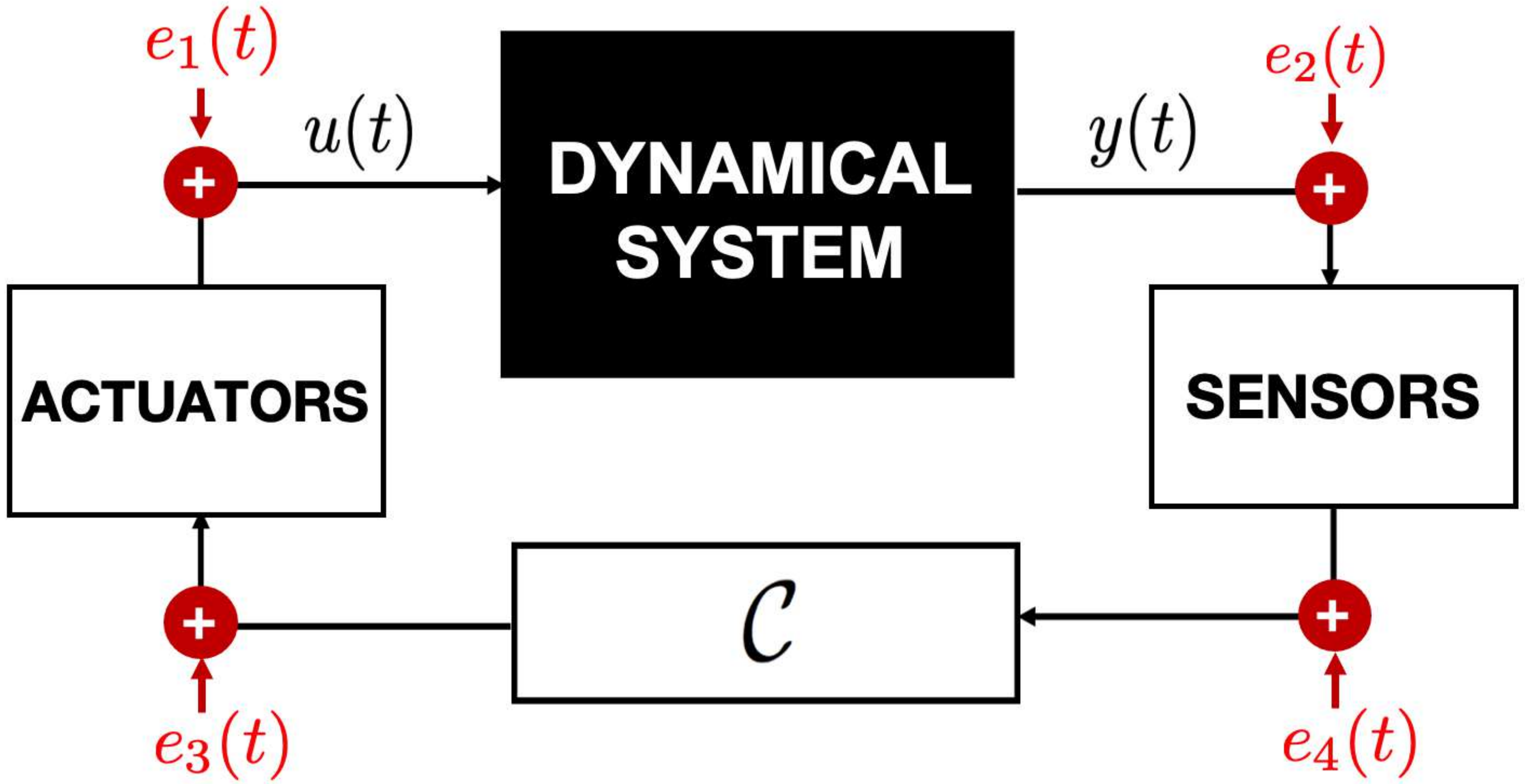
Equations given by the physics of the system

$$\dot{x}_1 = x_2$$
$$\dot{x}_3 = -f(x)\ sign(x_2) - bx_1$$
$$\dot{x}_3 = x_4$$
$$\dot{x}_4 = -cx_4 - dx_3 + u$$



Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

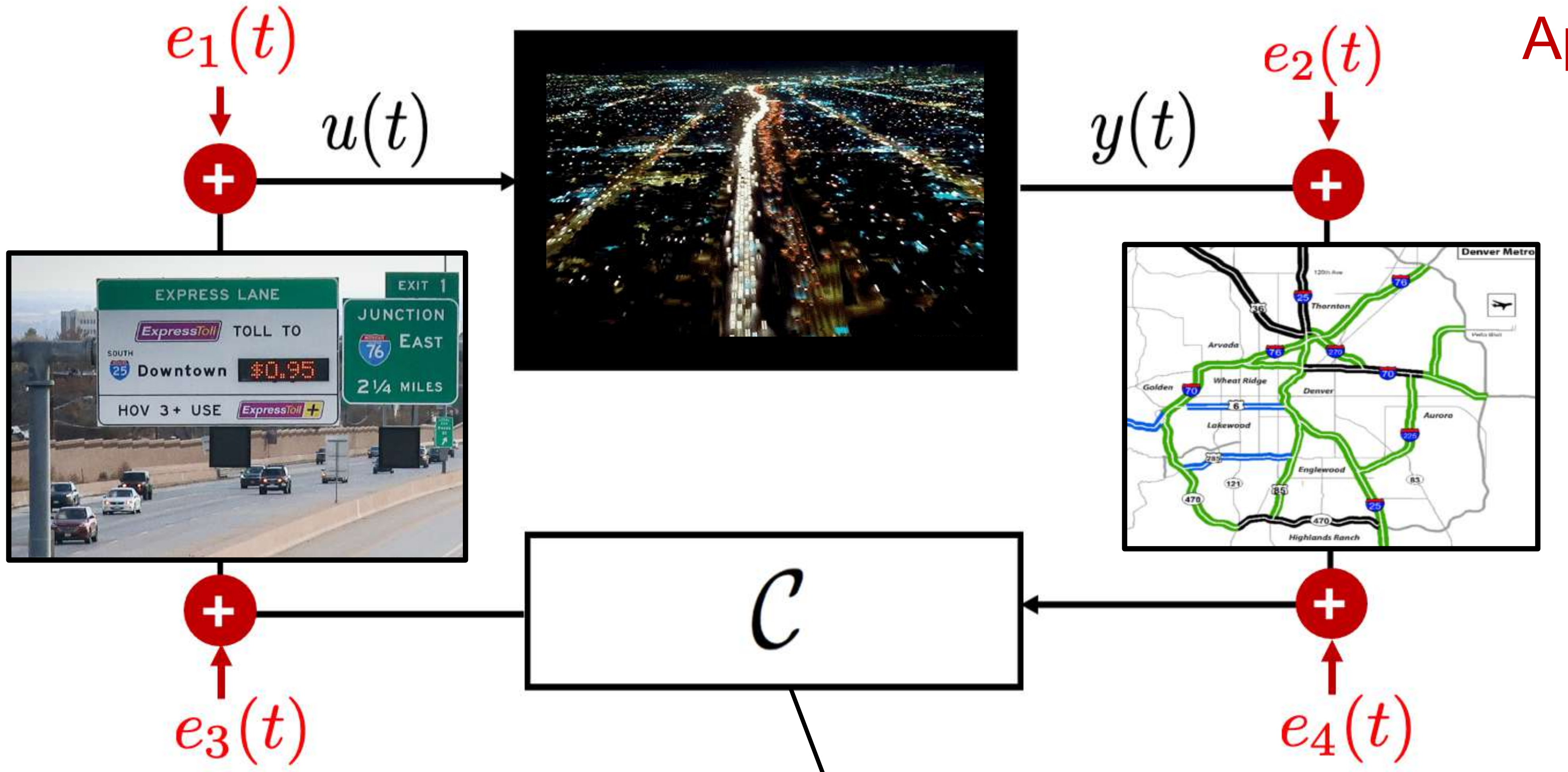


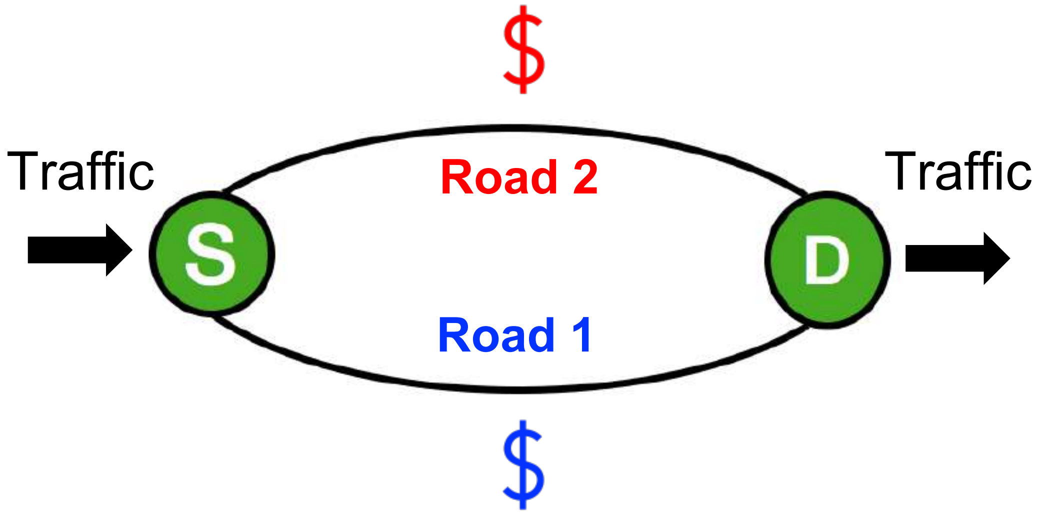
Applications in many domains:

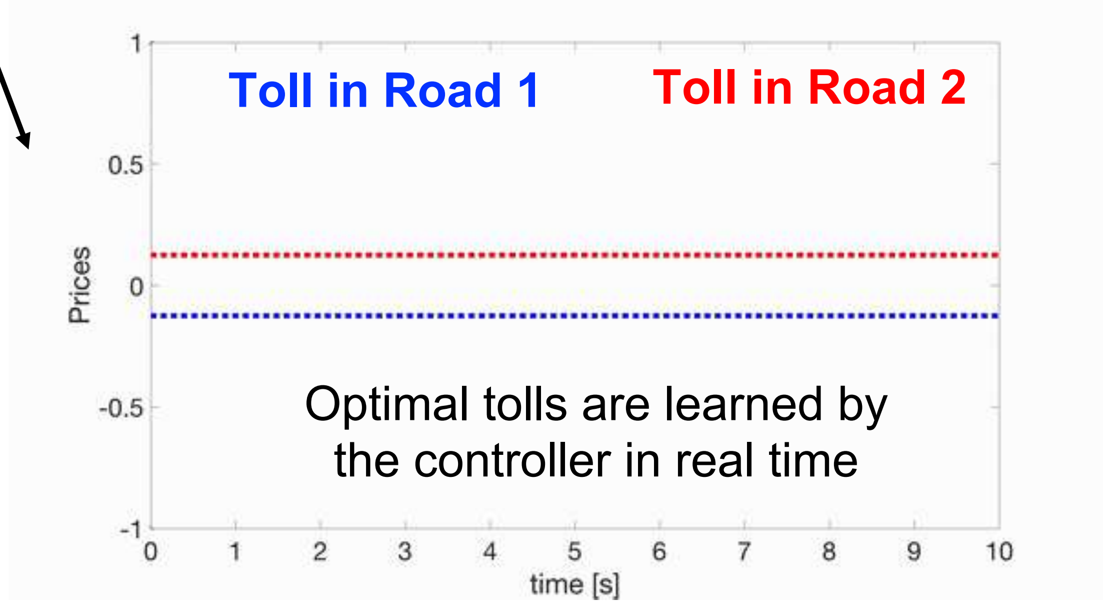
- ➢ Transportation systems (dynamic pricing and congestion control).

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

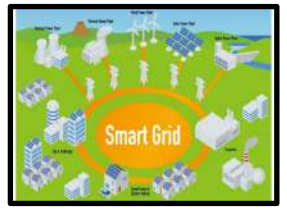




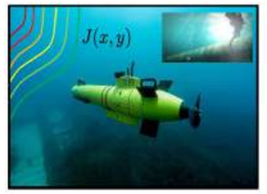


Applications in many domains:

- Transportation systems (dynamic pricing and congestion control).
- Power systems (voltage control, optimal power dispatch, wind turbines, photovoltaic systems).



- Robotic systems (source seeking, formation control).



- Manufacturing systems (semiconductor industry, industrial motion systems).

Data-Driven Robust Learning:

In this talk, we will use **differential equations** to model our algorithms:

Initial time Initial condition

$$\frac{dx(t)}{dt} = f(x(t), t), \qquad x(t_0) = x_0 \qquad (1)$$

Here, $f(\cdot,\cdot)$ is some function that we will design. Generally, this function is continuous, but this is not always required.

This equation does not mean anything unless we specify what do we mean by a “solution” (aka trajectory):

$$x(t) = x(t_0) + \int_{t_0}^{t} f(x(\tau), \tau)d\tau$$

Any continuously differentiable function $x(t)$ that satisfies this equation for all time t will be called a solution to (1).

Initial condition

Initial time

Example: $x(t) = t$ satisfies $\frac{dx(t)}{dt} = 1, \quad x(t_0) = 0, \quad t_0 = 0.$

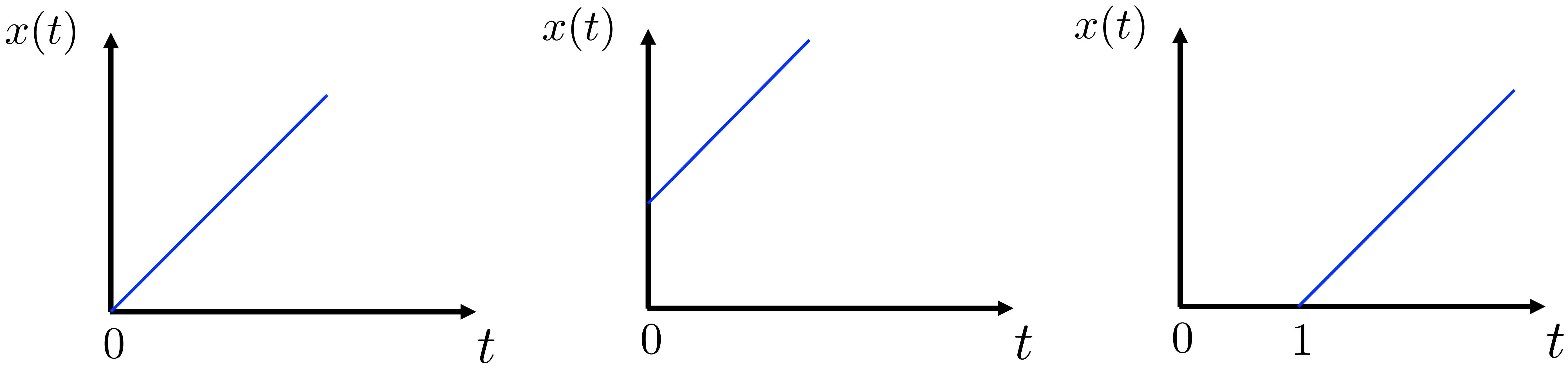
Data-Driven Robust Learning:

Note that, if we let the solutions to be parameterized by (x_0, t_0), then the differential equation describes a family of trajectories:

$$\frac{dx(t)}{dt} = 1, \quad x(t_0) = x_0 \quad \Rightarrow \quad x(t) = x_0 + \int_{t_0}^{t} 1 d\tau = x_0 + t - t_0$$

$$x(t) = x_0 + t - t_0$$

We can plot the function $x(t)$ to get an idea of the qualitative behavior of the trajectories of the system:

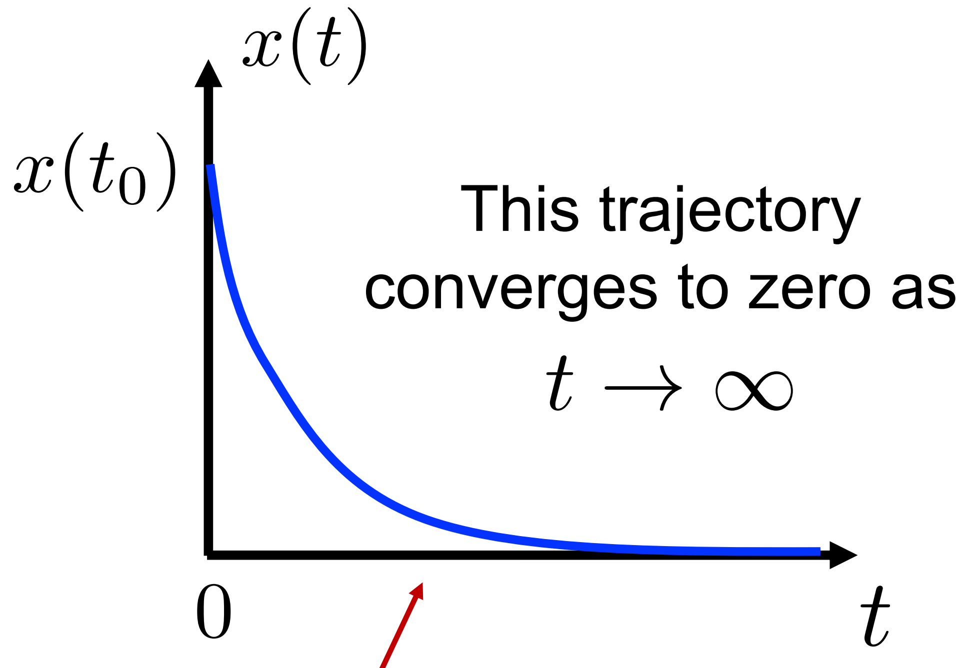


Data-Driven Robust Learning:

To simplify the notation, we will simply write differential equations as:

$$\dot{x} = f(x, t), \quad x(t_0) = x_0$$

How is this related to algorithms, estimation dynamics, and data?



Let's consider the following differential equation:

$$\dot{x} = -x$$

We can explicitly find the family of solutions:

$$\frac{dx}{x} = -dt \implies \ln(x) - \ln(x_0) = -(t - t_0) \implies \ln\left(\frac{x}{x_0}\right) = -(t - t_0)$$

$$\frac{x}{x_0} = e^{-(t-t_0)} \implies \boxed{x(t) = x(t_0)e^{-(t-t_0)}}$$

Data-Driven Robust Learning:

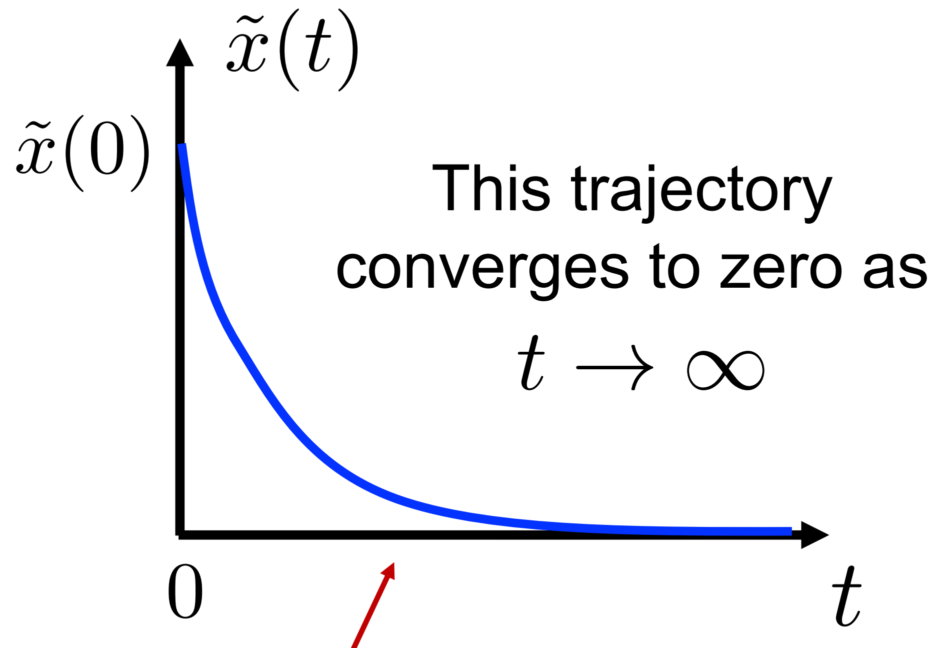
To simplify the notation, we will simply write differential equations as:

$$\dot{x} = f(x, t), \quad x(t_0) = x_0$$

How is this related to algorithms, estimation dynamics, and data?

Let's consider the following differential equation:

$$\dot{x} = -(x - x^*) \qquad x^* \in \mathbb{R}$$



Note that if I use: $\tilde{x} = x - x^* \implies \dot{\tilde{x}} = \dot{x} - \dot{x}^*$

Then, $\dot{\tilde{x}} = \dot{x} = -(x - x^*) = -\tilde{x}$

If $\tilde{x}(t) \to 0$ then $x(t) \to x^*$

$$\boxed{\tilde{x}(t) = \tilde{x}(t_0)e^{-(t-t_0)}}$$

Data-Driven Robust Learning:

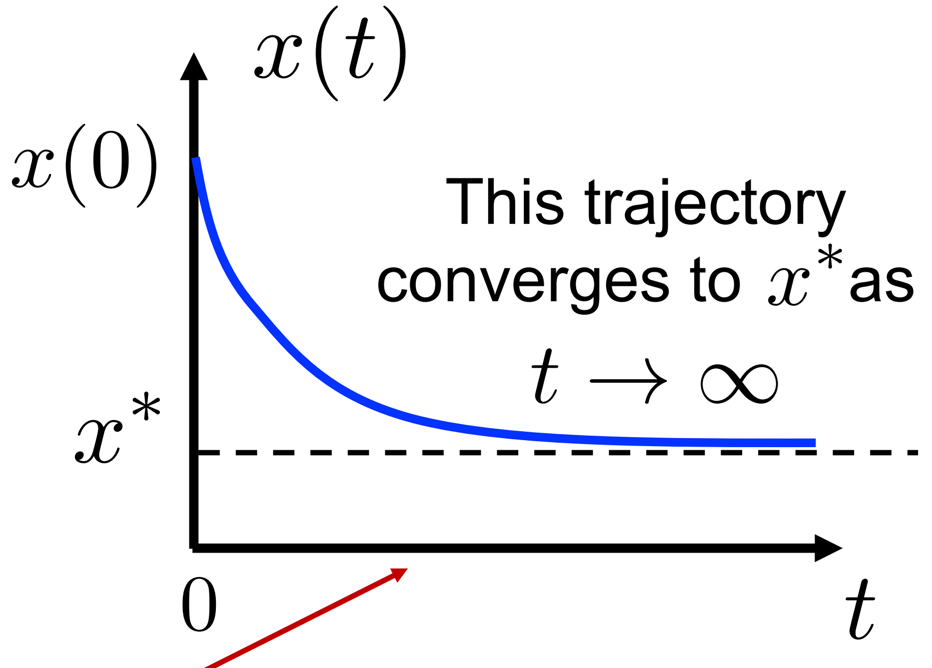
To simplify the notation, we will simply write differential equations as:

$$\dot{x} = f(x,t), \quad x(t_0) = x_0$$

How is this related to algorithms, estimation dynamics, and data?

Let's consider the following differential equation:

$$\dot{x} = -(x - x^*) \qquad x^* \in \mathbb{R}$$



Note that if I use: $\tilde{x} = x - x^* \implies \dot{\tilde{x}} = \dot{x} - \dot{x}^*$

Then, $\dot{\tilde{x}} = \dot{x} = -(x - x^*) = -\tilde{x}$

If $\tilde{x}(t) \to 0$ then $x(t) \to x^*$

$$\boxed{x(t) = (x(t_0) - x^*)e^{-(t-t_0)} + x^*}$$

Data-Driven Robust Learning:

To simplify the notation, we will simply write ordinary differential equations (ODEs) as:

$$\dot{x} = f(x, t), \quad x(t_0) = x_0$$

How is this related to algorithms, estimation dynamics, and data?

This behavior also holds if x is a vector, and the dynamics are pre-multiplied by a positive definite symmetric matrix:

$$\dot{x} = -Q(x - x^*) \qquad x^* \in \mathbb{R}$$

Note that the right-hand side of this equation is nothing but the gradient of a quadratic cost function:

$$J(x) = \frac{1}{2}(x - x^*)^\top Q(x - x^*)$$

And, in the general form, the dynamics can be written as:

$$\dot{x} = -\nabla J(x)$$

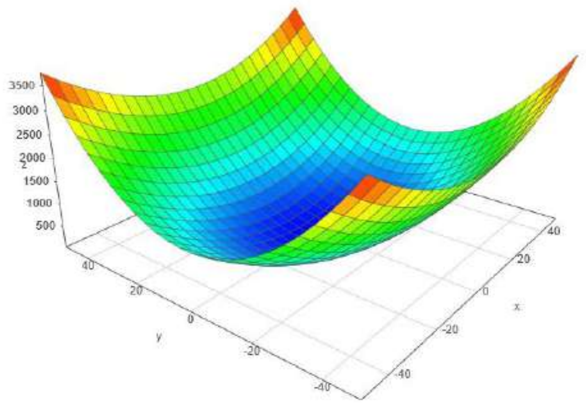
Data-Driven Robust Learning:

Theorem: For any smooth cost function J, with bounded level sets, and such that

$$\nabla J(x^*) = 0 \iff x^* \in \arg\ \min J(x)$$

the gradient-descent ODE:

$$\dot{x} = -\nabla J(x)$$



Generates solutions that converge to the set of minimizers of J.

Again: How is this result related to algorithms, estimation dynamics, and data?

The above result states a mathematical property of every trajectory generated by the ODE.

But these trajectories can be computed numerically using integration/discretization techniques.

Data-Driven Robust Learning:

$$\frac{dx(t)}{dt} = -\nabla J(x(t))$$

Simplest approach: Euler discretization

$$\frac{dx(t)}{dt} \approx \frac{x(t+\Delta T) - x(t)}{\Delta T} \qquad \frac{x(t+\Delta T) - x(t)}{\Delta T} = -\nabla J(x(t))$$

$$x(t+\Delta T) = x(t) + \underbrace{\Delta T}_{\text{Step Size}} \nabla J(x(t))$$

We can now run an "algorithm" that generates updates $x(\Delta T)$ $x(2\Delta T)$ $x(3\Delta T)$ that approximates the gradient-descent ODE:

Algorithm 1: "Hello World": Gradient Descent

$x(0) = x_0$

for i=0:1000

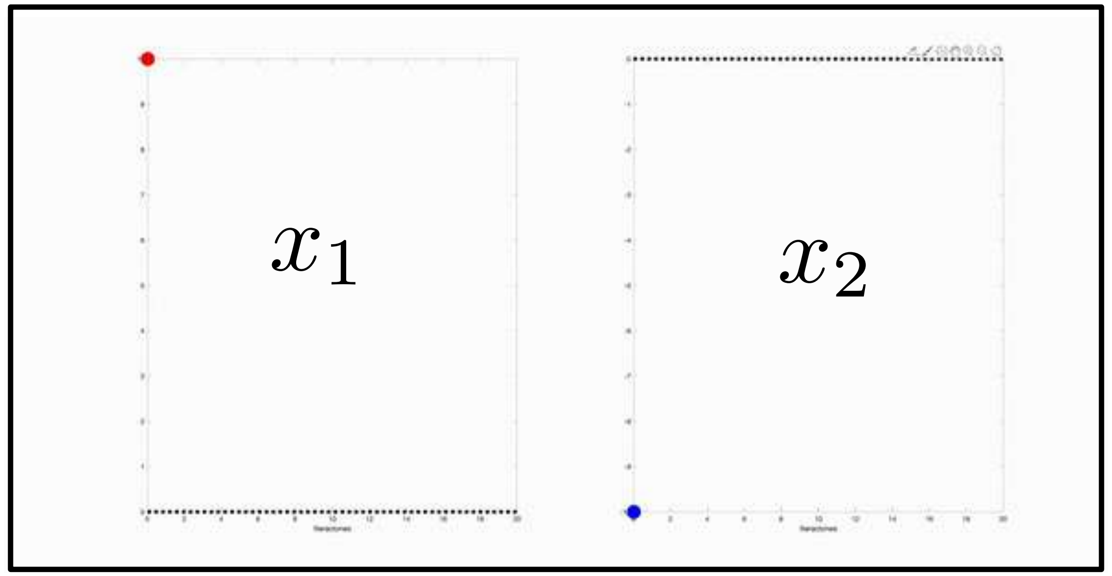
$$x((i+1)\Delta T) = x(i\Delta T) + \Delta T \nabla J(x(i\Delta T))$$

end

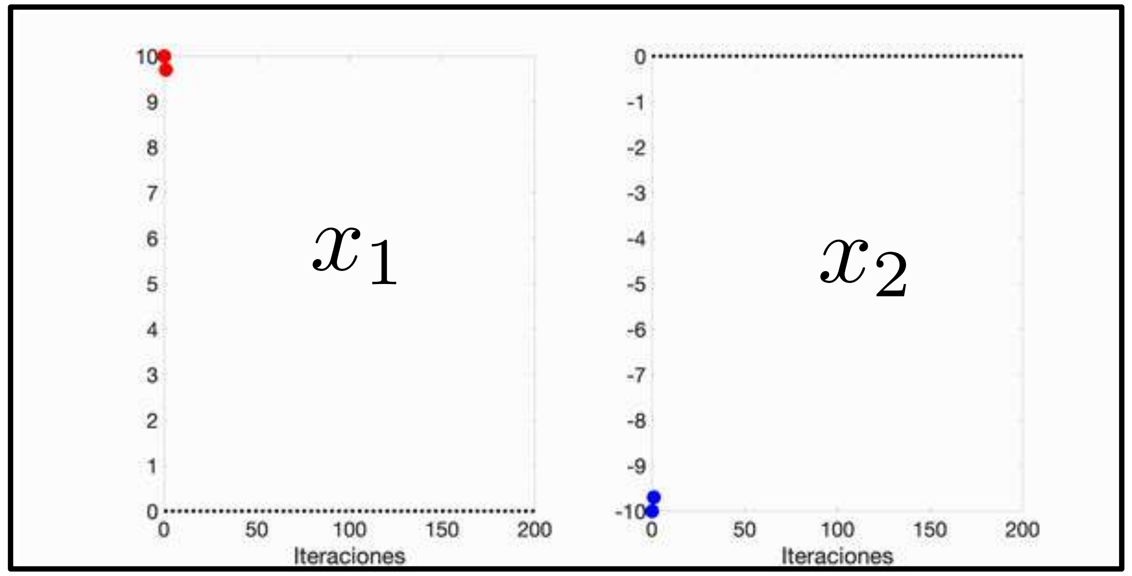
Data-Driven Robust Learning:

$$J(x) = \frac{1}{2} x^\top Q x$$

$$x \in \mathbb{R}^2$$

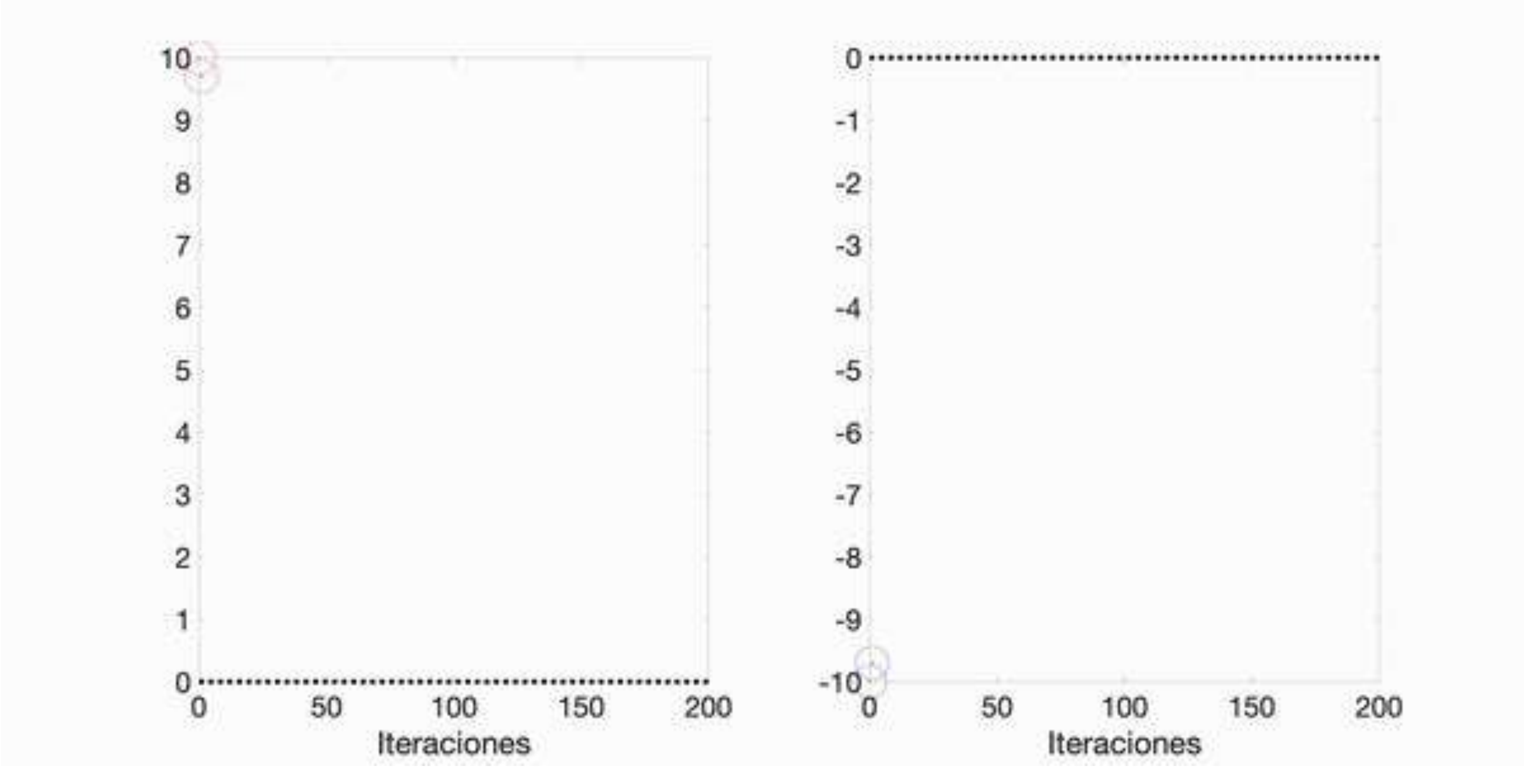


$$\Delta T = 0.1$$



$$\Delta T = 0.01$$

And as $\Delta T \to 0^+$ we recover the trajectories of the original ODE:

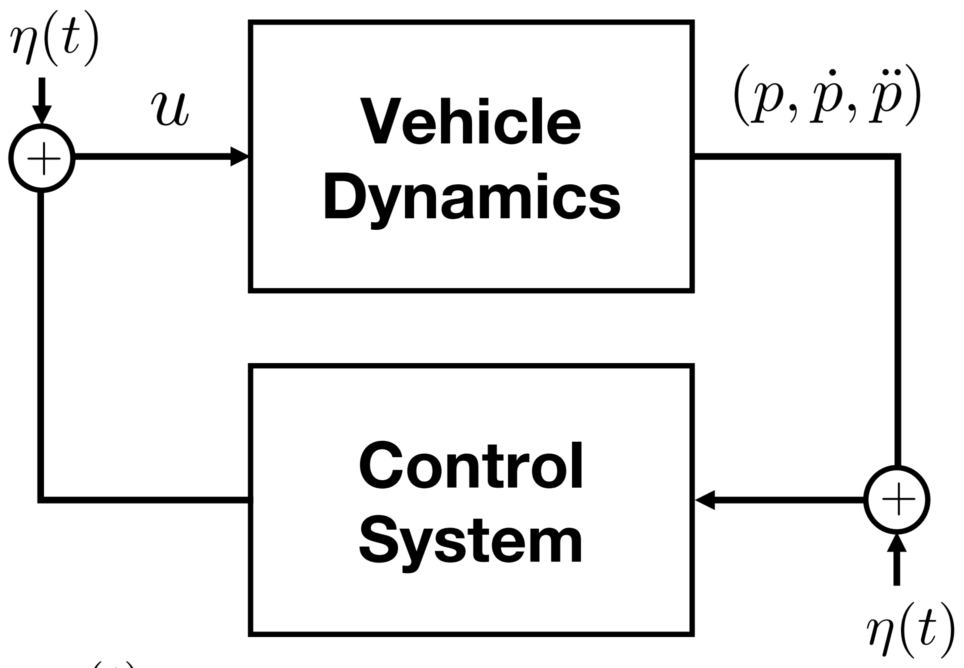


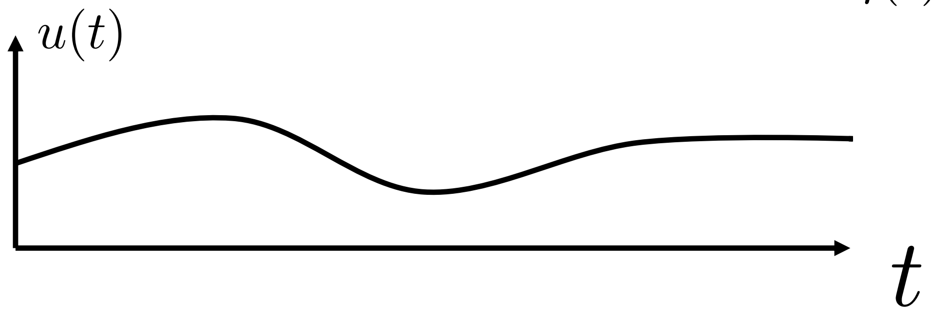
ODEs provide a good mathematical framework for the analysis and design of algorithms.

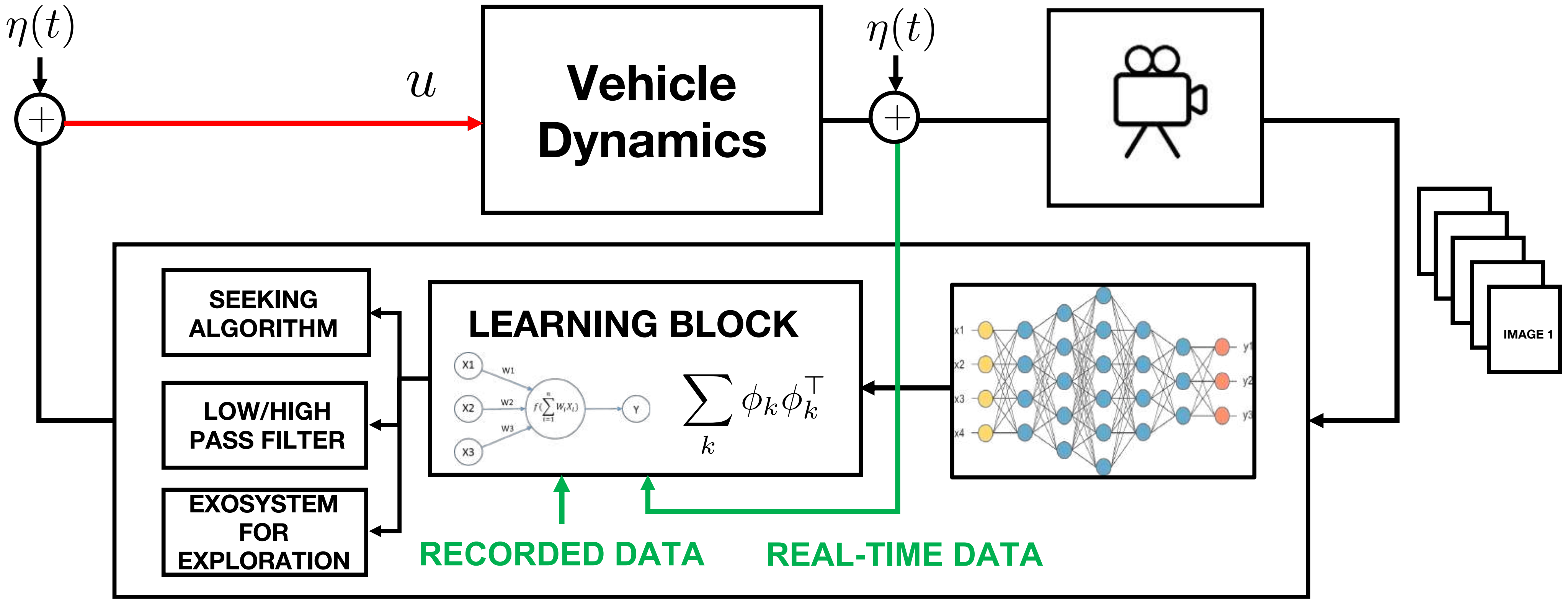
Area that studies how to “shape” or “manipulate” ODEs (and dynamical systems): **Control Theory**

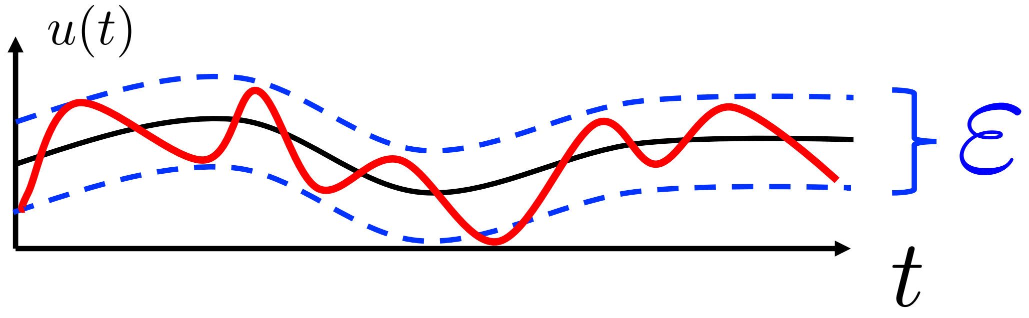
Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations









Feedback Loops Should be Intrinsically Robust:

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

STABLE NOMINAL SYSTEMS

$$\dot{x} = f(x + e) + e \qquad \sup_t |e(t)| \leq \varepsilon$$

"PRACTICAL STABILITY"

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

Typical problem: Learn an unknown parameter θ^* using real-time measurements and/or recorded data

$$y(t) = \phi(t)^\top \theta^*$$

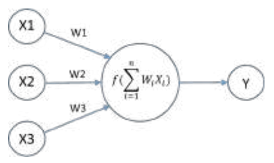
Real-Time Measurements

Known regressor

Unknown parameter

$\hat{\theta}(t) \approx \theta^*$

LEARNING BLOCK



$\sum_k \phi_k \phi_k^\top$

RECORDED DATA

REAL-TIME DATA

We can try to use a gradient-descent approach to solve this problem:

$$\hat{y} = \phi(t)^\top \hat{\theta} \qquad e = \hat{y} - y \qquad J(\hat{\theta}) = \frac{1}{2} e(\hat{\theta})^2$$

To minimize J, we use:

$$\dot{\hat{\theta}} = -\nabla J(\hat{\theta}) = -e(\hat{\theta}) \frac{\partial e}{\partial \hat{\theta}} = -e(\hat{\theta}) \phi(t)$$

Does this work?

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

$$y(t) = \phi(t)^\top \theta^* \qquad \hat{y} = \phi(t)^\top \hat{\theta} \qquad e = \hat{y} - y \qquad J(\hat{\theta}) = \frac{1}{2} e(\hat{\theta})^2$$

$$\begin{aligned}\dot{\hat{\theta}} = -e(\hat{\theta})\phi(t) \quad = -(\hat{y} - y)\phi(t) \quad &= -\phi(t)(\hat{y} - y) \\ &= -\phi(t)(\phi(t)^\top \hat{\theta} - \phi(t)^\top \theta^*) \\ &= -\phi(t)\phi(t)^\top (\hat{\theta} - \theta^*)\end{aligned}$$

We can now consider the error dynamics: $\tilde{\theta} := \hat{\theta} - \theta^* \implies \dot{\tilde{\theta}} = \dot{\hat{\theta}} = -\phi(t)\phi(t)^\top \tilde{\theta}$

$$\dot{\tilde{\theta}} = -\underbrace{\phi(t)\phi(t)^\top}_{Q(t)} \tilde{\theta} \qquad \implies \qquad \dot{\tilde{\theta}} = -Q(t)\tilde{\theta}$$

Does this work?

$$\phi(t) = 0, 1, \sin(t),$$

Depends on this signal

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

$$y(t) = \phi(t)^\top \theta^* \qquad \hat{y} = \phi(t)^\top \hat{\theta} \qquad e = \hat{y} - y \qquad J(\hat{\theta}) = \frac{1}{2} e(\hat{\theta})^2$$

$$\dot{\tilde{\theta}} = -Q(t)\tilde{\theta}$$

Intuitively, we want the matrix Q(t) to be positive definite "on average":

$$\int_t^{t+T} Q(\tau)d\tau \succ \beta I, \qquad \forall\, t \geq 0$$

For our problem, this is equivalent to:

$$\int_t^{t+T} \phi(\tau)\phi(\tau)^\top d\tau \succ \beta I, \quad \forall\, t \geq 0$$

This property is called "persistence of excitation (PE)"

Theorem: If $\phi(t)$ is PE, then all the trajectories of the following system converge to zero exponentially fast from any initial point:

$$\dot{\tilde{\theta}} = -\phi(t)\phi(t)^\top \tilde{\theta}$$

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

Typical problem: Learn an unknown parameter θ^* using real-time measurements and/or recorded data

$$y(t) = \phi(t)^\top \theta^*$$

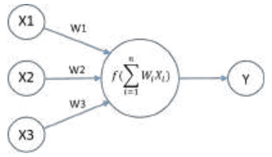
Real-Time Measurements

Known regressor

Unknown parameter

$\hat{\theta}(t) \approx \theta^*$

LEARNING BLOCK



$\sum_k \phi_k \phi_k^\top$

RECORDED DATA

REAL-TIME DATA

$$\hat{y} = \phi(t)^\top \hat{\theta} \qquad e = \hat{y} - y \qquad J(\hat{\theta}) = \frac{1}{2} e(\hat{\theta})^2 \qquad \dot{\hat{\theta}} = -\nabla J(\hat{\theta}) = -e(\hat{\theta}) \frac{\partial e}{\partial \hat{\theta}} = -e(\hat{\theta})\phi(t)$$

Works if the regressors are PE!

What can we do if the regressors are not PE?

One Option: Use Recorded Data

We now assume access to a sequence of past data: $y(t_i) = \phi(t_i)^\top \theta^* \qquad i \in \{1, 2, \ldots, N\}$

$$\hat{y}_i = \phi(t_i)^\top \hat{\theta} \qquad e_i = \hat{y} - y_i, \quad J(\hat{\theta}) = \frac{k_r}{2} e(\hat{\theta})^2 + \frac{k_d}{2} \sum_{i=1}^{N} e_i(\hat{\theta})^2$$

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

$$y(t_i) = \phi(t_i)^\top \theta^* \qquad \hat{y}_i = \phi(t_i)^\top \hat{\theta} \qquad e_i = \hat{y} - y_i, \quad J(\hat{\theta}) = \frac{k_r}{2} e(\hat{\theta})^2 + \frac{k_d}{2} \sum_{i=1}^{N} e_i(\hat{\theta})^2$$

Using again gradient descent, we now obtain:

$$\boxed{\dot{\hat{\theta}} = -k_r e(\theta)\phi(t) - k_d \sum_{i=1}^{N} e_i(\theta)\phi_i(t)}$$

We call this dynamics a “concurrent learning” algorithm

To study the convergence we can use again the change of variable $\tilde{\theta} = \hat{\theta} - \theta^*$:

$$\dot{\hat{\theta}} = -k_r \phi(t)\phi(t)^\top (\hat{\theta} - \theta^*) - k_d \sum_{i=1}^{N} \phi(t_i)\phi(t_i)^\top (\hat{\theta} - \theta^*)$$

$$\dot{\tilde{\theta}} = -k_r \underbrace{\phi(t)\phi(t)^\top}_{Q(t)} \tilde{\theta} - k_d \underbrace{\sum_{i=1}^{N} \phi(t_i)\phi(t_i)^\top}_{D} \tilde{\theta} \quad \Rightarrow \quad \boxed{\dot{\tilde{\theta}} = -(k_r Q(t) + k_d D)\tilde{\theta}}$$

Data-Driven Robust Learning:

- Adversarial signals
- Noisy measurements
- Faulty actuators
- Unmodeled dynamics
- Discretization errors
- Numerical approximations

$$\dot{\tilde{\theta}} = -\underbrace{(k_r Q(t) + k_d D)}_{A(t)}\tilde{\theta}$$

$$\tilde{\theta} \in \mathbb{R}^n$$

$$Q(t) \succeq 0 \qquad k_r \geq 0$$

$$D \succ 0 \qquad k_d > 0$$

$$\Rightarrow \quad A(t) \succ 0$$

So, we have just learned that in order to make this algorithm work without PE assumptions on the regressor, we need the data to satisfy $D \succ 0$:

$$\dot{\tilde{\theta}} = -k_r \underbrace{\phi(t)\phi(t)^\top}_{Q(t)} \tilde{\theta} - k_d \underbrace{\sum_{i=1}^{N} \phi(t_i)\phi(t_i)^\top}_{D \succ 0} \tilde{\theta}$$

And note that:

$$\sum_{i=1}^{N} \phi(t_i)\phi(t_i)^\top \succ 0 \iff \mathrm{rank}(M) = n \quad \text{where} \quad M = [\phi(t_1), \phi(t_2), \ldots \phi(t_N)]^\top$$

This rank condition can be verified a priori once the data is selected.

An Illustrative Application:

Study: Denver had 21st-worst traffic congestion in U.S. last year

THE DENVER POST

Traffic congestion will cost NYC $100B by 2023

A new study from the Partnership for New York City breaks it down

The New York Times

Los Angeles Times

L.A.'s traffic congestion is world's worst for sixth straight year, study says

Data-Driven Robust Learning of Dynamic Tolls:

Los Angeles, CA

There's only one way to fix L.A.'s traffic, and it isn't Elon Musk's tunnels. We need tolls — lots of them

New York City, NY

NYC congestion pricing plan could mean $12 charge for driving in Manhattan

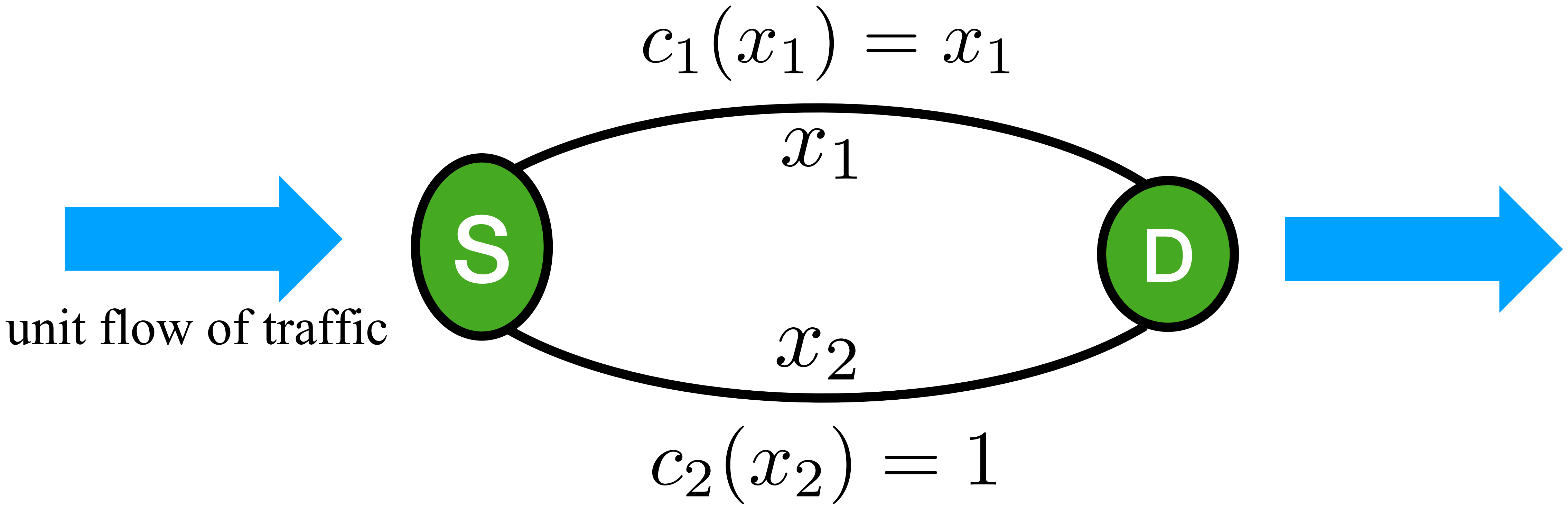
The proposed fee aims to abate traffic and raise funds for public transit fixes

Colorado

New I-70 Toll Lane Could Be Most Expensive per Mile in Nation

Aspen task force recommends congestion-base pricing, more transit to ease traffic woes

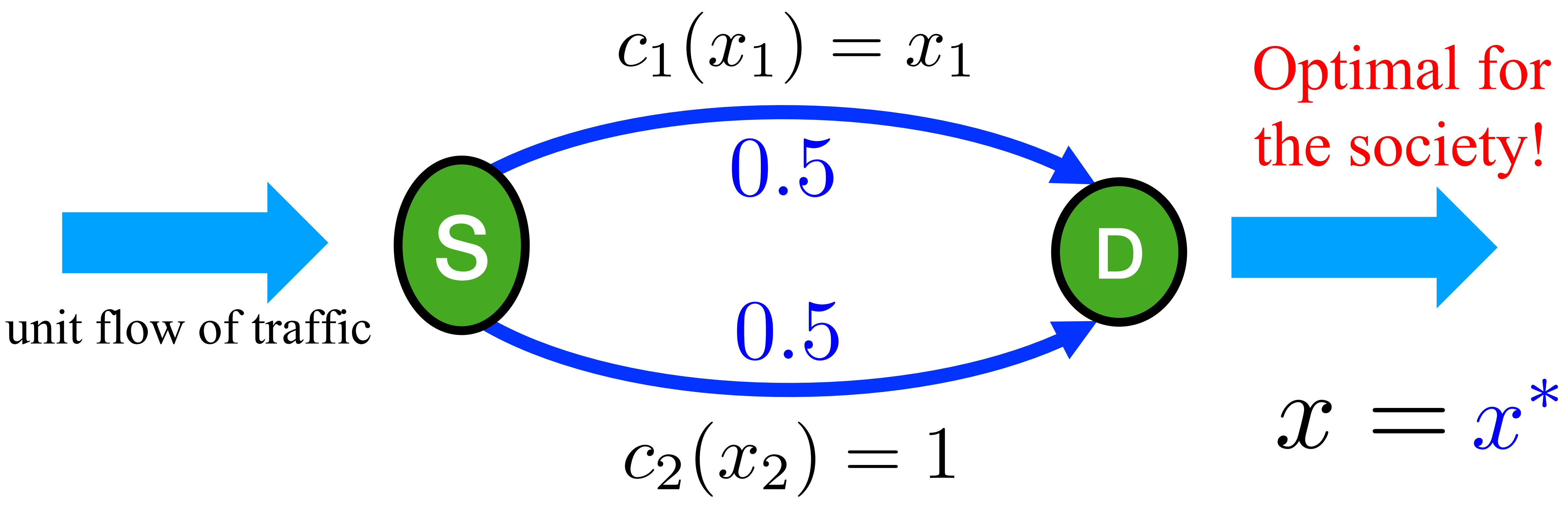
Data-Driven Robust Learning of Dynamic Tolls:



Social planner (e.g., mayor) wants to minimize the average delay in the network:

$$W(x) = c_1(x_1)x_1 + c_2(x_2)x_2$$

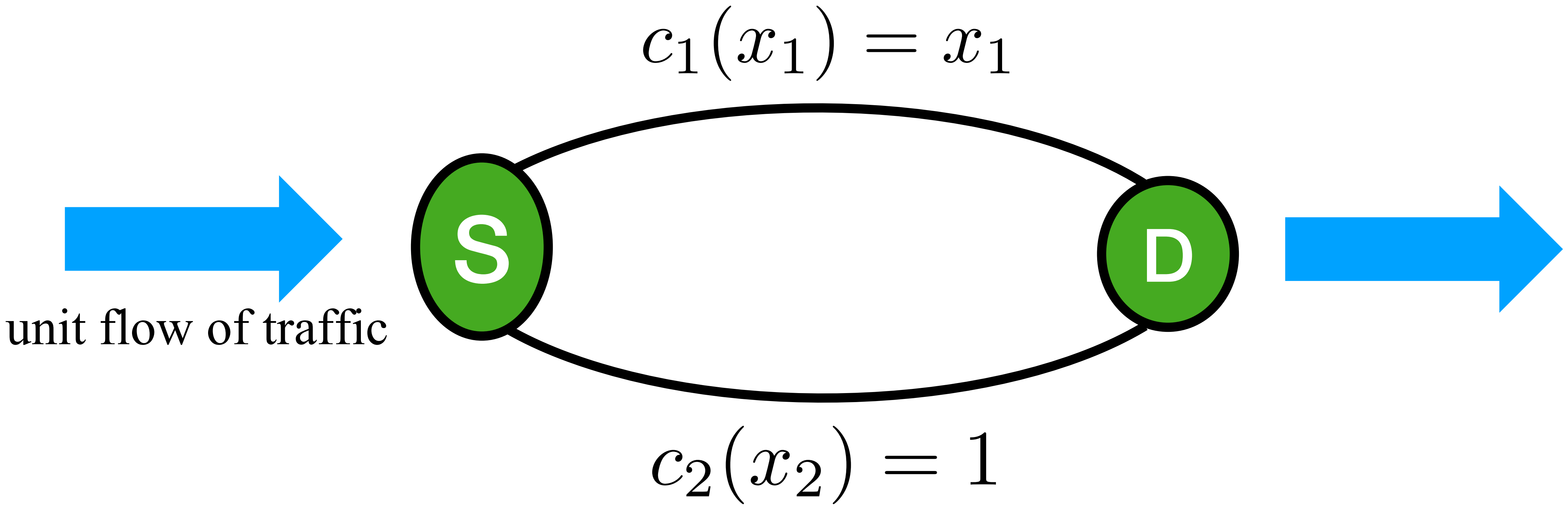
Data-Driven Robust Learning of Dynamic Tolls:



Social planner (e.g., mayor) wants to minimize the average delay in the network:

$$W(x) = c_1(x_1)x_1 + c_2(x_2)x_2$$
$$= (0.5 \times 0.5) + (1 \times 0.5) = 0.75$$

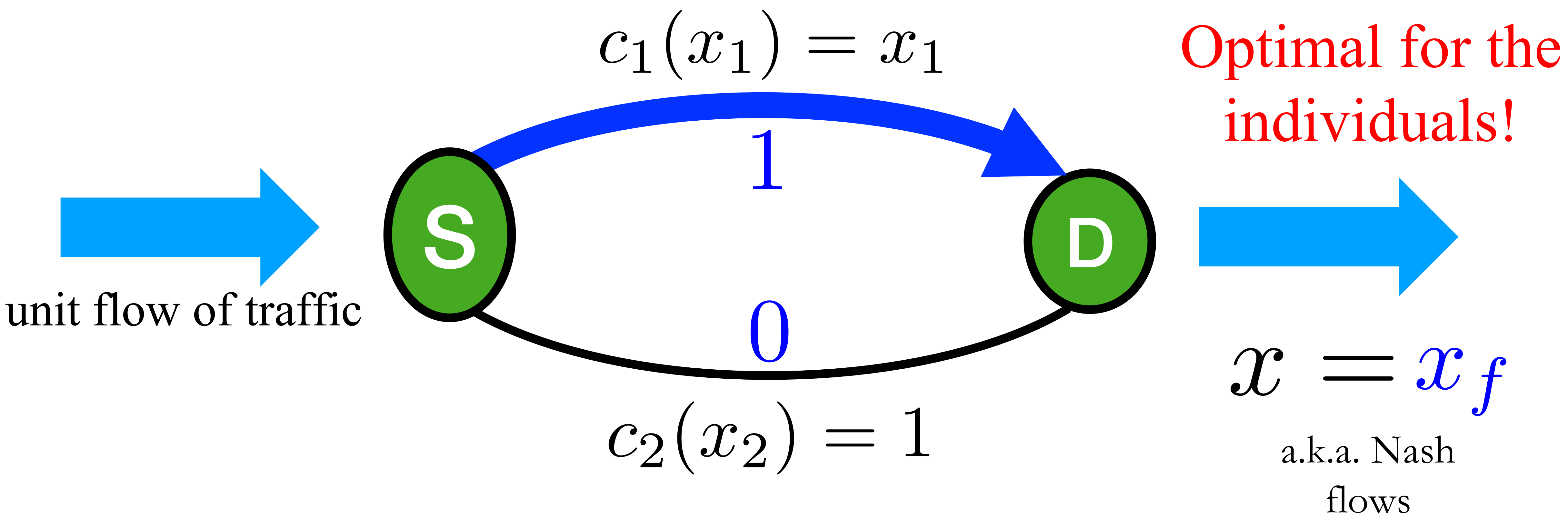
Data-Driven Robust Learning of Dynamic Tolls:



Social planner (e.g., mayor) wants to minimize the average delay in the network:

$$W(x) = c_1(x_1)x_1 + c_2(x_2)x_2$$

Data-Driven Robust Learning of Dynamic Tolls:

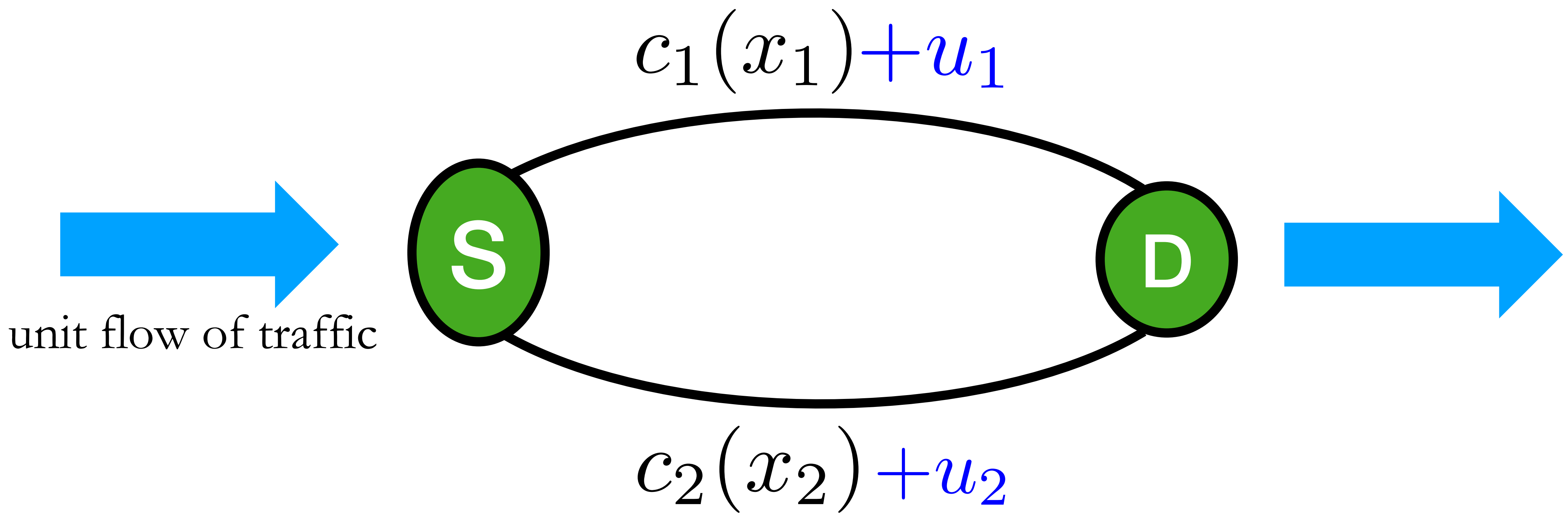


Social planner (e.g., mayor) wants to minimize the average delay in the network:

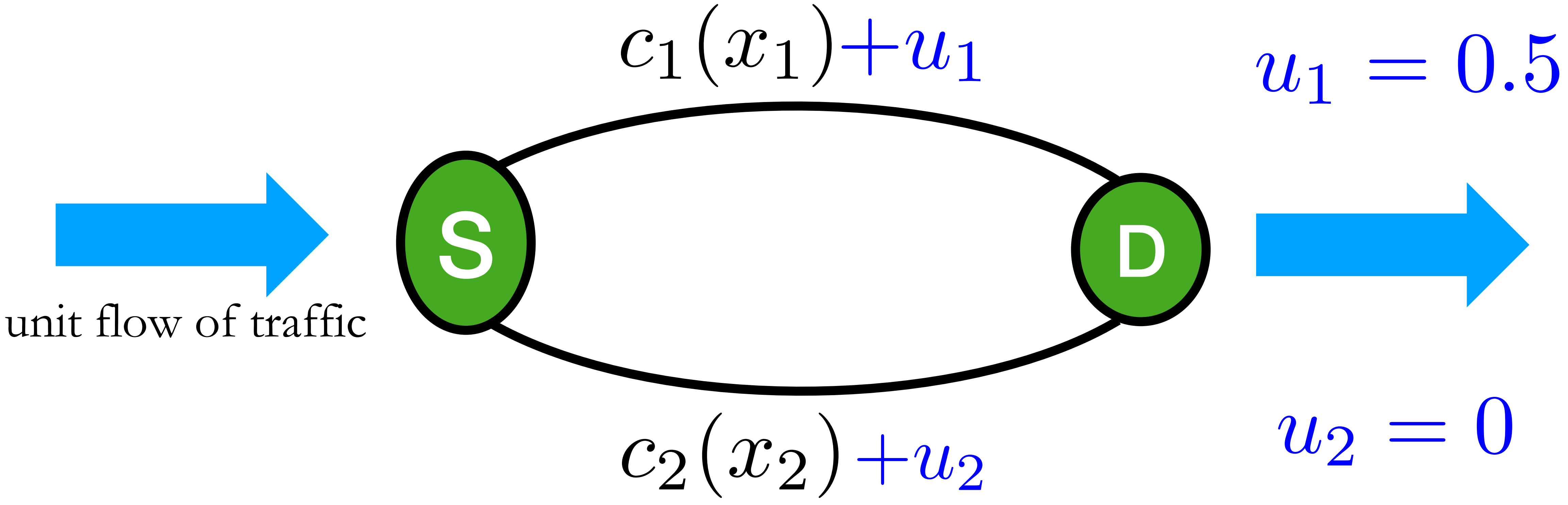
$$W(x) = c_1(x_1)x_1 + c_2(x_2)x_2$$
$$= (1 \times 1) + (1 \times 0) = 1 > 0.75$$

Moral: Selfish behavior can lead to poor social outcome.

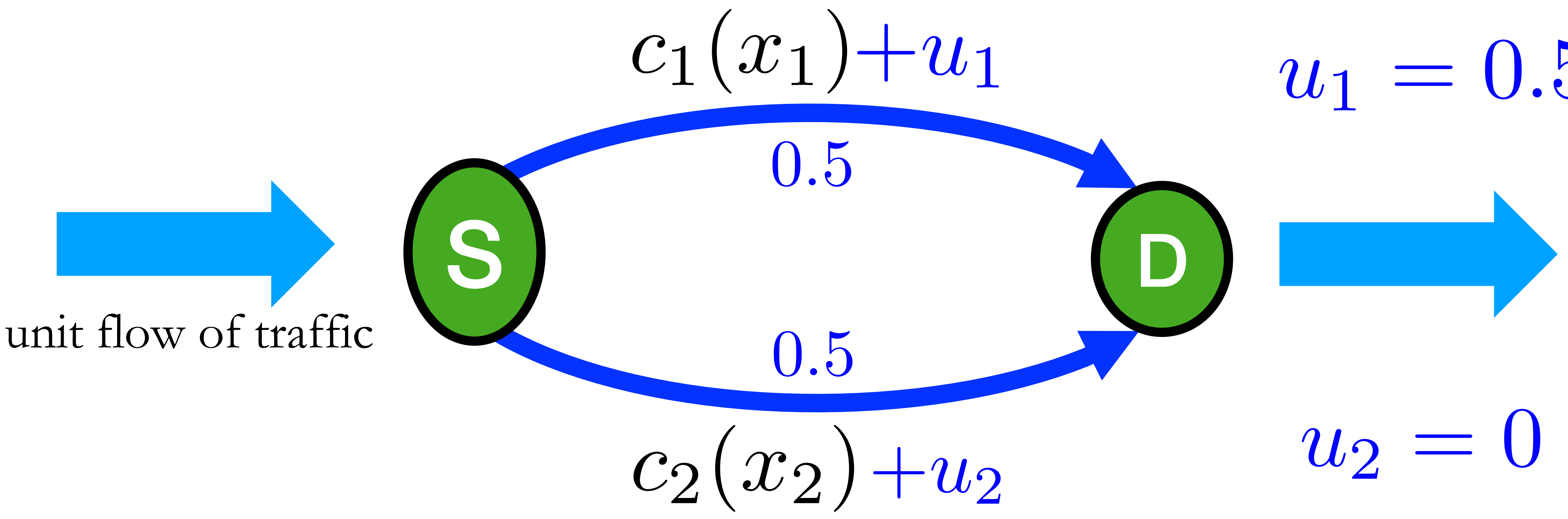
Data-Driven Robust Learning of Dynamic Tolls:



Data-Driven Robust Learning of Dynamic Tolls:



Data-Driven Robust Learning of Dynamic Tolls:



$u_1 = 0.5$

$u_2 = 0$

Optimal for the individuals

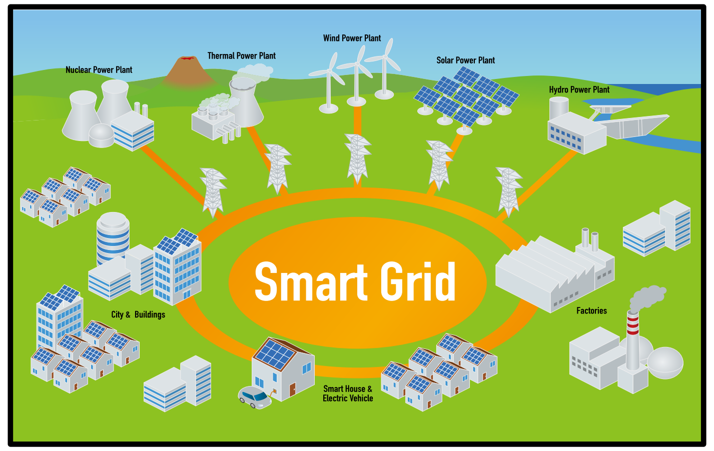
$$x_f(u) = x^*$$

Optimal for the society

We cannot directly control the actions of the users…

... but we can incentivize their behavior via u

Data-Driven Robust Learning of Dynamic Tolls:



How to control the prices?

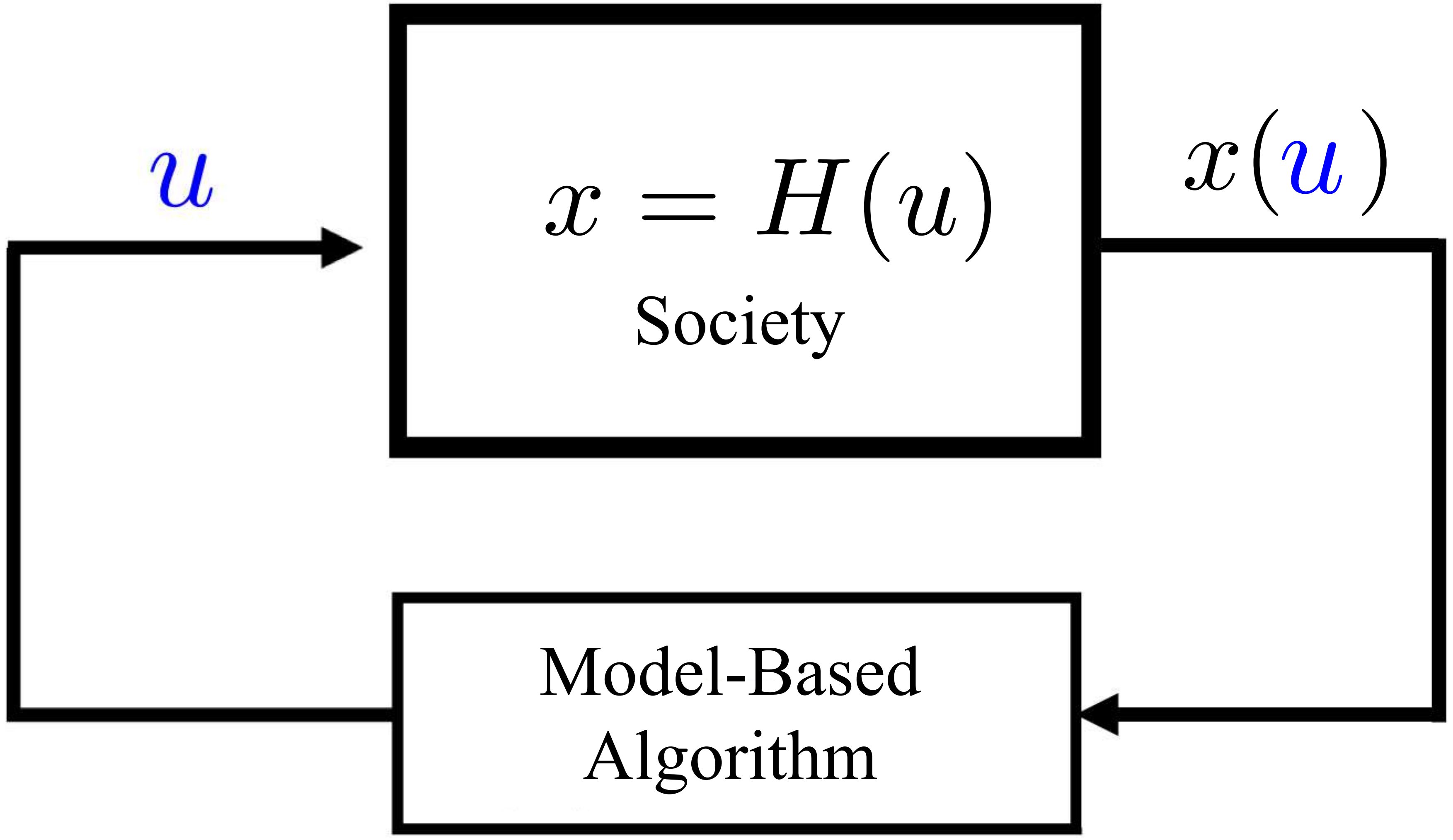
Decision-making problem with uncertainty and dynamical systems in the loop

Data-Driven Robust Learning of Dynamic Tolls:

Goal: Design a robust feedback mechanism to control u, such that:

$$u \rightarrow u^* \quad \text{such that} \quad x(u) \rightarrow x^* = \arg\max W(x)$$

Equilibrium Model: (common in economics and computer science)

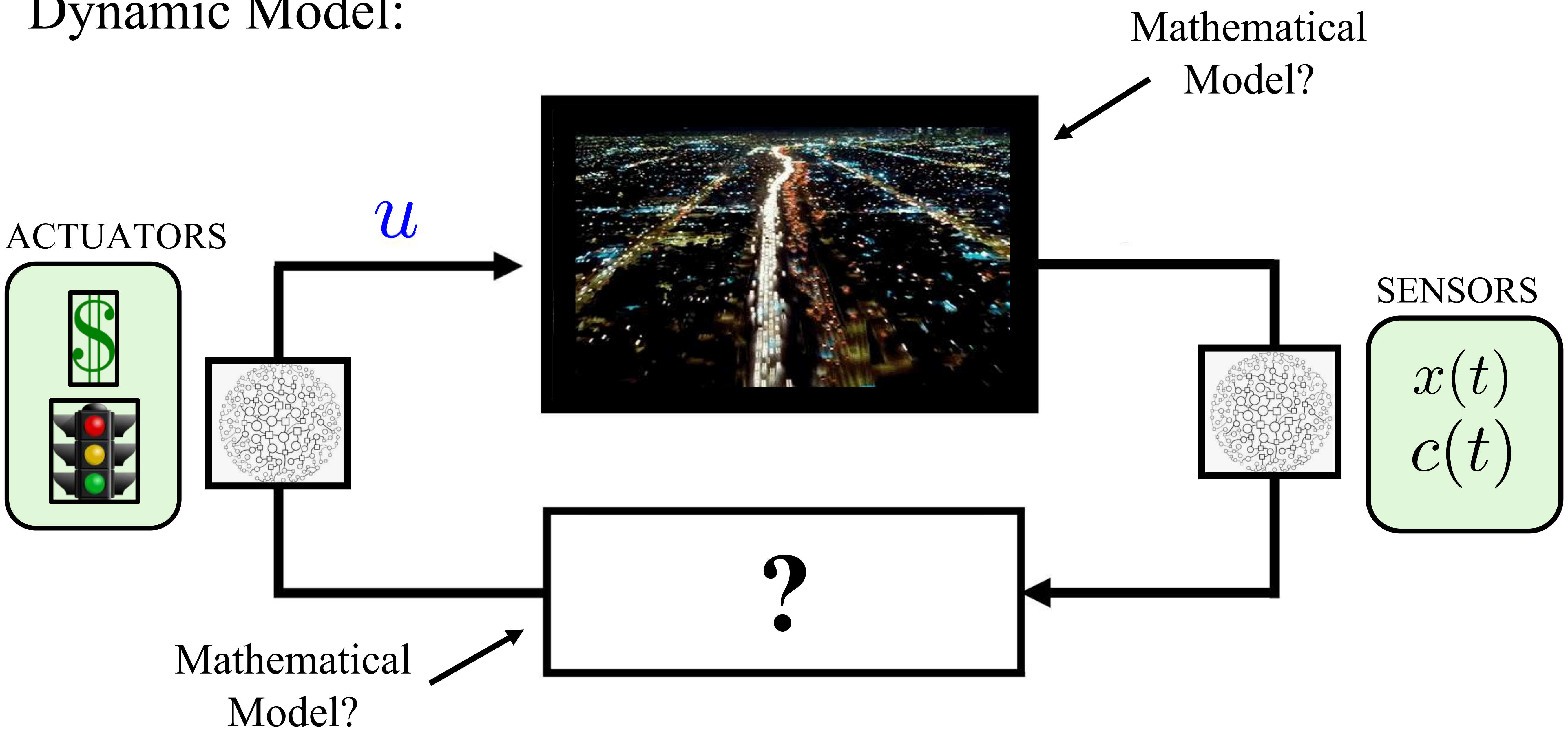


Data-Driven Robust Learning of Dynamic Tolls:

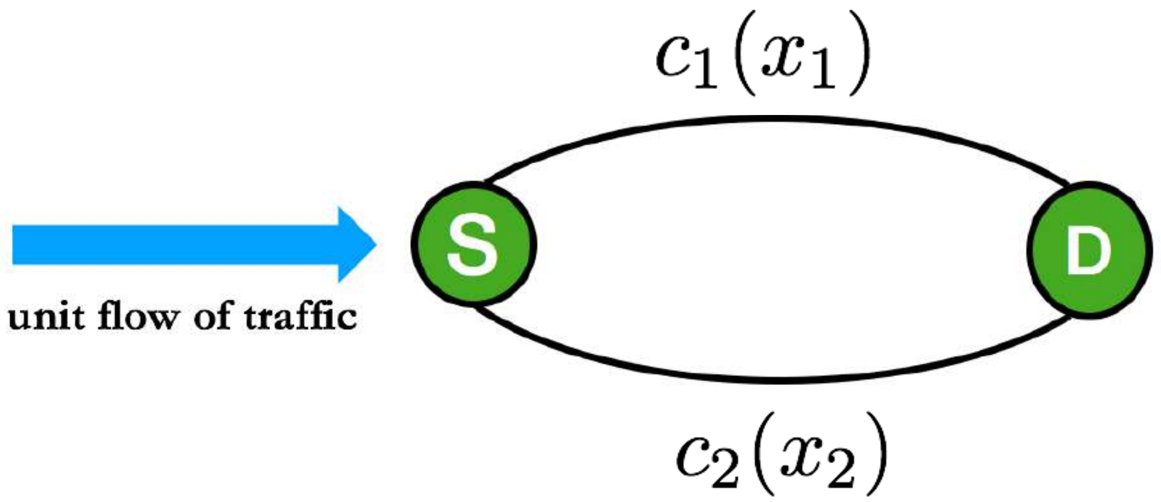
Goal: Design a robust feedback mechanism to control u, such that:

$$u \rightarrow u^* \quad \text{such that} \quad x(u) \rightarrow x^* = \arg\max W(x)$$

Dynamic Model:



Data-Driven Robust Learning of Dynamic Tolls:



- Transportation systems
- Power dispatch in smart grids
- Bandwidth allocation
- Advertising and auctions

Assumption 1:

1. A unit mass or flow of infinitesimally small decision makers
2. A finite number of actions N.
3. A N-vector of **unknown** but **measurable** cost functions:

$$c(x) = Ax + b, \quad A > 0$$

4. A **measurable** social state x defined on the simplex:

$$\Delta := \left\{ x \in \mathbb{R}^N_{\geq 0} : \sum_{i=1}^{N} x_i = 1 \right\}$$

Data-Driven Robust Learning of Dynamic Tolls:

We focus on SNS satisfying only two key properties:

Static Property

Users of the society face a decision-making problem modeled by a **Congestion Game** (CG)

Dynamic Property

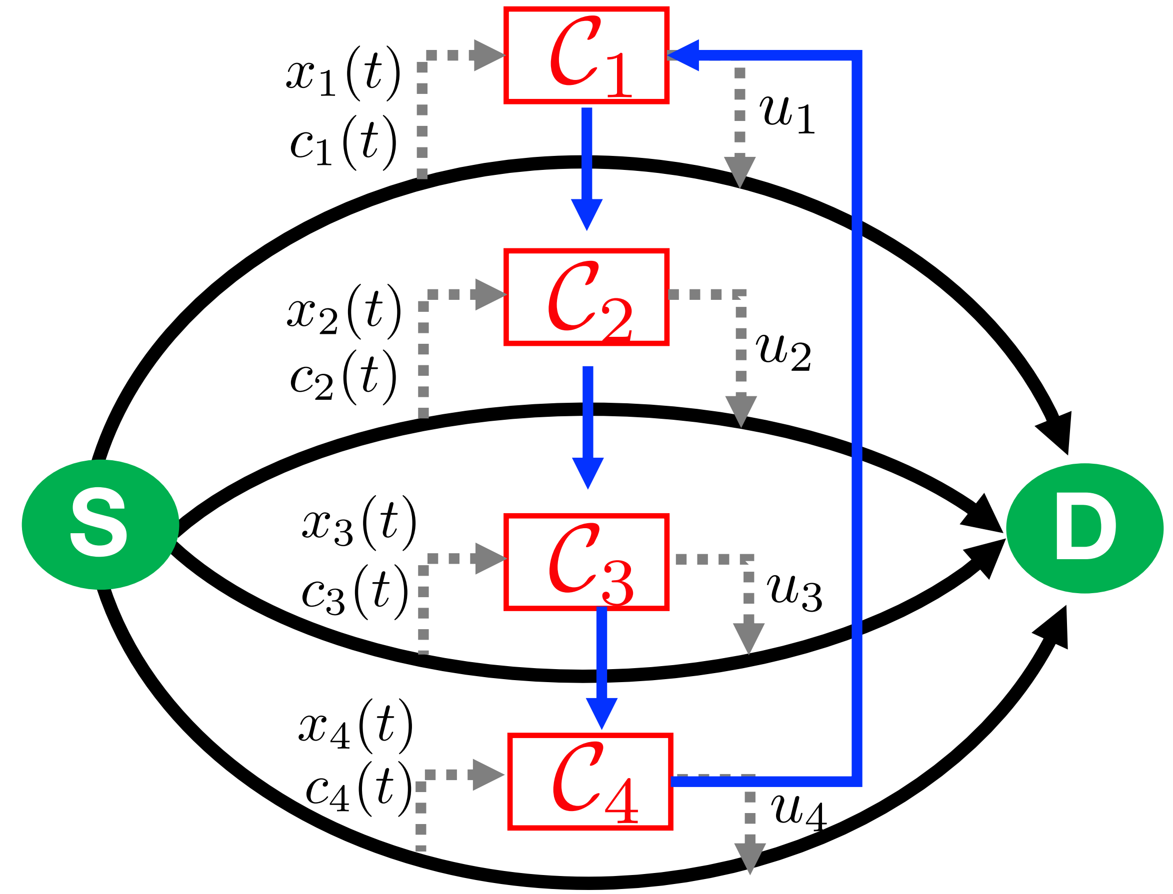
Social dynamics satisfy a **Stable Behavioral Model**

Data-Driven Robust Learning of Dynamic Tolls:

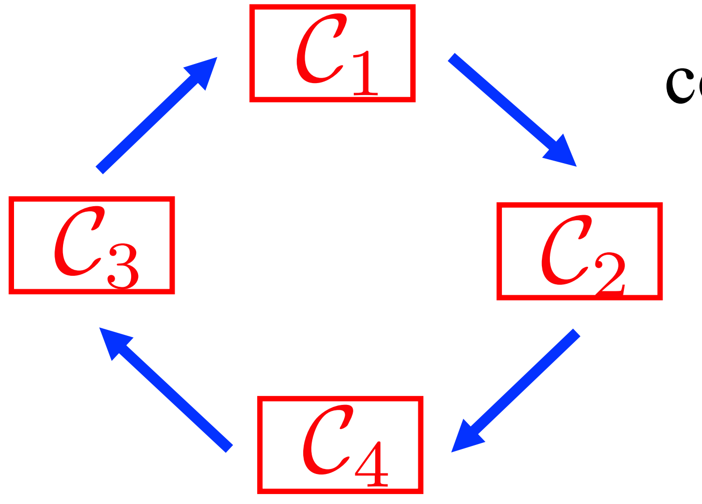
Captures existing models in the literature of evolutionary dynamics

Assumption 2: Users are selfish, that is: $x \to x_f(u)$

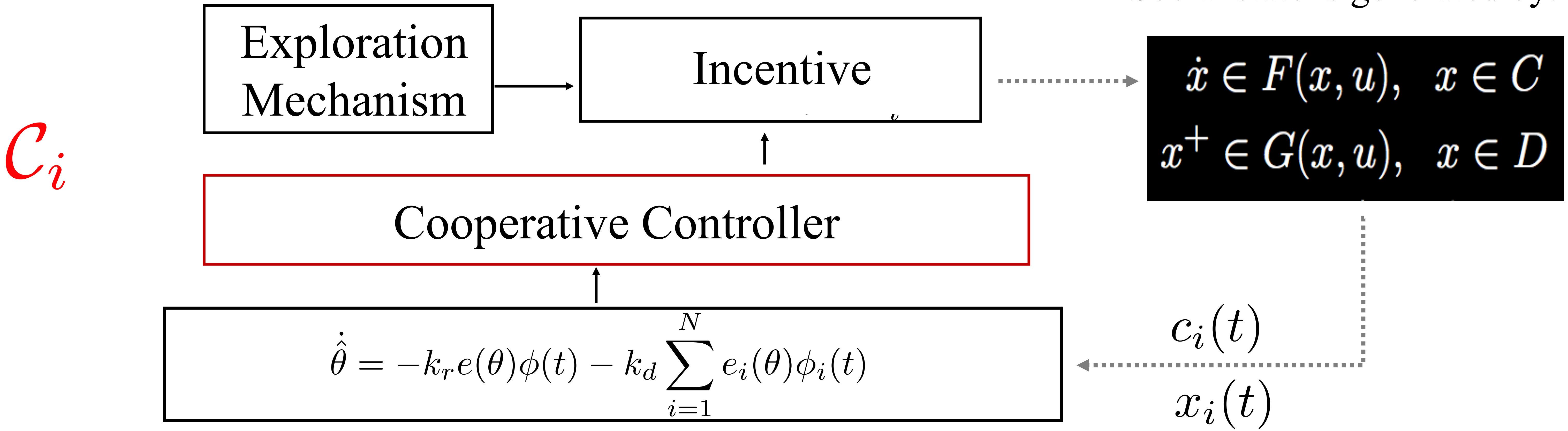
Data-Driven Robust Learning of Dynamic Tolls:



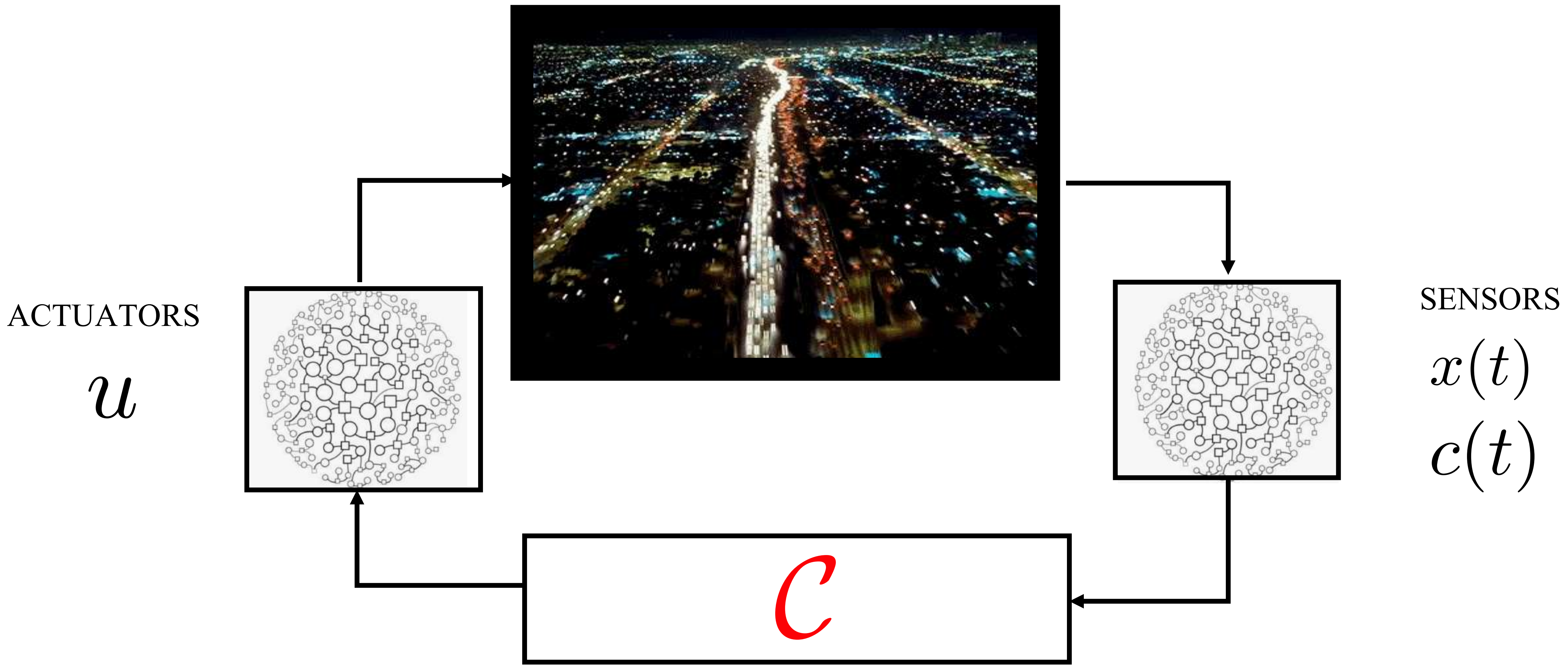
Control systems share information via a communication network



Social state is generated by:

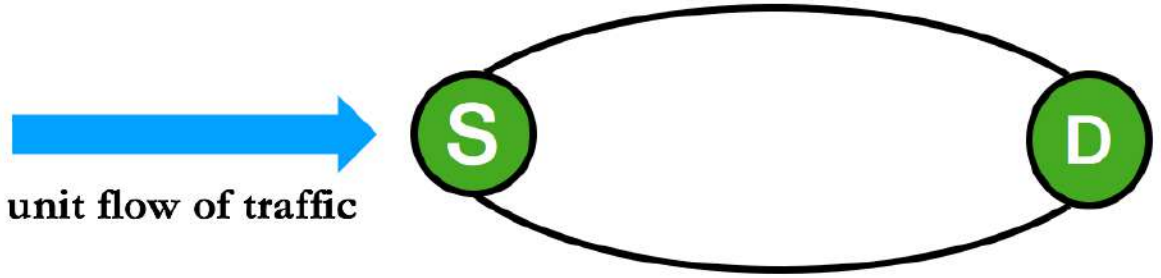


Data-Driven Robust Learning of Dynamic Tolls:



$$u \to u^* \text{ such that } x \to x^* = \arg\max W(x)$$

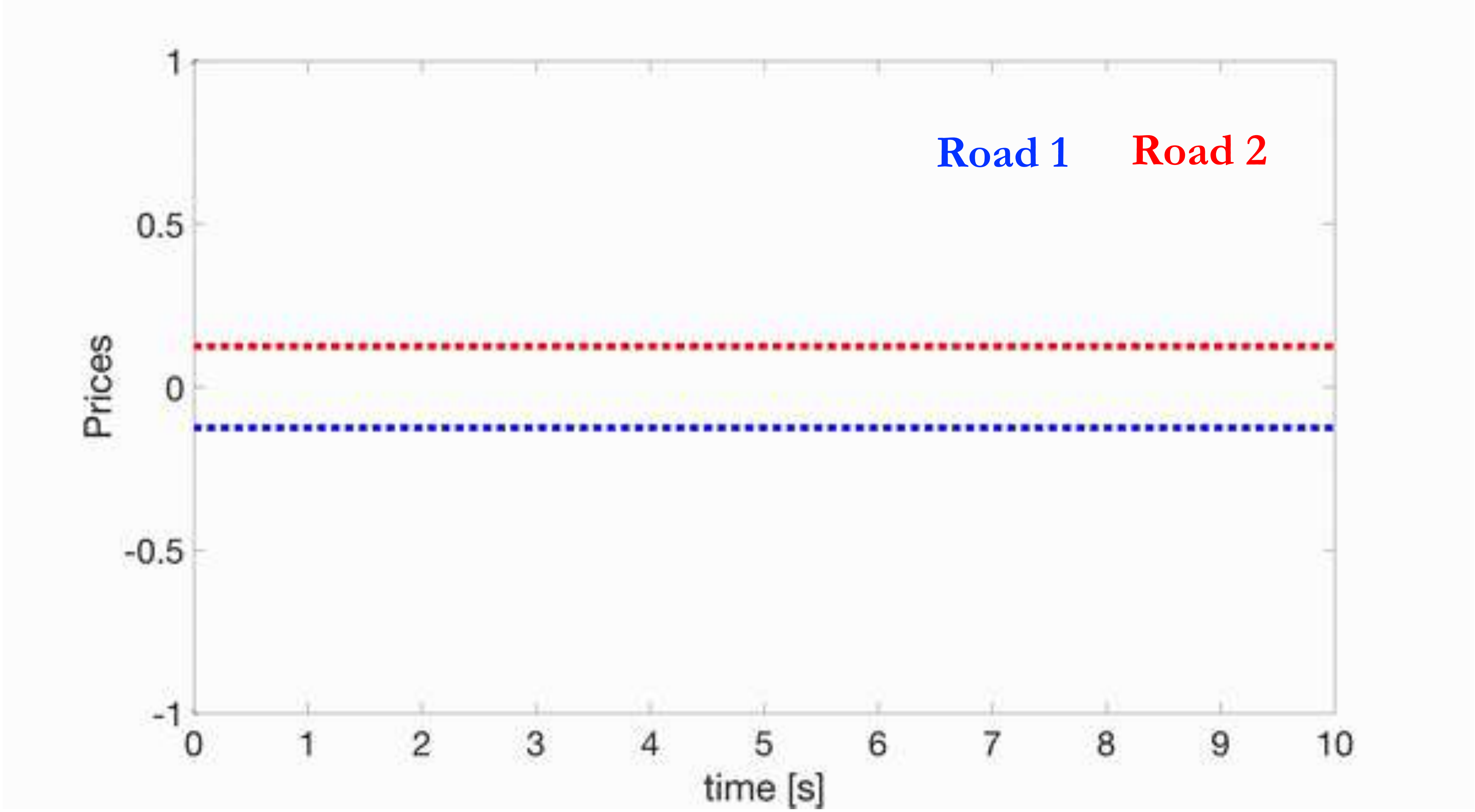
Data-Driven Robust Learning of Dynamic Tolls:



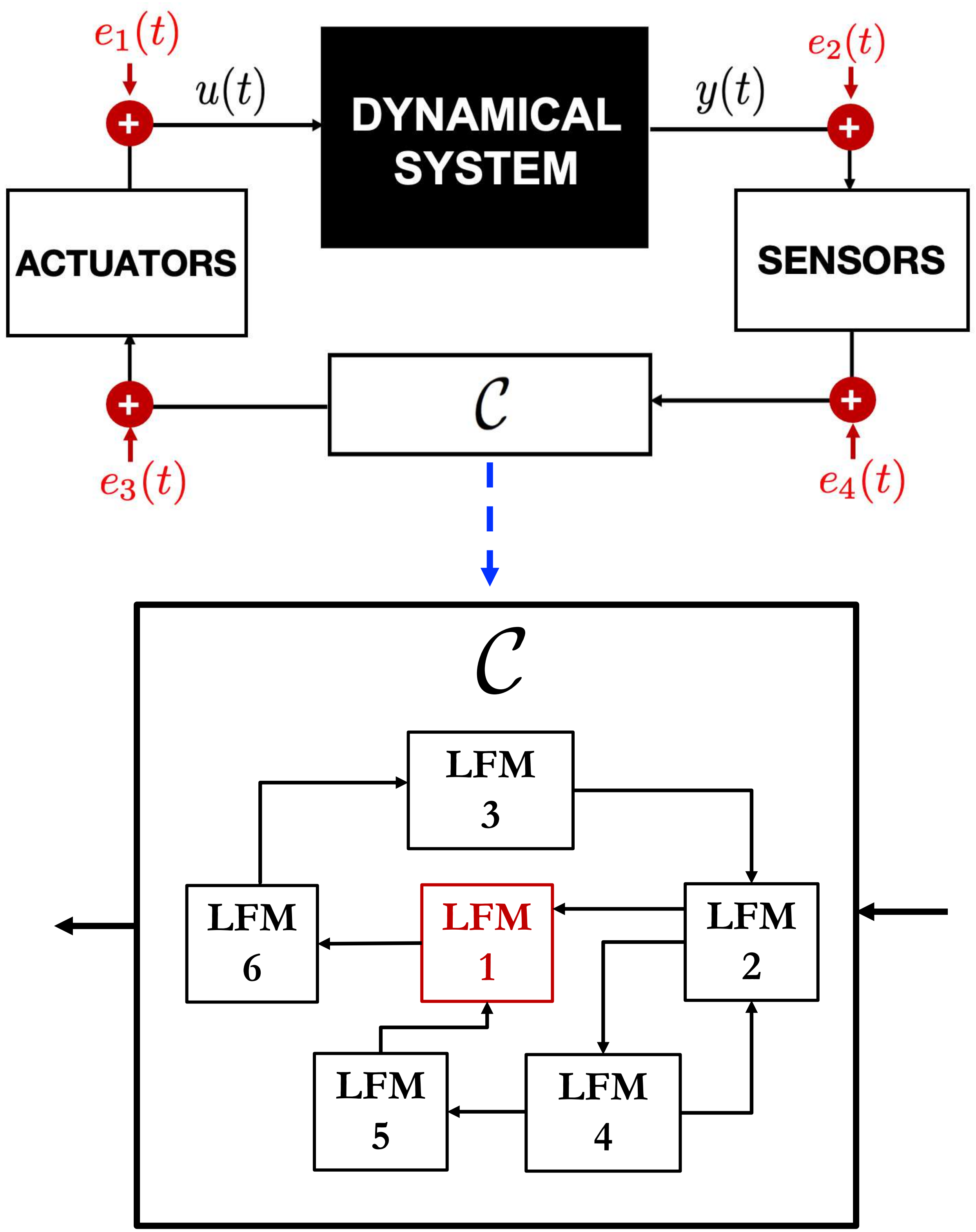
Set-valued Social Dynamics

$$\dot{x} \in -x + \mathrm{argmax}_{w \in \Delta}(w^\top(c(x) + u))$$

Sandholm, 2002



Fixed-Time Data-Driven Robust Learning:



Concurrent learning dynamics are used in many applications and algorithms:

$$\dot{\hat{\theta}} = -k_r e(\theta)\phi(t) - k_d \sum_{i=1}^{N} e_i(\theta)\phi_i(t)$$

$$\dot{\tilde{\theta}} = -(k_r Q(t) + k_d D)\tilde{\theta}$$

Examples of Controllers $\mathcal{C}$	**LFM 1**
Reinforcement Learning	Actor/Critic weights updates
Iterative Learning	Inputs updates per iteration
Extremum Seeking Control	Optimization of response map
Indirect Adaptive Control	Parameter estimation
Neural-based Control	Weights updates
Feedback Optimization	Optimization of steady-state cost
Consensus Dynamics	Gradient flow of Laplacian potential
Synergistic Hybrid Control	Gradient flow in each partition
Learning Dynamics for Games	Replicator-Pseudogradient dynamics

However, the convergence can be slow if the smallest eigenvalue of $A(t)$ is too small….

Data-Driven Robust Learning:

Can we improve the convergence rate of these dynamics?

$$\dot{\hat{\theta}} = -k_r e(\theta)\phi(t) - k_d \sum_{i=1}^{N} e_i(\theta)\phi_i(t)$$

Let's first focus on the case where we only use recorded data, i.e., $(k_r = 0)$.

$$\dot{\hat{\theta}} = -k_d \sum_{i=1}^{N} e_i(\theta)\phi_i(t) = D(\hat{\theta} - \theta^*) \qquad \text{or, equivalently} \qquad \dot{\tilde{\theta}} = -D\tilde{\theta}$$

Dynamical systems of this form (i.e., linear) can only achieve exponential convergence, in fact, their explicit solution can be computed:

$$\tilde{\theta}(t) = e^{A(t-t_0)}\tilde{\theta}(0)$$

Can we modify our dynamics to achieve faster convergence?

Fixed-Time Data-Driven Robust Learning:

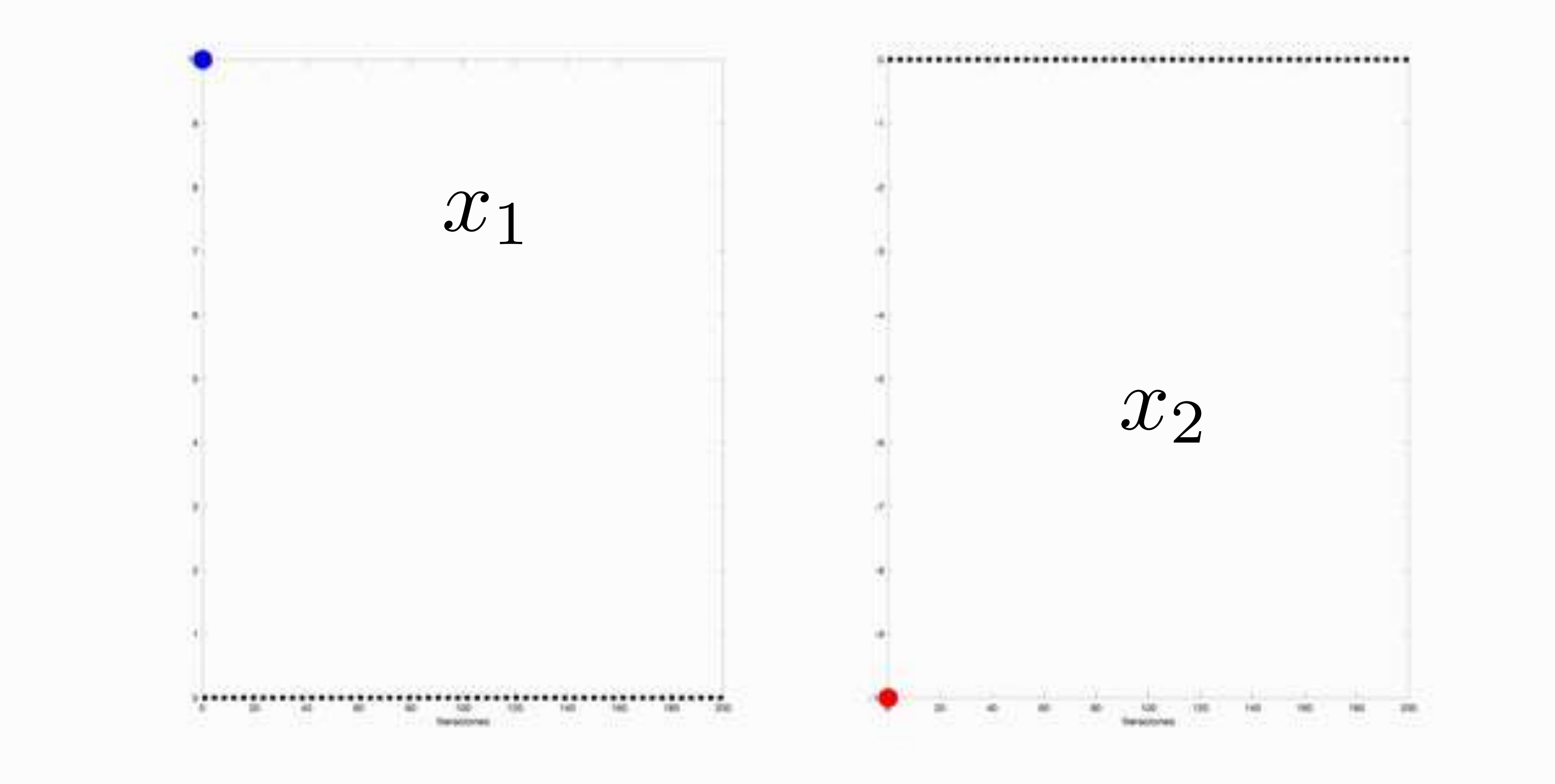
Example: Consider the ODE:

$$\dot{x} = -x^{\frac{1}{3}} - x^3$$

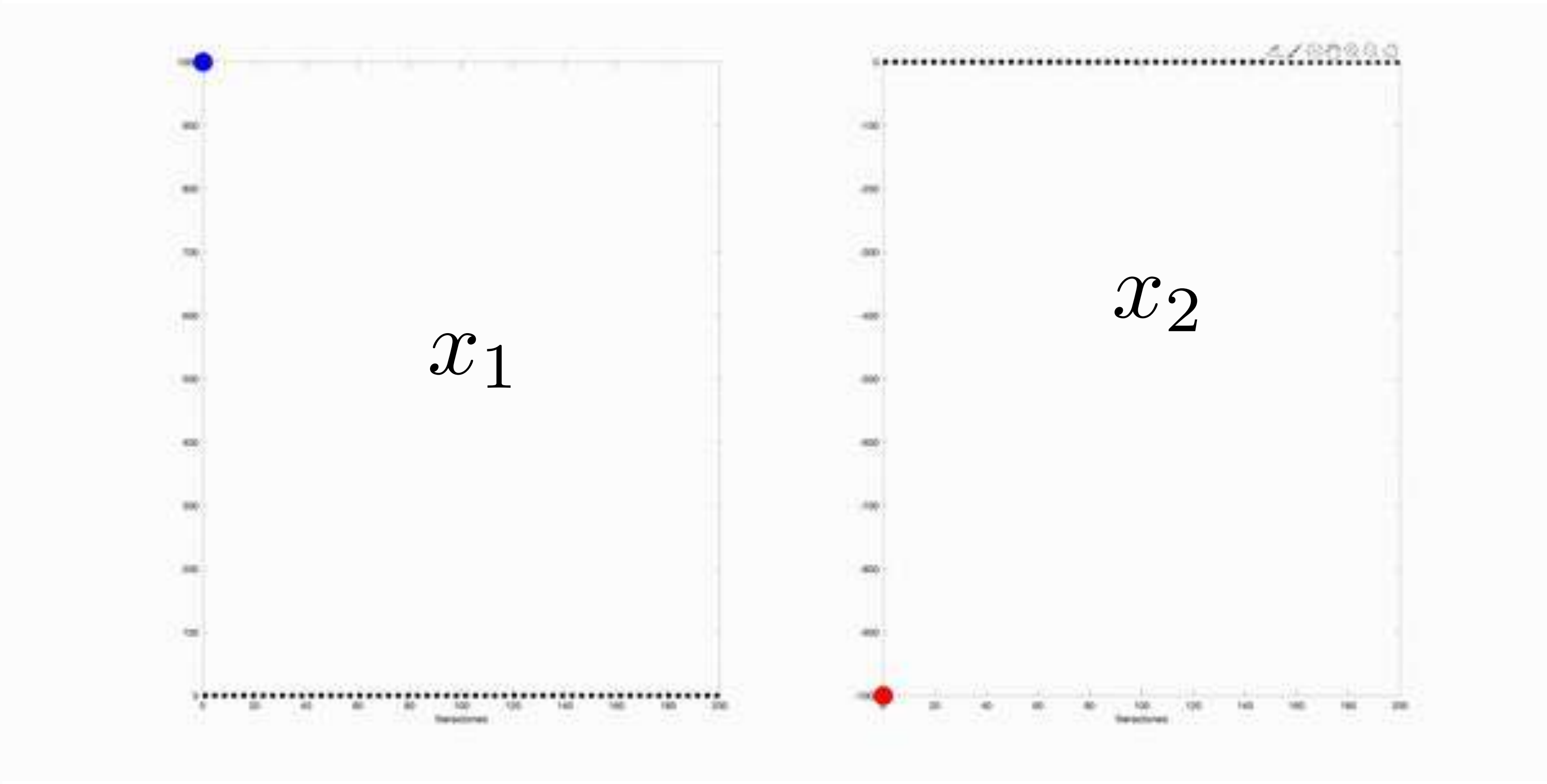
the solution of this system converges to zero, exactly before the time T = 2.5.

Comparison with traditional exponentially stable system:

For initial conditions close to zero we don't see much difference



But for initial conditions far away from zero….



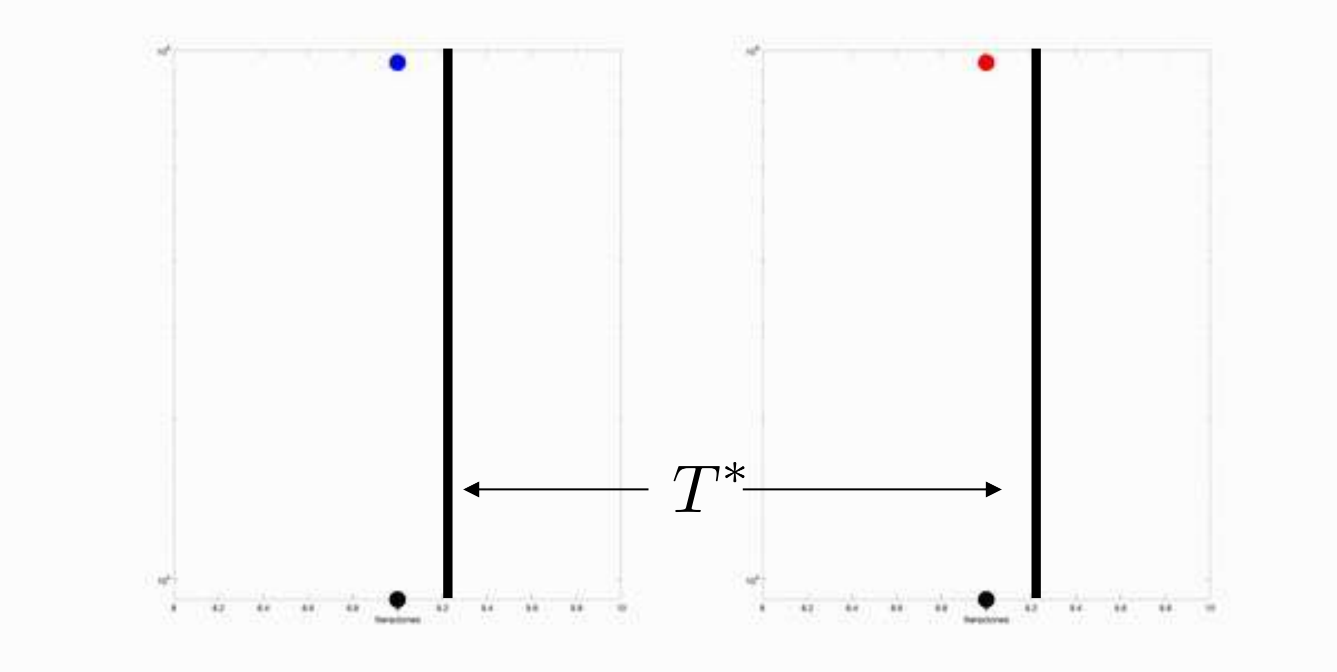
Fixed-Time Data-Driven Robust Learning:

Example: Consider the ODE:

$$\dot{x} = -x^{\frac{1}{3}} - x^3$$

the solution of this system converges to zero, exactly before the time T = 2.5.

Comparison with traditional exponentially stable system: In logarithmic scale



Dynamical systems with this rare property are called **fixed-time stable**

Fixed-Time Data-Driven Robust Learning:

Can we design data-driven estimation dynamics that are fixed-time stable?

We can start by considering fixed-time gradient flows of the form:

$$\dot{x} = -\frac{\nabla J(x)}{|\nabla J(x)|^{\alpha}} - \frac{\nabla J(x)}{|\nabla J(x)|^{-\alpha}} \qquad \alpha \in (0,1)$$

If $J(x) = x^\top Q x, \quad \lambda_{\min}(Q) > 0$ then $x(t) = 0 \ \ \forall\, t \geq T^* = \dfrac{\pi}{2\lambda_{\min}(Q)\alpha}$

With this result at hand, the design of a fixed-time estimation algorithm based on data is obvious:

Instead of: $\dot{\hat{\theta}} = -D(\hat{\theta} - \theta^*)$ $\Rightarrow$ $\dot{\hat{\theta}} = -\dfrac{D(\hat{\theta} - \theta^*)}{|D(\hat{\theta} - \theta^*)|^{\alpha}} - \dfrac{D(\hat{\theta} - \theta^*)}{|D(\hat{\theta} - \theta^*)|^{-\alpha}}$

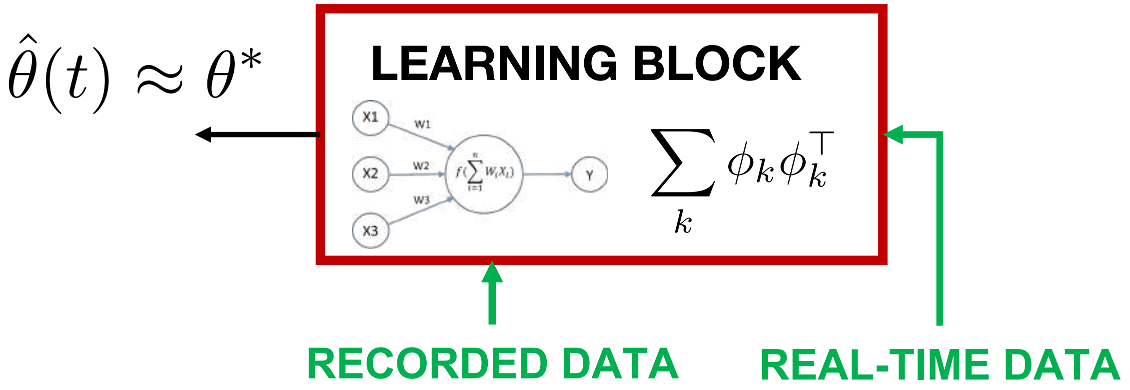
where $D = \sum_{i=1}^{N} \phi(t_i)\phi(t_i)^\top$

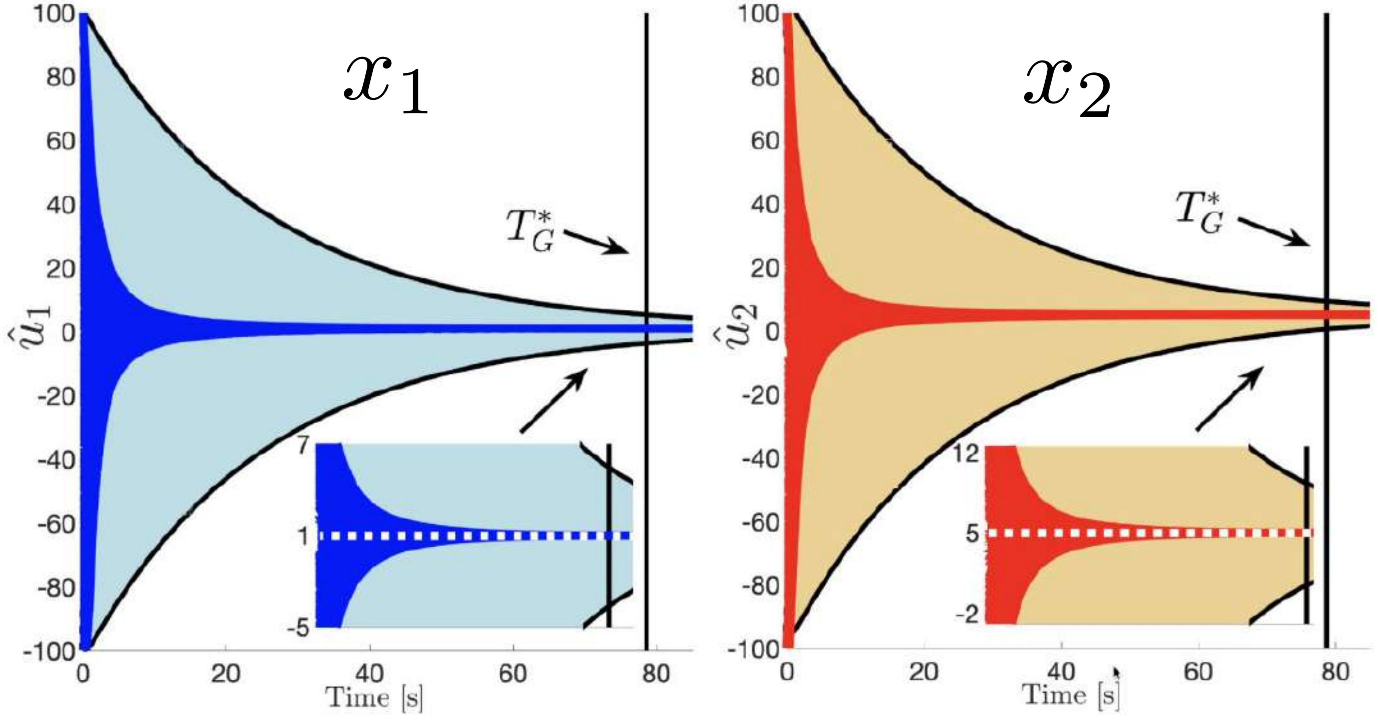
But we need to write the dynamics in terms of the signals that we can measure

Fixed-Time Data-Driven Robust Learning:

$$\dot{\hat{\theta}} = -\frac{D(\hat{\theta}-\theta^*)}{|D(\hat{\theta}-\theta^*)|^{\alpha}} - \frac{D(\hat{\theta}-\theta^*)}{|D(\hat{\theta}-\theta^*)|^{-\alpha}} \iff \dot{\hat{\theta}} = -\frac{\sum_{i=1}^{N} e_i\phi(t_i)}{|\sum_{i=1}^{N} e_i\phi(t_i)|^{\alpha}} - \frac{\sum_{i=1}^{N} e_i\phi(t_i)}{|\sum_{i=1}^{N} e_i\phi(t_i)|^{-\alpha}}$$

We have finally obtained a data-driven estimation algorithm with fixed-time convergence properties!





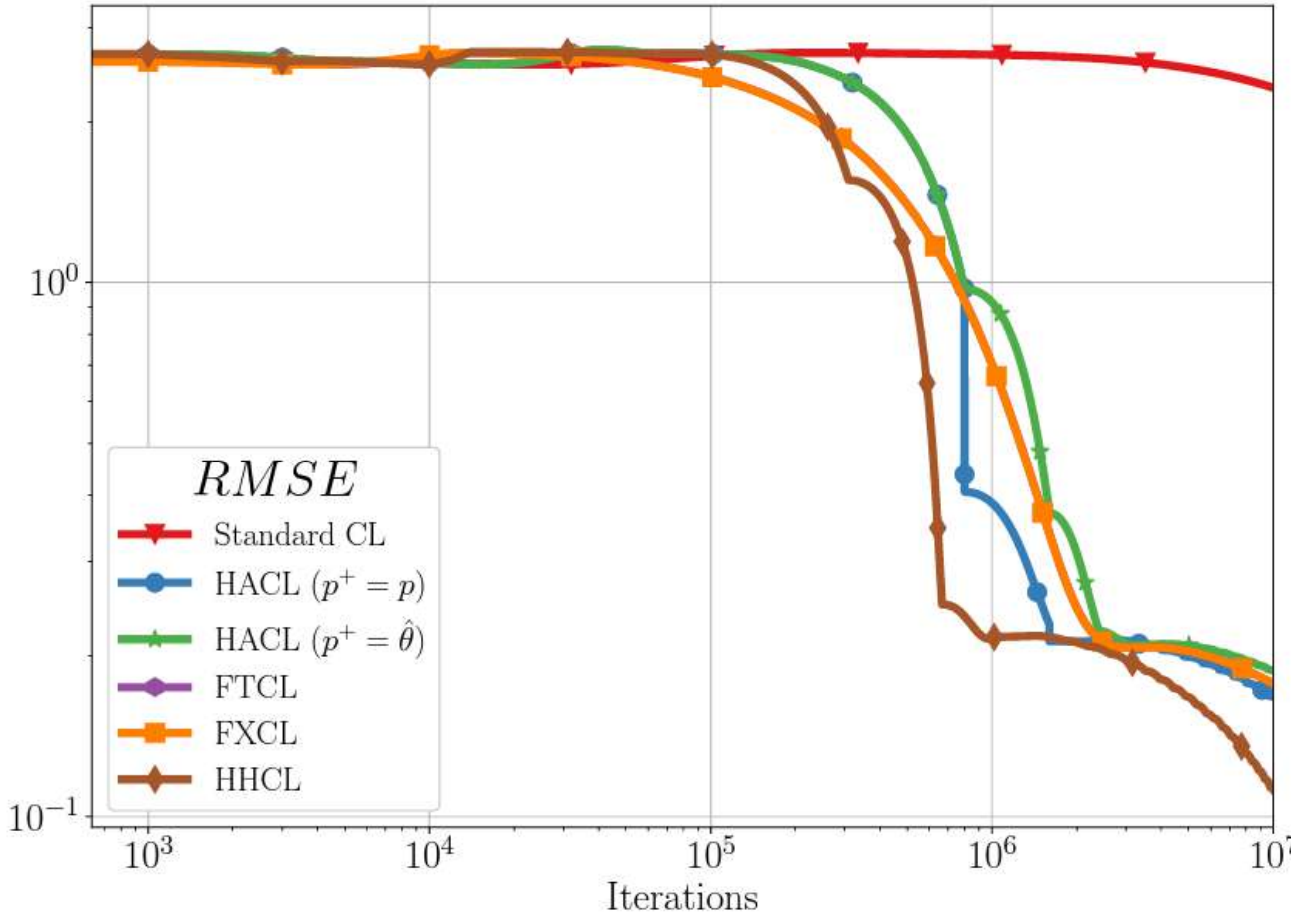
We could also add the term that uses real-time measurements of the signal

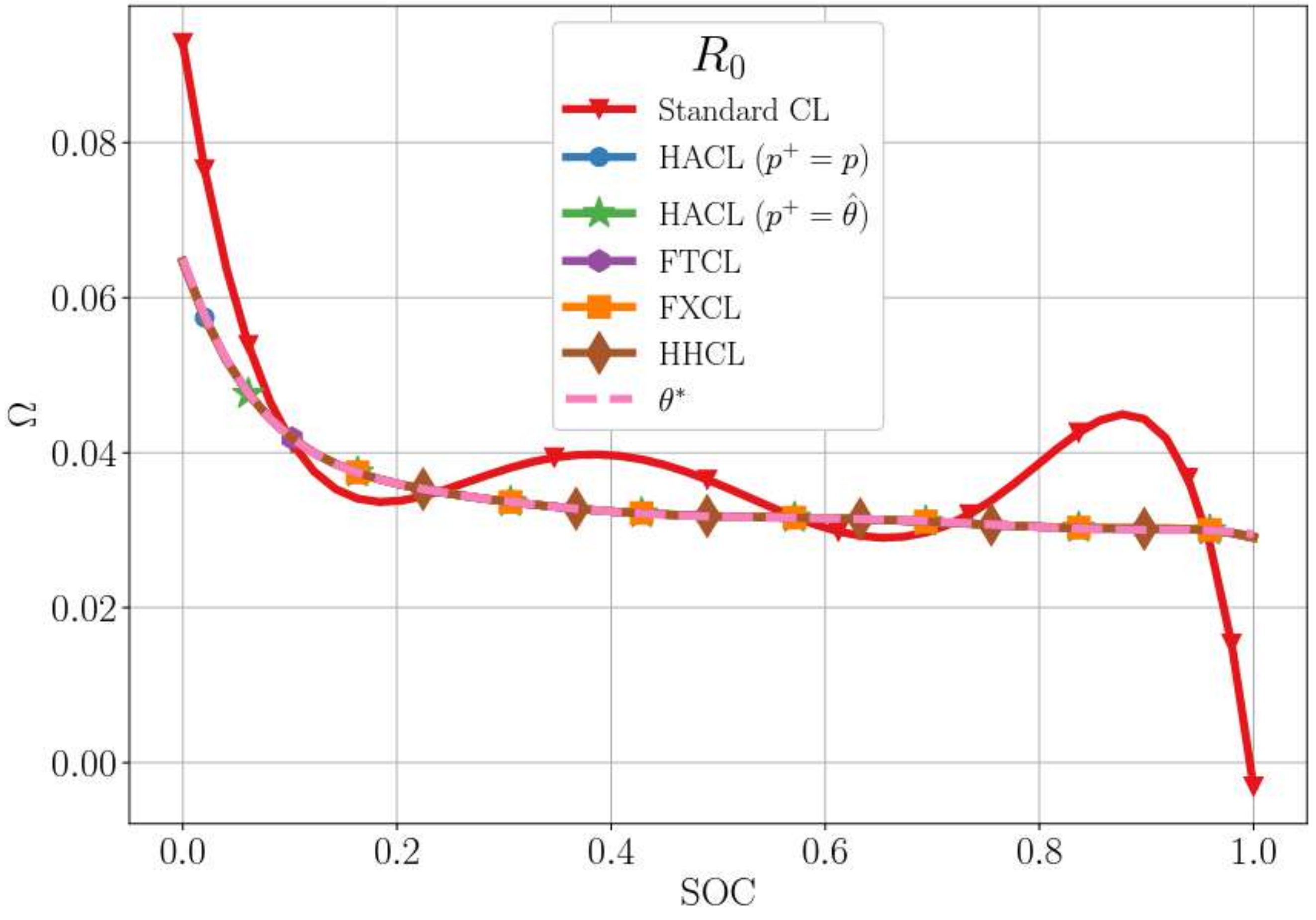
Fixed-Time Data-Driven Robust Learning: Application on Batteries

Problem: Characterization of the impedance parameters of an equivalent circuit model of a Lithium-Ion (Li-Ion) battery

$$y = \phi(t)^\top \theta^*$$

$$\dot{z} = \frac{I}{C_0} \qquad \dot{i}_1 = \frac{I - i_1}{R_1 C_1} \qquad V = \Phi(z) + I\boxed{R_0} + i_1 R_1$$

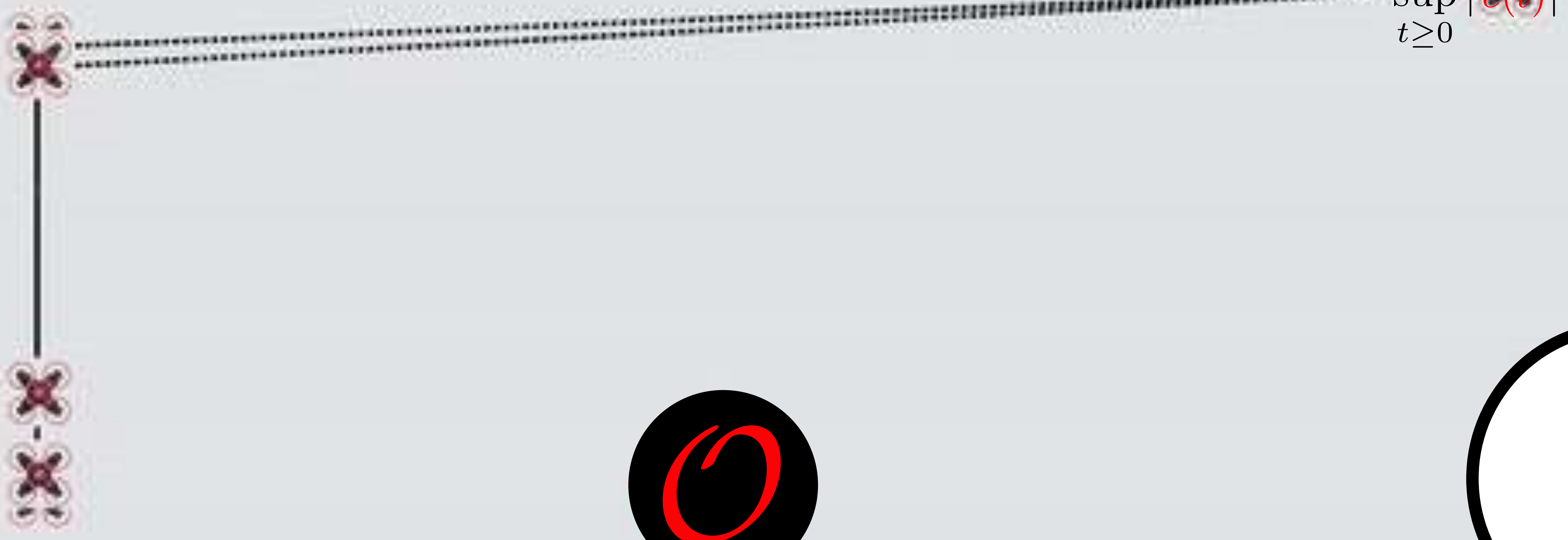




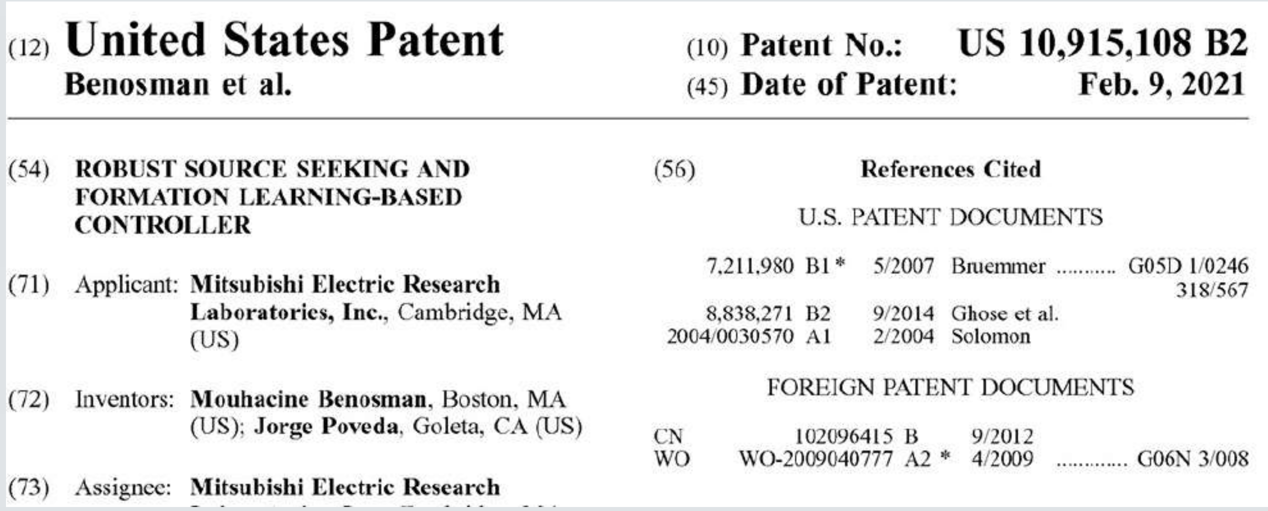
Collaboration with Robert Bosch, Mountain View, CA

BOSCH
Invented for life

Source Seeking in Mobile Robots



$$\sup_{t\geq 0} |e(t)| \leq \varepsilon$$


(12) **United States Patent**
Benosman et al.

(10) **Patent No.: US 10,915,108 B2**
(45) **Date of Patent: Feb. 9, 2021**

(54) **ROBUST SOURCE SEEKING AND FORMATION LEARNING-BASED CONTROLLER**

(71) Applicant: **Mitsubishi Electric Research Laboratories, Inc.**, Cambridge, MA (US)

(72) Inventors: **Mouhacine Benosman**, Boston, MA (US); **Jorge Poveda**, Goleta, CA (US)

(73) Assignee: **Mitsubishi Electric Research**

(56) **References Cited**

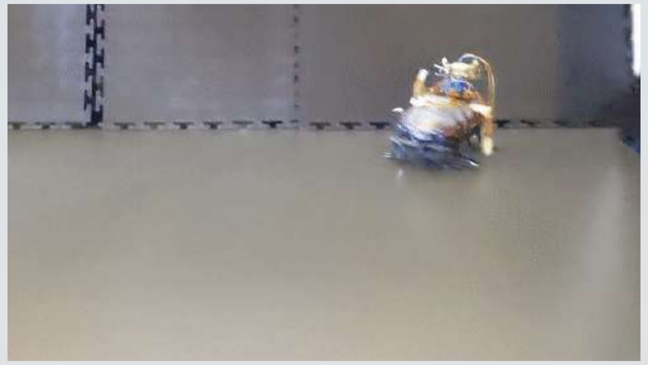
U.S. PATENT DOCUMENTS

7,211,980 B1 * 5/2007 Bruemmer G05D 1/0246 318/567
8,838,271 B2 9/2014 Ghose et al.
2004/0030570 A1 2/2004 Solomon

FOREIGN PATENT DOCUMENTS

CN 102096415 B 9/2012
WO WO-2009040777 A2 * 4/2009 .... G06N 3/008

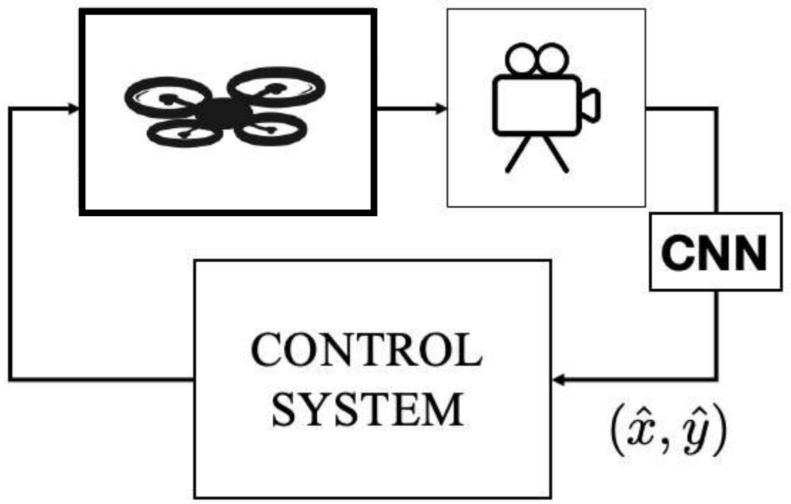
Bio-Inspired Hybrid Algorithms

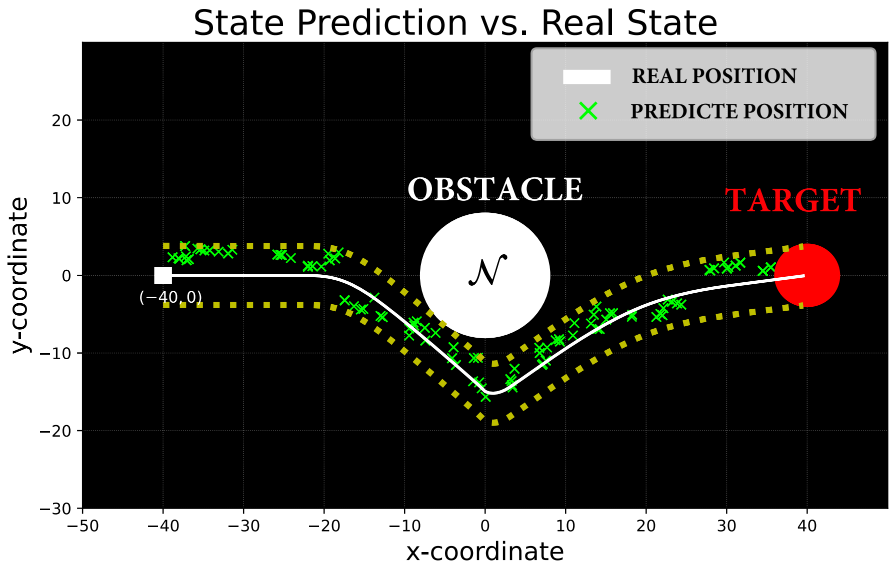


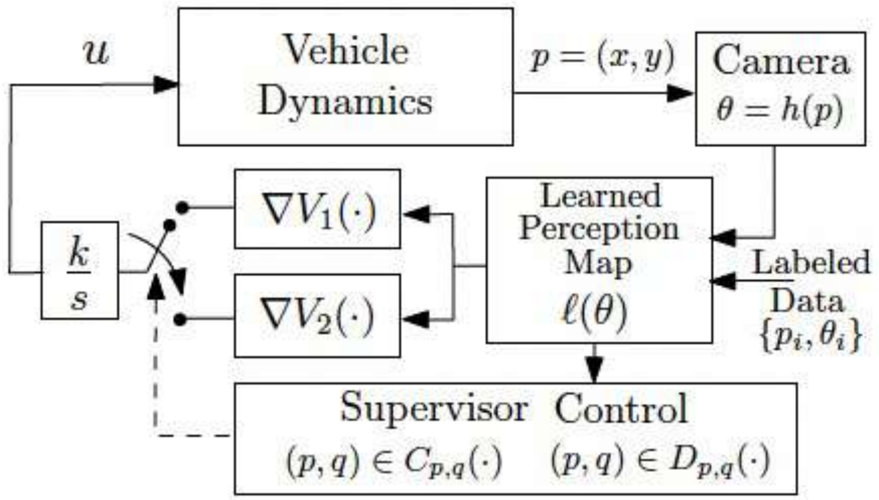
Work in collaboration with

MITSUBISHI ELECTRIC RESEARCH LABORATORIES

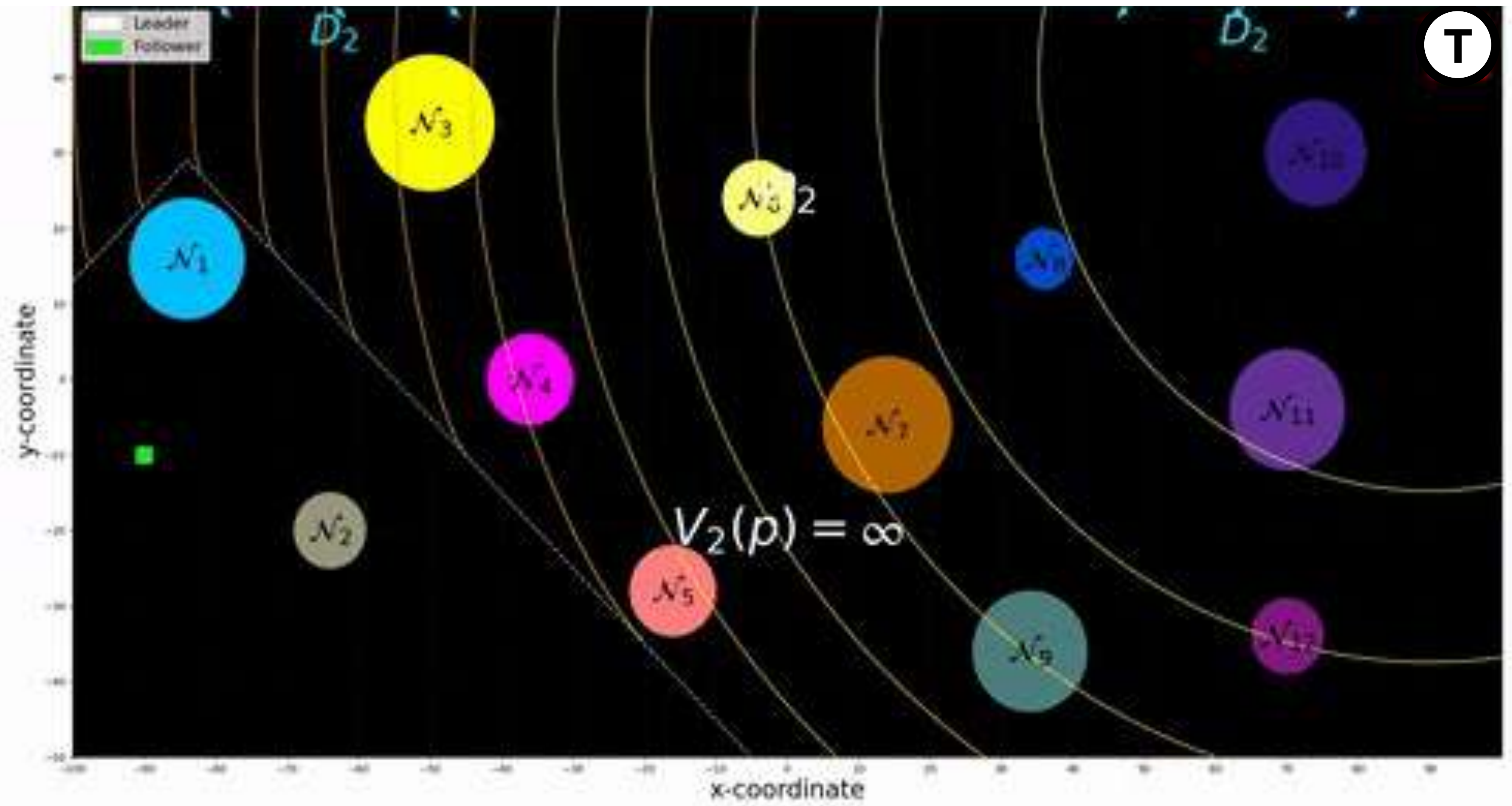
Learning Perception Maps for Navigation with Cameras:

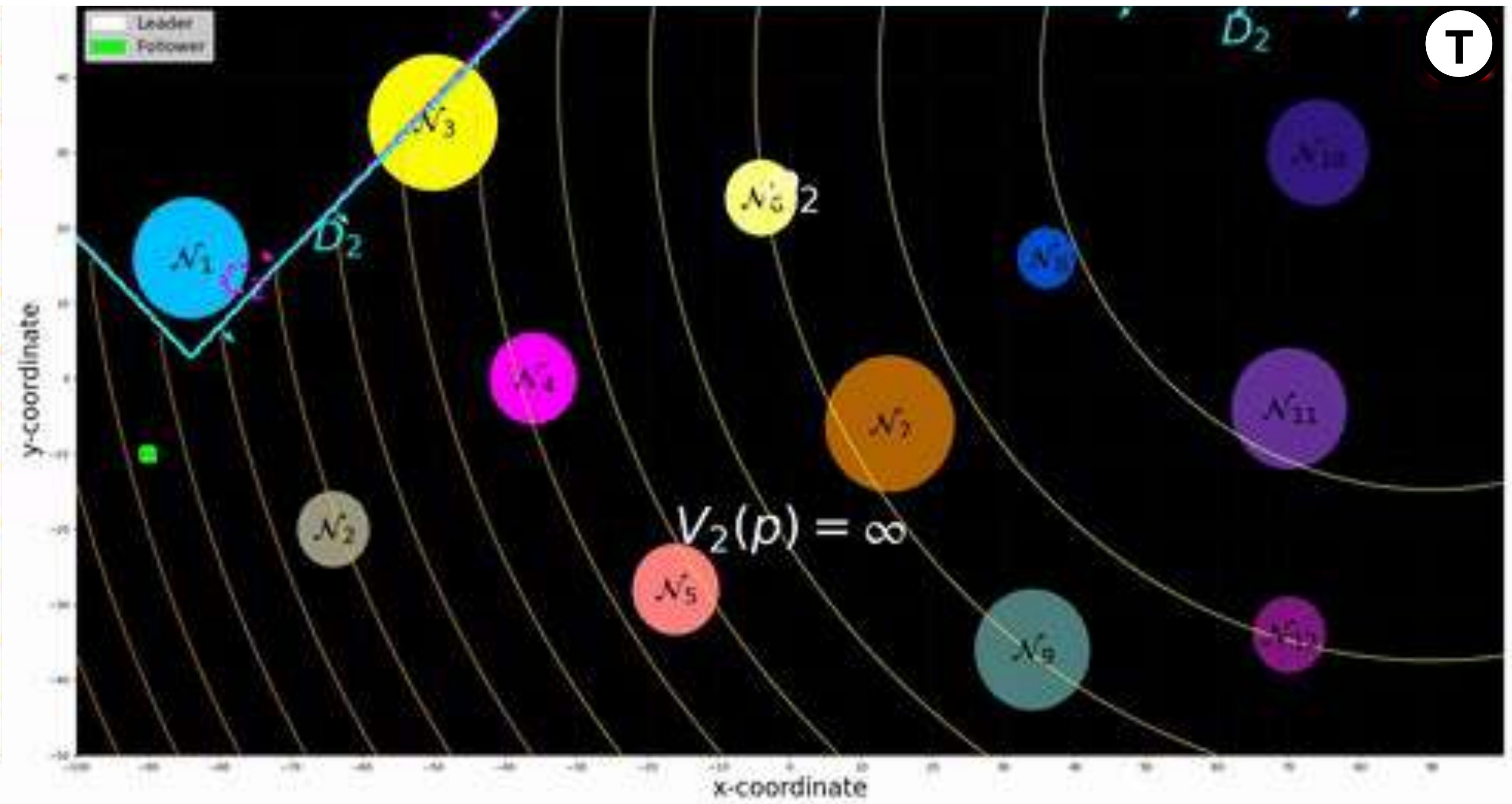






Murillo and Poveda, ACC '22.



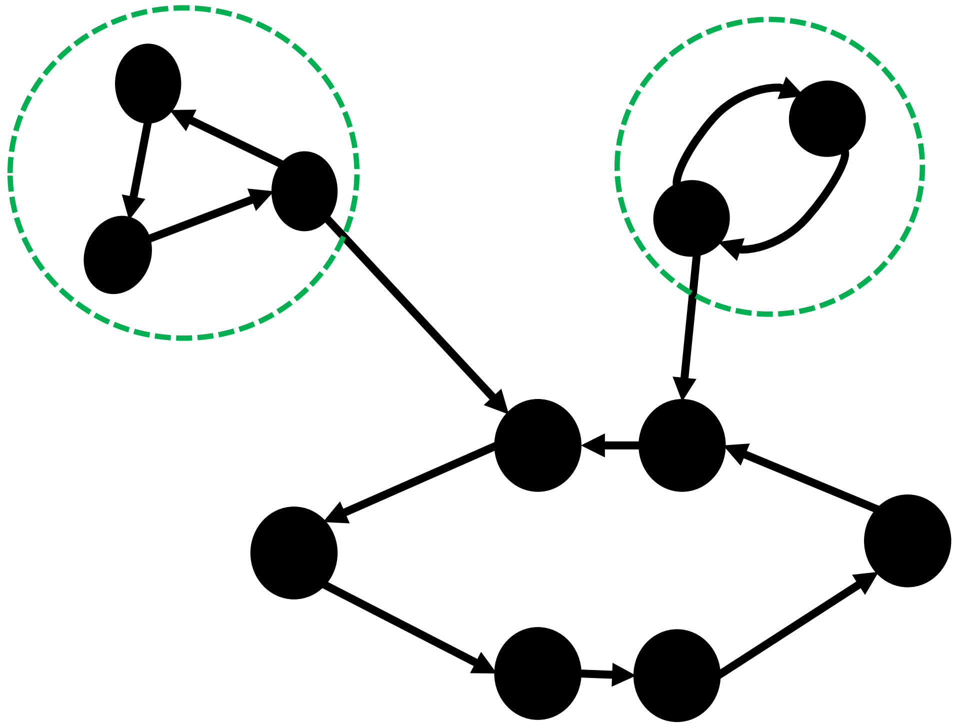


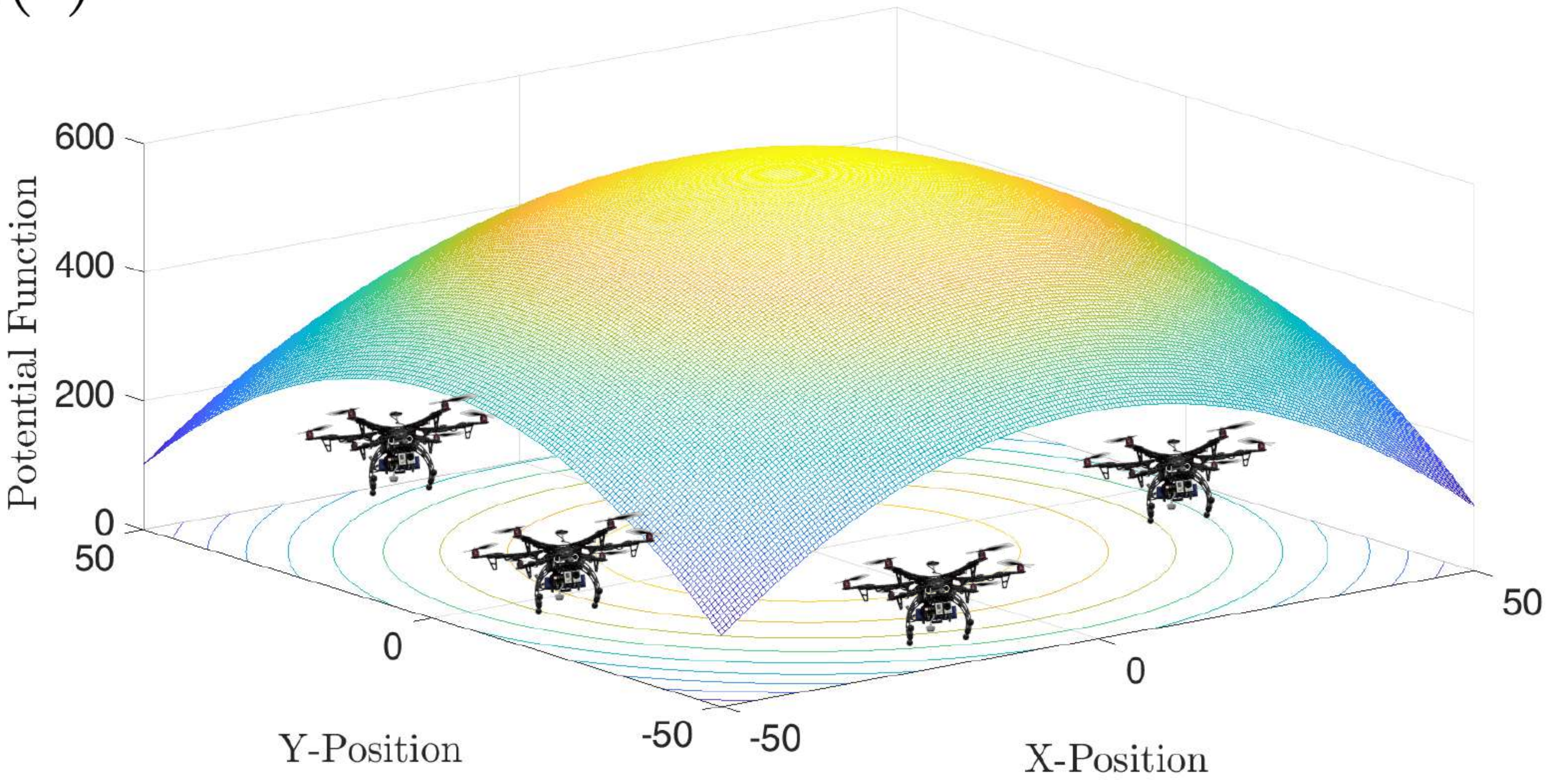
Learning Perception Maps for Navigation with Cameras:

The previous ideas can be extended to problems defined over networks:

$$y_i(t) := \phi\Big(X_i(t), Y_i(t)\Big)\theta^* = \phi_i(t)\theta^*$$

$$f(X,Y) := \phi(X,Y)\theta^* + \text{error}$$





What are the conditions on the communication graph? Can we still achieve fixed-time stability?