Introduction to Hybrid Systems

Karl Henrik Johansson
Department of Signals, Sensors and Systems
Royal Institute of Technology, Stockholm, Sweden

Formation en Automatique à Paris, April 2005

Outline

- Introduction
  - Motivating examples
- Models
  - Hybrid automata
  - Solutions
- Control
  - Stability
  - Stabilization
- Verification
  - Transition systems
  - Reachability
- Summary
  - Outlook, references

Introduction

What is a Hybrid System?

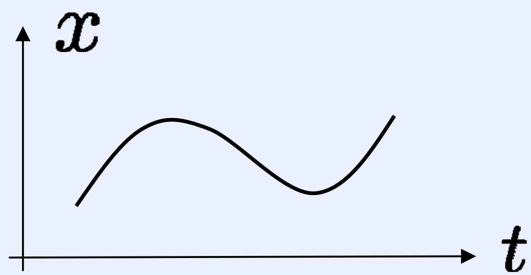
- *A hybrid system is a dynamical system with interacting time-triggered and event-triggered dynamics*

- E.g., differential equations and finite automata

$$\dot{x} = f(x, u) \quad \text{and} \quad q^+ = g(q, v)$$

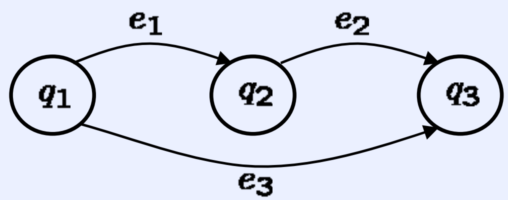
Control Systems

- Time-triggered $\dot{x} = f(x, u)$



$$x : [0, \infty) \to \mathbf{R}^n, u : [0, \infty) \to \mathbf{R}$$

- Event-triggered $q^+ = g(q, v)$

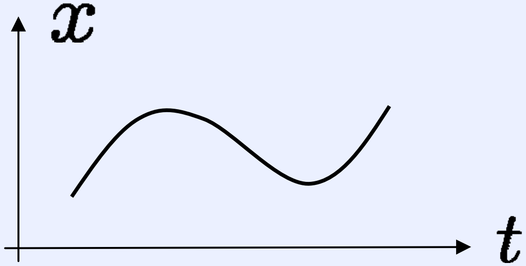


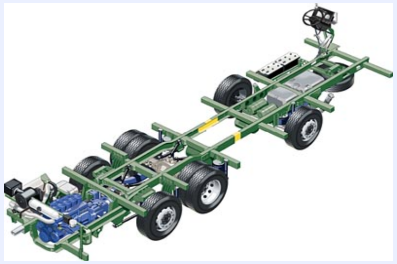
$$q_1, e_1, q_2, e_2, q_3$$

$$q : \mathbf{Z}^+ \to \{q_1, \ldots, q_N\}, v : \mathbf{Z}^+ \to \{e_1, \ldots, e_K\}$$

Control Systems

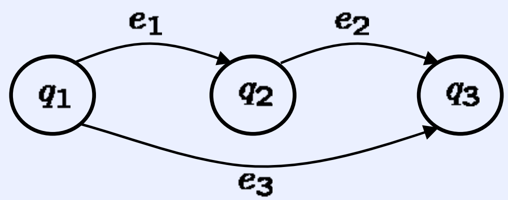
- Time-triggered $\dot{x} = f(x, u)$

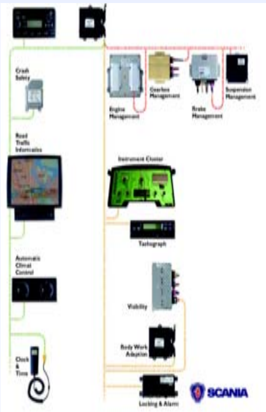




Electronics, physics, mechanics etc.

- Event-triggered $q^{+} = g(q, v)$

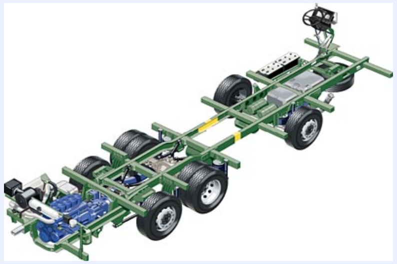




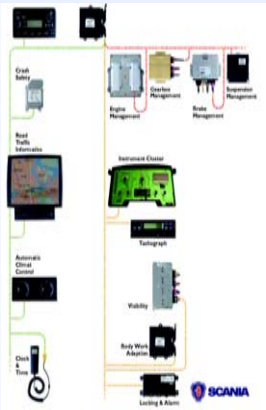
Digital circuits, logics, softwares etc.

Hybrid Control System

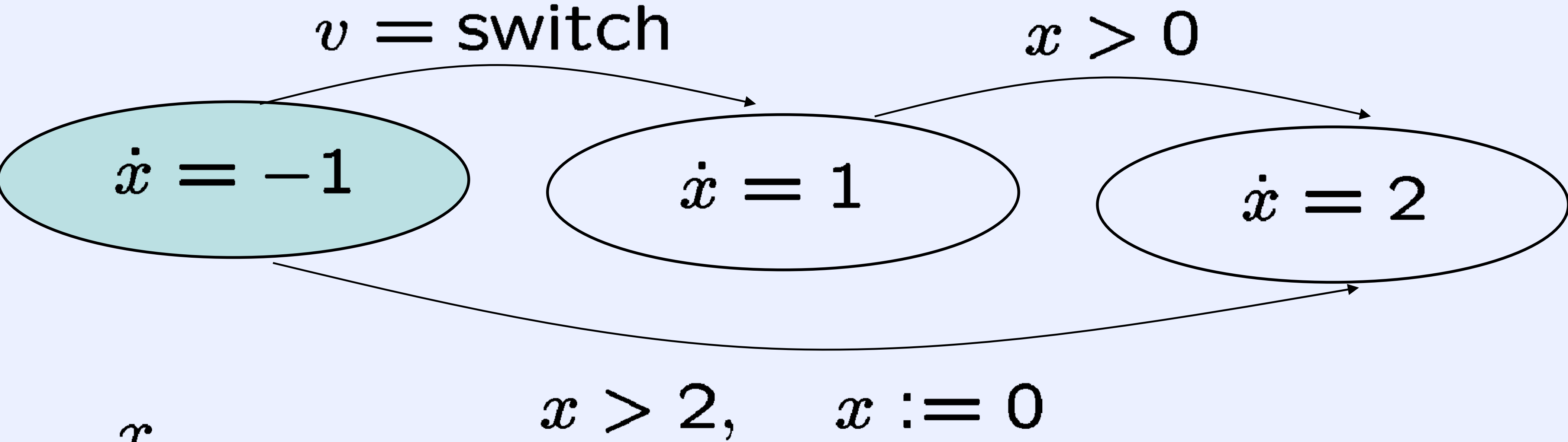
- Time-triggered

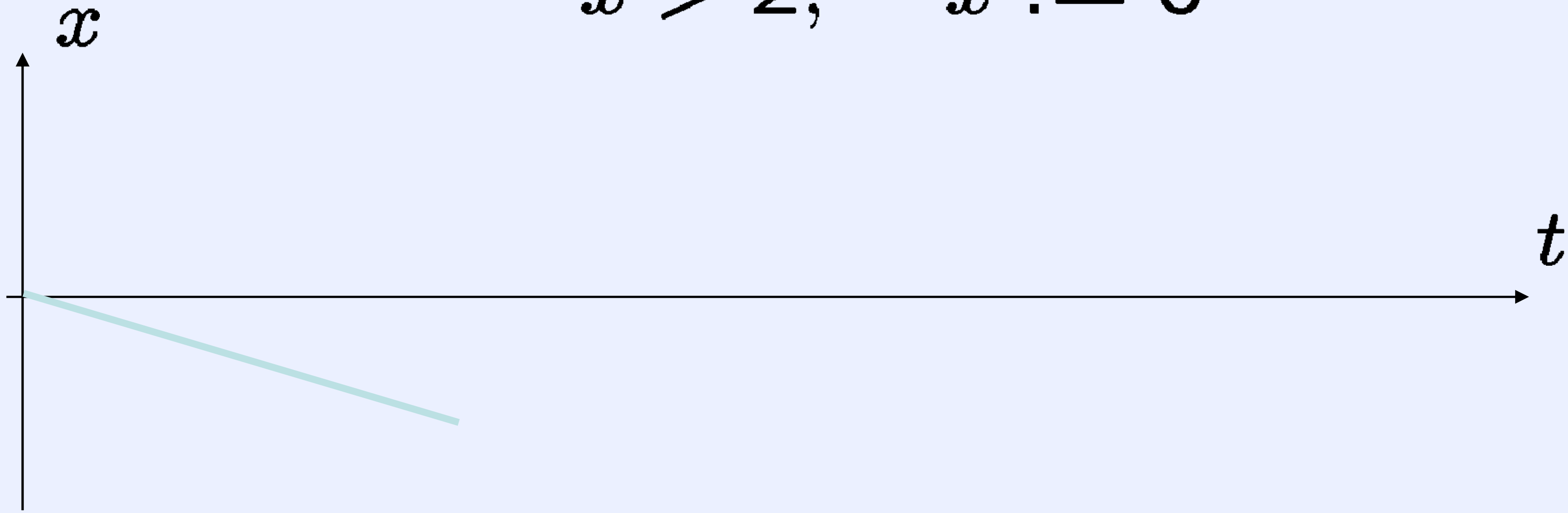


- Event-triggered

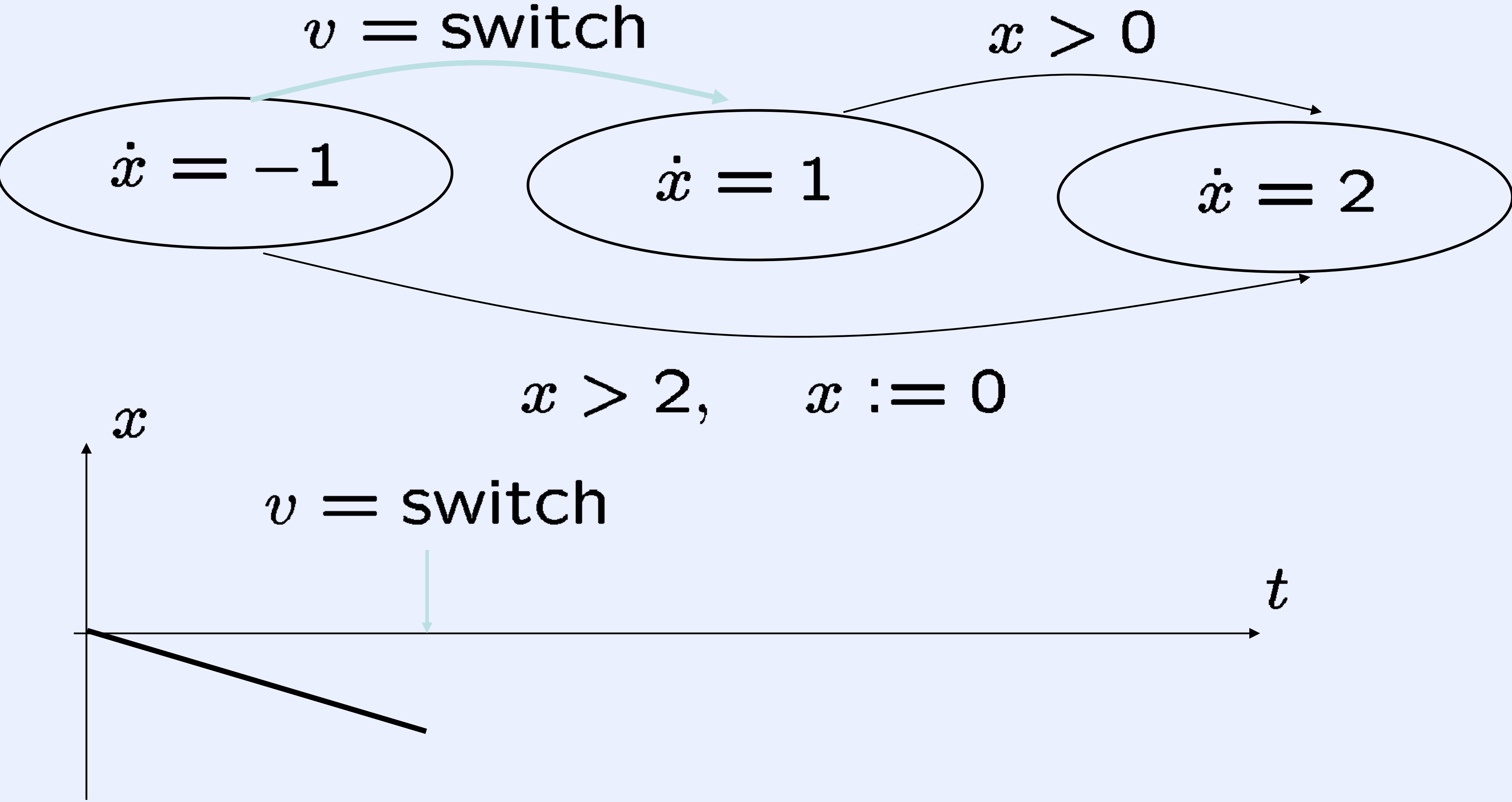


Example of a Hybrid System

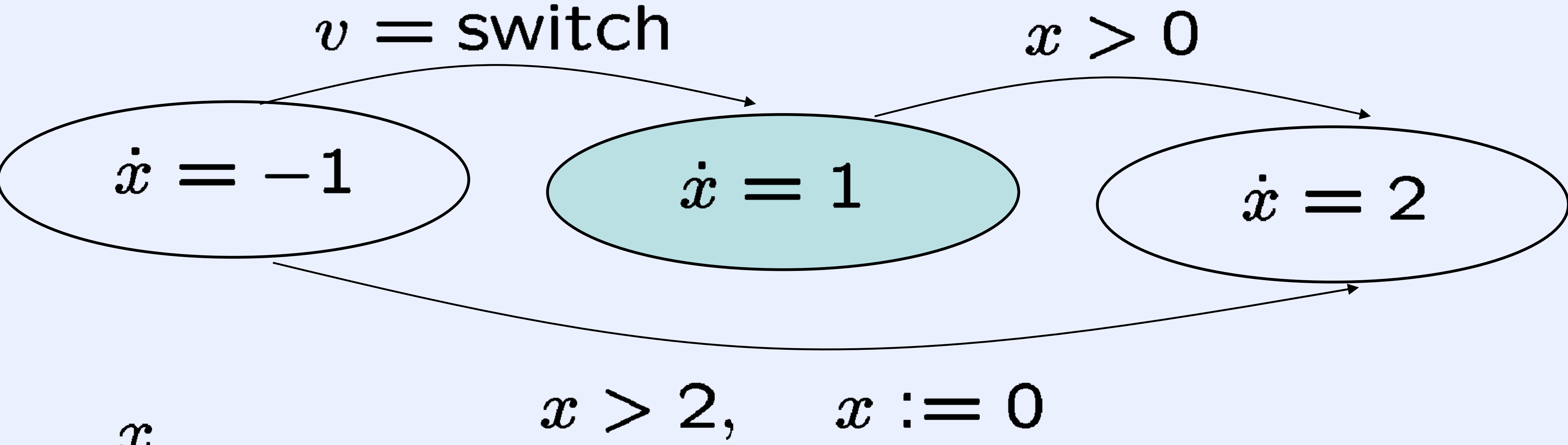


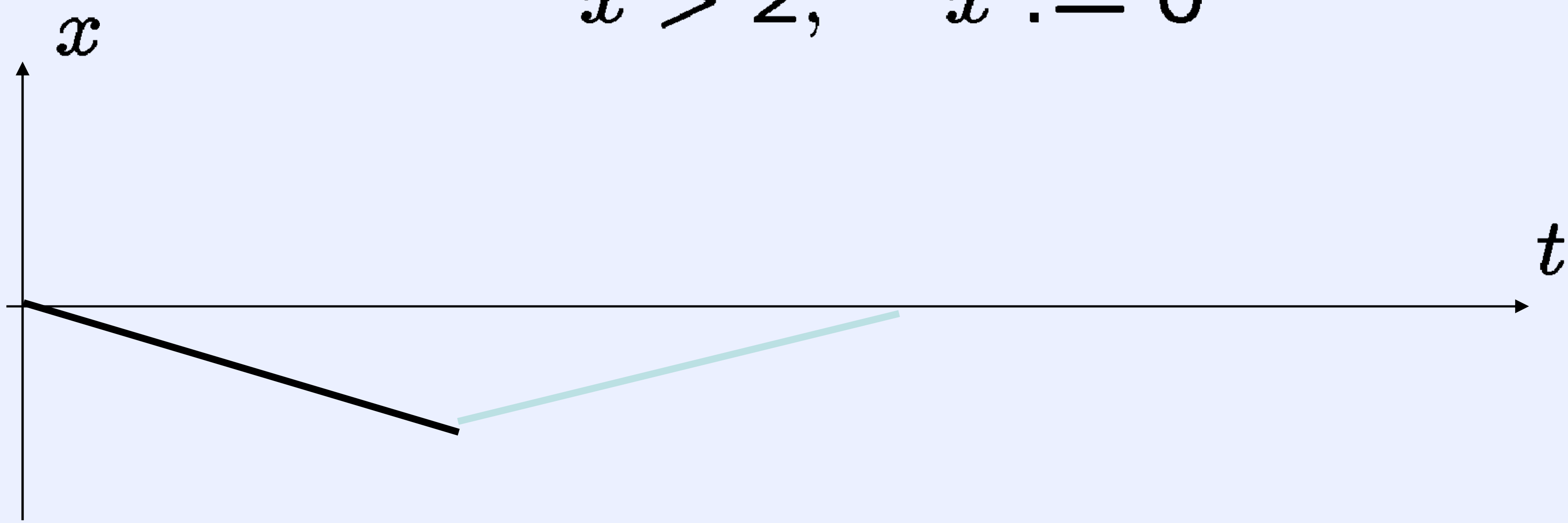


Example of a Hybrid System

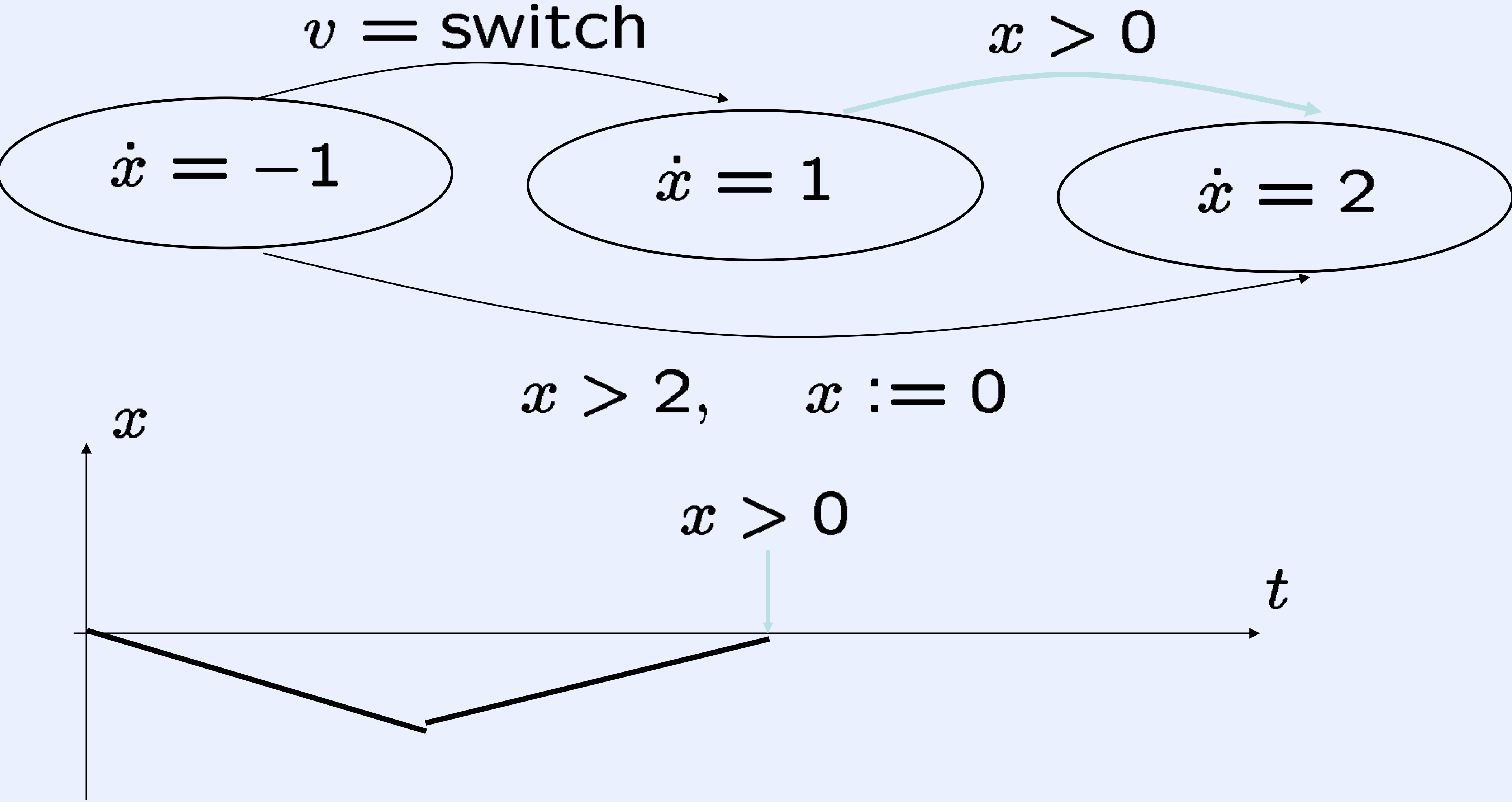


Example of a Hybrid System

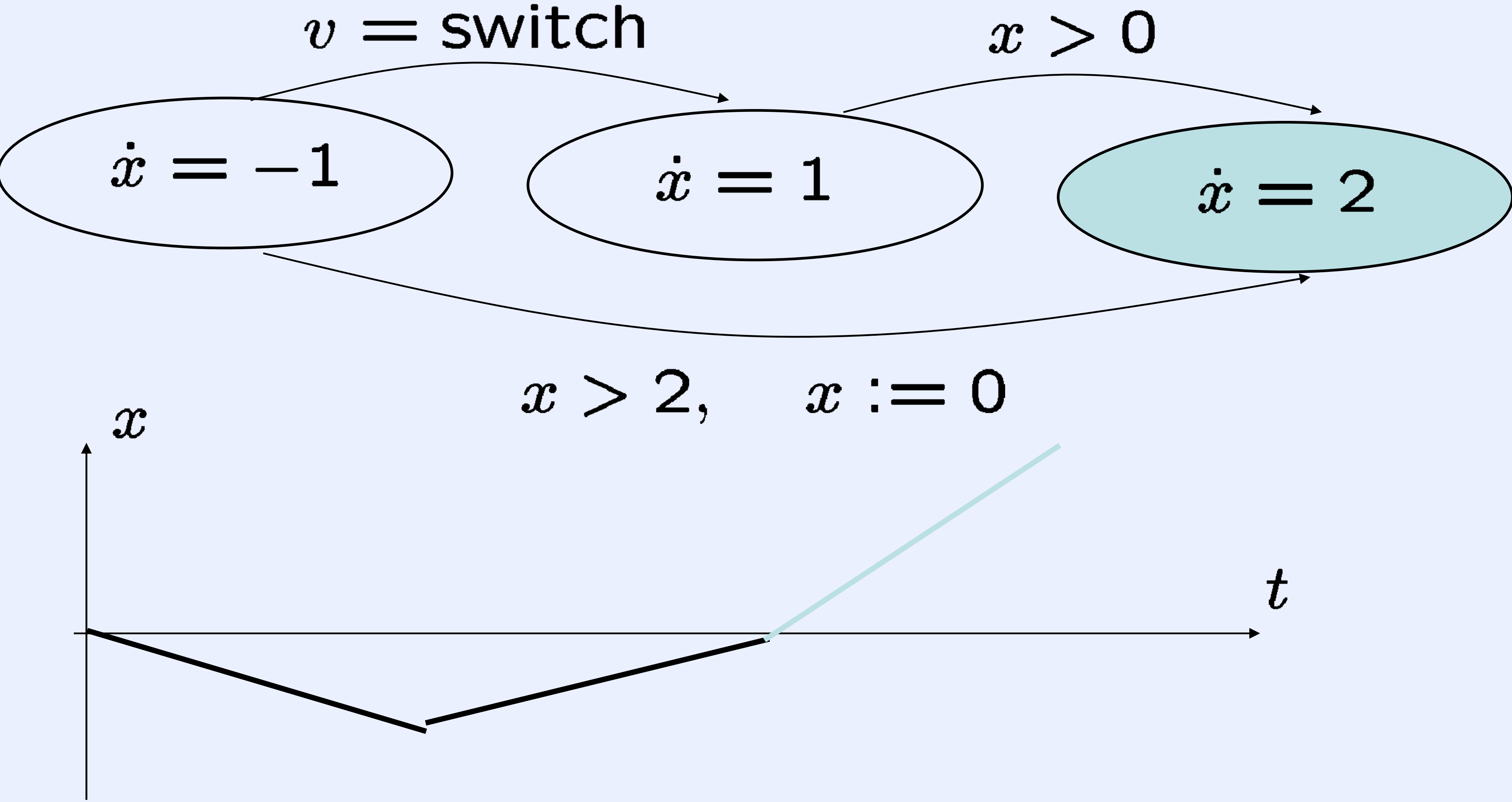




Example of a Hybrid System



Example of a Hybrid System



Why Hybrid Systems?

- Abstractions in design lead to hybrid dynamics
  - Time-scale separation, large scale systems
- Embedded computer systems are hybrid
  - Real-time software interacting with physical environment
- Control strategies are hybrid
  - On-off, optimal control, batch control, hierarchical control
- Improved performance
  - Brockett integrator, supervisory control, variable structure systems
- Nature is hybrid
  - Relays, impact mechanics, state constraints

Motivating Examples

- Automatic gear box
- Rocking block
- Internet congestion control
- Vacuum cleaning
- Multi-robot tracking
- Multi-robot flocking

Automatic Gear Box

Task: Design the control system for an automatic gearbox

x_1 is the longitudinal position of the car and x_2 its velocity

The dynamics (for a normalized car) can be written as

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \alpha_{\text{gear}}(x_2)u$$

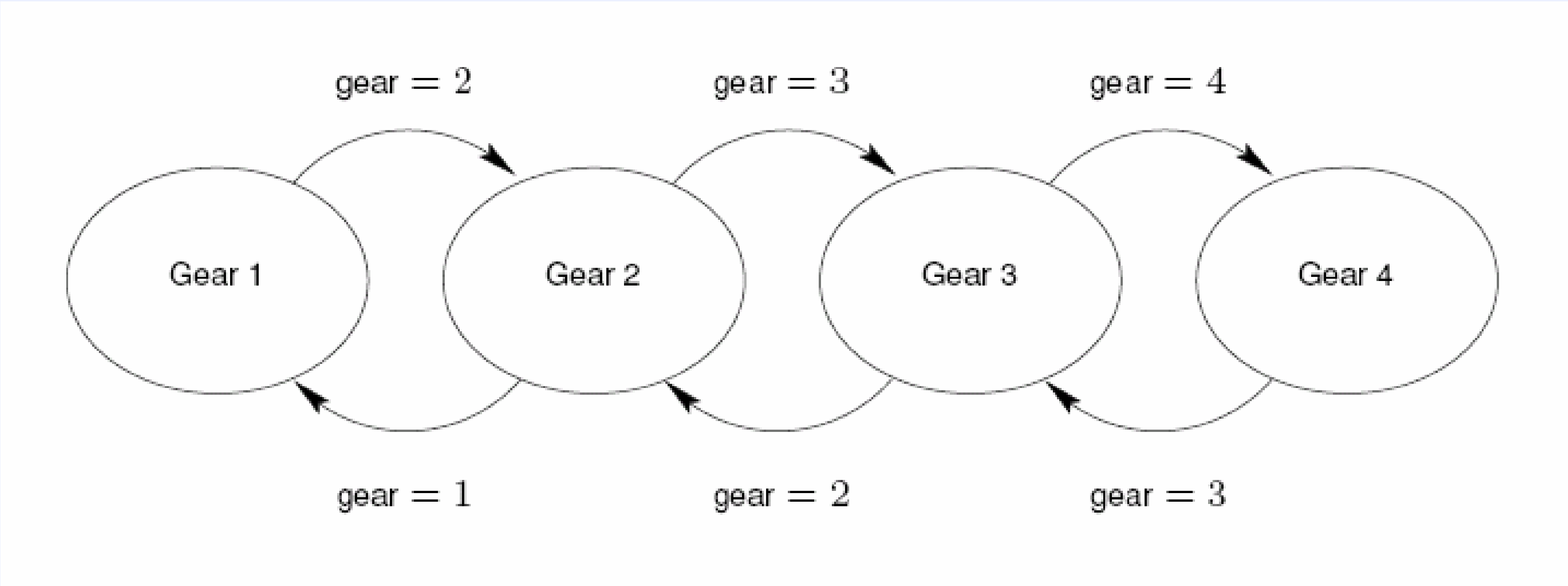
where u corresponds to the throttle position and

$\alpha_{\text{gear}}(\cdot)$ to the efficiency of a specific gear (draw a figure)

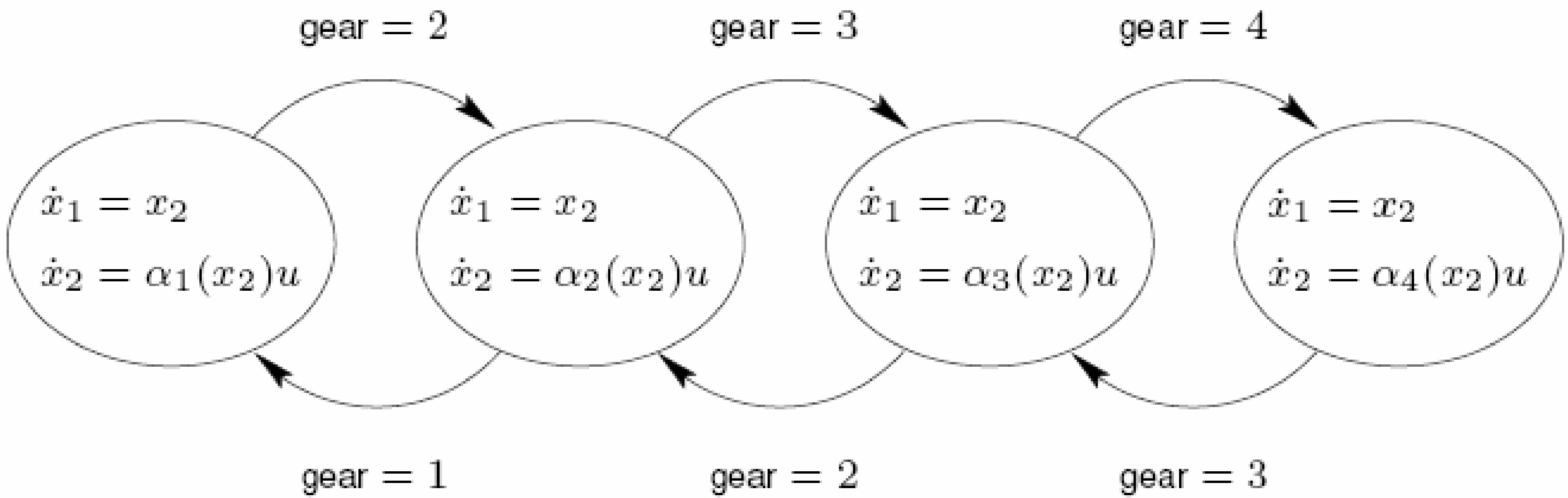
Note

- $u \in [0, u_{\max}]$ is a real-valued control
- gear $\in \{1, 2, 3, 4\}$ is an integer-valued control

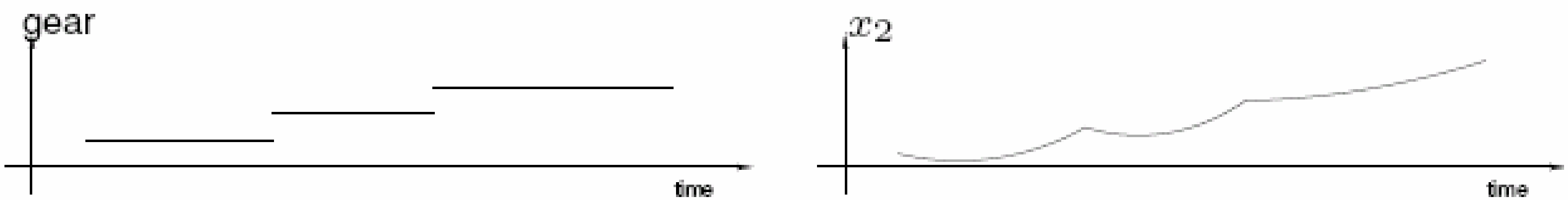
Discrete Event System



Hybrid System



Typical solutions:



How choose u and gear in a good (optimal) way?

Rocking Block

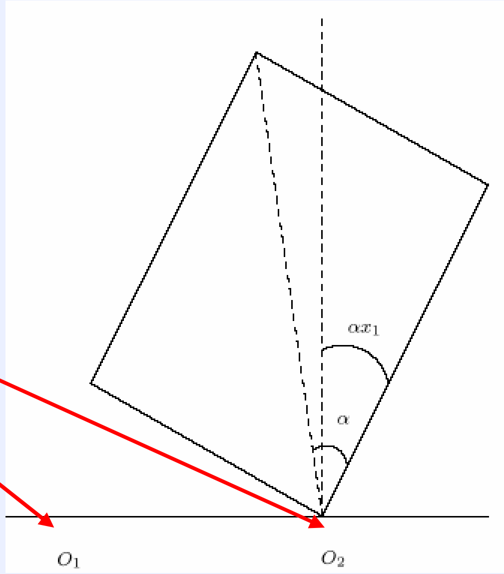
Hybrid models capture mechanical impacts and other discontinuous dynamics

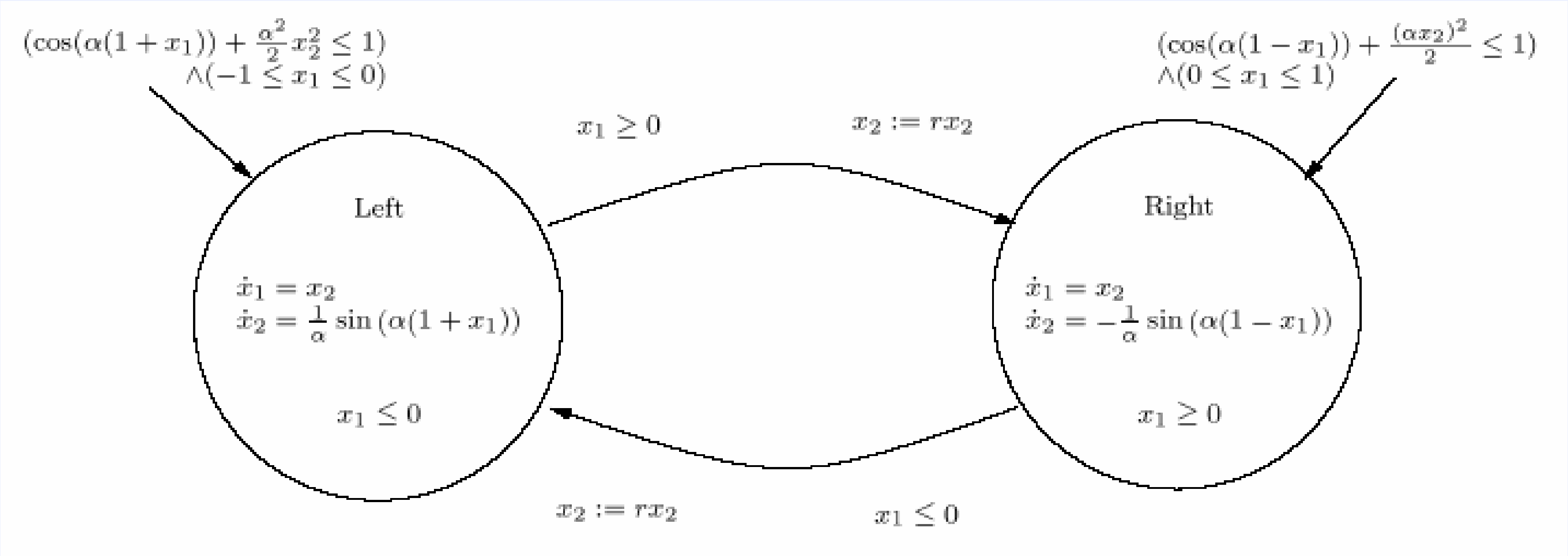
www.animats.com

Rocking Block

- Rocking block rotates around one of two pivot points
- Impacts represented as discrete transitions
- System may show complex dynamics

- Extensively studied as model for nuclear reactors, electrical transformers and tombstones

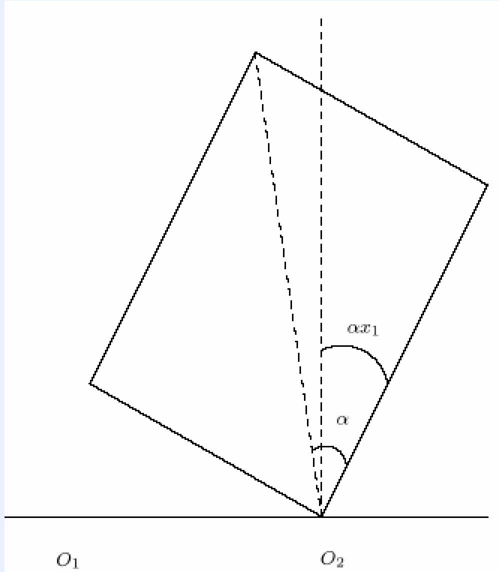


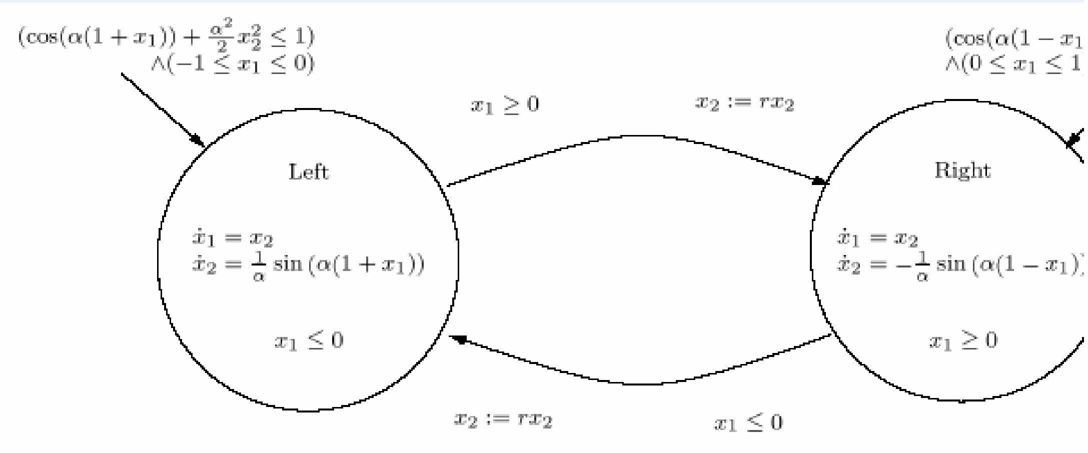


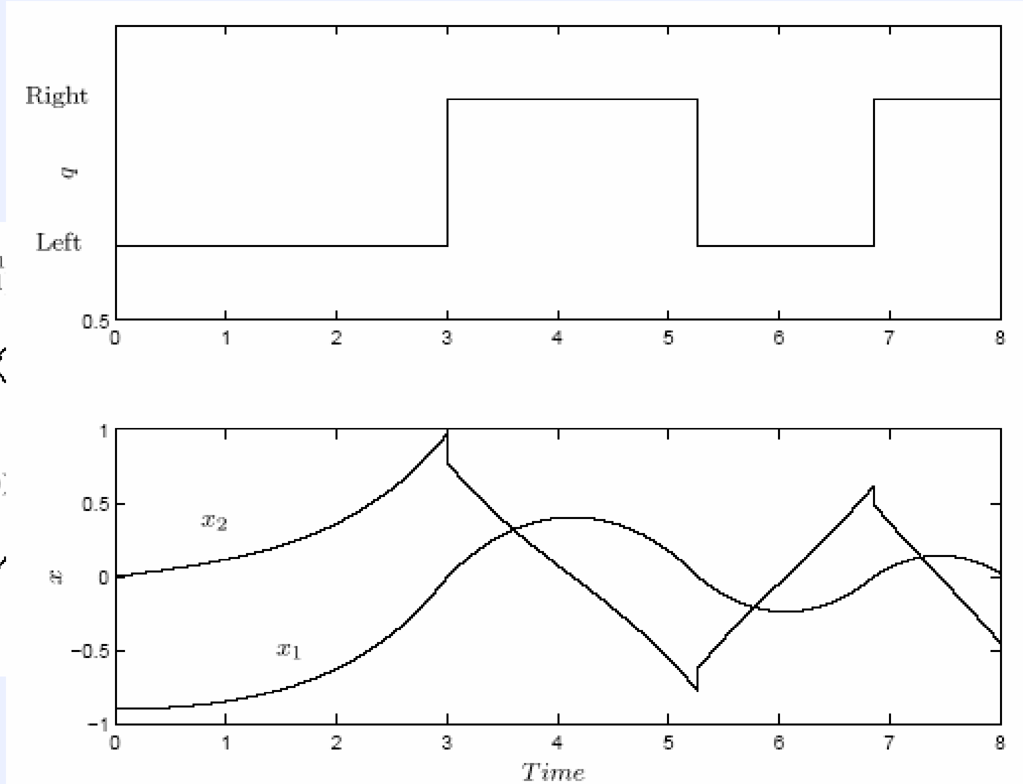
Rocking Block

- Rocking block rotates around one of two pivot points
- Impacts represented as discrete transitions
- System may show complex dynamics

- Extensively studied as model for nuclear reactors, electrical transformers and tombstones

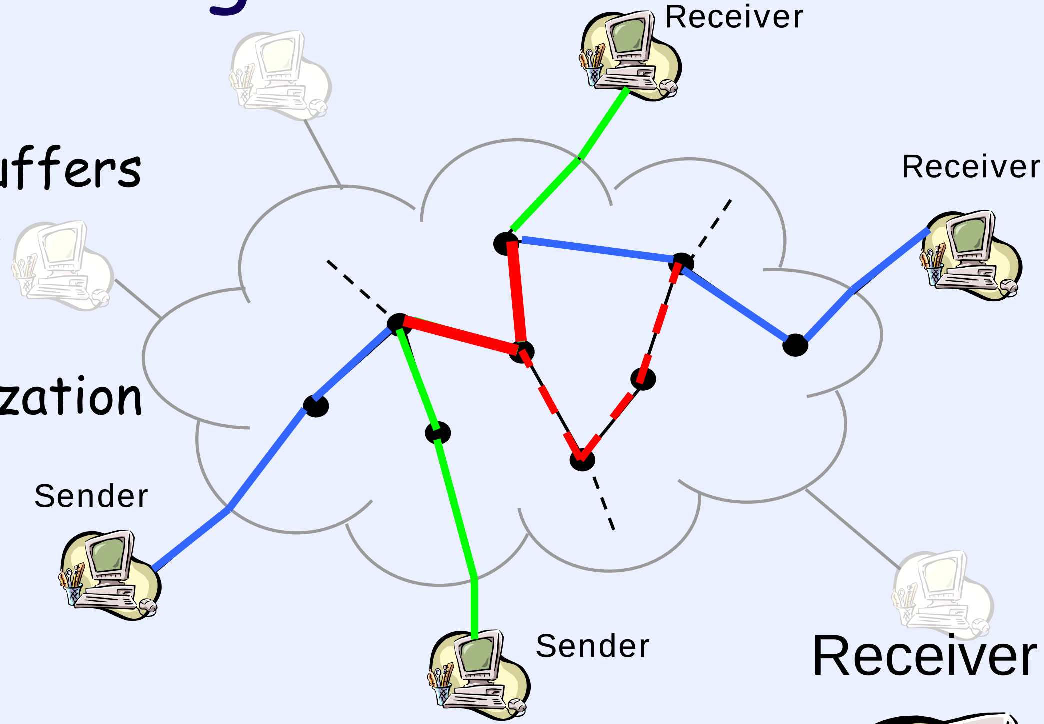


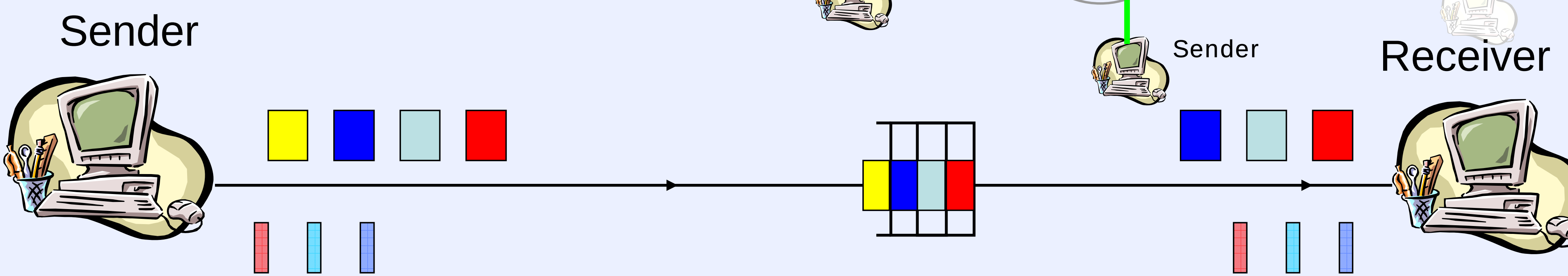


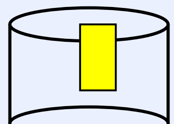


Network Traffic Congestion Control

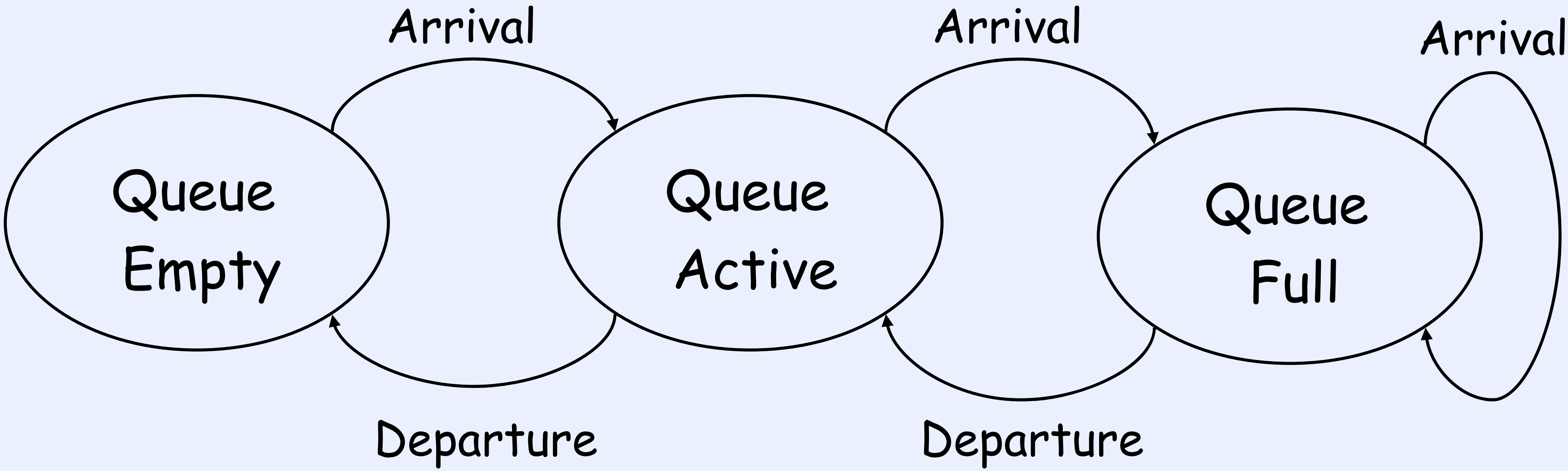
- Temporary storage of data in router buffers
- Sender receives data acknowledgments from receiver
- Regulate sending rate to maximize utilization

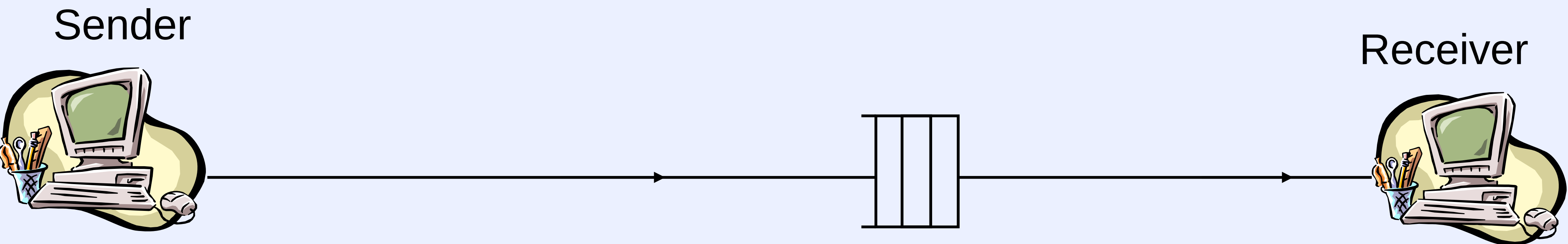






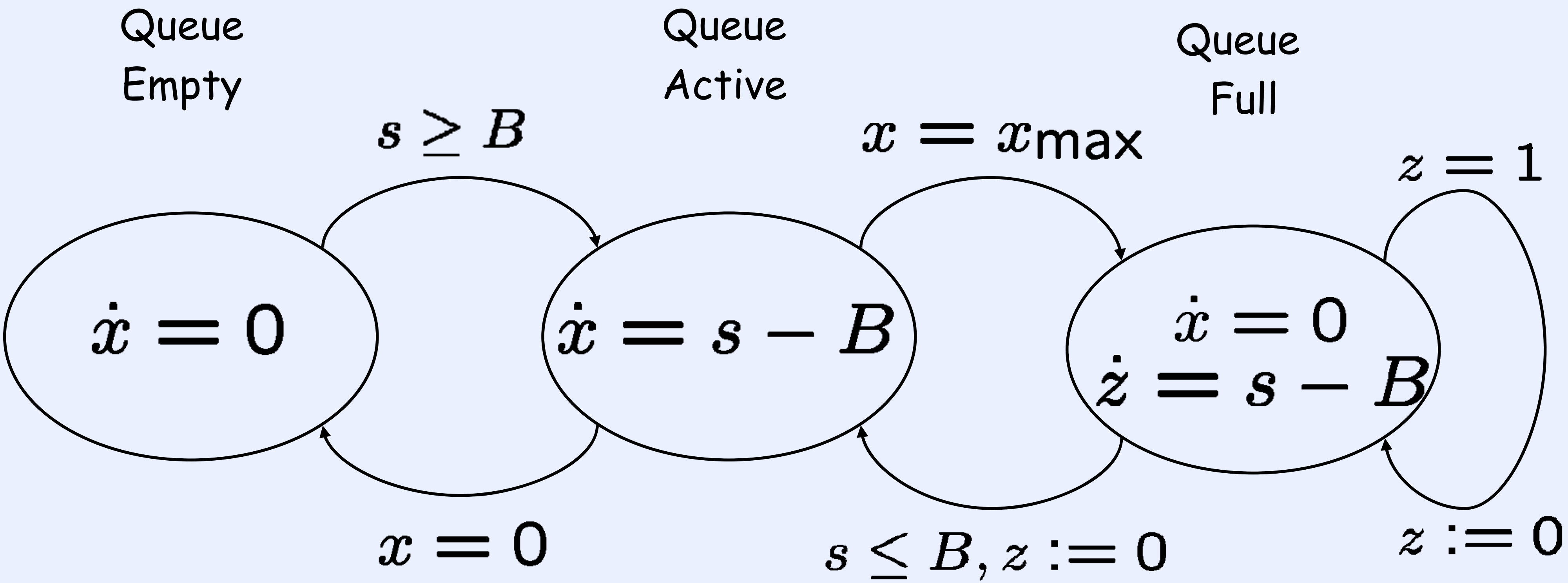
Queue Model for Router Node

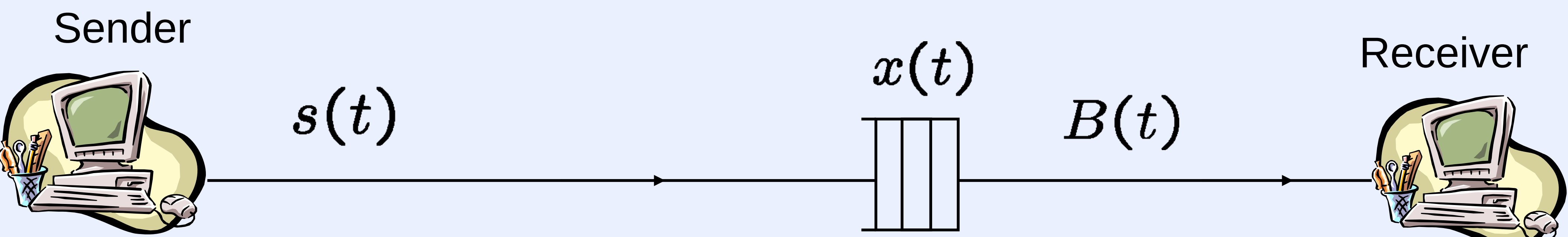




Hybrid Queue Model

Replace discrete variables (inputs and states) by continuous approximations



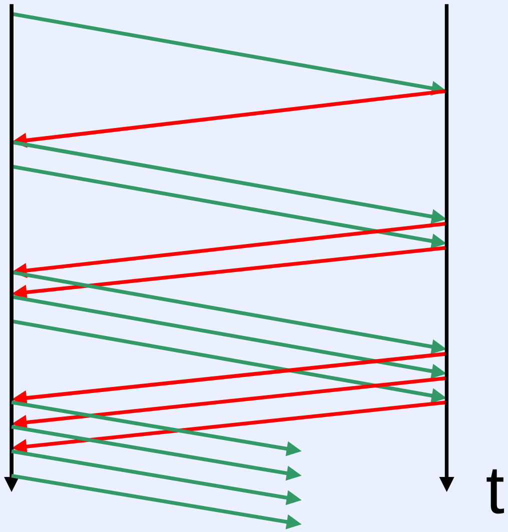


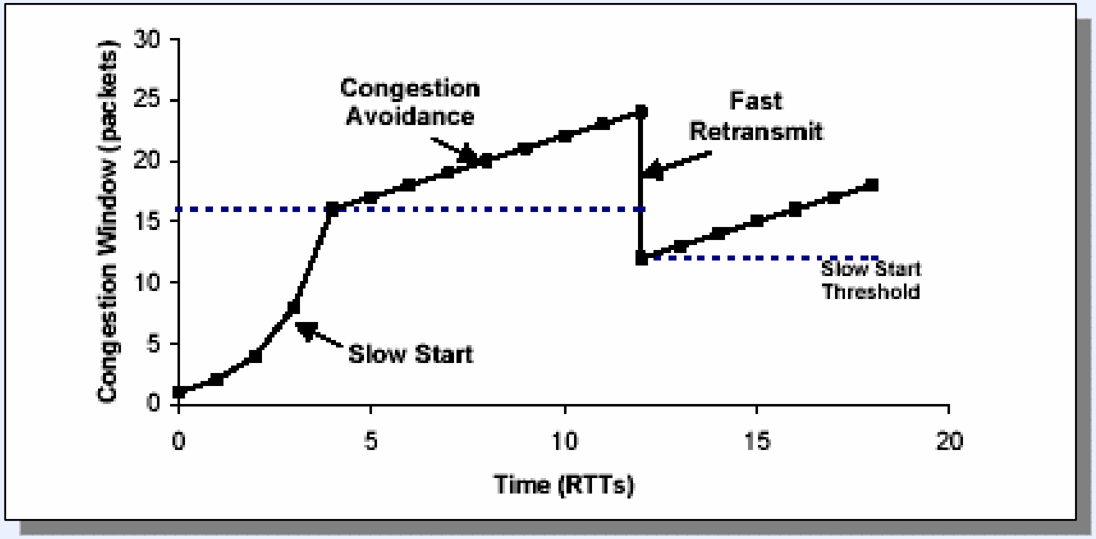
Transmission Control Protocol (TCP)

- Regulates transmission rate in each sender
- Receiver acknowledges received data (ACK's)
- Additive increase multiplicative decrease (AIMD)

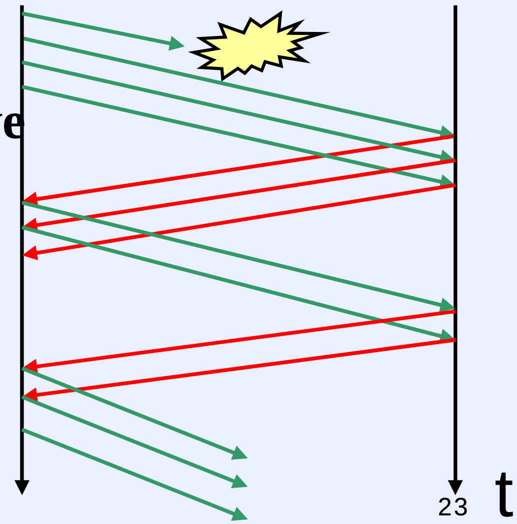
- Probes available bandwidth
- Implicit feedback of network state
- Packet drops should indicate traffic congestion

Additive increase

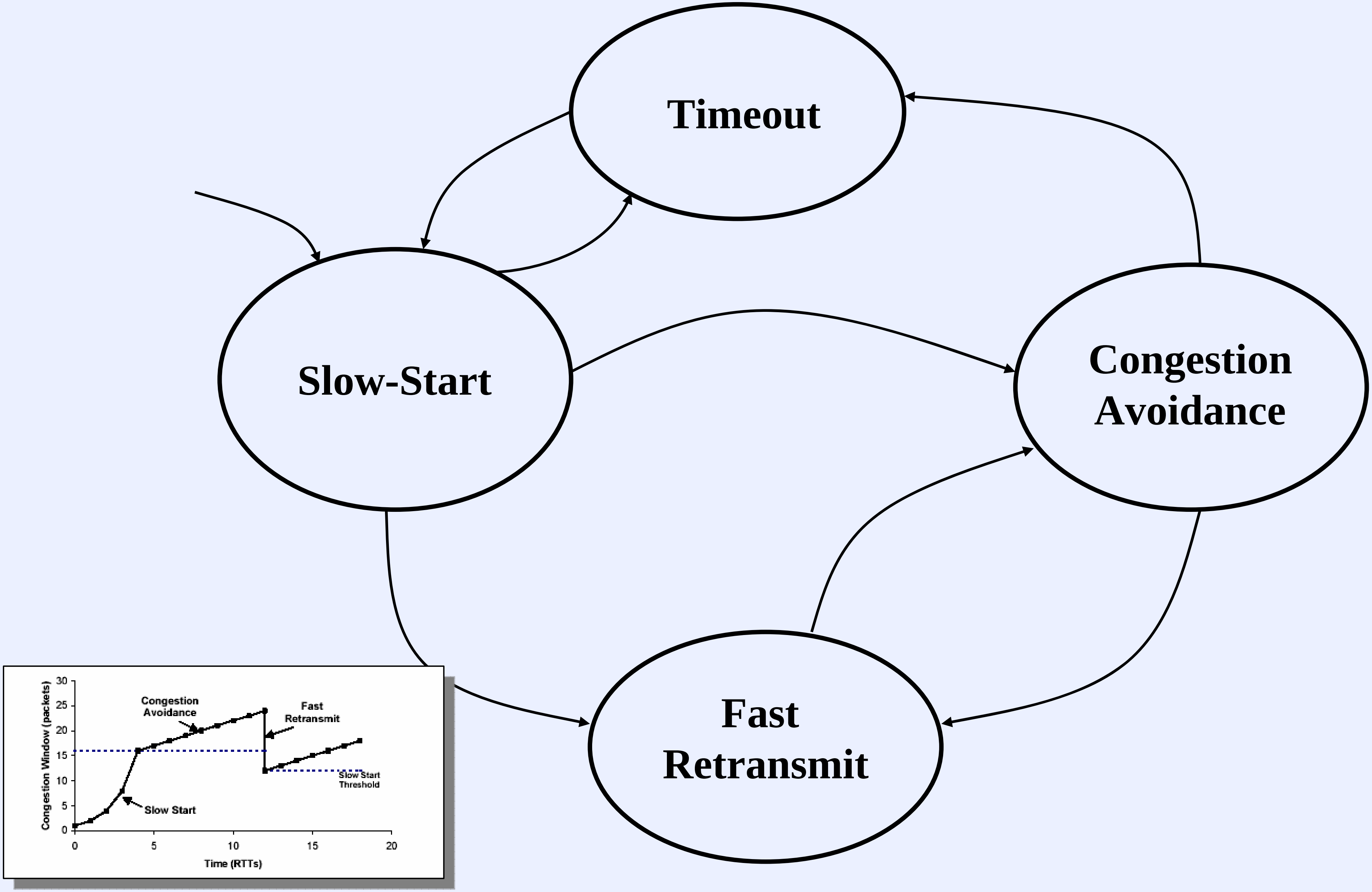




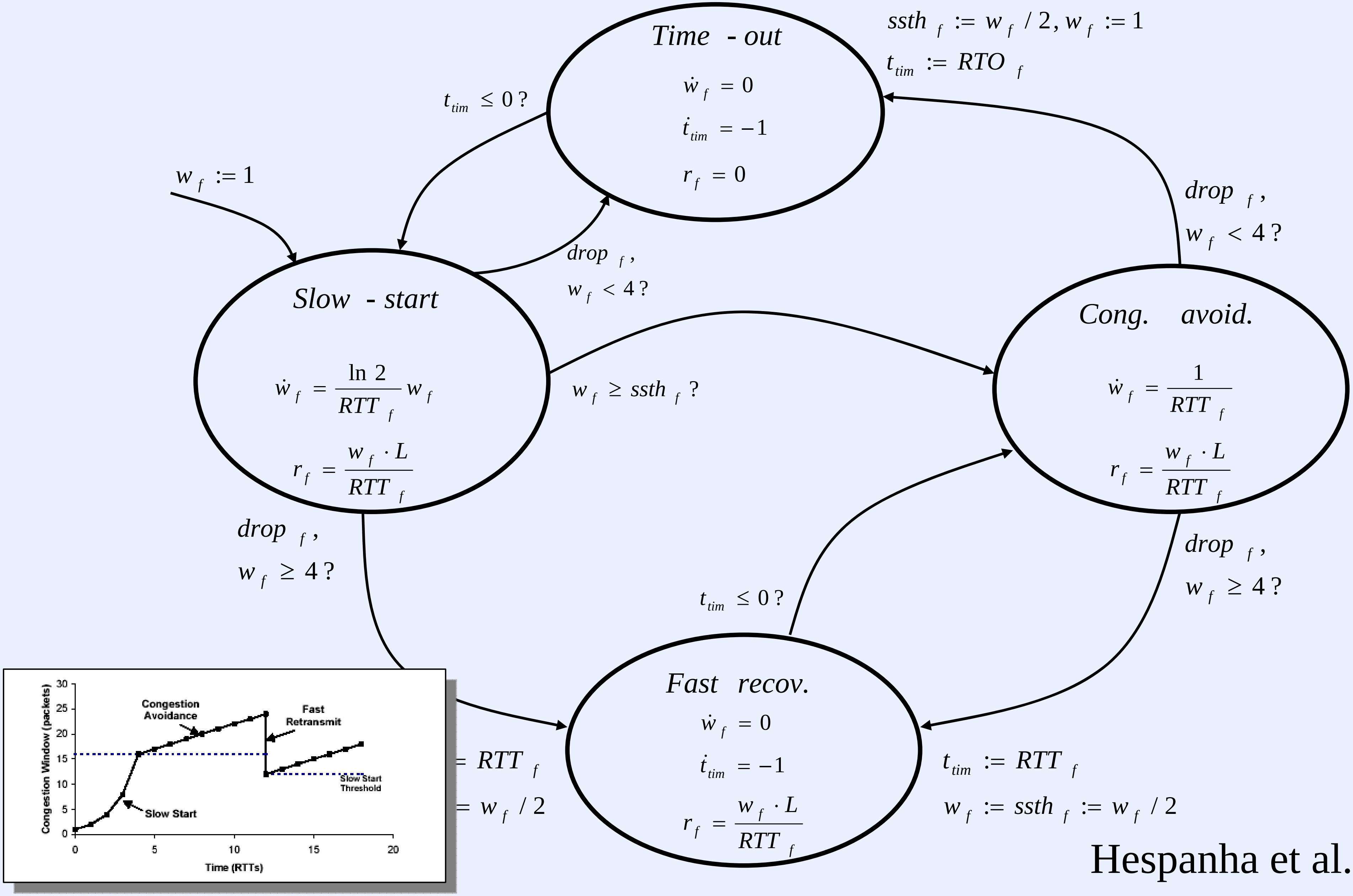
Multiplicative decrease



TCP Implements a Hybrid Controller

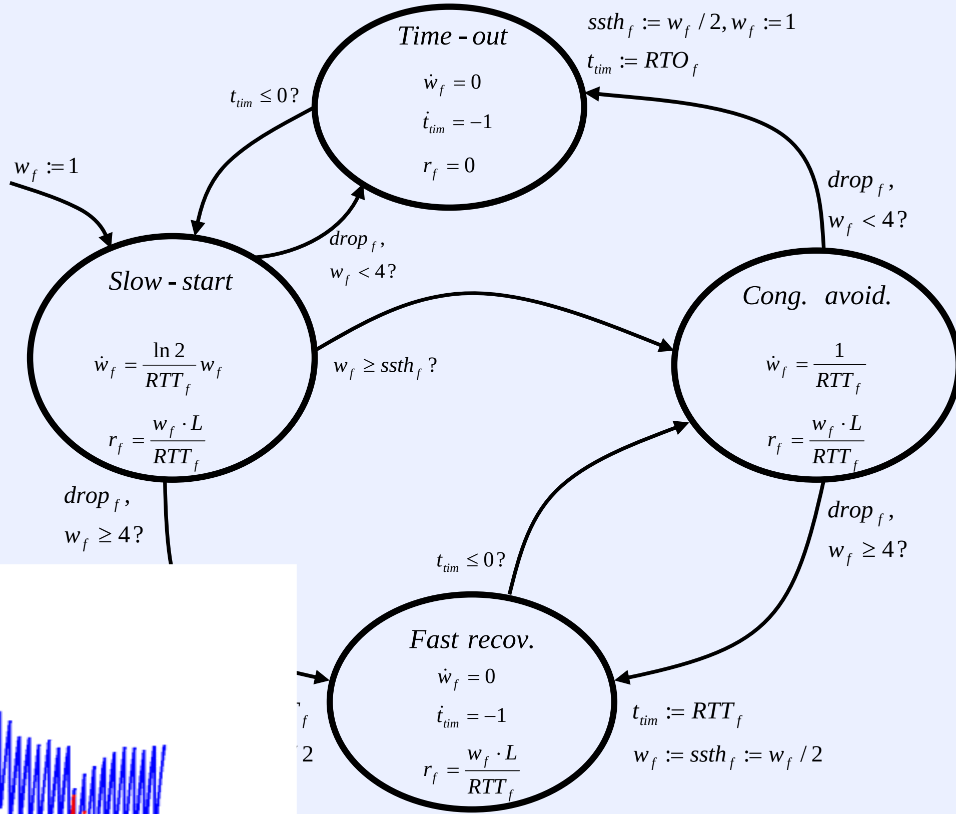


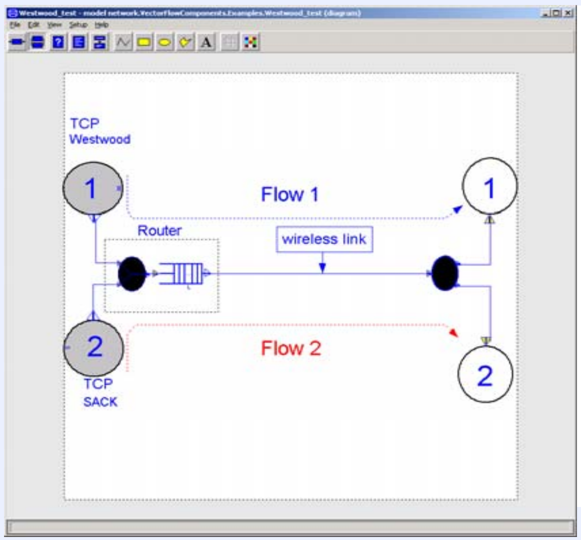
TCP Implements a Hybrid Controller

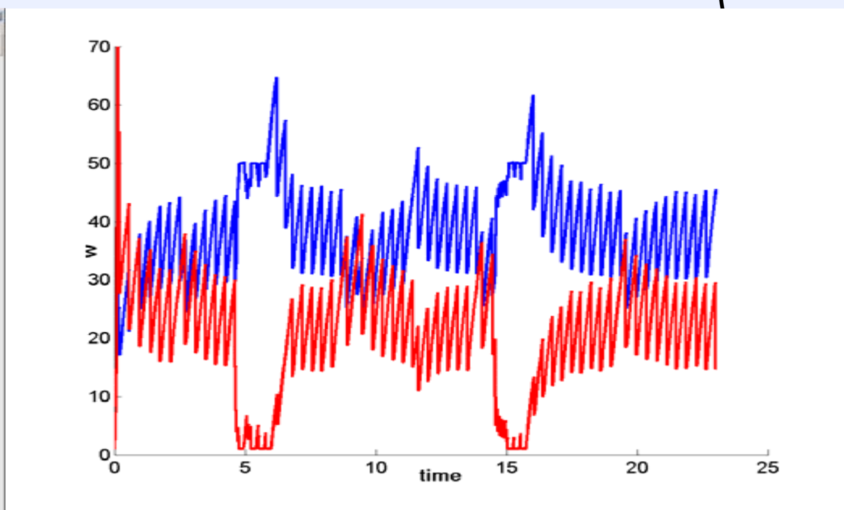


Time-Scale Separation

- Replace (fast) discrete window updates by continuous approximation
- Reasonable at time-scale of congestion control
- Enables analysis and more efficient simulations

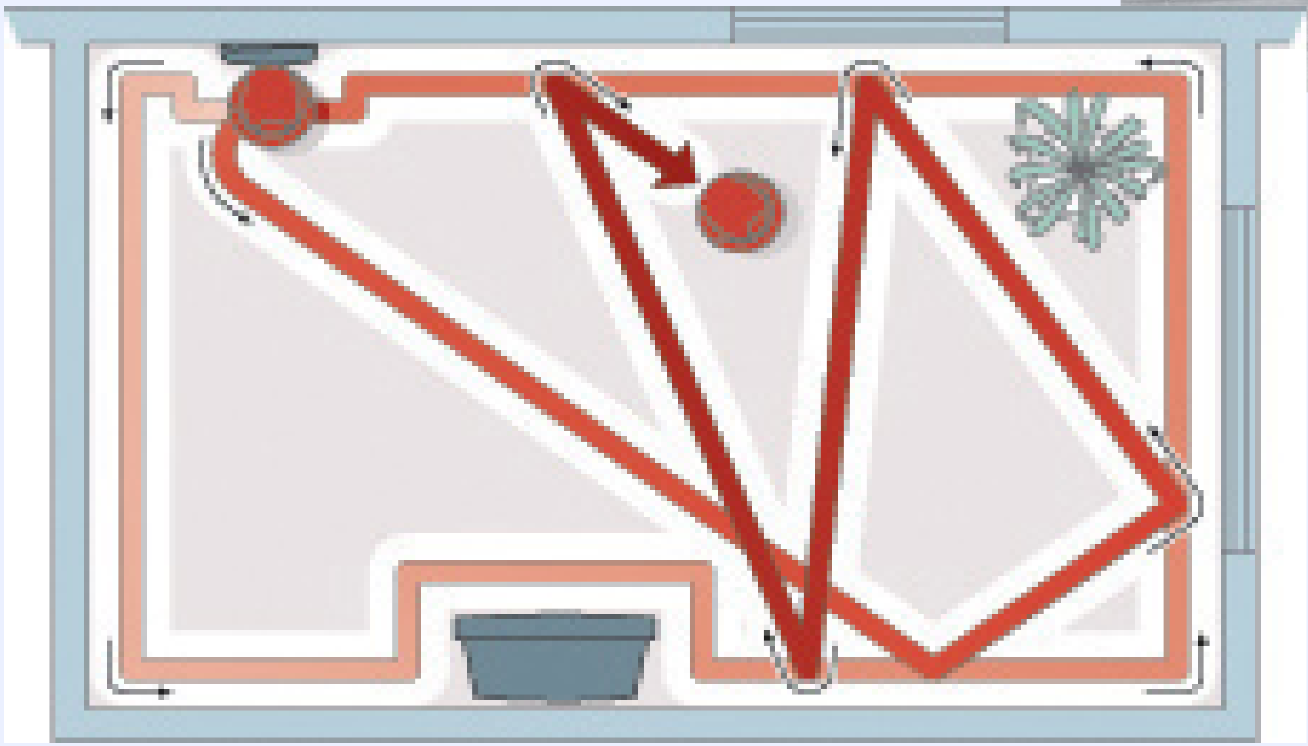






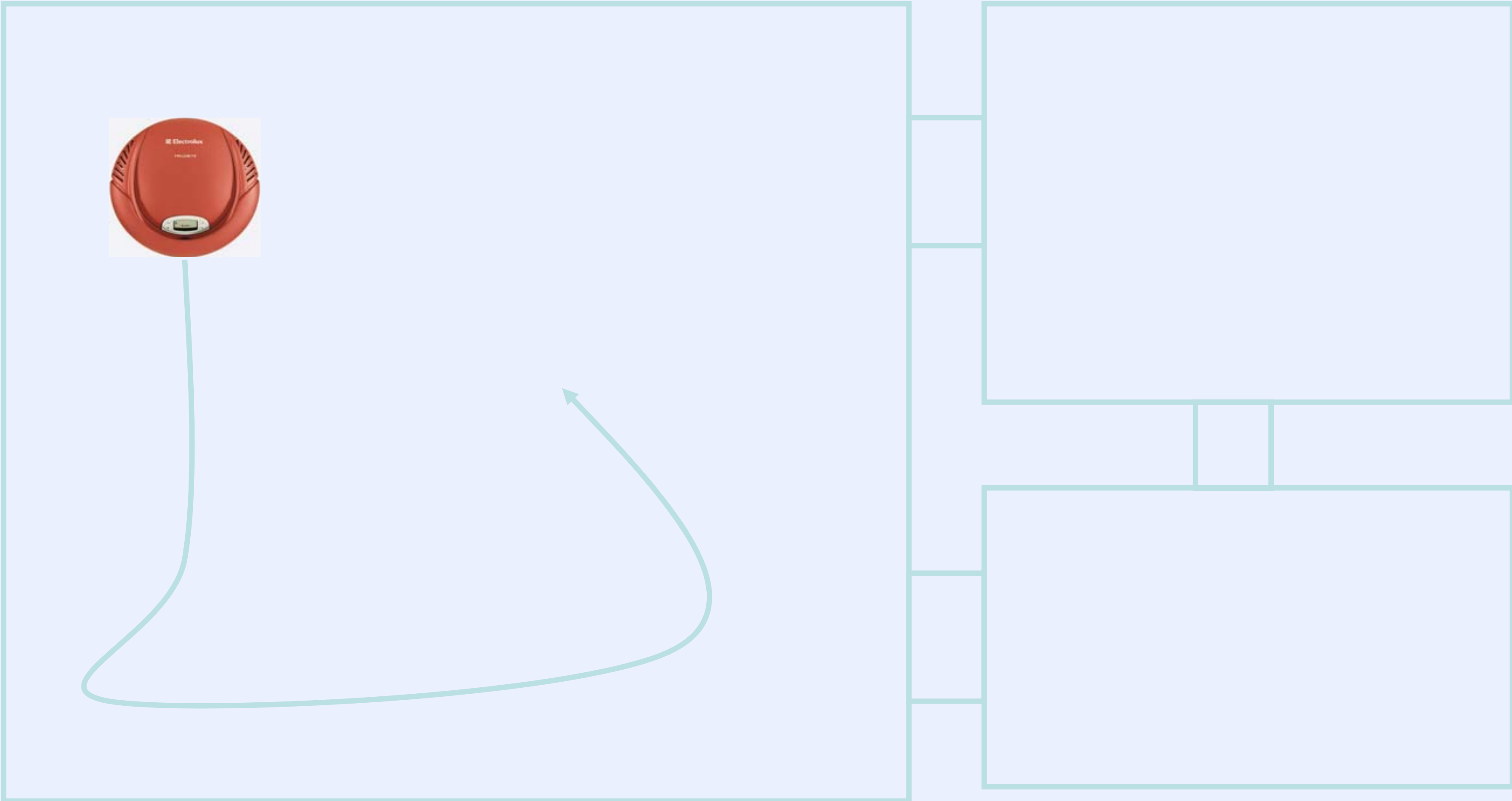
Vacuum Cleaning

- Find an efficient strategy for autonomous cleaning of an apartment

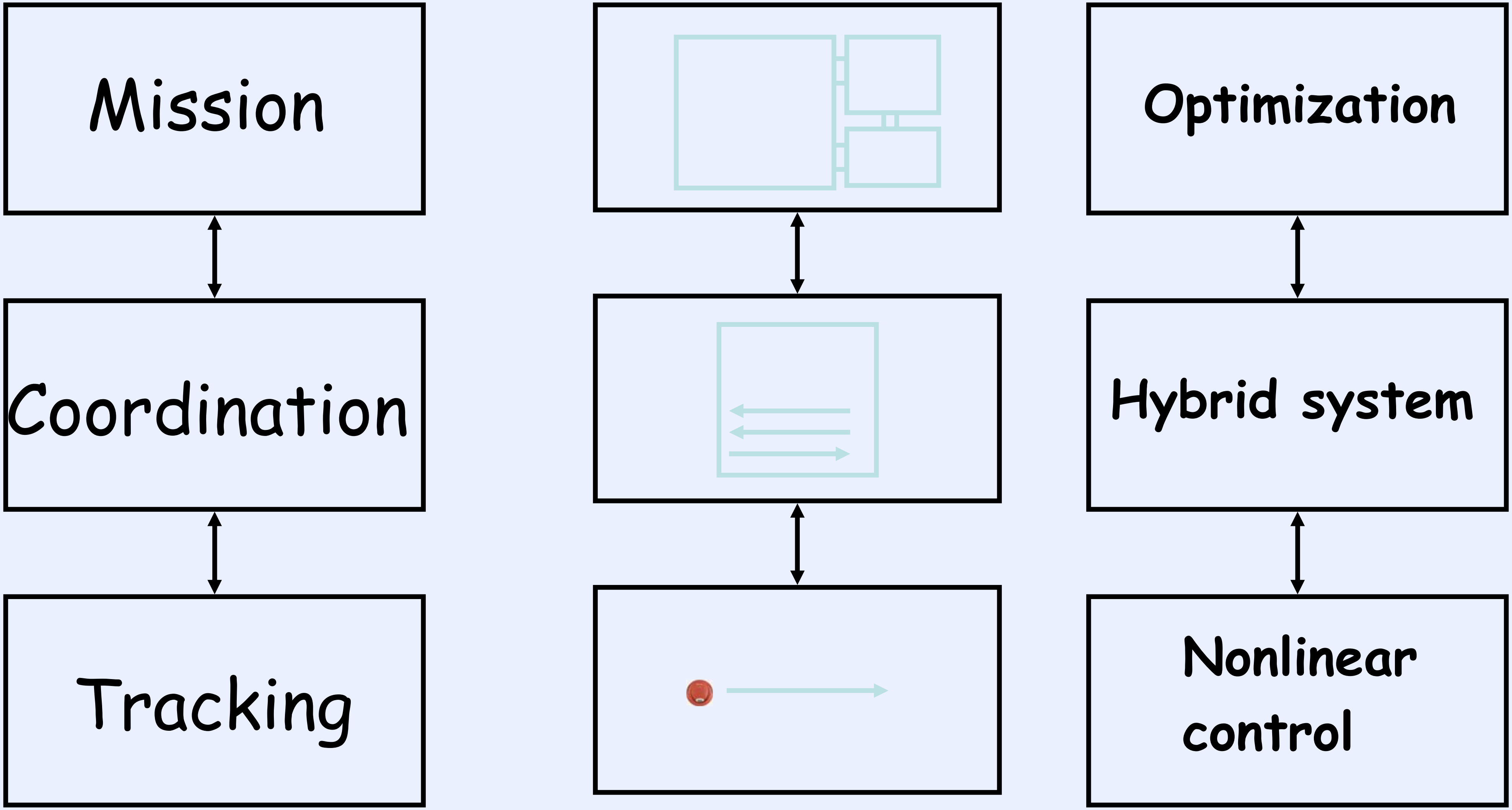


Efficient Area Coverage

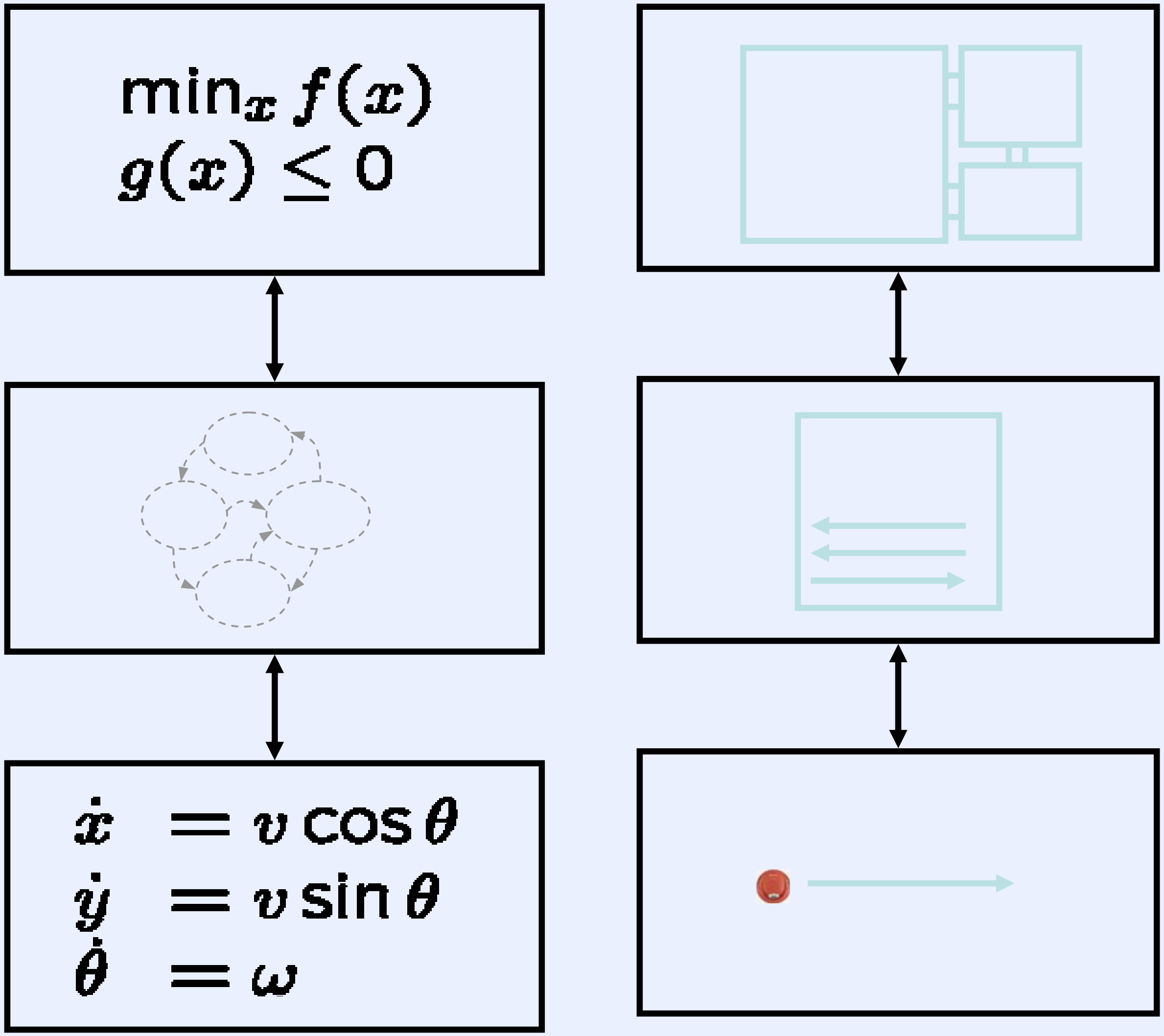
Constrained by nonlinear and uncertain dynamics, sensor noise, actuator limitations, unknown obstacles in environment etc.



Hierarchical Problem Structure



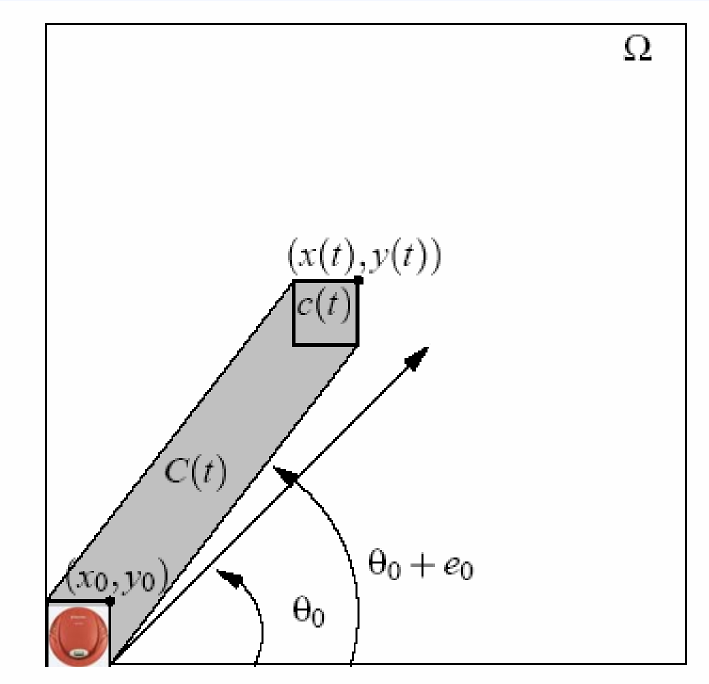
Hierarchical Control



- Natural to organize large systems into a hierarchy
- Divide into manageable layers
- Widely adopted approach in engineering: manufacturing, robotics, transportation etc.

- Cross-layer interaction results in hybrid systems

Area Coverage with Uncertain Heading



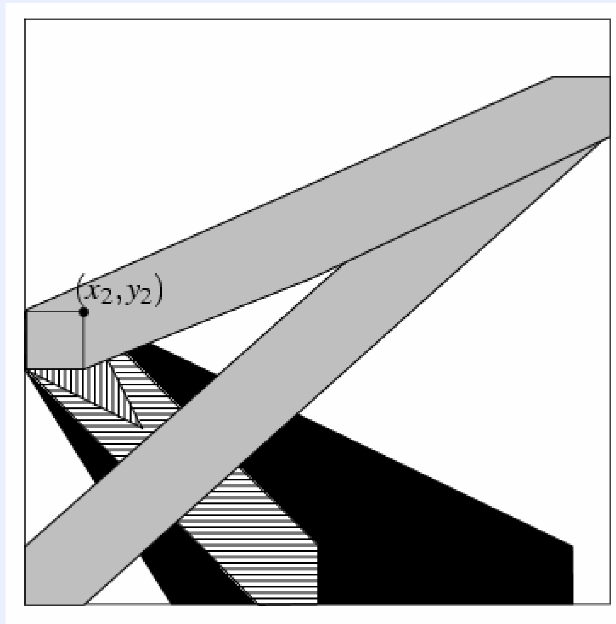
Robot motion governed by

$$\dot{x} = \cos(\theta + e)$$
$$\dot{y} = \sin(\theta + e)$$

where θ is heading, controlled when $c(t) \cap \partial\Omega \neq \emptyset$

$|e| < \varepsilon$ represents uncertainty in control

Problem: Given $\varepsilon > 0$, minimize number of turns N to cover Ω



Mazo & J, 04

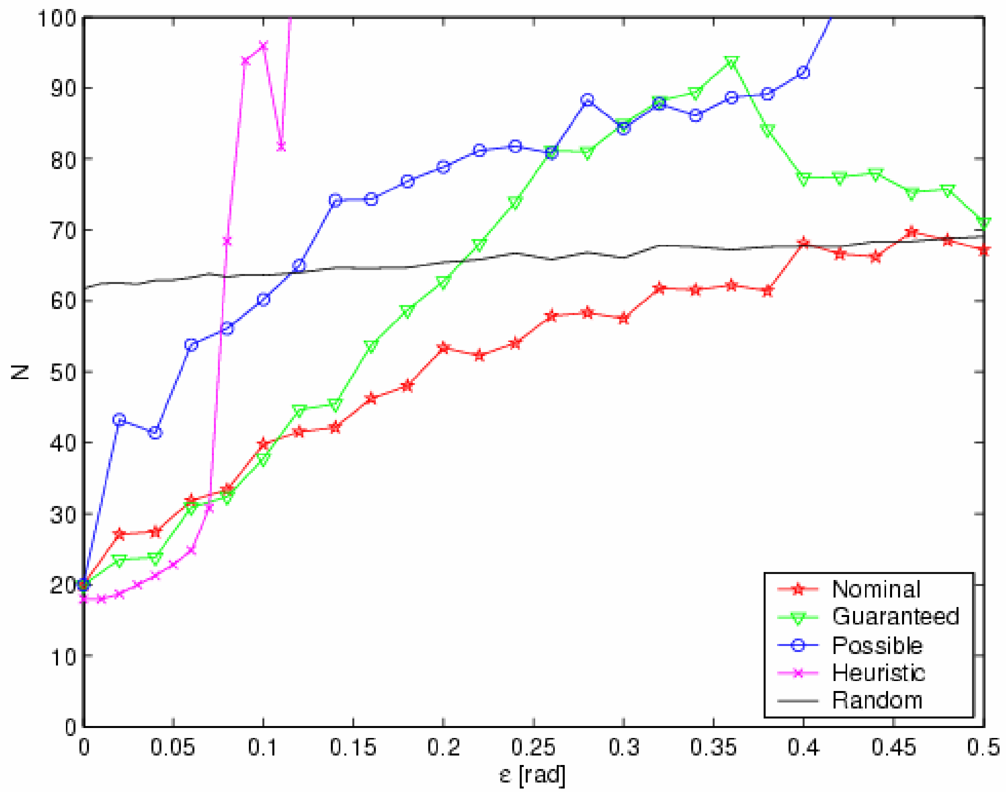
Comparison of Hybrid Controllers

Robot motion governed by

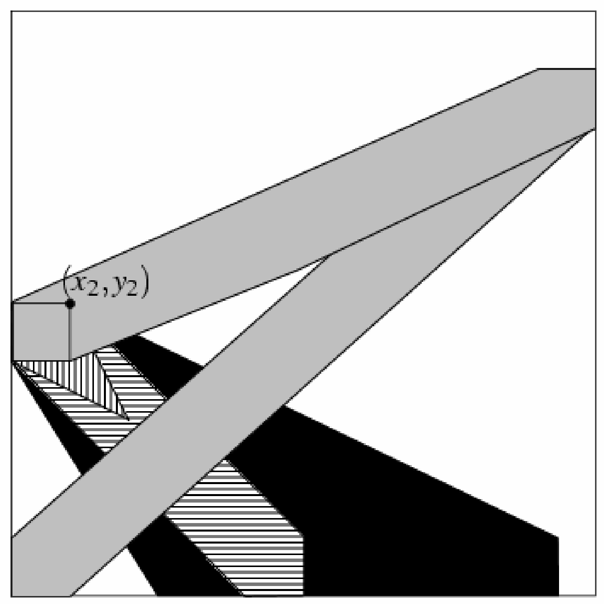
$$\dot{x} = \cos(\theta + e)$$
$$\dot{y} = \sin(\theta + e)$$

where θ is heading, controlled when $c(t) \cap \partial\Omega \neq \emptyset$

$|e| < \varepsilon$ represents uncertainty in control

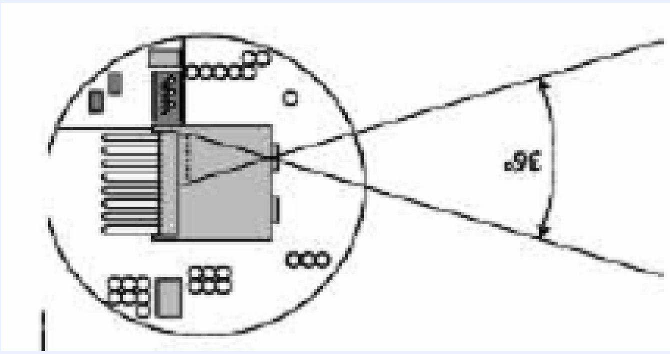


Given $\varepsilon > 0$, minimize number of turns N to cover Ω

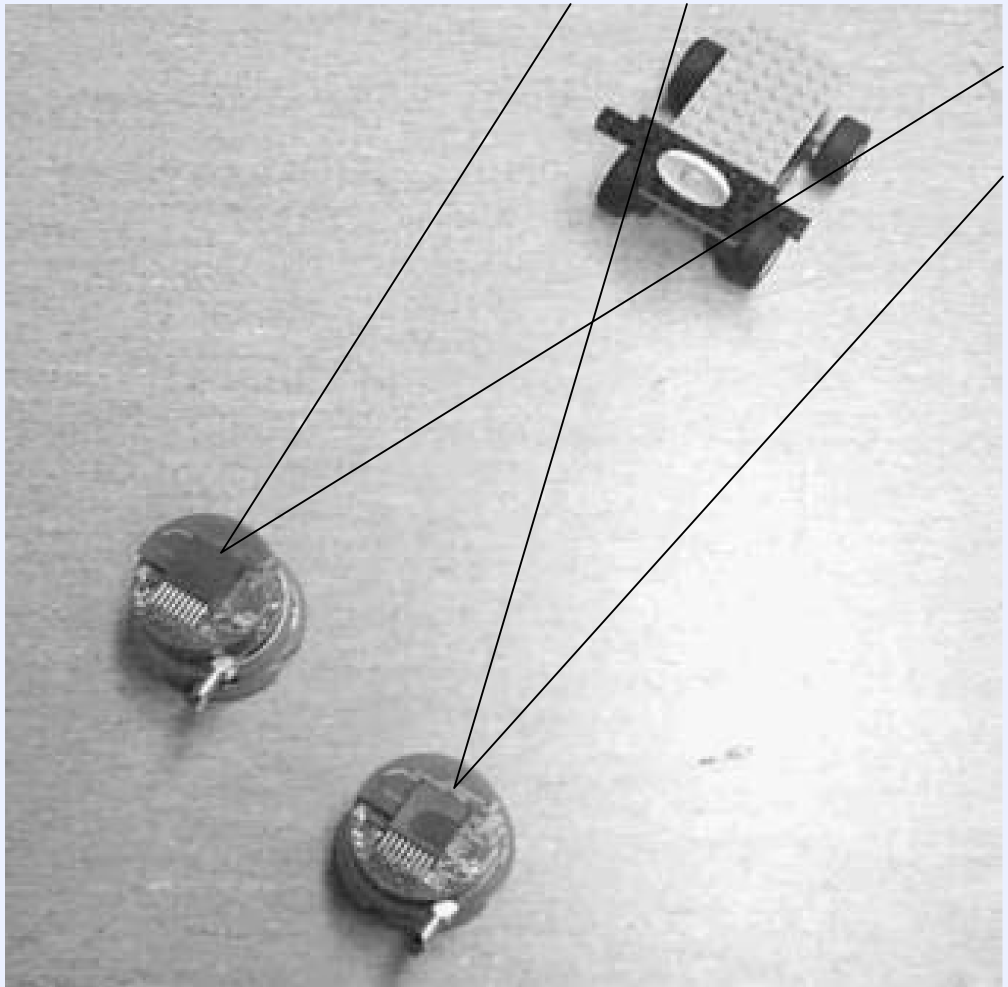


How Track a Moving Object with Directional Sensors?

- Target moving along smooth trajectory
- Two tracking unicycle robots (Khepera II) with inter-robot communication
- Directional sensors with limited range



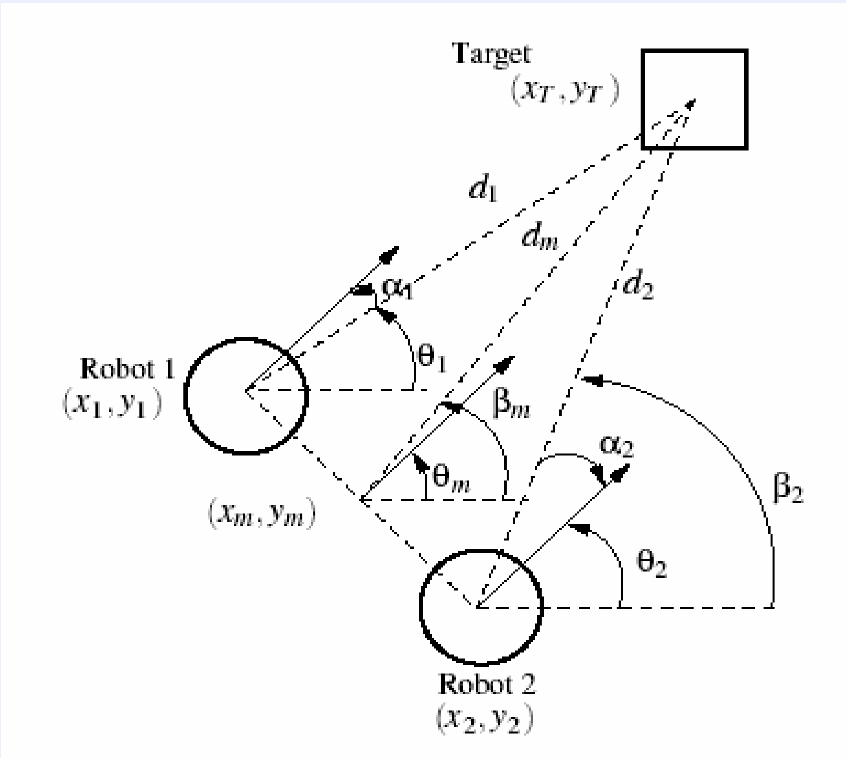
$$\dot{x}_i = v_i \cos\theta_i$$
$$\dot{y}_i = v_i \sin\theta_i \qquad i = 1, 2$$
$$\dot{\theta}_i = \omega_i$$

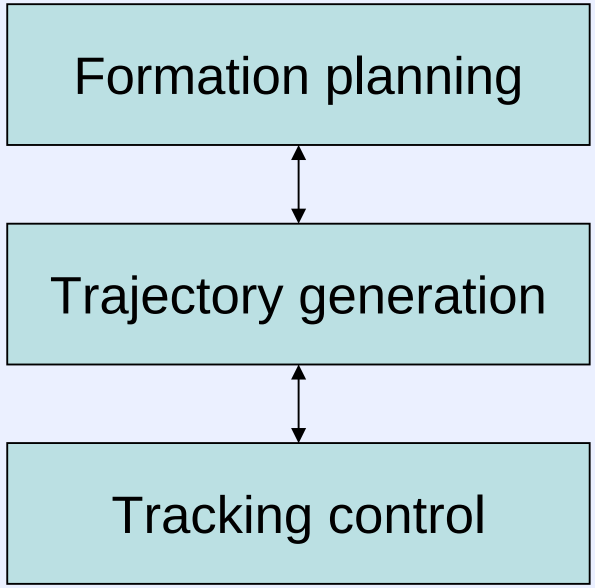


Mazo, Speranzon, J, Hu, 04

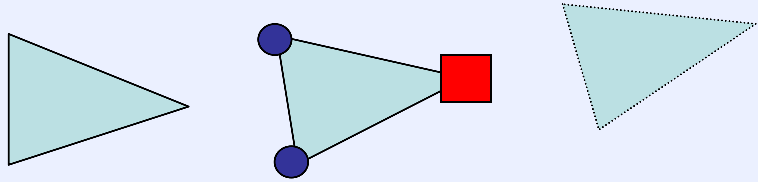
Hierarchical Control Strategy

- Plan desired robot formation that is suitable for robust sensing
- Generate trajectories that connect present and desired formations
- Track trajectories by low-level control



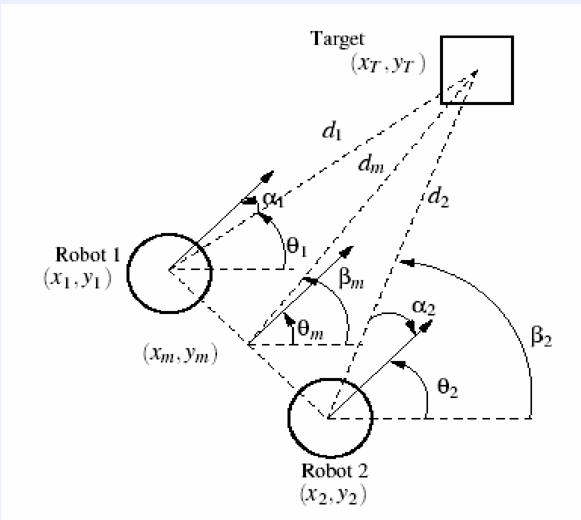


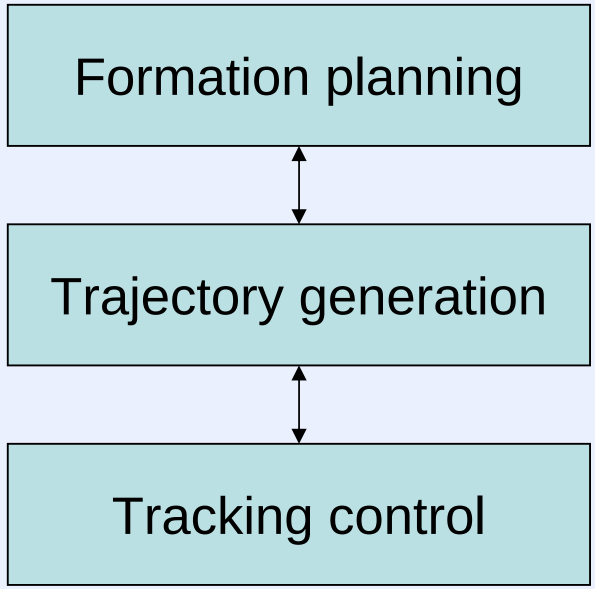
Formation Planning



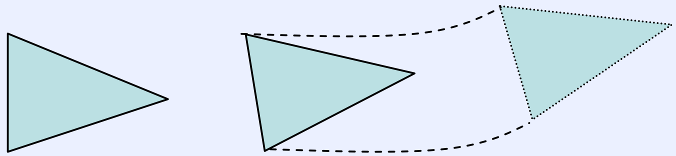
- Robot formation suitable for collaborative estimation of target position under directional sensor constraints
- Estimate target's current state $\hat{x}_T(t_k), \hat{y}_T(t_k), \hat{v}_T(t_k), \hat{\theta}_T(t_k)$ at discrete events t_k
- Predict target's position at next formation update

$$\begin{pmatrix} \hat{\hat{x}}_T(t_{k+1}) \\ \hat{\hat{y}}_T(t_{k+1}) \end{pmatrix} = \begin{pmatrix} \hat{x}_T(t_k) \\ \hat{y}_T(t_k) \end{pmatrix} + (t_{k+1} - t_k) \begin{pmatrix} \hat{v}_T(t_k) \cos \hat{\theta}_T(t_k) \\ \hat{v}_T(t_k) \sin \hat{\theta}_T(t_k) \end{pmatrix}$$





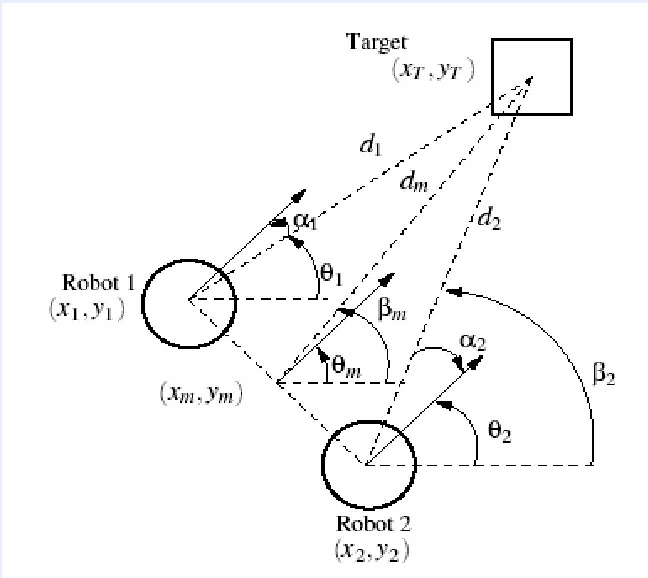
Trajectory Generation

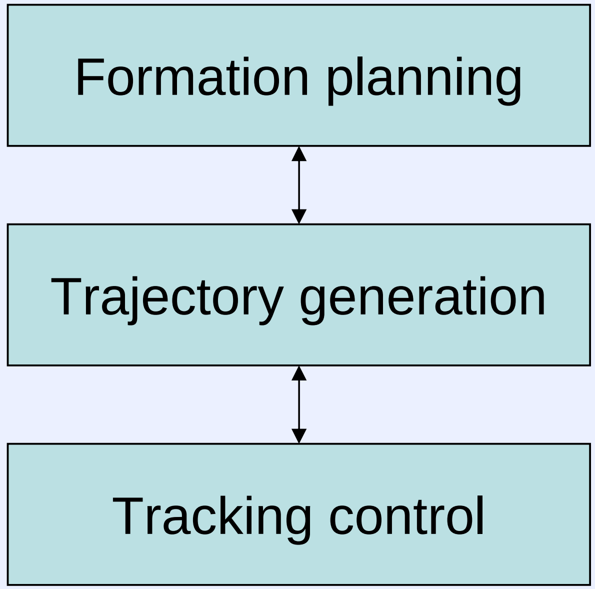


Generate reference trajectories for robots

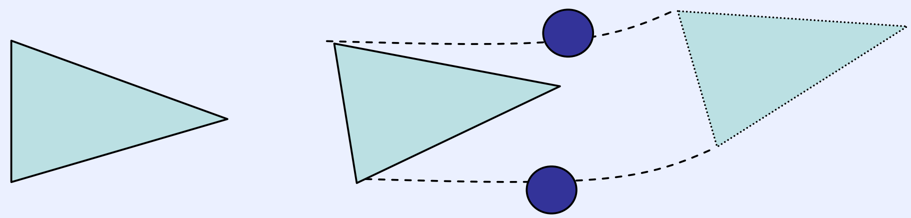
$$x_i^{ref}(t) = x_i^f(t_k) + \frac{v_i(t_k)}{\omega_i(t_k)}[\sin(\theta_i^f(t_k) + \omega_i(t_k)t) - \sin(\theta_i^f(t_k))]$$
$$y_i^{ref}(t) = y_i^f(t_k) - \frac{v_i(t_k)}{\omega_i(t_k)}[\cos(\theta_i^f(t_k) + \omega_i(t_k)t) - \cos(\theta_i^f(t_k))]$$

based on rigid formation: $\dot{p} = \dot{d}_1 = \dot{d}_2 = 0, \dot{\theta}_1 = \dot{\theta}_2$





Tracking Control



Track reference trajectory using virtual vehicle approach [Egerstedt et al.,01]:

Parameterize reference trajectories as $p_i(s_i) := x_i^{ref}(s_i), q_i(s_i) := y_i^{ref}(s_i)$

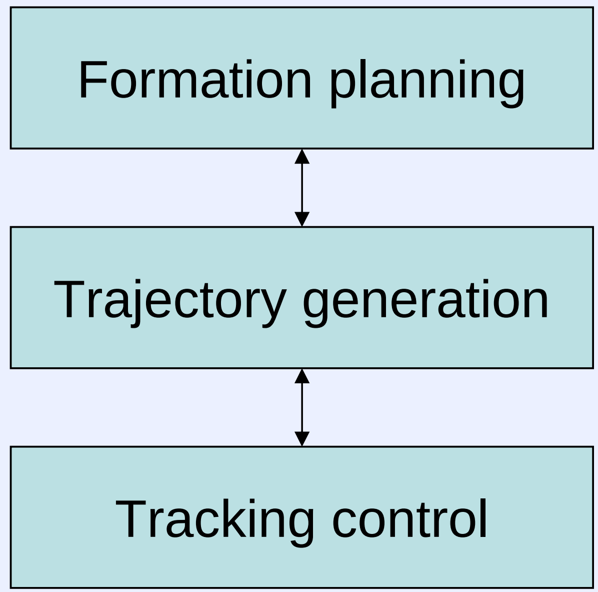
$$\dot{s}_i = \frac{ce^{-a\rho_i} v_i^0}{\sqrt{p_i'^2(s_i) + q_i'^2(s_i)}}$$

Tracking controller given by

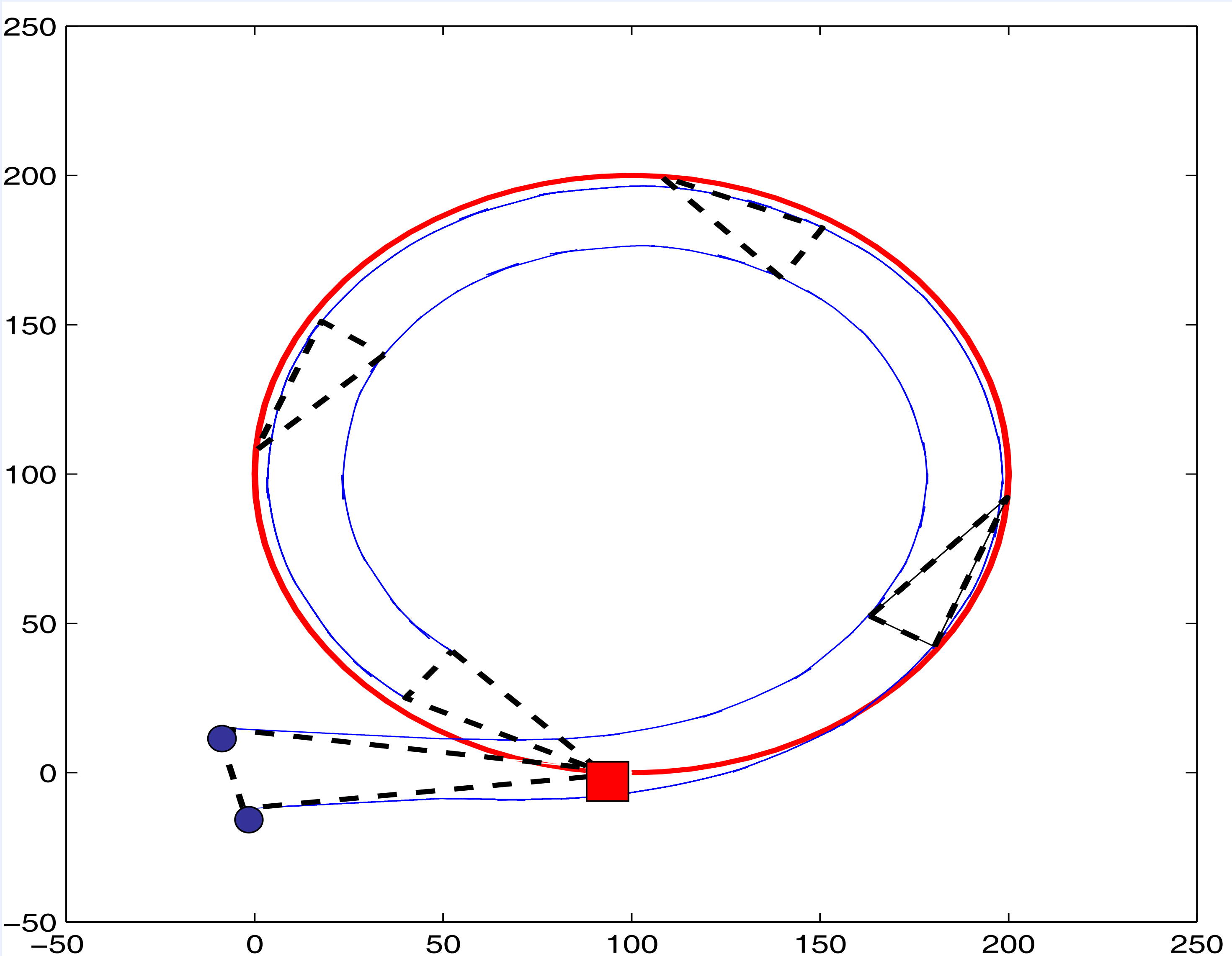
$$v_i(t) = \gamma \rho_i(t) \cos[\phi_i^d(t) - \theta_i(t)]$$
$$\omega_i(t) = k[\phi_i^d(t) - \theta_i(t)] + \dot{\phi}_i^d(t)$$

with

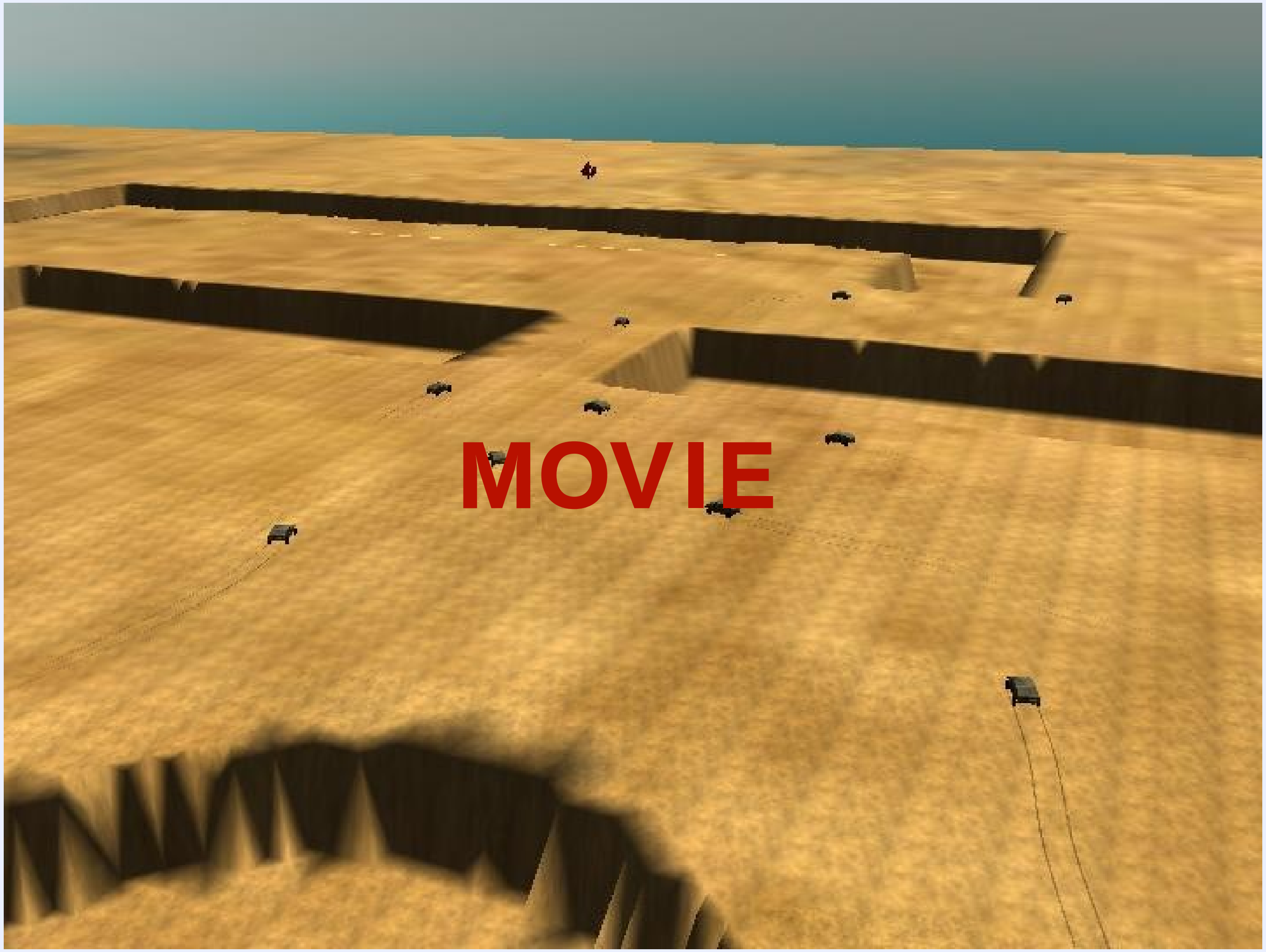
$$\rho_i(t) = \sqrt{(x_i^{ref}(s_i) - x_i(t))^2 + (y_i^{ref}(s_i) - y_i(t))^2}$$
$$\phi_i^d(t) = \arctan \frac{x_i^{ref}(s_i) - x_i(t)}{y_i^{ref}(s_i) - y_i(t)}$$



Simulation: Tracking Target on Circle

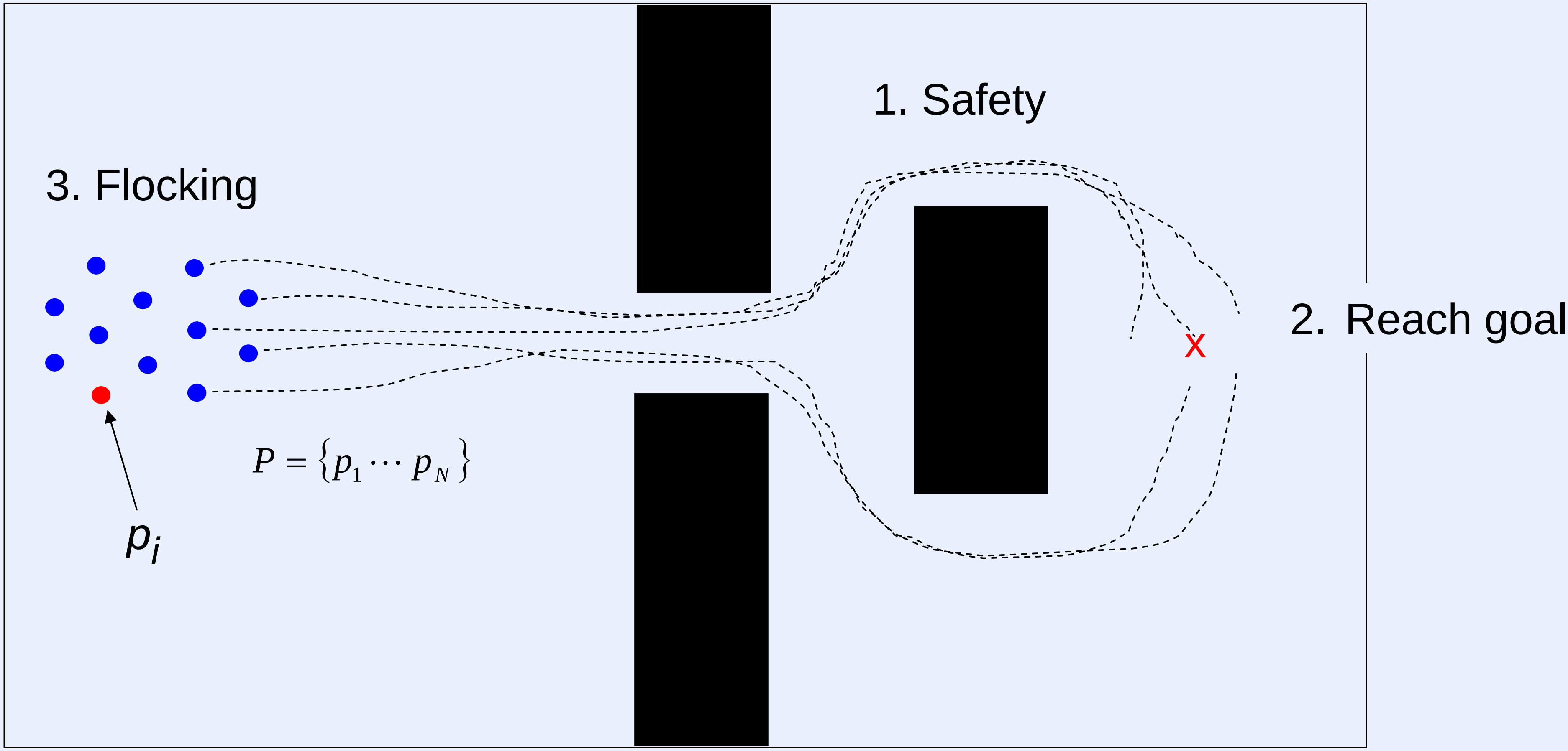


Multi-Robot Flocking with Obstacle Avoidance



Lindhé, Ögren, J, 04

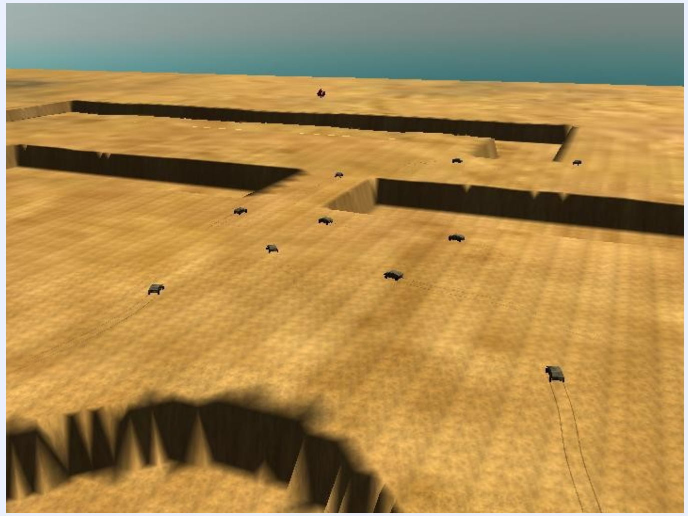
Multi-Robot Flocking with Obstacle Avoidance

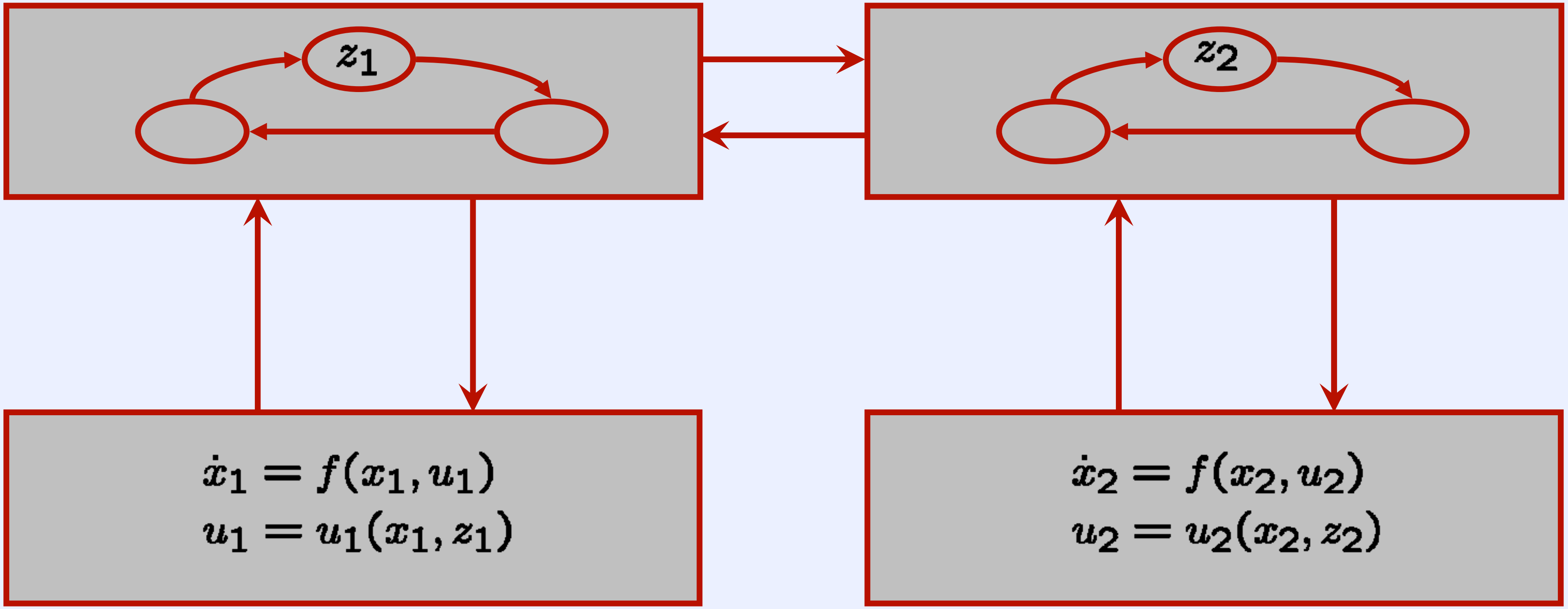


Event- and Time-Triggered Control

- Discrete-event waypoints generated by high-level algorithm
- Tracked by low-level continuous-time control

- **Safety** guaranteed by limited actuation
- **Goal reaching** from potential field (Lyapunov) argument
- **Flocking** due to distribution following Voronoi partition





Why Hybrid Systems?

- Abstractions in design lead to hybrid dynamics
  - **Time-scale separation**, **large scale systems**
- Embedded computer systems are hybrid
  - Real-time software interacting with physical environment
- Control strategies are hybrid
  - On-off, optimal control, batch control, **hierarchical control**
- Improved performance
  - Brockett integrator, supervisory control, variable structure systems
- Nature is hybrid
  - Relays, **impact mechanics**, state constraints