Decentralized Intrusion Detection in Cooperative Multi-Agent Systems

Adriano Fagiolini, Lucia Pallottino and Gianluca Dini

Interdep. Research Center "E. Piaggio" – University of Pisa

HSCC 07 Hybrid Systems Computation and Control

Introduction

We address the problem of **detecting faulty behaviors** of robots belonging to a multi-agent system. Our objective is to develop a **scalable architecture** that can be adopted to realize a **completely decentralized intrusion detector** monitoring the agents' behavior. We want the solution to be **independent from the set of "rules"** describing the interaction among the agents, **and from their dynamics**; ~~(non-invasive)~~ mainly **based on HW/SW components** that are already present on-board of each agent. We focus on systems with **decentralized cooperation schemes** where cooperation is obtained by **sharing a set of "rules"** by which each agent plans its **next "action"** and where some of the agents may act not according to the rules due to **spontaneous failure**, **tampering**, or **malicious introduction**.

Goals

- to realize intelligent transportation systems that are robust to malicious attacks;
- the "hybrid" nature of the agents' architecture, composed of a physical layer whose evolution is **time-driven**, and of a logical layer whose evolution is **event-driven**, leads quite naturally to the definition of a **hybrid observer**;
- (partially observable processes) due to the hypothesis of using only **local sensing** capability, each observer has **partial knowledge** of the monitored agent's neighborhood.

Modeling

Agent's Hybrid Architecture

The agent's hybrid architecture is composed of a **time-driven** "physical layer", and an **event-driven** "logical layer".

The lower level is the **physical layer** comprising:

- the **dynamics** $\dot{q}_i = f(q_i, u_i)$, where q_i is the robot "physical" state;
- the **controller** $u_i = g(\sigma_i)$ generating the input u_i that is necessary to execute the command σ_i

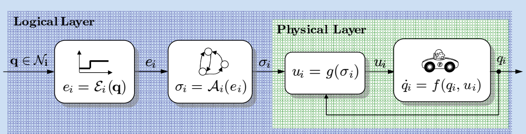
$$\sigma_i(t_k) = \mathcal{S}_i(\sigma_i(t_{k-1}), \mathbf{q}(t_k))$$

$\mathcal{N}_i = \{j \in \{0, 1, \ldots, N\} \,|\, \mathcal{R}(i,j) \text{ is active}\}$, time-varying set representing the i-th agent's neighborhood based on the decentralized cooperation rules $\mathcal{R}$

The higher level is the **logical layer** comprising:

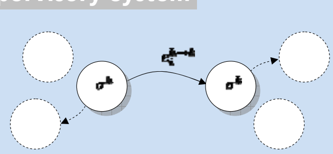
- the **event detector** $e_i = \mathcal{E}(\mathbf{q})$ checks the occurrence of enabled events e_i based on the state $\mathbf{q}$ of the agent's neighborhood $\mathcal{N}_i$;

the **finite state machine** (automaton) $\sigma_i = \mathcal{A}(e_i)$ planning/deciding the next "maneuver" σ_i on the basis of e_i



Hence, the **hybrid state** of the i-th agent is given by (q_i, σ_i)

Event-Driven Supervisory System

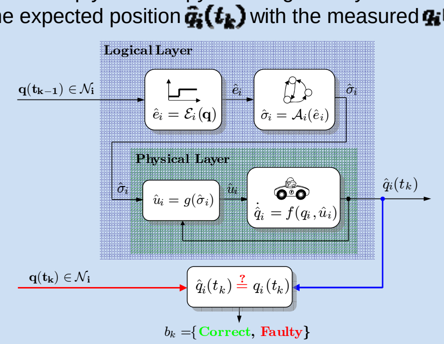


The generic input event $e_i^{h \to k}$ encodes the logical condition on the state of the agent's neighborhood requiring supervisor $\mathcal{S}_i$ to update its state σ_i from action σ^h to action σ^k

The agent virtually decomposes all events $e_i^{h \to k}$ as the disjunction of arbitrarily complex sub-events: $e_i^{h \to k} = \vee_l \, e_{i,l}^{h \to k}(\mathbf{q}_l)$

Exact and Complete Knowledge

In case the observing agent has **complete knowledge** of the monitored agent's neighborhood, it can simply use a copy of the agent's hybrid model, and make a comparison of the expected position $\hat{q}_i(t_k)$ with the measured $q_i(t_k)$

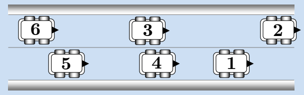


Case Study

The Automated Highway

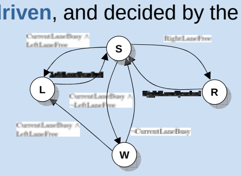
Group of vehicles traveling along a 2-lane automated highway. Each vehicle enters the highway in different initial positions, and moves with different maximum velocities to reach different final destinations.

To avoid collisions, vehicles are supposed to cooperate by executing a sequence of "**maneuvers**" in accordance with the **common driving rules**

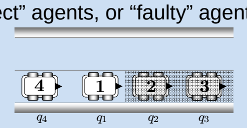


Let $q_i = (x_i, l_i)$ be the **state** of the i-th vehicle, where $x_i \in \mathbb{R}$ is the agent position along the current lane $l_i \in \mathbb{N}$. Let $\sigma_i \in \{S, L, R, W\}$ be the **maneuver** that the i-th vehicle **has planned to execute** according to the rules, and its neighborhood.

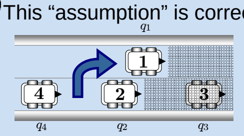
Each vehicle is a hybrid system: q_i has a **time-driven dynamics**, whereas the evolution of σ_i **is event-driven**, and decided by the finite state machine (**automaton**):



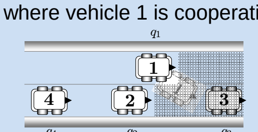
Vehicle 4 **(the observer)** is approaching to the others, and wants to establish whether they are "correct" agents, or "faulty" agents, by using only the **on-board sensing**.

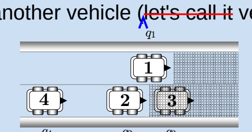


Since vehicle 4 sees only vehicle 1, it assumes that vehicle 1 is cooperating with zero other vehicles, $\mathcal{A}^0$. This "assumption" is correct up to this event:



Now the cooperation model $\mathcal{A}^0$ is not capable of **explaining the "left-turn" maneuver** of vehicle 1. Hence, vehicle 4 consider **a richer cooperation model** $\mathcal{A}^1$ where vehicle 1 is cooperating with another vehicle ~~(let's call it~~ vehicle 2).

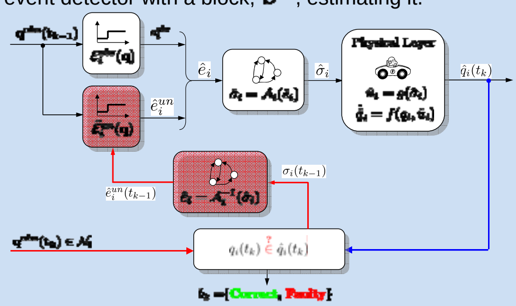




For the case on the left, model $\mathcal{A}^1$ is capable of explaining vehicle 1 behavior. For the case on the right, vehicle 4 have to consider a ~~more~~ richer cooperation model $\mathcal{A}^2$.

Exact But Partial Knowledge

In case of **partial knowledge**, some of the events become **indistinguishable**. This induces a decomposition of each event into observable and unobservable part, e.g. $e_i = e_{i1} \wedge e_{i2}$. We can obtain an observer by replacing the unknown part $\mathcal{E}^{un}$ of the event detector with a block, $\hat{\mathcal{E}}^{un}$, estimating it.



EC contract IST-2004-511368 (NoE HYCON)

EC contract IST-2004-004536 (IP RUNES)