Decentralized Intrusion Detection in Cooperative Multi-Agent Systems

Adriano Fagiolini, Lucia Pallottino and Gianluca Dini
Interdepartmental Research Center "E. Piaggio" – University of Pisa

HSCC07 Hybrid Systems Computation and Control

Introduction

We address the problem of **detecting faulty behaviors** of cooperative mobile agents. A novel **decentralized and scalable architecture** that can be adopted to realize a monitor of the agents' behavior is proposed. We consider agents which may perform different independent tasks, but cooperate to guarantee the entire system's safety. **Agents plan their actions** by following a **set of rules** R which is shared among them. Such rules are decentralized, i.e. they depend only on configurations of neighboring agents. Some agents may not be acting according to this **cooperation protocol**, due to spontaneous **failure** or **tampering**.

To detect such misbehaviors we propose a solution where each agent **runs a local observer** using **only locally available information**. The objective of the work is the definition of a basic framework to automatically realize decentralized intrusion detectors for (hybrid) multi-agent systems where interaction is modeled through logical cooperative protocols. Possible applications of this work may concern **intelligent transportation systems** which need to be robust to failures and malicious attacks.

Modeling The Hybrid Agent

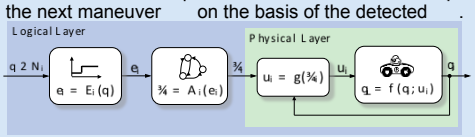
The agent's architecture is hybrid and composed of a **time-driven** physical layer, and an **event-driven** logical layer.

The lower level is the **physical layer** comprising:
- the **dynamics** q_i $\dot{q}_i = f(q_i, u_i)$ where q_i is the robot 's physical state; u_i $u_i = g(\sigma_i)$
- the **controller** σ_i generating the input that is necessary to execute the command .

In a **decentralized setting**, the agent is unaware of any global information about the system, and indeed the decision making process is based on only the locally available data.

We define the **active configuration space** as the set of configurations that actually affect the agent's behavior: , where R is a Boolean function, that ensues from the cooperation rule set . The **agent's neighborhood** is then the time-varying set:

The higher level is the **logical layer** comprising:
- the **event detector** q checks the occurrence of enabled events based on the state of the agent's neighborhood ;
- the **finite state machine** (automaton) planning / deciding the next maneuver on the basis of the detected .



Hence, the **hybrid state** of the i-th agent is given by .

Detecting Non-cooperative Behaviors Under Partial Knowledge

Each input event $e_i^{h!k}$ **encodes** a particular logical condition of the state of the agent's neighborhood requiring supervisor S_i to update its state σ_i from action σ^h to action σ^k. For our purpose, every event can be written in **disjunctive canonical form**:

$$e_i^{h!k} = \bigvee_l e_{i;l}^{h!k}(q_l)$$

where each sub-event is expressed as the conjunction of a **decidable** part $e_{i;l}^{o}$, depending only on known quantities q_i^o, with a **non-decidable** part $e_{i;l}^{u}$, depending also on unknown quantities q_i^u :

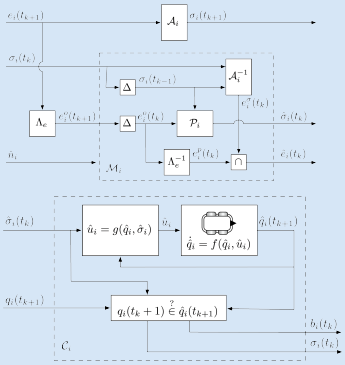
$$e_{i;l}(q_i) = e_{i;l}^{o}(q_i^o) \wedge e_{i;l}^{u}(q_i^o; q_i^u)$$

We define an **event observation map** $\eta_e : E_i \to E_i^o$ projecting any event of the original alphabet E_i into the observable alphabet E_i^o.

Misbehaviors of agent *i* can then be detected by using the following components:

- System M_i , a filter receiving as inputs the observable part of any event and the actual discrete state σ_i of the observed agent, and generating a prediction σ_i of possible future discrete states and an estimate of the complete e_i .

- System C_i - the classifier -, which first runs a multiple execution of the physical layer, in order to compute the set of all possible behaviors , and then checks whether the measured behavior of the agent was predicted or not.

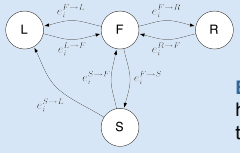


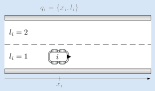
The Case Study - An Automated Highway

We consider groups 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** according to 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 $\sigma_i \in \{F; L; R; S\}$ be the **maneuver** that the i-th vehicle **has planned to execute** according to the rules and to its neighborhood.





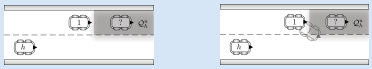
Each vehicle is a hybrid system: q_i has a **time-driven dynamics**, whereas the evolution of σ_i **is event-driven.**

An Example - Vehicle *h* **(a local observer)** is approaching the group of vehicles, and wishes to establish whether they are correctly behaving or not.

Since the observer is able to see only vehicle 1, it assumes that vehicle 1 is cooperating with zero other vehicles:

Whenever vehicle 1 starts a left-turn, the cooperation model considered by the observer becomes unable to **explain the agent's behavior**. Hence, it considers **a richer cooperation model** assuming vehicle 1 is cooperating with an unknown vehicle.

Now suppose vehicle 1 moves along the second lane under the following configuration:



Since no cooperation model exists such that the observation is consistent, **a faulty behavior is detected** by the observer.

Results

We presented a method for **decentralized intrusion detection for cooperative systems of mobile agents**. The method allows to detect a non-cooperative intruder based only on local information where the intruder's behavior is **inconsistent with observable data**.

Future work will address the **cooperation of local observers** that can exchange information in order to improve their detection capability.

EC contract IST-2004-511368 (NoE HYCON)
EC contract IST-2004-004536 (IP RUNES)