Automatic Generation of Distributed Misbehavior Detectors for the Security of Societies of Robots

Adriano Fagiolini, *Member, IEEE,* Gianluca Dini, and Antonio Bicchi, *Fellow, IEEE*

Abstract

This paper addresses the problem of detecting possible intruders in a robot society, i.e., a group of autonomous robots which coexist in a shared environment and interact with each other according to a set of "social behaviors", or common rules. Such rules specify what actions each robot is allowed to perform in the pursuit of its individual goals: rules are distributed, i.e. they can be evaluated based only on the state of the individual robot, and on local information that can be sensed directly or through communication with immediate neighbors. We consider intruders as robots which misbehave, i.e. do not follow the rules, because of either spontaneous failures or malicious reprogramming. Our goal is to detect intruders by observing the congruence of their behavior with the social rules as applied to the current state of the overall system. Moreover, in accordance with the fully distributed nature of the problem, the detection itself must be performed by individual robots, based only on local information. We present a formalism that allows to model uniformly a large variety of possible robot societies and propose an Intrusion Detection protocol that is based on two main components. The first is a *monitor* that an individual robot runs, using only locally available information, and applies to each of its neighbors, which constructs the set of all possible system states which might explain the observed behavior and are consistent with its own limited direct knowledge. The second component is a set–valued consensus algorithm allowing different "views" estimated by different local monitors to be combined through communication. We present simulative results of the application of our method to a system of vehicles moving within an urban–like environment, and experiments obtained in a real industrial scenario setup including autonomous cooperative forklifts.

Index Terms

Intrusion detection, security, multi-robot robotics, set–valued consensus algorithms, industrial autonomous systems.

I. Introduction

THE availability of distributed systems gave rise in the late 80s to a profound rethinking of many decision making problems [?], [?] and enabled solutions that were impossible before [?], [?]. A similar trend is now happening in Control and will soon enable a formidable number of new robotic applications. Various distributed control policies have been proposed for formation control, flocking, sensor coverage, and intelligent transportation (see e.g. [?], [?], [?], [?]). An intrinsic paradigm shift is indeed conveyed, from the idea of a distributed intelligent system as a collection of interacting software processes, to that of a network of physical robots that take information from the environment and act within the environment itself to change it. What will be meant by a *distributed intelligent robot* is the smart interconnection of heterogeneous units having different sensing, computation, and actuation abilities. It is foreseeable that, in a near future, number of robots produced by different makers will act within same environments, for which they will need to possess "social" interaction skills [?]. To this purpose, various theoretical issues are still unsolved, which include e.g. the definition of few standard formalisms to describe all possible *behaviors* of a cooperative robot, and whose translation into the corresponding control systems is rapid and

A. Fagiolini is with the Department of Electrical, Electronics, Telecommunication, Chemistry, Automation and Mathematical Models, Faculty of Engineering, Università degli Studi di Palermo, Italy, and with the Interdepartmental Research Center "E. Piaggio", Faculty of Engineering, Università di Pisa, Italy, fagiolini@unipa.it.

G. Dini is with the Department of Information Engineering, Faculty of Engineering, Università di Pisa, Italy, gianluca.dini@ing.unipi.it.

A. Bicchi is with the Interdepartmental Research Center "E. Piaggio", Faculty of Engineering, Università di Pisa, Italy, and with Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genova, Italy bicchi@centropiaggio.unipi.it.

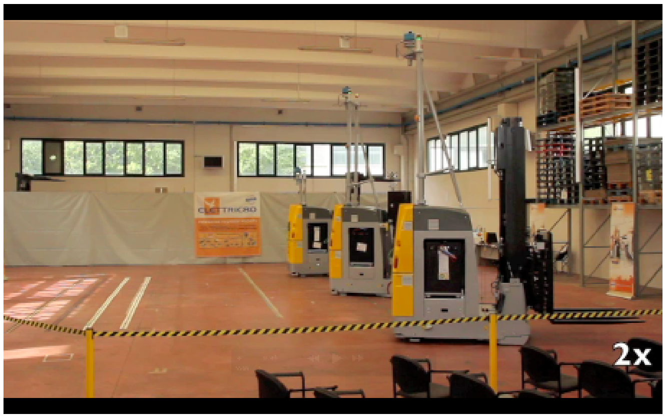


Figure 1. Autonomous forklifts simulating their operation in a factory's warehouse at Elettric80 S.p.A.'s premises.

systematic. Standards are common practice in Information Technology, their need in Robotics is perceived at every application level [**?**], although we believe that the larger variety and higher complexity of our interconnected cyber–physical systems may be the principal causes, explaining why they have not been established yet. Such formalisms should enable to specify e.g. how two robots with different mechanics can safely manipulate an object together.

In this vein we provide a formalism that consists in a hybrid model capturing the behavior of a general class of robots. According to our formalism, robots have the ability to interact with other neighboring robots based on event–based rules. For the sake of clarity, let us informally introduce all the components of our formalism through the example of an automated forklift moving in a factory's warehouse (Fig. 1). A forklift is a system described by a *configuration* vector and by a *dynamics*, including inertial and geometric parameters, that depend on its physical structure, its size and shape, etc. If the forklift is meant to be used within an environment with obstacles and other cooperative forklifts, the forklift itself is provided with a supervisory system, that can be implemented as an *automaton* and that allows it to perform a finite number of maneuvers. Each maneuver is represented by a different *discrete state* of the automaton and is continuously *decoded* into a suitable control value to be applied to the forklift's actuators. Moreover, a forklift has onboard sensors, such as cameras or infrareds, that determine its *visibility* capacities and provide information that is used to plan its trajectory and prevent collisions with obstacles or other forklifts. To this purpose, its sensor outputs are constantly *encoded* into a finite number of *events*, indicating e.g the presence or absence of another *neighboring* forklift, that may or may not require the forklift to change its current maneuver. As a whole, a cooperative forklift is a complex system, that can be described by the hybrid formalism that we propose in Section II, where all such components are formally defined.

Furthermore, such open dynamic scenarios may be unattended or possibly hostile. For this reason robots may become appealing targets for attackers aiming at degrading the system's QoS or even causing serious damage. In the example above, one attacker may tamper the supervisory system of a forklift, which may stop in a corridor or even crash into another forklift. The whole system is at risk also when only some robots' behaviors deviate from specification [**?**]. Our goal is to detect misbehavior in the physical motion of robots, by means of a distributed protocol that robots can run to determine and reach a consensus on whether their neighbors are cooperative or not. Robots are *cyber–physical systems* that embody complex intelligent links between perception and action [**?**], and thus their behavior is inherently determined by real–time physical dynamics, that give inputs to and receive outputs from their event–based supervisory systems. Not only this requires the use of a hybrid dynamic formalism, such as the one described above, but it also makes the problem we deal with much more complex than the detection of communication misbehavior in a Mobile Ad–hoc Network (MANET) or a Peer–to–Peer (P2P) system. In this last context, the discovery of e.g. selfish agents/nodes, that not to forward incoming messages to their recipients, or malicious agents, that send corrupted information to their neighbors, can be obtained by e.g. the use of the CONFIDANT protocol [?] or RCAR [?].

Misbehavior detection in these systems poses a number of theoretical challenges. For example, referring to the warehouse scenario, every forklift should know what and where are the neighbors of another target forklift, to use them as inputs of the cooperative model and verify if the observed actual behavior is congruent. Since some of the neighboring forklifts may be out of the observing robot's visibility region, an inversion of the hybrid nonlinear cooperative model of the target forklift would be necessary, which is possible only for specific systems [?], [?], [?]. Moreover, the neighbors of a target forklift dynamically change as they move in the factory, and thus their interaction topology, i.e. who is is interacting with whom, may be unknown to the observing forklift. On the contrary existing approaches to fault detection require this information to be fully known a priori [?], [?], [?], [?]. Finally, forklifts need to consent on information concerning the status of the target forklift's neighborhood, which is not simply represented via real numbers or vectors. Hence available approaches based on linear consensus strategies [?], [?] cannot be used.

We propose an Intrusion Detection System (IDS) that consists of a distributed protocol which is based on two components: a local monitor and a set–valued consensus algorithm. The local *monitor* reconstructs the information of free and occupied regions in the neighborhood of a target robot, by estimating the events that it should have observed and applying an inversion of the cooperative model at the automaton level. The set–valued consensus algorithm allows the robots to reach consensus on the estimated free/occupied regions in the neighborhood of the target. The protocol assumes a virtuous scenario hypothesis, in which the information exchanged among robots is correct, which can be guaranteed by the use of trusted software platforms [?]. No collusion exists between a robot executing an incorrect motion and another one trying to justify it. The problem of reaching consensus on information corrupted by intruders is a classical one [?] and is not investigated here. Obviously, detecting simultaneous motion and information misbehaviors is much more complex and is left for further studies. A preliminary study on the topic was presented in [?]. The problem we deal is new in the sense that previous works only cope with misbehavior detection at information level, while we consider systems where misbehavior can also be at action level. The paper is based on previous work by the authors, presented in [?], [?], and is extended here with a full formalization of the hybrid cooperative model, a formal proof of the IDS correctness, and the application to a real experimental system. Although any considered motion protocol is reasonably correct, i.e. it ensures absence of deadlocks and accidents if all robots correctly move, the proposed technique does not rely on such an assumption.

The paper is organized as follows. Section II formalizes the model of a cooperative robot, Section III describes the architecture of the local monitor, and Section IV defines the set–valued consensus protocol. Section V shows the application of the proposed method to a first example of cooperative system, consisting of a group of vehicles moving within an urban–like environment. Section VI shows an application of the technique to a real factory's warehouse and presents experimental results. Finally, the appendix contains the proof of correctness of the local observer's estimation procedure.

II. A Model of Cooperation Protocols for Physical Agents

Consider n agents, $\mathcal{A}_1, \ldots, \mathcal{A}_n$, sharing a state-space, or *environment* $\mathcal{Q}$. By cooperation protocol $\mathcal{P}$ we mean a formal description of the agents' constitutive elements, i.e. their perceptions and actions, and of the rules used to interconnect these elements. More precisely, we provide the following:

Definition 1: A *cooperation protocol* $\mathcal{P}$ consists in the specification, for each agent $\mathcal{A}_i$, of an octuple

$$\mathcal{P}_i := \{f_i, V_i, T_i, \Lambda_i, E_i, e_i, \Sigma_i, \delta_i, u_i\}\,,$$

where:

- $f_i : \mathcal{Q} \times \mathcal{U}_i \to T_{\mathcal{Q}}$ is the *dynamics* map of $\mathcal{A}_i$.
  Here, $\mathcal{U}_i$ denotes the set of admissible input values and $T_{\mathcal{Q}}$ the space tangent to $\mathcal{Q}$. The agent's state $q_i \in \mathcal{Q}$ evolves from its initial state q_i^0 according to the ODE
  $$\begin{cases} \dot{q}_i(t) = f_i(q_i(t), u_i(t)) \\ q_i(0) = q_i^0 \end{cases} , \; t \geq 0\,.$$
- $V_i : \mathcal{Q}^n \to 2^{\mathcal{Q}}$ is the *visibility map* of $\mathcal{A}_i$, describing the region observed by its sensors;
- $T_i = \{\eta_{i,1}, \cdots, \eta_{i,\kappa_i}\}$ is a set of *topologies* on $\mathcal{Q}$, with $\eta_{i,j} : \mathcal{Q} \to 2^{\mathcal{Q}}$.
  Topologies are basic to define a few further concepts: an agent's *neighborhood* is $N(q_i) = \cup_{j=1}^{\kappa_i} \eta_{i,j}(q_i)$; a *neighbor set* is $N_i = \{\mathcal{A}_k \,|\, q_k \in N(q_i)\}$; a *neighbor configuration set* is $I_i = \{q_k \in \mathcal{Q} \,|\, \mathcal{A}_k \in N_i\}$. Furthermore, we define an agent's *encoder* map as $s_i : \mathcal{Q} \times \mathcal{Q}^{n_i} \to \mathbb{B}^{\kappa_i}$, where $n_i = \mathrm{card}(I_i)$ and $\mathbb{B} \overset{\text{def}}{=} \{0, 1\}$, whose j–th component, $s_{i,j}$, is a logical–valued function returning 1 if any agent is in the j–th topology $\eta_{i,j}(q_i)$, i.e.
  $$\begin{array}{rcl} s_{i,j} & : & \mathcal{Q} \times \mathcal{Q}^{n_i} \to \mathbb{B} \\ & & (q_i, I_i) \mapsto \sum_{q_k \in I_i} \mathbf{1}_{\eta_{i,j}(q_i)}(q_k) \end{array} ,$$
  where $\sum$ represents the logical sum (*or*), and $\mathbf{1}_A(x)$ is the Indicator function of a set A;
- $\Lambda_i = \{\lambda_{i,1}, \cdots, \lambda_{i,h_i}\}$ is a set of *constants* with $\lambda_{i,j} : \mathcal{Q} \to 2^{\mathcal{Q}}$.
  Constants are basic to define an agent's *constant* map as $r_i : \mathcal{Q} \to \mathbb{B}^{h_i}$, whose j–th component, $r_{i,j}$, is a logical–valued function returning 1 if the agent's state belongs to the j–th constant $\lambda_{i,j}(q_i)$, i.e.
  $$\begin{array}{rcl} r_{i,j} & : & \mathcal{Q} \to \mathbb{B} \\ & & q_i \mapsto \mathbf{1}_{\lambda_{i,j}(q_i)} \end{array} ;$$
- $E_i = \{e^{i,1}, \cdots, e^{i,\nu_i}\}$ is a finite *alphabet of events*;
- e_i is an *event detector* map
  $$\begin{array}{rcl} e_i & : & \mathbb{B}^{\kappa_i} \times \mathbb{B}^{h_i} \to 2^{E_i} \\ & & (s_i, r_i) \mapsto \{e^{i,j} \in E_i \,|\, c_{i,j}(s_i) = 1\}\,, \end{array}$$
  where each detector condition $c_{i,j}$ is a logical function of the form
  $$\begin{array}{rcl} c_{i,j} & : & \mathbb{B}^{\kappa_i} \times \mathbb{B}^{h_i} \to \mathbb{B} \\ & & (s_i, r_i) \mapsto \Pi_{k \in \gamma_{i,j}} s_{i,k} \, \Pi_{k \in \rho_{i,j}} \neg s_{i,k} \, \cdot \\ & & \quad \cdot \, \Pi_{k \in \mu_{i,j}} t_{i,k} \, \Pi_{k \in \nu_{i,j}} \neg t_{i,k}\,, \end{array} \qquad (1)$$
  with $\lambda_{i,1}, \cdots, \lambda_{i,h_i}$ constants in $2^{\mathcal{Q}}$, $\gamma_{i,j} \cup \rho_{i,j} = \{1, \cdots, \kappa_i\}$ and $\gamma_{i,j} \cap \rho_{i,j} = \emptyset$, $\mu_{i,j} \cup \nu_{i,j} = \{1, \cdots, h_i\}$ and $\mu_{i,j} \cap \nu_{i,j} = \emptyset$, and Π and $\neg$ the logical product (*and*) and negation (*not*), respectively;
- $\Sigma_i = \{\sigma^{i,1}, \ldots, \sigma^{i,p}\}$ is a finite *discrete state set*;
- $\delta_i : \Sigma_i \times 2^{E_i} \to \Sigma_i$ is a deterministic *automaton* describing how the agent's discrete state is updated, i.e.
  $$\begin{cases} \sigma_i(t_{k+1}) = \delta_i(\sigma_i(t_k), e_i(t_{k+1}))\,, & t_k > 0 \\ \sigma_i(0) = \sigma_i^0 & \end{cases} ,$$

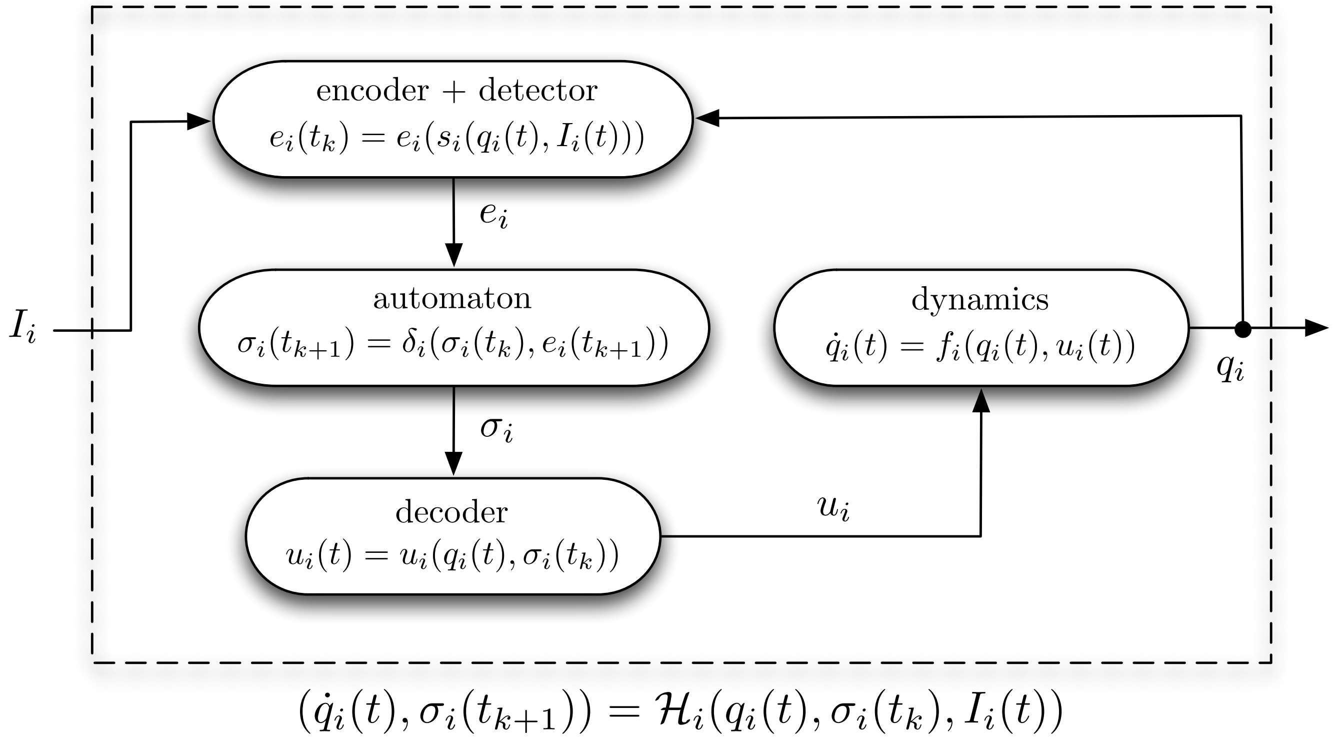

encoder + detector
$e_i(t_k) = e_i(s_i(q_i(t), I_i(t)))$
e_i
automaton
$\sigma_i(t_{k+1}) = \delta_i(\sigma_i(t_k), e_i(t_{k+1}))$
σ_i
decoder
$u_i(t) = u_i(q_i(t), \sigma_i(t_k))$
u_i
dynamics
$\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
q_i
I_i
$(\dot{q}_i(t), \sigma_i(t_{k+1})) = \mathcal{H}_i(q_i(t), \sigma_i(t_k), I_i(t))$

Figure 2. Architecture of a protocol–based cooperative physical agent.

where $\sigma_i^0 \in \Sigma_i$ is the initial discrete state, and t_k is the k–th instant t at which e_i detects a new event;

- $u_i : \mathcal{Q} \times \Sigma_i \to \mathcal{U}_i$ is a control *decoder* map, describing which control value is applied at different states of the system, i.e.

$$u_i(t) = u_i(q_i(t), \sigma_i(t_k)) \, . \quad \blacklozenge$$

According to Def. 1, the state of the generic *cooperative* agent $\mathcal{A}_i$ is $(q_i, \sigma_i) \in \mathcal{Q} \times \Sigma_i$ and its temporal evolution, or *behavior*, is described by the dynamic system:

$$\begin{array}{rcl} \dot{q}_i(t) & = & f_i(q_i(t), u_i(q_i(t), \sigma_i(t_k))) = \\ & = & f_i^*(q_i(t), \sigma_i(t_k)) \, , \\ \sigma_i(t_{k+1}) & = & \delta_i(\sigma_i(t_k), e_i(s_i(q_i(t), I_i(t)))) = \\ & = & \delta_i^*(\sigma_i(t_k), q_i(t), I_i(t)) \, . \end{array}$$

By introducing the i–th *hybrid dynamic map*

$$\mathcal{H}_i : \mathcal{Q} \times \Sigma_i \times \mathcal{Q}^{n_i} \to T_{\mathcal{Q}} \times \Sigma_i$$

the above system can written more concisely as

$$\left\{ \begin{array}{l} (\dot{q}_i(t), \sigma_i(t_{k+1})) = \mathcal{H}_i(q_i(t), \sigma_i(t_k), I_i(t)) \, , \\ (q_i(0), \sigma_i(0)) = (q_i^0, \sigma_i^0) \, . \end{array} \right. \tag{2}$$

Having denoted the history $I_i(\tau)$, for $\tau = 0, \cdots, t$, of $\mathcal{A}_i$'s neighbor configuration set with $\tilde{I}_i(t)$, the agent's behavior is obtained as the solution $\phi_{\mathcal{H}_i}(q_i^0, \sigma_i^0, \tilde{I}_i(t))$ of the dynamic system above where $\tilde{I}_i(t)$ acts as its input. Refer to Fig. 2 for a graphical representation of the dynamic model of $\mathcal{A}_i$.

In the following, we make the assumption that every agent has complete knowledge of the occupancy status of its own neighborhood through use of its own sensors, i.e.

$$N(q_i) \subseteq V_i(q_1, \cdots, q_n)\,.$$

III. Observer–based Intrusion Detection Protocol - Local Monitor

Consider how an observing agent $\mathcal{A}_h$ can learn whether another agent $\mathcal{A}_i$ is cooperative or not, by measuring the trajectory $\bar{q}_i(t)$ that it has executed, over successive observation periods $T_k = [t_k, t_{k+1})$, for $k = 0, 1, \cdots$. It usually happens that $\mathcal{A}_i$'s neighbor configuration set I_i is partially unknown to $\mathcal{A}_h$, since its neighborhood is not entirely included in $\mathcal{A}_h$'s visibility region, i.e.

$$N(q_i) \nsubseteq V_h(q_1, \cdots, q_n) \,.$$

If an uncertain yet complete estimate of I_i were available, $\mathcal{A}_h$ could say that $\mathcal{A}_i$ is cooperative when the curve $\bar{q}_i(t)$ lays within a tube around the solution of Eq. 2 with radius given by an accuracy ε. Our solution is a generalization of such a naive approach and consists of a local dynamic process, called *monitor*, which comprises a *hybrid observer* of $\mathcal{A}_i$'s state (q_i, σ_i), and an *occupancy estimator* of its neighbor configuration set I_i. During every observation period T_k, the local monitor receives as input the curve $\bar{q}_i(t)$ and the locally measured neighbor configuration set,

$$I_i^h = I_i \cap V_h(q_1, \cdots, q_n) \,,$$

and it returns a set–valued estimate $(\hat{q}_i, \hat{\sigma}_i)$ of the agent's hybrid state and and a set–valued estimate $\hat{I}_i^h$ of its neighbor configuration set.

A. The Hybrid Observer

Before formally describing the observer, we need to introduce the following five objects, derived from the components of the octuple $\mathcal{P}_i$:

- A *topology check* map $v_i : \mathcal{Q} \times 2^{\mathcal{Q}} \to \mathbb{B}^{\kappa_i}$, returning a binary vector whose j–th entry is 1 iff the j–th topology of $\mathcal{A}_i$ is entirely visible from $\mathcal{A}_h$, i.e.,

$$\begin{array}{rcl} v_{i,j} & : & \mathcal{Q} \times 2^{\mathcal{Q}} \to \mathbb{B} \\ & & (q_i, V_h) \mapsto \left\{ \begin{array}{ll} 1 & \text{if } \eta_{i,j}(q_i) \subseteq V_h \\ 0 & \quad\text{otherwise} \end{array} \right. \end{array} ;$$

- A *restricted encoder* map $\tilde{s}_i : \mathcal{Q} \times \mathcal{Q}^{\hat{n}_i} \to \mathbb{B}^{\kappa_i}$, with $\hat{n}_i = \text{card}(I_i^h)$, s.t. its j–th entry is a lower approximation of $s_{i,j}$ based on the locally available information of I_i:

$$\begin{array}{rcl} \tilde{s}_{i,j} & : & \mathcal{Q} \times \mathcal{Q}^{\hat{n}_i} \to \mathbb{B} \\ & & (q_i, I_i^h) \to \sum_{q_k \in I_i^h} \mathbf{1}_{\eta_{i,j}(q_i)}(q_k) \end{array}$$

  (note that $\tilde{s}_{i,j}(q_i, I_i^h) \le s_{i,j}(q_i, I_i)$ since $I_i^h \subseteq I_i$);
- An *event estimator* map $\tilde{e}_i : \mathbb{B}^{\kappa_i} \times \mathbb{B}^{\kappa_i} \to 2^{E_i}$ s.t. $\tilde{e}_i(\hat{s}_i, v_i)$ is an upper approximation of $e_i(s_i)$;
- A nondeterministic *automaton*

$$\begin{array}{rcl} \tilde{\delta}_i & : & 2^{\Sigma_i} \times 2^{E_i} \to 2^{\Sigma_i} \\ & & (\hat{\sigma}_i, \hat{e}_i) \to \{\bar{\sigma} \in \Sigma_i \,|\, \exists\, \sigma \in \Sigma_i, \sigma \subseteq \hat{\sigma}_i \,| \\ & & \qquad\qquad |\, \delta_i(\sigma, \hat{e}_i) = \bar{\sigma}\} \,, \end{array}$$

  describing how the estimated discrete state $\hat{\sigma}_i$ is updated starting from an initial estimate $\hat{\sigma}_i^0$, i.e.

$$\left\{ \begin{array}{l} \hat{\sigma}_i(t_{k+1}) = \tilde{\delta}_i(\hat{\sigma}_i(t_k), \hat{e}_i(t_{k+1})) \\ \hat{\sigma}_i(0) = \hat{\sigma}_i^0 \end{array} \right. ;$$

- A *controlled dynamics* map

$$\begin{array}{rcl} \tilde{f}_i^* & : & 2^{\mathcal{Q}} \times 2^{E_i} \to 2^{T_{\mathcal{Q}}} \\ & & (\hat{q}_i, \hat{\sigma}_i) \mapsto \{\dot{q} \in T_{\mathcal{Q}} \,|\, \exists\, \bar{q} \in \mathcal{Q}, \bar{q} \subseteq \hat{q}_i \,, \\ & & \qquad \exists\, \bar{\sigma} \in \Sigma_i \,,\, \bar{\sigma} \subseteq \hat{\sigma}_i \,|\, f_i^*(\bar{q}, \bar{\sigma}) = \dot{q}\} \,, \end{array}$$

describing how the estimated state $\hat{q}_i$ is updated starting from an initial estimate $\hat{q}_i^0$, i.e.

$$\left\{ \begin{array}{l} \dot{\hat{q}}_i(t) = \tilde{f}_i^*(\hat{q}_i(t), \hat{\sigma}_i(t_k)) \\ \hat{q}_i(0) = \hat{q}_i^0 \end{array} \right. .$$

Given the history $\tilde{I}_i^h(t)$ of the neighbor configuration set of $\mathcal{A}_i$ measured by $\mathcal{A}_h$, all $\mathcal{A}_i$'s *admissible cooperative behaviors*, i.e. the set of all temporal evolutions $(\hat{q}_i(t), \hat{\sigma}_i(t_k))$ that simultaneously complies with $\mathcal{P}_i$ and $\tilde{I}_i^h(t)$, are described by the set–valued dynamic system:

$$\begin{array}{rcl} \dot{\hat{q}}_i(t) & = & \tilde{f}_i^*(\hat{q}_i(t), \hat{\sigma}_i(t_k)) \, , \\ \hat{\sigma}_i(t_{k+1}) & = & \tilde{\delta}_i(\hat{\sigma}_i(t_k), \tilde{e}_i(\hat{s}_i(t_{k+1}), \hat{v}_i(t_{k+1}))) = \\ & = & \tilde{\delta}_i^*(\hat{\sigma}_i(t_k), \bar{q}_i(t), I_i^h(t)) \, , \end{array}$$

where

$$\begin{array}{l} \hat{v}_i(t_k) = v_i(\bar{q}_i(t_k), V_h(q_1(t), \cdots, q_n(t))) \, , \\ \hat{s}_i(t_k) = \tilde{s}_i(\bar{q}_i(t_k), I_i^h(t_k)) \, . \end{array}$$

In deriving the above model, the introduction of the topology check map v_i and the event estimator map $\tilde{e}_i$ is critical to ensure that no admissible behavior is possibly discarded. Clearly, the crucial step to obtain an effective observer is the event estimator map $\tilde{e}_i$, which should provide as tight an approximation as possible. To this purpose, the following result (whose proof is reported in Appendix A) is instrumental:

Theorem 1: The smallest event estimator compatible with an available topology check map v_i and a reduced encoder map $\hat{s}_i$ is given by

$$\begin{array}{rcl} \tilde{e}_i & : & \mathbb{B}^{\kappa_i} \to 2^{E_i} \\ & & \hat{s}_i \mapsto \{e^{i,j} \in E_i \,|\, \tilde{c}_{i,j}(\hat{s}_i, v_i) = 1\} \, , \end{array}$$

with

$$\begin{array}{rcl} \tilde{c}_{i,j} & : & \mathbb{B}^{\kappa_i} \times \mathbb{B}^{\kappa_i} \to \mathbb{B} \\ & & (\hat{s}_i, v_i) \to \Pi_{k \in \gamma_{i,j}} \left(\hat{s}_{i,k} v_{i,k} + \neg v_{i,k} \right) \cdot \\ & & \qquad \cdot \, \Pi_{k \in \rho_{i,j}} \neg \hat{s}_{i,k} \, \cdot \\ & & \qquad \cdot \, \Pi_{k \in \mu_{i,j}} \lambda_{i,k} \, \Pi_{k \in \tau_{i,j}} \neg \lambda_{i,k} \, ; \end{array} \tag{3}$$

We are now ready to describe the hybrid observer, whose execution consists of two distinct phases, *predict* and *update* (Fig. 3).

1) Predict Phase: During this phase, $\mathcal{A}_h$ computes a set–valued a–priori estimate $(\hat{q}_i(t|t_k), \hat{\sigma}_i(t_k|t_k))$ of $\mathcal{A}_i$'s behavior, based on the previously available estimate and the currently measured I_i^h.

At the very first execution ($k = 0$), the most conservative estimate is assumed: $\hat{q}_i(t_0|t_0) = \bar{q}_i(t_0)$ and $\hat{\sigma}_i(t_{-1}|t_0) = \Sigma_i$. For all successive executions, the propagation law is

$$\begin{array}{rcl} \hat{q}_i(t|t_k) & = & \phi_{\tilde{f}_i^*}(\hat{q}_i(t_k|t_k), \hat{\sigma}_i(t_k|t_k), \tilde{I}_i^h(t)) \, , \\ \hat{\sigma}_i(t_k|t_k) & = & \tilde{\delta}_i(\hat{\sigma}_i(t_{k-1}|t_k), \tilde{e}_i(\hat{s}_i(t_k|t_k), \hat{v}_i(t_k))) \, , \end{array}$$

for $t \geq t_k$, where $\hat{s}_i(t_k|t_k) = \tilde{s}_i(\bar{q}_i(t_k), I_i^h(t_k))$ is an a–priori estimate of the topology activation, and $\tilde{I}_i^h(t)$ is the history of I_i^h from t_k to t.

Moreover, to avoid explicit model inversion during the following update phase (which is impractical for general systems), the following set describing the "relationship" between predicted configuration trajectories and discrete states is computed and maintained as

$$L(t_k) = \hat{q}_i(t|t_k) \bowtie_{\tilde{f}_i^*} \hat{\sigma}_i(t_k|t_k) \, ,$$

where the operator $\bowtie_{\tilde{f}_i^*}$ is defined as

$$\begin{array}{rcl} \bowtie_{\tilde{f}_i^*} & : & 2^Q \times 2^{\Sigma_i} \to 2^{\mathcal{Q} \times \Sigma_i} \\ & & (\hat{q}_i(t), \hat{\sigma}_i) \mapsto \{(q(t), \sigma) \,|\, q(t) \subseteq \phi_{\tilde{f}_i^*}(\hat{q}_i(t_k), \sigma)\} \, . \end{array}$$

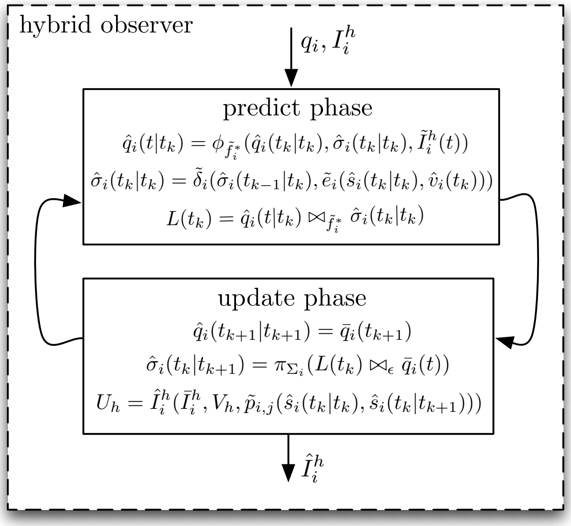

hybrid observer
q_i, I_i^h
predict phase
$\hat{q}_i(t|t_k) = \phi_{\tilde{f}_i^*}(\hat{q}_i(t_k|t_k), \hat{\sigma}_i(t_k|t_k), \tilde{I}_i^h(t))$
$\hat{\sigma}_i(t_k|t_k) = \tilde{\delta}_i(\hat{\sigma}_i(t_{k-1}|t_k), \tilde{e}_i(\hat{s}_i(t_k|t_k), \hat{v}_i(t_k)))$
$L(t_k) = \hat{q}_i(t|t_k) \bowtie_{\tilde{f}_i^*} \hat{\sigma}_i(t_k|t_k)$
update phase
$\hat{q}_i(t_{k+1}|t_{k+1}) = \bar{q}_i(t_{k+1})$
$\hat{\sigma}_i(t_k|t_{k+1}) = \pi_{\Sigma_i}(L(t_k) \bowtie_\epsilon \bar{q}_i(t))$
$U_h = \hat{I}_i^h(\bar{I}_i^h, V_h, \tilde{p}_{i,j}(\hat{s}_i(t_k|t_k), \hat{s}_i(t_k|t_{k+1})))$
$\hat{I}_i^h$

Figure 3. Operation phases of the hybrid observer in the local monitor.

It is worth noting that the set $L(t_k)$ does not raise any computation or representation issues, as it simply requires bidirectional pointers connecting every element of the finite set $\hat{\sigma}_i(t_k|t_k)$ with the corresponding curve $\hat{q}_i(t|t_k)$, computed by direct integration of the model.

2) Update Phase: At the end of the observation period, $\mathcal{A}_h$ computes a set–valued a–posteriori estimate of $\mathcal{A}_i$'s state, by combining the a–priori estimate with newly measured curve $\bar{q}_i(t)$ of $\mathcal{A}_i$'s state. The update step is given by

$$\begin{aligned} &\hat{q}_i(t_{k+1}|t_{k+1}) = \bar{q}_i(t_{k+1})\,, \\ &\hat{\sigma}_i(t_k|t_{k+1}) = \pi_{\Sigma_i}(L(t_k) \bowtie_\epsilon \bar{q}_i(t))\,, \end{aligned}$$

where π_A is the projector over the set A, and $\bowtie_\epsilon$ is an operator defined as

$$\begin{aligned} \bowtie_\epsilon \quad : \quad & 2^{Q \times \Sigma_i} \times \mathcal{Q} \to 2^{\mathcal{Q} \times \Sigma_i} \\ & (L, \bar{q}_i(t)) \mapsto \{(q(t), \sigma) \in L \,|\, ||q(t) - \bar{q}_i(t)|| \leq \epsilon\}\,. \end{aligned}$$

Intuitively, the operator drops out all behaviors where $\mathcal{A}_i$'s state q_i gets too far from the observation. Here, ϵ is a tolerance parameter, to be set depending on the accuracy of available sensors and nominal models.

Furthermore, an a–posteriori estimate of the encoder map outputs is obtained as follows:

$$\begin{aligned} \hat{s}_i(t_k|t_{k+1}) = \{&s^* \in \mathbb{B}^{\kappa_i} \,|\, s^* \geq \hat{s}_i(t_k|t_k)\,, \\ &e_i(s^*) \subseteq \hat{\sigma}_i(t_k|t_k) \bowtie_{\tilde{\delta}_i} \hat{\sigma}_i(t_k|t_{k+1})\,, \end{aligned}$$

where

$$\begin{aligned} \bowtie_{\tilde{\delta}_i} \quad : \quad & 2^{\Sigma_i} \times 2^{\Sigma_i} \to 2^{E_i} \\ & (\hat{\sigma}_i, \hat{\sigma}_i^+) \mapsto \{\hat{e}_i \in E_i \,|\, \exists\, \bar{\sigma}, \bar{\sigma}^+ \in \Sigma_i, \\ & \qquad \bar{\sigma} \subseteq \hat{\sigma}_i, \bar{\sigma}^+ \subseteq \hat{\sigma}_i^+,\, \bar{\sigma}^+ \subseteq \tilde{\delta}_i(\bar{\sigma}, e)\}\,. \end{aligned}$$

B. Occupancy Estimator

The output of the local occupancy estimator is a conservative estimate $\hat{I}_i^h(t_k)$ of the neighbor configuration set I_i during the current observation period. Intuitively, this is obtained by splitting each $s_{i,j}$ as the sum of a term $\tilde{s}_{i,j}$ (the value of the restricted encoder map) that depends only on quantities that are known to $\mathcal{A}_h$ and a term $p_{i,j}$ that is unknown and must be estimated:

$$s_{i,j}(q_i, I_i) = \tilde{s}_{i,j}(q_i, I_i^h) + p_{i,j}(q_i, I_i)\,,$$

where

$$p_{i,j}(q_i, I_i) = \sum_{q_j \in I_i \setminus V_h} \mathbf{1}_{\eta_{i,j}(q_i)}(q_j)\,.$$

The computation of this last term involves information related to the portion of the j–th topology $\eta_{i,j}$ that is out of $\mathcal{A}_h$'s visibility region. Its value can be inferred by a simple logics that compare the a–priori and a–posteriori estimates of the encoder map. If e.g. the a–priori value $\hat{s}_i(t_k|t_k)$ equals 0 and the a–posteriori one $\hat{s}_i(t_k|t_{k+1})$ equals 1, then an agent must lay in the portion of $\eta_{i,j}$ that is out of visibility. In that case, an estimate of the region where the hidden agent lays is obtained by subtracting $\mathcal{A}_h$'s visibility region from the topology $\eta_{i,j}$.

This intuition is formalized as follows. A conservative yet optimal (in virtue of Theorem 1) estimate of $\mathcal{A}_i$'s neighbor configuration set I_i is obtained through the map

$$\begin{array}{rcl} \hat{I}_i^h & : & 2^{\mathcal{Q}} \times 2^{\mathcal{Q}} \times 2^{\mathbb{B}^{\kappa_i}} \to 2^{2^{\mathcal{Q}}} \\ & & (\bar{I}_i^h, V_h, \hat{p}_i) \mapsto \\ & & \qquad \{(\hat{\eta}_{i,1}(\bar{I}_i^h, V_h, p_{i,1}), \cdots, \hat{\eta}_{i,\kappa_i}(\bar{I}_i^h, V_h, p_{i,\kappa_i}) \\ & & \qquad \;|\, p = (p_{i,1}, \cdots, p_{i,\kappa_i}) \in \mathbb{B}^{\kappa_i},\, p \subseteq \hat{p}_i\}\ , \end{array}$$

where

- the estimated occupancy of the j–th topology is

$$\begin{array}{rcl} \hat{\eta}_{i,j} & : & 2^{\mathcal{Q}} \times 2^{\mathcal{Q}} \times \mathbb{B} \to 2^{\mathcal{Q}} \\ & & (\bar{I}_i^h, V_h, 0) \mapsto \bar{I}_i^h \cap \eta_{i,j}(q_i)\,, \\ & & (\bar{I}_i^h, V_h, 1) \mapsto (\bar{I}_i^h \cap \eta_{i,j}(q_i)) \cup (\eta_{i,j}(q_i) \setminus V_h)\,; \end{array}$$

- the estimate of the unknown term is obtained as

$$\hat{p}_i = \tilde{p}_i(\hat{s}_i(t_k|t_k), \hat{s}_i(t_k|t_{k+1})\,,$$

  where $\tilde{p}_i : \mathbb{B}^{\kappa_i} \times 2^{\mathbb{B}^{\kappa_i}} \to 2^{\mathbb{B}^{\kappa_i}}$ is s.t. its j–th component is

$$\begin{array}{rcl} \tilde{p}_{i,j} & : & \mathbb{B} \times 2^{\mathbb{B}} \to 2^{\mathbb{B}} \\ & & (0,0) \mapsto 0\,,\ (0,1) \mapsto 1 \\ & & (0,\{0,1\}), (1,1) \mapsto \{0,1\}\,. \end{array}$$

  (Note that $\tilde{p}_{i,j}$ is undefined for the inputs $(1,0)$, $(1,\{0,1\})$ as the a–posteriori estimates of $s_{i,j}$ are obviously greater than the a–priori ones);
- $\bar{I}_i^h$ is an over–approximation of $\mathcal{A}_h$'s measures taking into account of its sensor inaccuracy, i.e.

$$\bar{I}_i^h = \{\hat{q}_k \in 2^{\mathcal{Q}} \,|\, \hat{q}_k = B_\epsilon(q_k) \text{ for some } q_k \in I_i^h\}\,,$$

  where $B_\epsilon(q_k)$ is a ball with radius ϵ and center at q_k;
- V_h is the current visibility region of $\mathcal{A}_h$.

IV. Set–valued Consensus Protocol for Monitor Agreement

Consider m_i observing agents, $\mathcal{A}_{i_1}, \cdots, \mathcal{A}_{i_{m_i}}$, trying to reach an agreed decision on the cooperativeness of a common neighbor $\mathcal{A}_i$. Here we assume that agents are connected through a communication topology, described by an *undirected* graph $G(V_G, E_G)$, where V_G is a node set representing the agents and E_G is an edge set representing agents that are within communication range. Recall that a graph is undirected when the fact that $\mathcal{A}_h$ can send a message to $\mathcal{A}_k$, implies also the reverse. We also assume that they exchange correct information, whether or not they move according to the motion protocol $\mathcal{P}_i$. We show here how the observing agents can construct a unique global estimate I_i^* of the occupancy map I_i only through one–hop message exchange.

As already stated in the introduction, off–the–shelf solutions for network agreement are inadequate to our context, as they typically work on data represented by real numbers or vectors and use very simple combination rules. As the outputs of local monitors are continuous sets, we need to attack the consensus problem from a more general perspective. In this vein, we consider that the consensus domain is $\mathcal{Q}$ and that each agent participating in the estimation process has a set–valued state $X_h \in 2^{\mathcal{Q}}$, for $h = 1, \cdots, m_i$. We also assume that a generic *merging function* $F : 2^{\mathcal{Q}} \times 2^{\mathcal{Q}} \to 2^{\mathcal{Q}}$ is intended to be used to combine any two states X_h, X_k of two different agents into a new state value $F(X_h, X_k)$. A composed merging function can be introduced as

$$\begin{array}{rcl} F^{(l)} & : & 2^{\mathcal{Q}^l} \to 2^{\mathcal{Q}} \\ & & (X_1, \cdots, X_l) \mapsto F(\cdots F(X_1, X_2) \cdots, X_l)\,. \end{array}$$

A hypothetical centralized process having full knowledge of all agents' initial estimates would be able to compute in one step the following estimate

$$X^* = F^{(m_i)}(X_1(0), \cdots, X_{m_i}(0))\,. \tag{4}$$

If F is both *commutative*, i.e., $F(X_1, X_2) = F(X_2, X_1)$ for all X_1, X_2, and *associative*, i.e., $F(X_1, F(X_2, X_3)) = F(F(X_1, X_2), X_3)$ for all X_1, X_2, X_3, the set–valued estimate X^* is well–defined, since it is independent of the order by which the estimates are processed. We also require that F be *idempotent* if $F(X_1, X_1) = X_1$ for all X_1. Moreover, let $CV_h(p)$ be the set of agents that can transmit a message to $\mathcal{A}_h$ by passing through at most $p - 1$ other agents, i.e. $CV_h(p) = \{j \in V_G \mid \mathrm{dist}(i, j) \leq p\}$, where $\mathrm{dist}(h, k)$ is the so–called geodesic distance of $\mathcal{A}_h$ from $\mathcal{A}_k$, i.e. the shortest path length between the two agents. Recall the notion of graph diameter being the maximum distance between any two nodes in the graph, i.e. $\mathrm{diam}(G) = \max_{i,j \in V_G} \mathrm{dist}(i, j)$.

We are now ready to prove the following result, which has a theoretical importance going beyond the scope of this paper, and involving the convergence of a class of set–valued consensus protocol systems:

Theorem 2 (Set–Valued Consensus Protocol): A collection of m_i agents running a consensus protocol described by the dynamic system

$$\left\{ \begin{array}{rcl} X_h(k+1) & = & F^{(p_h(1))}(X_{h,1}(k), \cdots, X_{h,p_h(1)}(k))\,, \\ X_h(0) & = & U_h\,, \end{array} \right.$$

where $p_h(k) = \mathrm{card}(CV_h(k))$, for all agents h, converges to the centralized consensus state in at most $\tilde{n} = \mathrm{diam}(G)$ steps, i.e.,

$$X(\tilde{n}) = \mathbf{1}_{m_i} X^*\,,$$

if F is commutative, associative, and idempotent, and if the communication graph G is connected.

Proof: Let us first prove, by induction, that the consensus state of an agent $\mathcal{A}_h$ after k consensus steps is

$$X_h(k) = F^{(p_h(k))}(X_{h,1}(0), \cdots, X_{h,p_h(k)}(0))\,.$$

The property is trivially satisfied after one consensus step:

$$X_h(1) = F^{(p_h(1))}(X_{h,1}(0), \cdots, X_{h,p_h(1)}(0))\,.$$

By assuming that the property holds after k steps, we want to prove its satisfaction after the $(k+1)$–th step. We have:

$$X_h(k+1) = F^{(p_h(1))}(J_1(k), \cdots, J_{p_h(1)}(k))\,, \tag{5}$$

where $J_i(k) = F^{(p_i(k))}(X_{i,1}(0), \cdots, X_{i,p_i(k)}(0))$ by the inductive hypothesis. Moreover, note that the order by which every estimate is processed is irrelevant, thanks to F's associativity and commutativity, and that multiple occurrence of the same estimate $X_{i,j}(0)$ can be simplified through F's idempotency. Eq. 5 involves the states of all agents $l \in CV_j(k)$ where $j \in CV_h(1)$, whose union gives by definition $CV_h(k+1)$, the set of agents that can send a message to $\mathcal{A}_h$ via a communication path of at most $k+1$ other agents, which proves the property. To finally prove the theorem, it is sufficient to note that, for all $k \geq \tilde{n}$, $CV_h(k) = V_G$

and hence $p_h(k) = m_i$, as the communication graph G is connected. Therefore, we have

$$\begin{array}{rcl} X_h(k) & = & F^{(p_h(\tilde{n}))}(X_{h,1}(0), \cdots, X_{h,p_h(\tilde{n})}(0)) = \\ & = & F^{(m_i)}(U_1 \cdots, U_{m_i}) = X^* , \end{array}$$

for all h and all $k \geq \tilde{n}$, which concludes the proof. ■

We can now move to the main implication of Theorem 2 as for what it concerns our intrusion detection problem. A hypothetical *centralized observer* receiving all m_i estimates would be able to compute in a single step the following merged estimate

$$I_i^c = \hat{I}_i^{(i_1)} \cap \hat{I}_i^{(i_2)} \cap \cdots \cap \hat{I}_i^{(i_{m_i})} ,$$

where $\cap$ is the set–theoretic intersection. Moreover, we can also state the following:

Corollary 1 (Monitor Agreement Protocol): A set–valued consensus protocol where

- the communication graph G is connected,
- the generic consensus state X_h is initialized with the locally estimated occupancy map, i.e.

$$U_h = \hat{I}_i^h(t_k | t_{k+1}) \,, \text{ and}$$

- $F^{(p_h(1))}$ is defined through the merging function

$$\begin{array}{rcl} \cap^* & : & 2^{\mathcal{Q}} \times 2^{\mathcal{Q}} \to 2^{\mathcal{Q}} \\ & & (X_1, X_2) \mapsto \{x \,|\, \exists\, x_1 \in X_1 \setminus \emptyset, x_2 \in X_2 \setminus \emptyset \,| \\ & & \qquad\qquad x = x_1 \cap x_2\} , \end{array}$$

converges in finite time to a consensus state $X^* = \mathbf{1_{m_i}} I_i^*$ with $I_i^* = I_i^c$.

Moreover, the very same decision on $\mathcal{A}_i$'s cooperativeness computed by the centralized observer is reached in finite time by all agents.

Proof: The operator $\cap^*$ satisfies the hypotheses of Theorem 2. ■

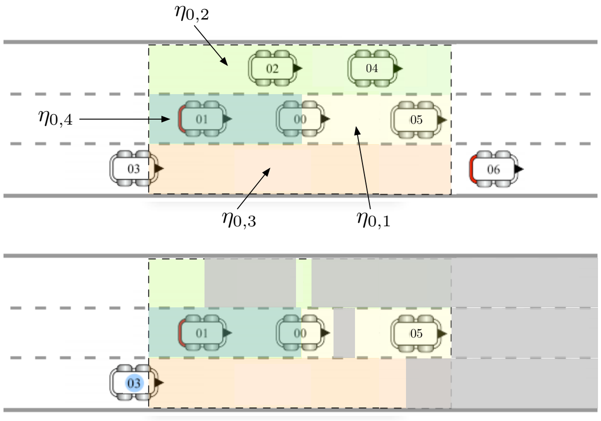


Figure 4. Neighborhood topologies of an agent (above) and partial visibility of a local monitor (below) in the highway example. For clarity's sake, only the first four topologies are shown in this picture.

V. Application to Automated Transportation Systems in Urban Environment

In this section we describe the application of our technique to a first cooperative system, consisting of a group of n cars moving along an m lane highway (see Fig. 4 for a depiction with $m = 3$). The system can be formalized as an instance of the cooperation protocol $\mathcal{P}$ presented in Section II. Cars have their own dynamics f_i and local controllers u_i, but their pilots are supposed to follow the European, right–hand traffic rules to avoid collisions. Every car $\mathcal{A}_i$ is assigned a $v_{des,i}$, which is a constant state variable $(\dot{v}_{des,i} = 0)$ indicating the speed at which each car desires to travel and is supposed to be measured as any other state variable. Each car must decide on a suitable motion maneuver, i.e., accelerate (FAST) or decelerate (SLOW), change to the next left lane (LEFT) or to the right one (RIGHT), based on the presence or absence of other cars in its neighborhood. E.g., the presence of a slower car in the front, and a free lane on the left requires the execution of an overtake that is a change from a FAST to a LEFT maneuver. In addition to these four maneuvers, in this system we examine the possibility of a fifth one called PLATOON, which defines a mode characterized by an oscillating longitudinal motion between the preceeding and/or following car. The rules require the introduction of a topology $\eta_{i,1}(q_i)$ representing a region in the immediate front of an agent $\mathcal{A}_i$ and longitudinal size d_f, a topology $\eta_{i,2}(q_i)$ for a region on its left and longitudinal size $d_f + d_b$, a topology $\eta_{i,3}(q_i)$ for a region on its right and longitudinal size $d_f + d_b$, and a topology $\eta_{i,4}(q_i)$ for a region on its back and longitudinal size d_b (see again, Fig. 4), a topology $\eta_{i,5}(q_i)$ for a region in front of it and longitudinal size d_{int}, $\eta_{i,6}(q_i)$ for a region on its back and longitudinal size d_{int}, where d_f and d_b are a forward and backward safety distances, d_{int} is the interaction distance, smaller than the visible distance but larger than d_f and d_b. We also define three more topologies $\eta_{i,7}(q_i)$, $\eta_{i,8}(q_i)$, $\eta_{i,9}(q_i)$ which are not simple areas in the highway plane, but also depend on another state variable (v_{des}). These topologies respectively describe the state-space area in front of the agent, on its back and on its right with compatible v_{des} (for the first two) and smaller v_{des} (for the last one). The last one also intuitively refers to the state-space region of a car that can be overataken by $\mathcal{A}_i$. What we mean by 'compatible' and 'smaller' will be better expressed with the formal definition of the topologies.

The system can be described as an instance of $\mathcal{P}$ with the environment, the configuration q_i and the dynamic map $f_i : \mathcal{Q} \times \Sigma_i \to T_\mathcal{Q}$ described in Section II. The above introduced topologies can be formalized as follows:

$$
\begin{array}{rcl}
\eta_{i,1} := \eta_{i,forwardBlock} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i \leq x \leq x_i + d_f, \\
& & \qquad \lfloor \frac{y_i}{w} \rfloor w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w\} \ , \\
\eta_{i,2} := \eta_{i,leftBlock} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i - d_b \leq x \leq x_i + d_f, \\
& & \qquad \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 2\right) w\} \ , \\
\eta_{i,3} := \eta_{i,rightBlock} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i - d_b \leq x \leq x_i + d_f, \\
& & \qquad \left(\lfloor \frac{y_i}{w} \rfloor - 1\right) \leq y \leq \lfloor \frac{y_i}{w} \rfloor w\} \ , \\
\eta_{i,4} := \eta_{i,backBlock} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i - d_b \leq x \leq x_i, \\
& & \qquad \lfloor \frac{y_i}{w} \rfloor w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w\} \ , \\
\eta_{i,5} := \eta_{i,forwardPresent} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i \leq x \leq x_i + d_{int}, \\
& & \qquad \lfloor \frac{y_i}{w} \rfloor w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w\} \ , \\
\eta_{i,6} := \eta_{i,backPresent} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i - d_{int} \leq x \leq x_i, \\
& & \qquad \lfloor \frac{y_i}{w} \rfloor w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w\} \ , \\
\eta_{i,7} := \eta_{i,forwardCompatible} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i \leq x \leq x_i + d_{int}, \\
& & \qquad \lfloor \frac{y_i}{w} \rfloor w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w \\
& & \qquad (1 - f)v_{i,des} \leq v_{des} \leq (1 + f)v_{i,des}\} \, , \\
\eta_{i,8} := \eta_{i,backCompatible} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i - d_{int} \leq x \leq x_i, \\
& & \qquad \lfloor \frac{y_i}{w} \rfloor w \leq y \leq \left(\lfloor \frac{y_i}{w} \rfloor + 1\right) w \\
& & \qquad (1 - f)v_{i,des} \leq v_{des} \leq (1 + f)v_{i,des}\} \, , \\
\eta_{i,9} := \eta_{i,rightOvertakeable} & : & \mathcal{Q} \to 2^{\mathcal{Q}} \\
& & q_i \mapsto \{(x, y, \theta, v, v_{des}) \,|\, x_i - d_{int} \leq x \leq x_i, \\
& & \qquad \left(\lfloor \frac{y_i}{w} \rfloor - 1\right) \leq y \leq \lfloor \frac{y_i}{w} \rfloor w \\
& & \qquad v_{des} \leq (1 - f)v_{i,des}\}
\end{array}
$$

where w is the lane width, f is a constant indicating the relative tolerance in v_{des} and $\lfloor \cdot \rfloor$ returns the nearest lower integer of the argument. Thus, the encoder map is $s_i : \mathcal{Q} \times \mathcal{Q}^{n_i} \to \mathbb{B}^9$, $s_i = (s_{i,1}, \cdots, s_{i,9})$, and the agent's neighborhood is $N(q_i) = \eta_{i,1}(q_i) \cup \cdots \cup \eta_{i,9}(q_i)$. Moreover, we need to introduce two constants $\lambda_{i,1}, \lambda_{i,2}$ representing the left–most and right–most lanes, respectively, and two constants $\lambda_{i,3}, \lambda_{i,4}$ representing the current target lane's left and right edges, respectively. Finally, we define a $\lambda_{i,5}$ indicating if the agent is lined up with its lane:

$$
\begin{array}{rcl}
\lambda_{i,1} := \lambda_{i,minLane} & = & \{(x, y, \theta, v, v_{des}) \,|\, (m - 1)w \leq y \leq m\, w\} \ , \\
\lambda_{i,2} := \lambda_{i,maxLane} & = & \{(x, y, \theta, v, v_{des}) \,|\, 0 \leq y \leq w\} \ , \\
\lambda_{i,3} := \lambda_{i,targetLeftLane} & = & \left\{(x, y, \theta, v, v_{des}) \,|\, y \geq \left(\left\lfloor \frac{y_i(t_k)}{w} \right\rfloor + 1\right) w\right\} \ , \\
\lambda_{i,4} := \lambda_{i,targetRightLane} & = & \left\{(x, y, \theta, v, v_{des}) \,|\, y \leq \left(\left\lfloor \frac{y_i(t_k)}{w} \right\rfloor\right) w\right\} \ , \\
\lambda_{i,5} := \lambda_{i,linedUp} & = & \{(x, y, \theta, v, v_{des}) \,|\, |y - \lfloor \frac{y}{w} \rfloor w - \frac{w}{2}| \leq \Delta_y \ , \\
& & \qquad |\theta| \leq \Delta_\theta\}
\end{array}
$$

where Δ_x are constant tolerances of the variable x, and t_k is the instant at which an event was detected. Through these constants we define the constant map $r_i : \mathcal{Q} \to \mathbb{B}^5$, $r_i = (r_{i,1}, \cdots, r_{i,5})$. The event alphabet is $E_i = \{e^{i,1}, \cdots, e^{i,21}\}$ and the detector map $e_i \in \mathbb{B} \to 2^{E_i}$ is characterized by the event conditions

$$
\begin{aligned}
&c_{i,1} = s_{i,1}\, s_{i,2} \neg s_{i,7}\, r_{i,5}\,, \quad c_{i,2} = s_{i,1} \neg s_{i,7}\, r_{i,2}\, r_{i,5}\,, \quad c_{i,3} = s_{i,1} \neg s_{i,7} \neg r_{i,5}\,, \quad c_{i,4} = \neg s_{i,1}\, s_{i,3}\, s_{i,7} \neg s_{i,9}\,,\\
&c_{i,5} = s_{i,1} \neg s_{i,2} \neg s_{i,7} \neg r_{i,2}\, r_{i,5}\,,\\
&c_{i,6} = \neg s_{i,1} \neg s_{i,3} \neg r_{i,1}\, r_{i,5}\,,\\
&c_{i,7} = \neg s_{i,1}\, s_{i,9}, \quad c_{i,8} = \neg s_{i,1} \neg s_{i,3}\,,\\
&c_{i,9} = s_{i,1} \neg s_{i,2} \neg r_{i,2}\, r_{i,5}\,,\\
&c_{i,10} = \neg s_{i,7}\, r_{i,3}\,,\\
&c_{i,11} = \neg s_{i,7}\, r_{i,4}\,,\\
&c_{i,12} = \neg s_{i,7} \neg s_{i,8}, \quad c_{i,13} = \neg s_{i,3} \neg s_{i,5}\, s_{i,6}\, r_{i,1},\\
&c_{i,14} = \neg s_{i,1}\, s_{i,3} \neg s_{i,9}\\
&c_{i,15} = s_{i,1} \neg s_{i,2} \neg s_{i,7} \neg r_{i,2}\\
&c_{i,16} = s_{i,7} \quad c_{i,17} = s_{i,3}\, s_{i,8} \quad c_{i,18} = s_{i,1}\, s_{i,2} \neg s_{i,7}\, s_{i,8} c_{i,19} = s_{i,1} \neg s_{i,7}\, s_{i,8}, r_{i,2}\\
&c_{i,20} = s_{i,7}\, r_{i,3}\\
&c_{i,21} = s_{i,7}\, r_{i,4}
\end{aligned}
$$

The finite set of discrete states is $\Sigma_i = \{\mathsf{FAST}, \mathsf{SLOW}, \mathsf{LEFT}, \mathsf{RIGHTPLATOON}\}$ ($p = 5$) and the automaton's dynamics is

$$
\begin{aligned}
\delta_i \;:\; & \Sigma_i \times 2^{E_i} \to \Sigma_i\\
& ((\mathsf{FAST}, \neg(e^{i,1} + e^{i,2} + e^{i,3} + e^{i,4} + e^{i,5} + e^{i,6} + e^{i,16} + e^{i,17} + e^{i,18} + e^{i,19})) \mapsto \mathsf{FAST}\,,\\
& (\mathsf{FAST}, e^{i,1}), (\mathsf{FAST}, e^{i,2}), (\mathsf{FAST}, e^{i,3}), (\mathsf{FAST}, e^{i,4}) \mapsto \mathsf{SLOW}\,,\\
& (\mathsf{FAST}, e^{i,5}) \mapsto \mathsf{LEFT},\\
& (\mathsf{FAST}, e^{i,6}) \mapsto \mathsf{RIGHT},\\
& (\mathsf{FAST}, e^{i,16}), (\mathsf{FAST}, e^{i,17}), (\mathsf{FAST}, e^{i,18}), (\mathsf{FAST}, e^{i,19}) \mapsto \mathsf{PLATOON},\\
& ((\mathsf{FAST}, \neg(e^{i,7} + e^{i,8} + e^{i,9})) \mapsto \mathsf{SLOW}\,,
\end{aligned}
$$

with initial state $\sigma_i^0 = \mathsf{FAST}$.

The decoder map is $u_i : \mathcal{Q} \times \Sigma_i \to \mathcal{U}_i$, $u_i = (a_i, \omega_i)$, with

$$
\begin{aligned}
a_i \;:\; & \mathcal{Q} \times \Sigma_i \to \mathbb{R}\\
& \begin{matrix}(q_i, \mathsf{FAST}), (q_i, \mathsf{LEFT}),\\ (q_i, \mathsf{RIGHT})\end{matrix} \mapsto \begin{cases} \bar{a} & \text{if } v_i < v^i_{max}\\ 0 & \text{otherwise}\end{cases}\,,\\
& (q_i, \mathsf{SLOW}) \mapsto \begin{cases} -\bar{a} & \text{if } v_i > 0\\ 0 & \text{otherwise}\end{cases}\,,
\end{aligned}
$$

$$
\begin{aligned}
\omega_i \;:\; & \mathcal{Q} \times \Sigma_i \to \mathbb{R}\\
& \begin{matrix}(q_i, \mathsf{FAST}),\\ (q_i, \mathsf{SLOW})\end{matrix} \mapsto \left((y^*(q_i) - y_i)\, \tfrac{\sin\theta_i}{\theta_i} - \mu\,\theta_i\right) v_i\,,\\
& (q_i, \mathsf{LEFT}), \mapsto \begin{cases} \bar{\omega} & \text{if } \theta_i < \theta_{max}\\ 0 & \text{otherwise}\end{cases}\,,\\
& (q_i, \mathsf{RIGHT}) \mapsto \begin{cases} -\bar{\omega} & \text{if } \theta_i > -\theta_{max}\\ 0 & \text{otherwise}\end{cases}\,,
\end{aligned}
$$

where $y^*(q_i) = \left(\left\lfloor \frac{y_i}{w} \right\rfloor + \frac{1}{2}\right) w$ is the current lane center, θ_{max} is the agent's maximum curvature angle, and μ, $\bar{a}$ and $\bar{\omega}$ are positive constants.

Finally, the visibility map returns the set of configurations laying within a distance R_i and that are not hidden by other cars (see e.g. the known sweeping line algorithm in [**?**] for its computation, and the examples in Fig. 5). A formal description of the map is avoided for space reasons.

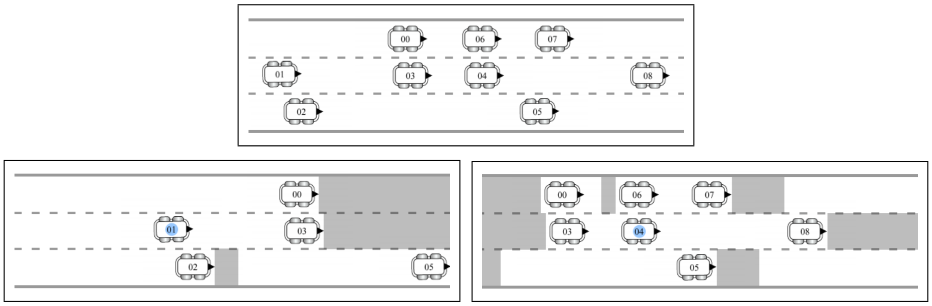


Figure 5. Sensing model in the highway example: from left to right, complete state of the system and views of agents $\mathcal{A}_1$ and $\mathcal{A}_4$, respectively.

A. Local Monitors

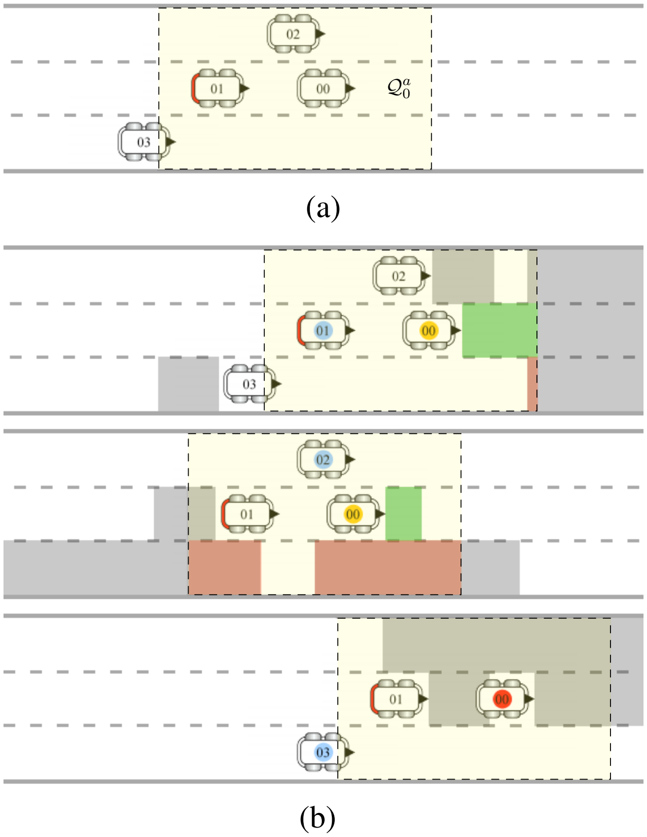
Consider four cars in the highway example (Fig. 6–a). Misbehavior of car 0, running a FAST maneuver along the second lane, while its next right lane is free, has to be detected (the car should start a RIGHT maneuver to return to the first lane). A FAST maneuver of a car in the second lane implies that the region on its right is occupied by another car. Three local monitors on the other cars try to learn whether the car 0 is cooperative or not, but have only partial view of the car's neighborhood. By means of the proposed local monitor, the three agents are able to compute estimates, $\hat{I}_0^1$, $\hat{I}_0^2$, and $\hat{I}_0^3$, of the occupancy map of car 0's neighborhood, which are reported in Fig. 6–b). However, the figure shows that all monitors are still unable to decide on the cooperativeness of the car 0, since there exist possible behaviors that comply with the cooperative model and their partial visibility.

As a second example, consider eight cooperative cars in the highway and focus on the local view of car 0's monitor (Fig. 7). The presence of car 07 is detected (region a), based on the fact that car 06 is executing a SLOW maneuver. The presence of car 5 is detected (regions e, and f), based on the FAST maneuvers on the second lane executed by cars 3 and 4. This also allows the detection of car absence in front of car 3 (region b) and car 4 (region c). To the local monitor all these neighboring cars are uncertain, except car 1 that is certainly cooperative. The example is used to show the fact that — although this goes beyond the scope of the paper — a local monitor's uncertainty in the classification of a neighbor can be reduced by cross–correlating maps of occupancies of different neighbors: the occupancy map $\hat{I}_3^0$ contains a free region (b in the figure) in front of car 3, and an occupied region (the union of d with e) on its right, while $\hat{I}_2^0$ contains a free region (same d in the figure) in front of it. Therefore the region d in $\hat{I}_3^0$ must be removed and the only possibly occupied region must be (e).

B. Monitor Agreement via Set–valued Consensus

Consider the example in Fig. 8 where four cars (2, 3, 4, and 5) are trying to reach a consensus on the misbehavior of a car (1 in the figure) remaining in the second lane. The cars can share their own local estimates of the occupancy map of car 1's neighborhood, by sending one–hop (immediate neighbor) messages through a communication network described by a connected graph $G = (V_G, E_G)$, with $V_G = \{2, 3, 4, 5\}$ and $E_G = \{e_{2,2}, e_{2,3}, e_{2,5}, e_{3,3}, e_{3,4}, e_{4,4}, e_{5,5}\}$ (note that $\mathrm{diam}(G) = 3$). The corresponding set–valued consensus protocol specializes to the following dynamic system:

$$\begin{cases} X_2(k+1) = F^{(3)}(X_2(k), X_3(k), X_5(k)) = \\ \qquad\qquad = X_2(k) \cap^* X_3(k) \cap^* X_5(k)\,, \\ X_3(k+1) = F^{(3)}(X_2(k), X_3(k), X_4(k)) = \\ \qquad\qquad = X_2(k) \cap^* X_3(k) \cap^* X_4(k)\,, \\ X_4(k+1) = F^{(2)}(X_3(k), X_4(k)) = X_3(k) \cap^* X_4(k)\,, \\ X_5(k+1) = F^{(2)}(X_2(k), X_5(k)) = X_2(k) \cap^* X_5(k)\,. \end{cases}$$



(a)

(b)

Figure 6. Misbehavior of car 0, running a FAST maneuver along the second lane, while its next right lane is free, has to be detected (a). Local maps of occupancy, $\hat{I}_0^1$, $\hat{I}_0^2$, and $\hat{I}_0^3$, which local monitors on the cars 1, 2, and 3 have reconstructed (b). The yellowish area dashed box outlines the target agent neighborhood; a blue circle specifies the current monitor; red (green) areas are non–visible regions, where the presence (absence) of a car is required. A colored circle around the target robot (green, yellow, or red) specifies its estimated cooperativeness (cooperative, uncertain, or uncooperative, respectively).

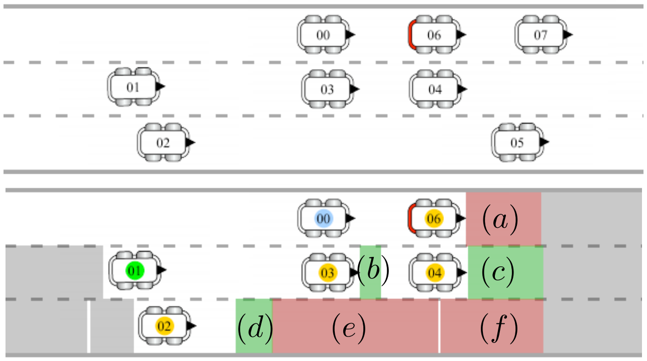


Figure 7. Eight cooperative cars (above) and view of the monitor on car 00 (below). A local monitor's uncertainty in the classification of a neighbor can be reduced by cross–correlating maps of occupancies of different neighbors.

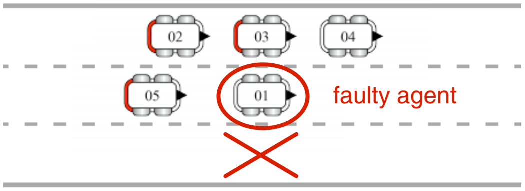


Figure 8. The misbehaving car 1 is executing a FAST maneuver on the second lane, while its next right lane is free.

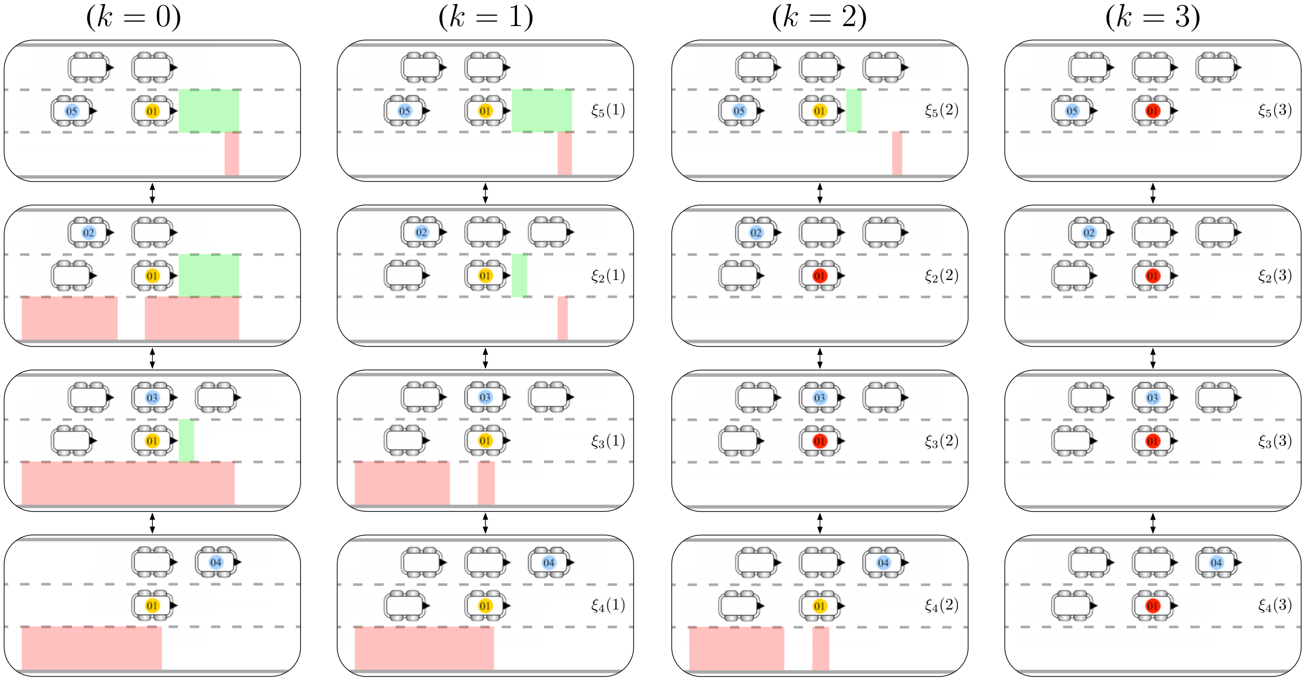


Figure 9. Misbehavior of car 1 is detected by the set–valued consensus algorithm, although no single local monitor was initially able to do it.

The system's evolution is reported in Fig. 9, where the i–th row represents the evolution of $X_i(t)$ (from left to right). No single local monitor has initially detected the misbehavior, which is instead iteratively obtained by car 2 and 3 after two consensus steps and then by the other two cars. As expected from theory, all local monitors consent to the centralized estimated occupancy map (last column in the figure)

$$X^* = \hat{I}_1 = F^{(4)}(\hat{I}_1^2, \hat{I}_1^3, \hat{I}_1^4, \hat{I}_1^5)$$

after at most 3 steps.

VI. Application to a Real Industrial Automated Warehouse

We now move on to a second example consisting of a factory's warehouse, where n autonomous forklifts are used to move products from carrier tapes to storage piles. The example is related to a real industrial automated system, yet it allows us to explicitly show the construction of a local monitor and its operation while detecting the noncooperation of a forklift due to a failure in its encoder.

The warehouse system is composed of a matrix of cells and macro-cells. *Cells* are square regions that can be exclusively occupied by a single forklift to prevent collisions, and *macro-cells* are sequence of cells, representing e.g. corridors or narrow paths, whose access from the forklifts needs to be exclusively handled to prevent deadlocks. Forklifts are assigned with *paths*, being sequences of adjacent cells or macro-cells, that may intersect. Forklifts are required to travel at a maximum speed v_{max}, if the current

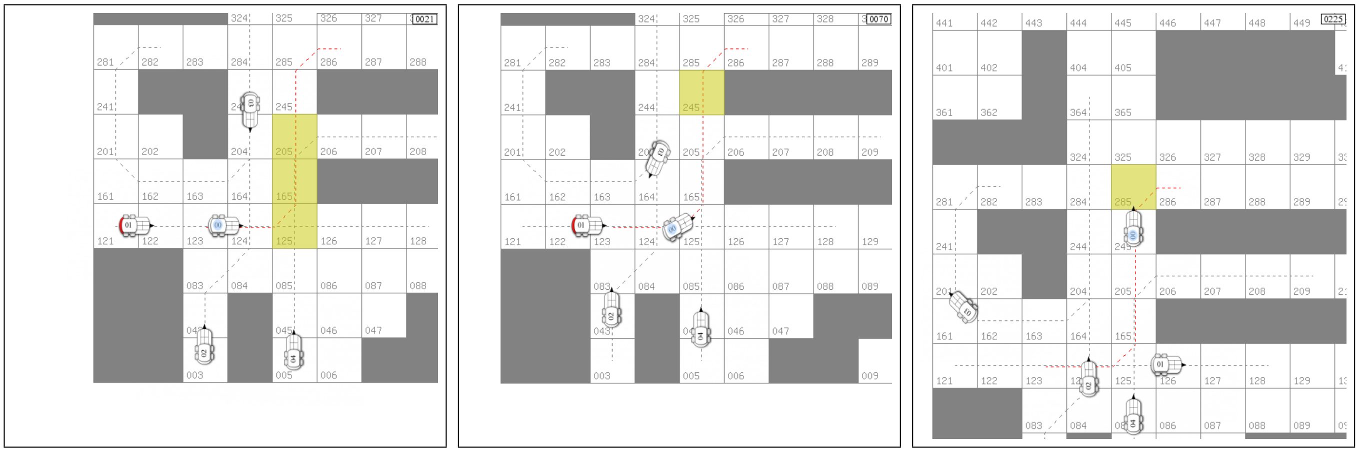


Figure 10. Snapshots from the simulation of five forklifts that cooperatively plan their motions to avoid collisions and deadlocks.

cell or macro–cell is free, or decelerate if another forklift is approaching from a path on its right. To detect neighbors forklifts are provided with 360–degree cameras with visibility range R_i.

For the sake of clarity, consider the scenario in Fig. 10 including five forklifts that must solve conflicts at cell and macro–cell levels. At cell level, forklifts 00 and 01 need to negotiate the access to the cells 123 and 125, forklifts 01 and 02 need to negotiate the access to the cell 124; forklifts 01 and 04 need to negotiate the access to the cell 125; at macro–cell level, forklifts 00 and 04 need to negotiate the access to the macro–cell $\{125, 165, 205\}$.

In the following we describe the cooperation protocol of the warehouse's example, and we exemplify the construction of the local monitor and of its operation. We also present results from experiments in a real factory hangar where 3 forklifts follow the above protocol and use the distributed IDS to discover one forklift's noncooperation.

A. *The Warehouse Protocol*

The system's cooperation protocol $\mathcal{P}$ can be described as follows. The environment is $\mathcal{Q} = R^2 \times SO(2) \times R$. An agent state is $q_i = (x_i, y_i, \theta_i, v_i)$ and, based on its input $u_i = (a_i, \omega_i)$, is updated through the dynamic map

$$\begin{aligned} f_i \;:\; & \mathcal{Q} \times \mathcal{U}_i \to T_{\mathcal{Q}} \\ & (q_i, u_i) \mapsto (v_i \cos\theta_i,\; v_i \sin\theta_i,\; \omega_i,\; a_i)^T \end{aligned} ,$$

The topology set is

$$\begin{aligned} \eta_{i,1} \;:\; & \mathcal{Q} \to 2^{\mathcal{Q}} \\ & q_i \mapsto \{(x, y, \theta, v) \in \mathcal{Q} \,|\, (x - x_i)^2 + (y - y_i)^2 \le d_i, \\ & \quad -\tfrac{\pi}{2} \le \arctan\left(\tfrac{y - y_i}{x - x_i}\right) - \theta_i \le \tfrac{\pi}{4}\right\} \end{aligned}$$

where d_i is a safety distance, and the corresponding encoder map is $s_i = s_{i,1}$ ($\kappa_i = 1$) with

$$\begin{aligned} s_{i,1} \;:\; & \mathcal{Q} \times \mathcal{Q}^{n_i} \to \mathbb{B} \\ & (q_i, I_i) \mapsto \textstyle\sum_{q_k \in I_i} \mathbf{1}_{\eta_{i,1}(q_i)}(q_k) \,. \end{aligned}$$

Thus, the agent's neighborhood is $N(q_i) = \eta_{i,1}(q_i)$. The event alphabet is $E_i = \{e^{i,1}, e^{i,2}\}$ and the detector map $e_i \in \mathbb{B} \to 2^{E_i}$, with $2^{E_i} = \{\emptyset, e^{i,1}, e^{i,2}, \{e^{i,1}, e^{i,2}\}\}$, is characterized by the event conditions $c_{i,1}, c_{i,2} : \mathbb{B} \to \mathbb{B}$, with $\gamma_{i,1} = \emptyset$, $\rho_{i,1} = \{1\}$, $\mu_{i,1} = \nu_{i,1} = \emptyset$, $\gamma_{i,2} = \{1\}, \rho_{i,2} = \mu_{i,1} = \nu_{i,1} = \emptyset$ ($\lambda_{i,j}$ need not be defined), i.e.,

$$c_{i,1} = \neg s_{i,1} \,, \quad c_{i,2} = s_{i,1} \,,$$

and thus

$$\begin{array}{rcl} e_i & : & \mathbb{B} \to 2^{E_i} \\ & & 0 \mapsto \{e^{i,1}\}\,, \;\; 1 \mapsto \{e^{i,2}\}\,. \end{array}$$

The finite set of discrete states is $\Sigma_i = \{\mathsf{ACC}, \mathsf{DEC}\}$ ($p = 2$) and the automaton's dynamics is

$$\begin{array}{rcl} \delta_i & : & \Sigma_i \times 2^{E_i} \to \Sigma_i \\ & & (\mathsf{ACC}, e^{i,1}) \mapsto \mathsf{ACC}\,, \\ & & (\mathsf{ACC}, e^{i,2}) \mapsto \mathsf{DEC}\,, \\ & & (\mathsf{DEC}, e^{i,1}) \mapsto \mathsf{ACC}\,, \\ & & (\mathsf{DEC}, e^{i,2}) \mapsto \mathsf{DEC}\,, \end{array}$$

with initial state $\sigma_i^0 = \mathsf{DEC}$. The decoder map is

$$\begin{array}{rcl} u_i & : & \mathcal{Q} \times \Sigma_i \to \mathcal{U}_i \\ & & (q_i, \mathsf{ACC}) \mapsto (-\mu(v_i - v_{max}), 0)^T\;, \\ & & (q_i, \mathsf{DEC}) \mapsto (-\mu\, v_i, 0)^T\;, \end{array}$$

where μ is a positive constant, which implies that the configuration q_i evolves according to the controlled dynamic map

$$\begin{array}{rcl} f_i^* & : & \mathcal{Q} \times \Sigma_i \to T_{\mathcal{Q}} \\ & & (q_i, \mathsf{ACC}) \mapsto (v_i \cos\theta_i,\; v_i \sin\theta_i,\; 0,\; -\mu(v_i - v_{max}))\;, \\ & & (q_i, \mathsf{DEC}) \mapsto (v_i \cos\theta_i,\; v_i \sin\theta_i,\; 0,\; -\mu\, v_i)\;. \end{array}$$

The solution $q_i(t) = \phi_{f_i^*}(q_i(t_k), \sigma_i(t_k))$, for $t \geq t_k$, of the controlled dynamics is

$$\left\{ \begin{array}{rcl} x_i(t) & = & x_i(t_k) + \Delta(\sigma_i(t_k), t) \cos(\theta_i(0))\;, \\ y_i(t) & = & y_i(t_k) + \Delta(\sigma_i(t_k), t) \sin(\theta_i(0))\;, \\ \theta_i(t) & = & \theta_i(0)\,, \\ v_i(t) & = & V(\sigma_i(t_k), t) + v_i(t_k)\, e^{-\mu(t-t_k)}\,, \end{array} \right. \tag{6}$$

with

$$\begin{array}{c} \Delta(\mathsf{ACC}, t) = v_{max}\,(t - t_k) + \frac{v_i(t_k) - v_{max}}{\mu}\left(1 - e^{-\mu(t-t_k)}\right)\;, \\ \Delta(\mathsf{DEC}, t) = \frac{v_i(t_k)}{\mu}\left(1 - e^{-\mu(t-t_k)}\right)\;, \\ V(\mathsf{ACC}, t) = v_{max}\left(1 - e^{-\mu(t-t_k)}\right)\;,\; V(\mathsf{DEC}, t) = 0\,. \end{array}$$

Finally, the visibility map is

$$\begin{array}{rcl} \mathcal{V}_i & : & \mathcal{Q}^n \to 2^{\mathcal{Q}} \\ & & (q_1, \cdots, q_n) \mapsto \{q \in \mathcal{Q} \,|\, (x - x_i)^2 + (y - y_i)^2 \leq R_i\}\,, \end{array}$$

with $R_i > d_i$.

B. Monitor Construction

Consider a forklift $\mathcal{A}_h$ trying to learn whether another forklift $\mathcal{A}_i$ is cooperative or not. The corresponding monitor is constructed as follows. The topology visibility map is $v_i = (v_{i,1})$, with

$$\begin{array}{rcl} v_{i,1} & : & \mathcal{Q} \times 2^{\mathcal{Q}} \to \mathbb{B} \\ & & (q_i, V_h) \mapsto \left\{ \begin{array}{ll} 1 & \text{if } \eta_{i,1}(q_i) \subseteq V_h\,, \\ 0 & \quad\text{otherwise}\,, \end{array} \right. \end{array}$$

and the encoder map is $\tilde{s}_i = \tilde{s}_{i,1}$ with

$$\begin{array}{rcl} \tilde{s}_{i,1} & : & \mathcal{Q} \times \mathcal{Q}^{\hat{n}_i} \to \mathbb{B} \\ & & (q_i, I_i^h) \to \sum_{q_k \in I_i^h} \mathbf{1}_{\eta_{i,1}(q_i)}(q_k)\,. \end{array}$$

The detection conditions are determined by the values of $\gamma_{i,j}$, $\rho_{i,j}$, $\mu_{i,j}$, and $\nu_{i,j}$, for $j = 1, 2$:

$$\begin{array}{rcl} \tilde{c}_i & : & \mathbb{B} \times \mathbb{B} \to \mathbb{B}^2 \\ & & (\hat{s}_i, v_i) \mapsto \left(\begin{array}{c} \neg \hat{s}_{i,1} \\ \hat{s}_{i,1} v_{i,1} + \neg v_{i,1} \end{array} \right) . \end{array}$$

The automaton's initial state is $\hat{\sigma}_i^0 = \{\text{ACC}, \text{DEC}\}$ and its nondeterministic dynamics is

$$\begin{array}{rcl} \tilde{\delta}_i & : & 2^{\Sigma_i} \times 2^{E_i} \to 2^{\Sigma_i} \\ & & \begin{array}{ll} \begin{array}{l} (\text{ACC}, e^{i,1}), (\text{DEC}, e^{i,1}), \\ (\{\text{ACC}, \text{DEC}\}, e^{i,1}) \end{array} & \mapsto \text{ACC} \, , \\ \begin{array}{l} (\text{ACC}, e^{i,2}), (\text{DEC}, e^{i,2}), \\ (\{\text{ACC}, \text{DEC}\}, e^{i,2}) \end{array} & \mapsto \text{DEC} \, , \\ \begin{array}{l} (\text{ACC}, \{e^{i,1}, e^{i,2}\}), \\ (\text{DEC}, \{e^{i,1}, e^{i,2}\}), \\ (\{\text{ACC}, \text{DEC}\}, \{e^{i,1}, e^{i,2}\}) \end{array} & \mapsto \{\text{ACC}, \text{DEC}\} \, , \end{array} \end{array}$$

and the controlled dynamic map is

$$\begin{array}{rcl} \tilde{f}_i^* & : & 2^{\mathcal{Q}} \times 2^{\Sigma_i} \to 2^{T_{\mathcal{Q}}} \\ & & (\hat{q}_i, \text{ACC}) \mapsto \Big(\hat{v}_i \cos^* \hat{\theta}_i, \; \hat{v}_i \sin^* \hat{\theta}_i, \; 0, \; -\mu(\hat{v}_i - v_{max}) \Big) \\ & & (\hat{q}_i, \text{DEC}) \mapsto \Big(\hat{v}_i \cos^* \hat{\theta}_i, \; \hat{v}_i \sin^* \hat{\theta}_i, \; 0, \; -\mu \hat{v}_i \Big) \; , \\ & & (\hat{q}_i, \{\text{ACC}, \text{DEC}\}) \mapsto \left(\begin{array}{c} \hat{v}_i \cos^* \hat{\theta}_i \\ \hat{v}_i \sin^* \hat{\theta}_i \\ 0 \\ \{-\mu(\hat{v}_i - v_{max}), -\mu \hat{v}_i\} \end{array} \right) \; , \end{array}$$

where $\cos^*$ and $\sin^*$ are

$$\begin{array}{rcl} \cos^* & : & 2^{\mathbb{R}} \to 2^{\mathbb{R}} \\ & & \hat{\alpha} \mapsto \{\alpha \in \mathbb{R} \,|\, \exists \bar{\alpha} \in \mathbb{R}, \bar{\alpha} \subseteq \hat{\alpha} \mid \cos(\bar{\alpha}) = \alpha\} \, , \\ \sin^* & : & 2^{\mathbb{R}} \to 2^{\mathbb{R}} \\ & & \hat{\alpha} \mapsto \{\alpha \in \mathbb{R} \,|\, \exists \bar{\alpha} \in \mathbb{R}, \bar{\alpha} \subseteq \hat{\alpha} \mid \sin(\bar{\alpha}) = \alpha\} \, . \end{array}$$

C. Dealing with Corrupted Encoders

Consider an attack undertaken by a misbehaving forklift $\mathcal{A}_i$ whose neighborhood $N(q_i)$ is free of other forklifts, i.e. $I_i(t) = \emptyset$ for all t, while the agent simulates the existence of a forklift $\mathcal{A}_j$ s.t. $q_j(t) \in \eta_{i,1}(q_i)$ for $t \geq 2T$. Hence, the neighbor configuration set that $\mathcal{A}_i$ pretends to be subject to is

$$I_i(t) = \left\{ \begin{array}{ll} \emptyset & t < 2T \\ q_j(t) & t \geq 2T \end{array} \right. .$$

The agent's encoder map $s_i(t) = s_{i,1}(t)$ correspondingly takes the values $s_{i,1}(t) = 0$, for $t \in [0, 2T)$ and $s_{i,1}(t) = 1$ for $t \geq 2T$, which implies

$$e_i(t_k) = e_i(s_i(t)) = \left\{ \begin{array}{ll} e^{i,1} & \text{if } t_k = 0, T \\ e^{i,2} & \text{if } t_k \geq 2T \end{array} \right. .$$

Given the agent's initial state, $q_i(0) = (3.2, 4.1, \pi/4, v_{max})$ and $\sigma_i(0) = \text{ACC}$, its behavior is computed as follows. The configuration's evolution is

$$\begin{array}{rcl} q_i(t) & = & \phi_{f_i^*}(q_i(0), \sigma_i(0)) = \\ & = & \Big(3.2 + \frac{\sqrt{2}}{2} v_{max} \, t, 4.1 + \frac{\sqrt{2}}{2} v_{max} \, t, \pi/4, v_{max} \Big) \; , \end{array}$$

for $t < 2T$, since it also holds $\sigma_i(T) = \mathsf{ACC}$. Moreover, we have $\sigma_i(2T) = \mathsf{DEC}$ and hence

$$\begin{array}{rcl} q_i(t) & = & \phi_{f_i^*}(q_i(2T), \sigma_i(2T)) = \\ & = & \left(3.2 + A, 4.1 + A, \pi/4, v_{max}\, e^{-\mu(t-2T)}\right) \,, \end{array}$$

for $t \geq 2T$, where $A = \sqrt{2}\, v_{max}\, T + \bar{\Delta}(2T)$ and $\bar{\Delta}(t_k) = \frac{v_{max}}{\mu}\left(1 - e^{-\mu(t-t_k)}\right)$.

Consider another agent $\mathcal{A}_h$ trying to learn whether $\mathcal{A}_i$ is cooperative or not. Assume that $\mathcal{A}_h$ has only partial view of the region $\eta_{i,1}(q_i)$, i.e. q_h is s.t. $\eta_{i,1}(q_i) \not\subseteq V_h(q_1, \cdots, q_n)$, and thus $v_i = 0$. At $t = 0$, the local monitor reads the measures, $\bar{q}_i(0) = q_i(0)$ and $I_i^h(0) = \emptyset$, and initializes the estimate of $\mathcal{A}_i$'s states as

$$\begin{array}{l} \hat{q}_i(0|0) = \bar{q}_i(0) = q_i(0)\,, \\ \hat{\sigma}_i(t_{-1}|0) = \Sigma_i = \{\mathsf{ACC}, \mathsf{DEC}\}\,. \end{array}$$

Given that $\hat{s}_i(0|0) = 0$, the predicted behavior of the agent during the observation period T_0 is

$$\begin{array}{rcl} \hat{\sigma}_i(0|0) & = & \tilde{\delta}_i(\hat{\sigma}_i(t_{-1}|0), \tilde{e}_i(\hat{s}_i(0|0), v_i(0)) = \\ & = & \tilde{\delta}_i(\{\mathsf{ACC}, \mathsf{DEC}\}, \{e^{i,1}, e^{i,2}\}) = \{\mathsf{ACC}, \mathsf{DEC}\}\,, \\ \hat{q}_i(t|0) & = & \phi_{\tilde{f}_i^*}(\hat{q}_i(0|0), \hat{\sigma}_i(0|0)) = \{q_i^{\mathsf{ACC}}(t|0), q_i^{\mathsf{DEC}}(t|0)\}\,, \end{array}$$

where

$$\begin{array}{l} q_i^{\mathsf{ACC}}(t|0) = \left(3.2 + \frac{\sqrt{2}}{2}\, v_{max}\, t, 4.1 + \frac{\sqrt{2}}{2}\, v_{max}\, t, \frac{\pi}{4}, v_{max}\right)\,, \\ q_i^{\mathsf{DEC}}(t|0) = \left(3.2 + \bar{\Delta}(0), 4.1 + \bar{\Delta}(0), \frac{\pi}{4}, v_{max}\, e^{-\mu t}\right)\,. \end{array}$$

We also have $L(0) = \{\left(q_i^{\mathsf{ACC}}(t|0), \mathsf{ACC}\right), \left(q_i^{\mathsf{DEC}}(t|0), \mathsf{DEC}\right)\}$. At $t = T$, the monitor reads the measures $\bar{q}_i(T) = q_i(T)$ and $I_i^h(T) = \emptyset$, and the agent's predicted state can be corrected as follows:

$$\begin{array}{l} \hat{q}_i(T|T) = \bar{q}_i(T)\,, \\ \hat{\sigma}_i(0|T) = \pi_{\Sigma_i}\left(L(0) \bowtie_\epsilon q_i(t)\right) = \mathsf{ACC}\,, \end{array}$$

which gives, along with the fact that $\hat{\sigma}_i(0|0) \bowtie_{\tilde{\delta}_i} \hat{\sigma}_i(0|T) = e^{i,1}$, the a posteriori topology activation estimate $\hat{s}_i(0|T) = 0$. The estimate of the unknown topology activation is $\hat{p}_{i,1}(\hat{s}_i(0|0), \hat{s}_i(0|T)) = 0$, which means the agent's behavior is cooperative if, and only if, no other forklift is present also in the portion of its neighborhood that is out of the monitor's visibility, i.e. in $\eta_{i,1}(q_i) \setminus V_h(q_1, \cdots, q_n)$. This finally gives the estimated neighbor configuration set $\hat{I}_i^h(0|T) = \emptyset$.

Similar computation is performed during the observation periods T_1 and T_2, which is omitted here for space reasons. At $t = 3T$, the local monitor reads the measures $\bar{q}_i(3T) = q_i(3T)$ and $I_i^h(3T) = \emptyset$, which gives the following a posteriori estimates

$$\hat{q}_i(3T|3T) = \bar{q}_i(3T)\,, \quad \hat{\sigma}_i(2T|3T) = \mathsf{DEC}\,.$$

Moreover, as $\hat{\sigma}_i(2T|2T) \bowtie_{\tilde{\delta}_i} \hat{\sigma}_i(2T|3T) = e^{i,2}$, we have $\hat{s}_i(2T|3T) = 1$ and $\hat{p}_{i,1}(\hat{s}_i(2T|2T), \hat{s}_i(2T|3T)) = 1$, which implies that the current behavior of forklift $\mathcal{A}_i$'s is compatible if, and only if, there is another forklift in the non–visible portion of $\eta_{i,1}(q_i)$. The estimated neighbor configuration set is indeed

$$\hat{I}_i^h(2T|3T) = \eta_{i,1}(q_i) \setminus V_h(q_1, \cdots, q_n)\,.$$

Note that an observer onboard the local monitor is unable to check if this estimated hypothesis is correct or not, which can be overcome as described in the following section.

D. Simulative Evaluation

In this section we perform a simulative evaluation of the proposed distributed IDS. To this purpose suppose that forklift 00 incorrectly stops, thus causing a deadlock in the system. Fig. 11 shows how the local monitors on the forklifts 01, 02, 03 and 04, are able to obtain an estimate the occupancy map, that

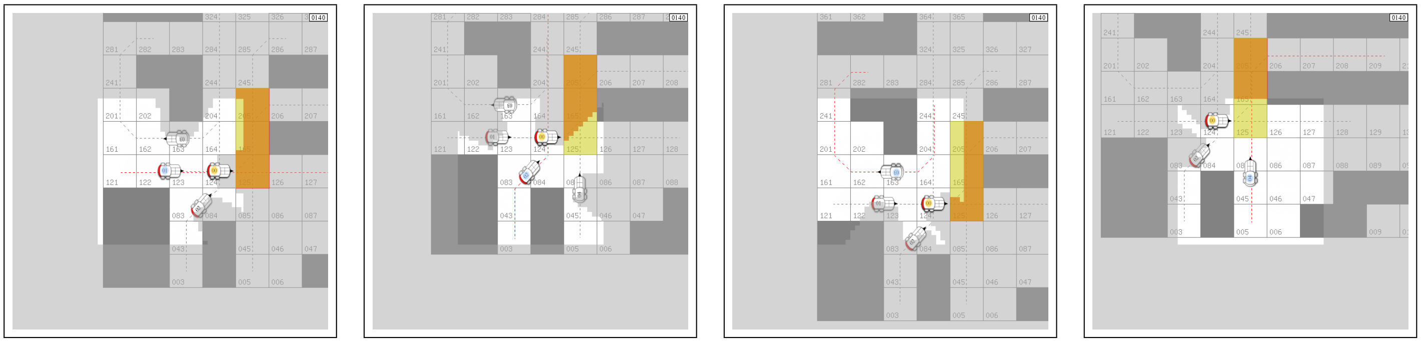


Figure 11. Occupancy maps of the forklift 00 reconstructed by the other forklifts (a blue circle indicates the subject forklift).

# monitors	# detections	# false negatives
2	8.1 %	78.3 %
3	13.9 %	81.2 %
4	24.2 %	80.7 %
5	29.9 %	75.9 %
6	33.3 %	76.6 %
> 6	~ 34 %	~ 79 %

(a)

# monitors	# detections	# false negatives
2	23.2 %	66.7 %
3	48.7 %	38.6 %
4	66.7 %	17.7 %
5	75.8 %	14.8 %
6	89.3 %	12.5 %
> 6	~ 93 %	~ 7 %

(b)

Table I

EFFECTIVENESS AND RELIABILITY OF THE PROPOSED MOTION MISBEHAVIOR DETECTION SYSTEM WITHOUT COMMUNICATION (A) AND WITH COMMUNICATION (B).

try to explain the behavior of forklift 00. The figures report each local monitor's view, in which light gray represents region where the local monitor cannot see, green (orange) represent regions where the absence (presence) of forklifts is required. The figures also show that all four monitors cannot successfully detect forklift 00's uncooperative behavior, and "optimistically" consider it as possibly cooperative.

The performance of the proposed IDS has been evaluated through execution of a large number of simulations, in which parameters, such as the forklifts initial positions and paths, have been randomly generated. Simulations including no cell or macro–cell conflicts have been discarded. In every simulation, only one forklift was programmed to uncooperatively stop while accessing to the shared cell or macro–cell. The number of local monitors was also varied. For every simulation, we have measured if at least one local monitor has discovered the misbehavior of the uncooperative forklift (this situation is graphically represented by a red circle on the uncooperative forklift). By assuming the optimistic approach, implying that an uncertain forklift for which an explanation to its behavior still exists is considered cooperative, we have measured the percentage of false negatives. This is summarized in Table I–a. The table shows that the number of local monitors affects the effectiveness and reliability of the detection system only for small numbers (< 6). For larger numbers of local monitors, the monitors themselves are even unable to see the misbehavior forklift, which is hided by some other forklifts, and thus cannot effectively participate in the detection. It is also apparent that, without monitor communication, the IDS is ineffective.

We finally show how and in what amount the ability of the detection system is improved by means of communication. Before doing this, we show how local monitors can capitalize on the possibility to share locally estimated occupancy maps by using the proposed *set–valued consensus*. Referring to the example above, Fig. 12, one for each local monitor, show how the occupancy maps are iteratively improved, which finally allows to detect the presence of the uncooperative forklift. In each figure, the first picture on the left represents the estimation that the local monitor has computed based on only locally available information, and the k–th picture (with $k > 1$) shows how the same occupancy map is improved after k steps of the consensus algorithm. By comparing the last picture on the right of the four figures, it is possible to show that all local monitors consent on the absence of an forklift in the neighborhood of forklift 00, which can be consequently categorized as uncooperative.

Table I–b reports the performance evaluation of the proposed motion misbehavior detection system while monitor communication is active. The same simulations that were selected for the evaluation of the previous section were considered. By comparing these results with the ones described in the previous section (with no monitor communication), we can see that the detection capability of the system is much further improved.

E. Experiments in a Real Industrial Plant

To prove the effectiveness of the proposed cooperation model and of the IDS, the following experiment has been done. Three forklifts are supposed to follow the cooperation rules described above. They start from cells adjacent to three different stretch–wrappers and reach final positions at the three storing locations. Their paths share a number of cells, which requires that the forklifts negotiate their access to prevent collisions. The video attached to the paper shows in the first part how a simple stopping failure of one of the forklifts can cause deadlock, if the IDS is inactive. In the second part of the video, the remaining two correct forklifts can cooperatively detect the noncooperation of the faulty one by using the proposed IDS, and can thus solve the deadlock. After consenting on the third robot's noncooperation, the other two forklifts temporarily exclude it from the cell negotiation and proceed further in their paths. In the last part of the video, the faulty robot is restored, the other two forklifts realize of it, and thus they can readmit it to the cell cooperation. By using the proposed IDS, the three forklifts are indeed able to conclude their paths. Fig 13 reports some snapshots from the experiment.

VII. Conclusion

The problem of detecting misbehaving robots in a decentralized setting was addressed in this work. Robots are supposed to interact with each other based on "rules" that depend on the status of their neighborhoods. The literature on DES for fault diagnosability and state observability (see e.g. [**?**], [**?**], [**?**]) addresses similar problems, but those solutions are not applicable mainly because the interaction topology in our systems is time–varying and unknown. Furthermore, the fault diagnosis approaches based on excitation of the model and observation of the error between expected and actual outputs [**?**] could be possible, but their would require changing the cooperation rules. The proposed solution is a distributed IDS where every agent first runs a local monitor to obtain a subjective map of free and occupied regions and then a consensus algorithm to agree on a unique shared map. We assumed that the information exchanged is always correct. Future extension of the work will address the important generalization to the case where robots can send false information due to communication failure or tampering. Preliminary but promising results toward this direction are reported in [**?**].

Acknowledgment

The authors wish to thank Elettric80 S.p.A. for providing their forklifts, Marco Casarini and Massimiliano Magnani for their knowledge and expertise, Simone Martini, Davide Di Baccio, Ida M. Savino, and Leonardo Rocchi for their fruitful work on the experimental setup.

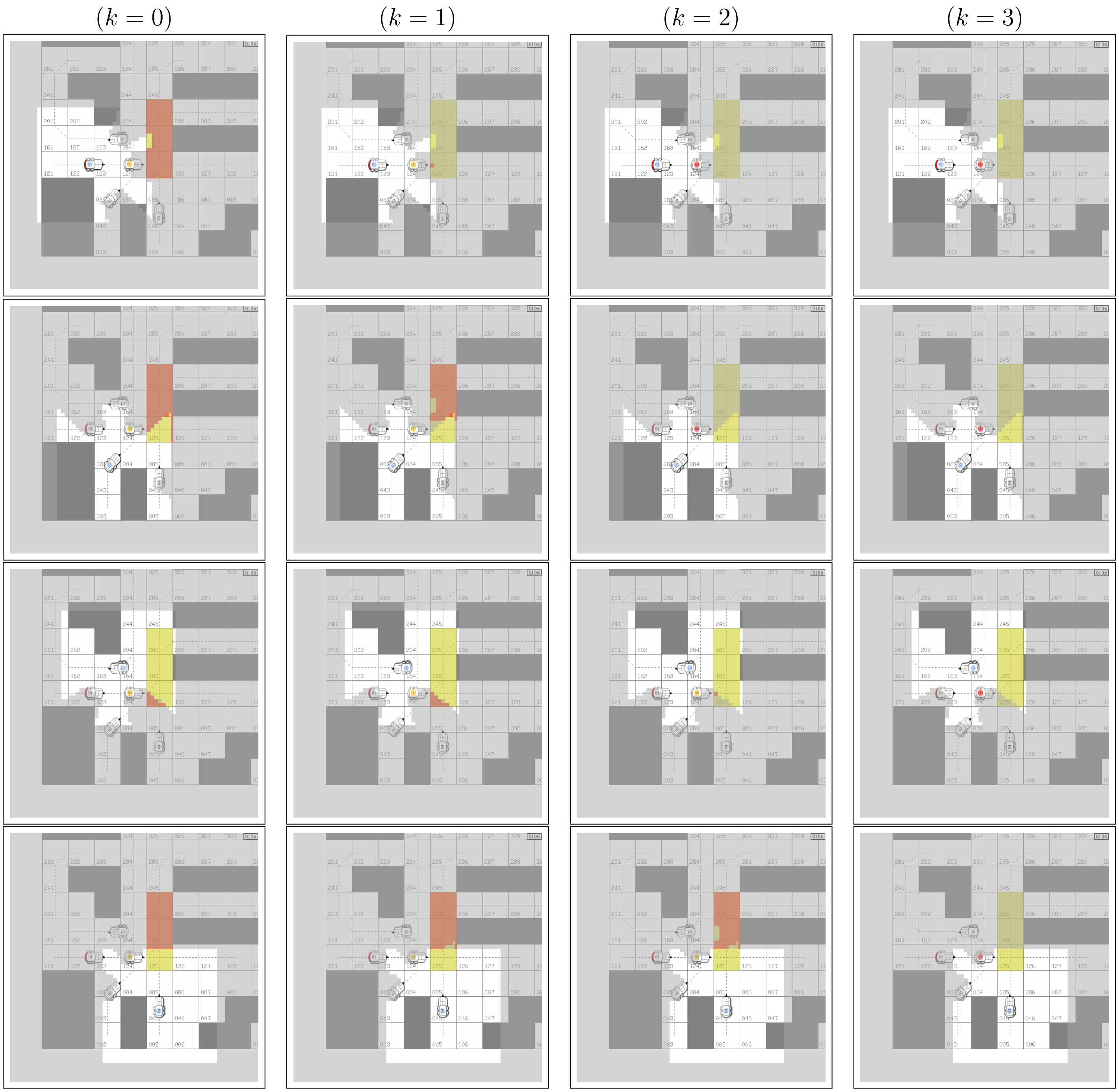


Figure 12. Consensus run of the local monitors: forklift 01 communicates with forklift 02 and forklift 03, forklift 02 communicates with forklift 01 and forklift 04, forklift 03 communicates with forklift 01, forklift 04 communicates with forklift 02.

Appendix

A. *Event Estimation with Incomplete, Time–Varying Visibility*

A proof of the formula used for the observer's detector map of Eq. 3 is given in this section. This result, along with the procedure presented in Section III for the construction of the nondeterministic automaton $\tilde{\delta}_i$ extends available solutions (see e.g. [**?**]) in sofar as that it shows that an observer for discrete event systems with uncertain events can be efficiently estimated also with incomplete, time–varying visibility.

First consider the following propositions:

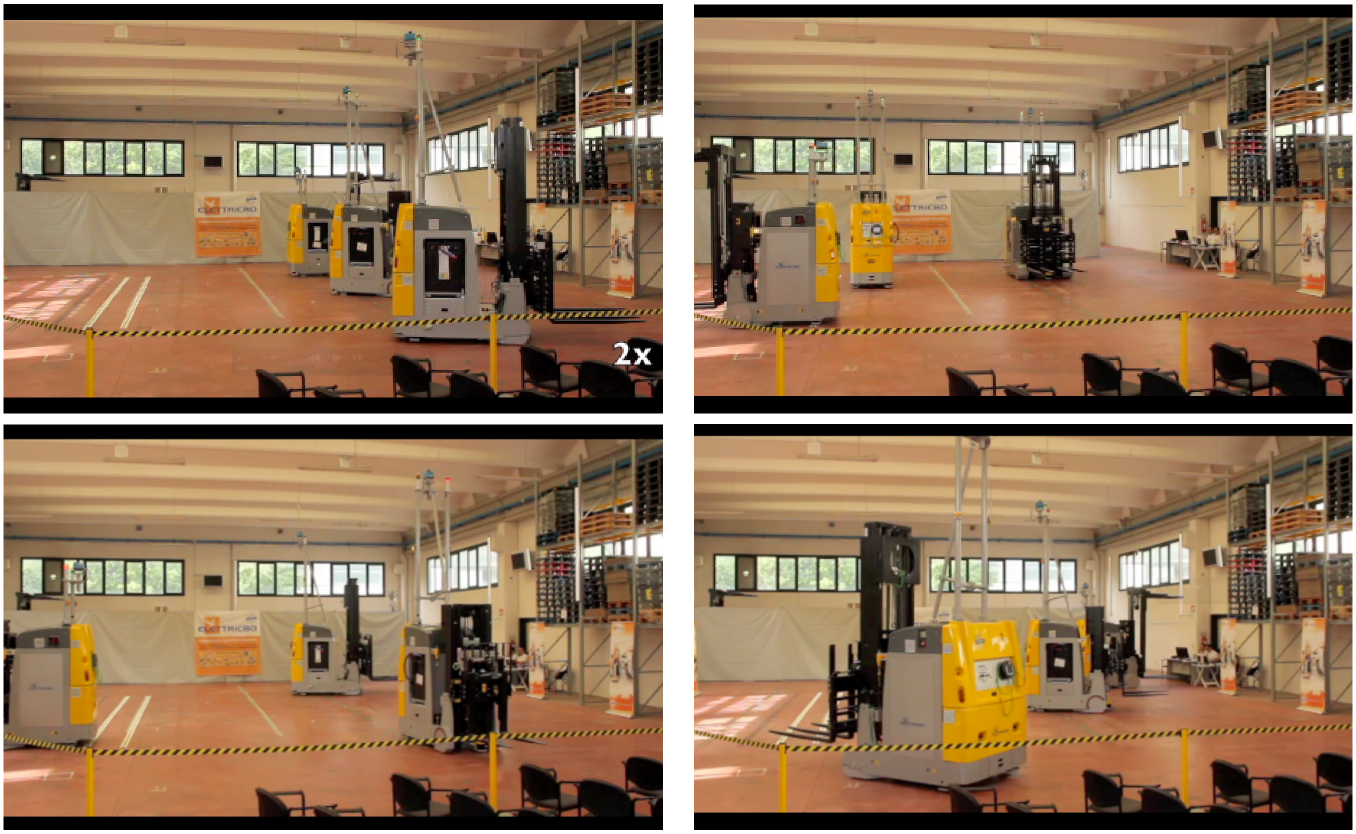


Figure 13. Real experiments with 3 forklifts running a cooperative motion protocol to avoid collisions and the proposed intrusion detection system to avoid deadlocks. Experiments have been performed at the premises of Elettric80 S.p.A.

Proposition 1: The smallest upper approximation of a detector condition $c_{i,j} = s_{i,k}$ ($\gamma_{i,j} = \{k\}, \rho_{i,j} = \mu_{i,j} = \pi_{i,j} = \emptyset$), based on an observer's topology check v_h and an available encoder map $\tilde{s}_{i,k}$, is

$$\tilde{c}_{i,j} = \tilde{s}_{i,k}\, v_{h,k} + \neg v_{h,k}\,.$$

Proof: Based on the observer's visibility region V_h, the encoder map $s_{i,k}$ can be written as

$$\begin{array}{rcl} s_{i,k}(q_i, I_i) & = & \tilde{s}_{i,k}(q_i, I_i^h) + \sum_{q_k \in I_i \setminus V_h} \mathbf{1}_{\eta_{i,k}(q_i)}(q_k) = \\ & = & \tilde{s}_{i,k}(q_i, I_i^h) + \tilde{p}_{i,k}(q_i, I_i)\,, \end{array}$$

that can be conveniently factorized as follows. If $\tilde{p}_{i,k} = 0$, the expression reduces to $c_{i,j} = \tilde{s}_{i,k}$, whereas if $\tilde{p}_{i,k} = 1$, it becomes $c_{i,j} = \tilde{s}_{i,k} + 1 = 1$. Then, the detector condition can be factorized as $c_{i,j} = \tilde{s}_{i,k}\,\neg\tilde{p}_{i,k} + 1\,\tilde{p}_{i,k}$. Moreover, if the observer has complete visibility of the k–th topology ($v_{i,k} = 1$), $\tilde{p}_{i,k} = 0$ since $I_i \setminus V_h = \emptyset$, which implies $c_{i,j} = \tilde{s}_{i,k}$, whereas nothing can be said on the value of $\tilde{p}_{i,k}$ if $v_{i,k} = 0$. Therefore, $c_{i,j}$ can be factorized w.r.t. the observer's topology check as

$$c_{i,j} = \tilde{s}_{i,k}\, v_{i,k} + (\tilde{s}_{i,k}\,\neg\tilde{p}_{i,k} + \tilde{p}_{i,k})\,\neg v_{i,k}\,.$$

Its visibility–based smallest upper approximation is

$$\tilde{c}_{i,j} \quad = \quad \max_{\tilde{p}_{i,k} \in \mathbb{B}} c_{i,j} = \tilde{s}_{i,k}\, v_{i,k} + A\,\neg v_{i,k}\,,$$

with $A = \max_{\tilde{p}_{i,k} \in \mathbb{B}} (\tilde{s}_{i,k}\,\neg\,\tilde{p}_{i,k} + \tilde{p}_{i,k}) = \max\{\tilde{s}_{i,k}, 1\} = 1$, which proves the thesis. ■

Proposition 2: The smallest upper approximation of a detector condition $c_{i,j} = \neg s_{i,k}$ ($\gamma_{i,j} = \emptyset, \rho_{i,j} = \{k\}, \mu_{i,j} = \pi_{i,j} = \emptyset$), based on an observer's topology check v_h and an available encoder map $\tilde{s}_{i,k}$, is

$$\tilde{c}_{i,j} = \neg\tilde{s}_{i,k}\,.$$

Proof: As in Prop. 1, based on the observer's visibility region V_h, the detector condition $c_{i,j}$ can be written as

$$\begin{array}{rcl}\neg s_{i,k}(q_i, I_i) & = & \neg\left(\tilde{s}_{i,k}(q_i, I_i^h) + \tilde{p}_{i,k}(q_i, I_i)\right) = \\ & = & \neg\tilde{s}_{i,k}(q_i, I_i^h)\,\neg\tilde{p}_{i,k}(q_i, I_i)\,,\end{array}$$

where De Morgan's law is used. If $\tilde{p}_{i,k} = 0$, the expression reduces to $c_{i,j} = \neg\tilde{s}_{i,k}$, whereas if $\tilde{p}_{i,k} = 1$, it becomes $c_{i,j} = 0$. Then, $c_{i,j}$ can be factorized as $c_{i,j} = \neg\tilde{s}_{i,k}\,\neg\tilde{p}_{i,k} + 0\,\tilde{p}_{i,k} = \neg\tilde{s}_{i,k}\,\neg\tilde{p}_{i,k}$. Moreover, if $v_{i,k} = 1$, $\tilde{p}_{i,k} = 0$ that implies $c_{i,j} = \neg\tilde{s}_{i,k}$, whereas nothing can be said on its value otherwise. Therefore, $c_{i,j}$ can be factorized w.r.t. the observer's topology check as

$$c_{i,j} = \neg\tilde{s}_{i,k}\,v_{i,k} + \neg\tilde{s}_{i,k}\,\neg\tilde{p}_{i,k}\,\neg v_{i,k}$$

Its visibility–based smallest upper approximation is

$$\begin{array}{rcl}\tilde{c}_{i,j} & = & \neg\tilde{s}_{i,k}\,v_{i,k} + \neg\tilde{s}_{i,k}\max_{\tilde{p}_{i,k}\in\mathbb{B}}\left(\tilde{p}_{i,k}\right)\neg v_{i,k} = \\ & = & \neg\tilde{s}_{i,k}\,v_{i,k} + \neg\tilde{s}_{i,k}\,\neg v_{i,k} = \\ & = & \neg\tilde{s}_{i,k}\left(v_{i,k} + \neg v_{i,k}\right) = \neg\tilde{s}_{i,k}\,,\end{array}$$

which gives the thesis. ■

Proposition 3: The smallest upper approximation of a detector condition $c_{i,j} = s_{i,k}\,\neg s_{i,m}$ ($\gamma_{i,j} = \{k\}, \rho_{i,j} = \{m\}, \mu_{i,j} = \pi_{i,j} = \emptyset$), based on an observer's topology check v_h, is

$$\tilde{c}_{i,j} = \left(\tilde{s}_{i,k}\,v_{h,k} + \neg v_{h,k}\right)\,\neg\tilde{s}_{i,m}\,.$$

Proof: Based on the observer's visibility region V_h, the detector condition can be written as

$$\begin{array}{rcl}c_{i,j} & = & \left(\tilde{s}_{i,k} + \tilde{p}_{i,k}\right)\left(\neg\tilde{s}_{i,m}\,\neg\tilde{p}_{i,m}\right) = \\ & = & \tilde{s}_{i,k}\neg\tilde{s}_{i,m}\neg\tilde{p}_{i,m} + \tilde{p}_{i,k}\,\neg\tilde{s}_{i,m}\neg\tilde{p}_{i,m}\,.\end{array}$$

By enumerating all possible combinations of $\tilde{p}_{i,k}$ and $\tilde{p}_{i,m}$, $c_{i,j}$ can be factorized as

$$c_{i,j} = \left(\neg\tilde{s}_{i,m}\right)\,\tilde{p}_{i,k}\,\neg\tilde{p}_{i,m} + \left(\tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}\right)\,\neg\,\tilde{p}_{i,k}\neg\tilde{p}_{i,m}\,.$$

Moreover, based on the observer's topology check (recall that $v_{i,k} = 1$ implies $\tilde{p}_{i,k} = 0$, and $v_{i,m} = 1$ implies $\tilde{p}_{i,m} = 0$), the expression can be further factorized as

$$\begin{array}{rcl}c_{i,j} & = & A\,v_{i,k}\,v_{i,m} + B\,v_{i,k}\,v_{i,m} + \\ & + & C\,\neg v_{i,k}\,v_{i,m} + D\,\neg v_{i,k}\,\neg v_{i,m}\,,\end{array}$$

with $A = \tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}$, $B = \tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}\,\neg\tilde{p}_{i,m}$, $C = \neg\tilde{s}_{i,m}\,\tilde{p}_{i,k} + \left(\tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}\right)\neg\tilde{p}_{i,k}$, and $D = \neg\tilde{s}_{i,m}\,\tilde{p}_{i,k}\,\neg\tilde{p}_{i,m} + \tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}\neg\tilde{p}_{i,k}\neg\tilde{p}_{i,m}$. Its visibility–based smallest upper approximation is

$$\begin{array}{rcl}\tilde{c}_{i,j} & = & \tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}\left(v_{i,k}\,v_{i,m} + v_{i,k}\,\neg v_{i,m}\right) + \\ & + & \neg\tilde{s}_{i,m}\left(\neg v_{i,k}\,v_{i,m} + \neg v_{i,k}\,\neg v_{i,m}\right) = \\ & = & \tilde{s}_{i,k}\,\neg\tilde{s}_{i,m}\,v_{i,k} + \neg\tilde{s}_{i,m}\neg v_{i,k}\,,\end{array}$$

which easily gives the thesis. ■

We can now readily give a proof of Theorem 1 as follows. W.r.t. the above propositions, an event estimator map e_i with detector conditions of the form of Eq. 1 is characterized by a generic combination of the sets $\gamma_{i,j}, \rho_{i,j} \in \{1, \cdots, \kappa_i\}$ and $\mu_{i,j}, \pi_{i,j} \in \{1, \cdots, h_i\}$. It is sufficient to show that the above propositions also extend to the general case.

Proof: (of Theorem 1) Let us proceed by induction. Consider the case with only $\gamma_{i,j} \neq \emptyset$ and card$(\gamma_{i,j}) \geq 1$. Assume $\gamma_{i,j} = \{1, \cdots, l\}$, which is always possible upon reordering of the encoder map's components. The case with $l = 1$ is proved by Prop. 1. By assuming that the thesis holds for $l = m$, i.e., that the smallest upper approximation of $c_{i,j} = \Pi_{k\in\gamma_{i,j}} s_{i,k} = \Pi_{k=1}^{m} s_{i,k}$ is $\tilde{c}_{i,j} = (\Pi_{k=1}^{m}\,\tilde{s}_{i,k}\,v_{i,k} + \neg v_{i,k})$, the inductive step requires proving it for $l = m + 1$. Indeed, the detector condition $c_{i,j} = \Pi_{k=1}^{m+1} s_{i,k}$ can be written as

$$\underbrace{\left(\Pi_{m=1}^{k} s_{i,k}\right)}_{z} s_{i,m+1} = z\, s_{i,m+1} = z\,\left(\tilde{s}_{i,m+1} + \tilde{p}_{i,m+1}\right)\,,$$

that can be factorized as follows. If $\tilde{p}_{i,m+1} = 0$, the expression reduces to $c_{i,j} = z\,\tilde{s}_{i,m+1}$, whereas if $\tilde{p}_{i,m+1} = 1$, it becomes $c_{i,j} = z$, thus giving the expression

$$c_{i,j} = z\,\tilde{s}_{i,m+1}\,\neg\tilde{p}_{i,m+1} + z\,\tilde{p}_{i,m+1}\,.$$

The detector condition can be factorized w.r.t. the observer's topology check $v_{i,m+1}$ as follows. If $v_{i,m+1} = 1$, we have $\tilde{p}_{i,m+1} = 0$ and $c_{i,j} = z\,\tilde{s}_{i,m+1}$, whereas if $v_{i,m+1} = 0$ nothing can be said on its value. This yields

$$c_{i,j} = z\,A\,v_{i,m+1} + z\,B\,\neg v_{i,m+1}\,,$$

with $A = \tilde{s}_{i,m+1}$, and $B = \tilde{s}_{i,m+1}\,\neg\,\tilde{p}_{i,m+1} + \tilde{p}_{i,m+1}$. Its visibility–based, smallest upper approximation is

$$\begin{gathered}\tilde{c}_{i,j} = \max_{\tilde{p}_{i,1},\cdots,\tilde{p}_{i,m+1}\in\mathbb{B}} c_{i,j} = \\ = \max_{\tilde{p}_{i,1},\cdots,\tilde{p}_{i,m}} z\,\cdot\,\max_{\tilde{p}_{i,m+1}}\left(A\,v_{i,m+1} + B\,\neg v_{i,m+1}\right) = \\ = \left(\Pi_{k=1}^{m}\,\tilde{s}_{i,k}\,v_{i,k} + \neg v_{i,k}\right)\left(\tilde{s}_{i,m+1}v_{i,m+1} + \neg v_{i,m+1}\right),\end{gathered}$$

which proves the thesis in the first considered case.

Consider the case with only $\rho_{i,j} \neq \emptyset$, $\rho_{i,j} = \{1,\cdots,l\}$. As above, we want to proceed by induction. The case with $l = 1$ is proved by Prop. 2. By assuming that the thesis holds for $l = m$, i.e., that the smallest upper approximation of $c_{i,j} = \Pi_{k\in\rho_{i,j}}\neg s_{i,k} = \Pi_{k=1}^{m}\neg s_{i,k}$ is $\tilde{c}_{i,j} = \Pi_{k=1}^{m}\,\neg\tilde{s}_{i,k}$, the inductive step requires proving it for $l = m+1$. Indeed, the detector condition

$$c_{i,j} = \Pi_{k=1}^{m+1}\neg s_{i,k} = z\,\neg\tilde{s}_{i,m+1}\,\neg\tilde{p}_{i,m+1}$$

can be factorized as follows. If $\tilde{p}_{i,m+1} = 0$, the expression reduces to $c_{i,j} = z\,\neg\tilde{s}_{i,m+1}$, whereas, if $\tilde{p}_{i,m+1} = 1$, it becomes $c_{i,j} = 0$, thus giving the expression

$$c_{i,j} = z\,\neg\tilde{s}_{i,m+1}\,\neg\tilde{p}_{i,m+1}\,.$$

The detector condition can be factorized w.r.t. the observer's topology check $v_{i,m+1}$ as follows. If $v_{i,m+1} = 1$, we have $\tilde{p}_{i,m+1} = 0$ and $c_{i,j} = z\,\neg\tilde{s}_{i,m+1}$, whereas if $v_{i,m+1} = 0$ nothing can be said on its value. This yields

$$\begin{aligned}c_{i,j} &= z\,\neg\tilde{s}_{i,m+1}\,v_{i,m+1} + z\,\neg\tilde{s}_{i,m+1}\,\neg\tilde{p}_{i,m+1}\,\neg v_{i,m+1} = \\ &= z\,\neg\tilde{s}_{i,m+1}\left(v_{i,m+1} + \neg\tilde{p}_{i,m+1}\,\neg v_{i,m+1}\right)\,.\end{aligned}$$

Its visibility–based, smallest upper approximation is $\tilde{c}_{i,j} = \left(\Pi_{k=1}^{m}\neg s_{i,k}\right)\neg\tilde{s}_{i,m+1}\,C$, with

$$\begin{aligned}C &= \max_{\tilde{p}_{i,m+1}}\left(v_{i,m+1} + \neg\tilde{p}_{i,m+1}\,\neg v_{i,m+1}\right) = \\ &= \max\{v_{i,m+1}, 1\} = 1\,,\end{aligned}$$

which proves the thesis also in this second case.

The cases with $\gamma_{i,j}, \rho_{i,j} \neq \emptyset$ and their cardinality greater than the unity straightforwardly follow from the discussion above and recursive application of Prop. 3. Finally, the estimated value of every application $\lambda_{i,k}$, affecting $c_{i,j}$ if $\mu_{i,j}, \tau_{i,j} \neq \emptyset$, coincides with its real value as they only depend on the configuration q_i of the monitored agent $\mathcal{A}_i$, that is measurable from $\mathcal{A}_h$ by assumption. ■