Abstract — This paper addresses a security problem in robotic multi-agent systems, where agents are supposed to cooperate according to a shared protocol. A distributed Intrusion Detection System (IDS) is proposed here, that detects possible non-cooperative agents. Previous work by the authors showed how single monitors embedded on-board the agents can detect non-cooperative behavior, using only locally available information. In this paper, we allow such monitors to share the collected information in order to overcome their sensing limitation. In this perspective, we show how an agreement on the type of behavior of a target–robot may be reached by the monitors, through execution of a suitable consensus algorithm. After formulating a consensus problem over non–scalar quantities, and with a generic update function, we provide conditions for the consensus convergence and an upper bound to its transient duration. Effectiveness of the proposed solution is finally shown through simulation of a case study.

I. INTRODUCTION

In the last few years, there has been a great effort to define decentralized and cooperative control strategies for applications, such as intelligent transportation, surveillance, etc., requiring the employment of teams of robots (see e.g. [1], [2]). The development of such strategies is motivated by the so–called divide et impera principle, according to which the original problem is reduced to find solutions for sub–problems of less complexity, and indeed the actions of each robot can be seen as a partial contribution to solving the complete problem.

Furthermore, the redundant number of robots allows a higher level of robustness against simple faults to be reached e.g. by a possible task–reallocation whenever a faulty robot is discovered within the system. However, in the absence of a centralized monitoring infrastructure, byzantine behaviors [3] of a robot, arbitrarily deviating from the nominal cooperation strategy, may remain undiscovered for a long time. As a matter of fact, a malicious robot may “play” with the model of cooperation and deceive any of its neighbors monitoring its behavior, by leveraging on their partial knowledge of the system’s state.

We focus on systems where cooperation is obtained by sharing a common set of decentralized rules \( R \), i.e. we consider systems where each robot plans its motion based on rules that dictate actions depending on the configuration of the robot itself and of its neighbors (see e.g. [4]–[6]). The challenge in these systems is to find strategies to detect possible non–cooperative robots, without the use of any form of centralization. Bearing this in mind, our objective is to develop a synthesis technique that makes it possible to build a distributed Intrusion Detection System (IDS) [7], [8] for securing the considered class of robotic multi–agents. The proposed IDS consists of two main “ingredients”: a decentralized monitoring mechanism, by which every robot assigns all its neighbors with a direct reputation, a measure of their cooperativeness, and an agreement mechanism, by which all of such monitors sharing locally collected information can “converge” to a unique network decision.

The concept of reputation is normally employed in Peer–To–Peer (P2P) systems, and in Mobile Ad–hoc NETworks (MANET), where a form of cooperation is required, e.g. for establishing a message routing service that enables the communication among all agents. In these systems — see e.g. the works of LeBoudec [9], [10] —, each agent assigns its neighbors with a reputation rate that depends on whether they display a collaborative behavior, e.g. with respect to message forwarding. Our problem is different and more difficult due to the fact that each robot has only partial knowledge of the system’s state, and thus it can not establish with certainty whether a given behavior of one of its neighbors is cooperative or not. The challenge of a robot acting as a decentralized monitor is indeed to distinguish a faulty or malicious robot in its neighborhood from a correctly cooperating robot whose actions may be influenced by other robots out of the monitor’s range. Furthermore, the fact that the topology of interaction and exchange of information among mobile robots is changing and unknown should be taken into account. These reasons make the problem we deal with quite distinct from those tackled in the current Security and Fault Detection [11]–[16] literatures, and indeed a very challenging one.

In previous work [17], [18], we proposed a scheme by which each robot can independently establish a reputation of all its neighbors, using only locally available information. This paper addresses the problem of reaching an agreement on such reputations, and indeed the possibility that the monitors share locally collected information is considered. To achieve this, the flourishing literature on distributed consensus algorithms [19]–[21] represents a quite natural framework under which the problem should be treated. Indeed, the system–theoretic approach (used e.g. in Murray’s works) to represent the dynamic behavior of such algorithms makes it possible to find useful results on the rate of convergence,
and on the conditions under which an agreement can be
established. However, such algorithms involve the exchange
of scalar quantities and allows the use of very simple rules
only, such as weighted average, to combine measures of
different distributed sensors. In our application scenario,
robots need to exchange locally reconstructed “evidences”
of the reputation of their neighbors that are not scalars, as
it will be discussed afterward, and hence a more complex
combination rule is required. In this vein, the works on
set-membership [22] and the so-called Marzullo’s algorithm
[23] define rules to combine sets or intervals, respectively,
estimated by different sensors. Such works may indeed
provide useful hints to solve our problem. Due to this fact, we
believe that the consensus literature can still be enriched, and
we present a convergence result when more general functions
are used to combine different measures, which may represent
a first step in this direction.

II. HYBRID MODEL OF ROBOTIC AGENTS

The class of robotic systems of interest is represented
by teams of robots that plan their motions according to a
set of decentralized and cooperative rules \( \mathcal{R} \). In particu-
lar, we assume that the set \( \mathcal{R} \) defines \( \kappa \) possible actions
\( \Sigma = \{ \sigma^1, \sigma^2, \ldots, \sigma^\kappa \} \) that robots can perform, and specifies \( \nu \) logical conditions on the state of their neighborhoods
requiring a change of maneuver. Let \( E = \{ e^1, e^2, \ldots, e^\nu \} \)
be the set of discrete events associated with such conditions.

For the sake of clarity, consider as an example the case of \( n \) cars moving on a multi–laned highway. Such cars
are supposed to have the same dynamics, and pilots are
supposed to decide the current maneuver based on its goal,
the configurations of the car and of other neighboring cars.
In this example, the actions defined by \( \mathcal{R} \) are accelerate,
decelerate, and change to the next left or right lane. The
logical conditions for a change of maneuver are represented
by e.g., a slower car in the front, and a free lane on the left
requiring the execution of an overtake.

Robotic systems composed of a physical plant and a
control system implementing such a kind of cooperation rules
\( \mathcal{R} \) can be modeled as hybrid systems. The components of
such hybrid models \( \mathcal{H} \) are depicted in Fig. 1 and explained in
the following. Let \( q_i \in \mathcal{Q} \) be a vector describing the physical
state of the \( i \)-th robot and taking value in the configuration
space \( \mathcal{Q} \), and let \( \sigma_i \in \Sigma \) be the maneuver that the robot
is currently performing. The \( i \)-th robot’s configuration \( q_i \) has
a continuous dynamics

\[
\dot{q}_i = f(q_i, u_i),
\]

where \( u_i \in \mathcal{U} \) is a control input. In particular, \( u_i \) is a
feedback law generated by a low–level controller \( g : \mathcal{Q} \times \Sigma \rightarrow \mathcal{U} \), i.e.

\[
u_i = g(q_i, \sigma_i),
\]

so that the robot’s trajectory \( q_i(t) \) corresponds to the desired
current maneuver \( \sigma_i \). The \( i \)-th robot’s current maneuver has
a discrete dynamics \( \delta : \Sigma \times E \rightarrow \Sigma \), i.e.

\[
\sigma_i^+ = \delta(\sigma_i, e),
\]

where \( e \) is an event requiring a change of maneuver from
\( \sigma_i \) to \( \sigma_i^+ \). Event activation is detected by a static map \( \mathcal{D} : \mathcal{Q} \times \mathcal{Q}^p \times \mathcal{Z} \rightarrow E \), where \( p \) is the maximum number of
neighbors whose configurations may affect the robot, and
\( \zeta_i \in \mathcal{Z} \) is a parameter that may be reset at any maneuver
transition. Map \( \mathcal{D} \) encodes conditions such as the presence
of a slower car in the front, and a free lane on the left. The
currently detected event is then

\[
e = \mathcal{D}(q_i, v_i, \zeta_i),
\]

where \( v_i = (q_{i_1}, \ldots, q_{i_p}) \) is a vector impiling the configura-
tions of the \( i \)-th robot’s neighbors. In conclusion, the hybrid
dynamics of the \( i \)-th robot is

\[
q_i = \mathcal{H}(q_i, q_{i_1}, \ldots, q_{i_p}),
\]

where \( \mathcal{H} : \mathcal{Q} \times \mathcal{Q}^p \rightarrow \mathcal{Q} \), and \( i_1, \ldots, i_p \) are the indices
of its neighbors. Hence, \( q_{i_1}, \ldots, q_{i_p} \) represents \( \mathcal{H} \)’s input and
\( q_i \) its output.

III. CONSTRUCTION OF LOCAL MONITORS FOR
INTRUSION DETECTION

We first give the following

**Definition 1:** A non–cooperative robot, or intruder, is a
faulty or malicious robot whose behavior arbitrarily deviates
from the one imposed by the cooperation rules \( \mathcal{R} \).

In practice, the \( i \)-th robot is deemed non–cooperative if
its trajectory \( \tilde{q}_i(t) \) differs from the output \( \hat{q}_i(t) \) of the hybrid
model \( \mathcal{H} \) derived from \( \mathcal{R} \) and excited by the configurations
\( q_{i_1}(t), \ldots, q_{i_p}(t) \) of its neighbors. In formula, the condition
is the following:

\[
\tilde{q}_i(t) \neq \hat{q}_i(t) = \mathcal{H}(q_i(t), q_{i_1}(t), \ldots, q_{i_p}(t)).
\]

The problem of a robot \( h \) acting as a monitor of the
behavior of robot \( i \) is due to its partial knowledge of \( i \)’s
neighborhood. In the example in study, some cars affecting
the behavior of robot \( i \) may be out of robot \( h \)'s sensing range since they remain hidden by other cars (see Fig. 2). To model this, we first partition the configuration space \( Q \) according to the \( h \)-th monitor’s visibility:

\[
Q = Q^\text{obs}_h \cup Q^\text{unobs}_h,
\]

where \( Q^\text{obs}_h \) and \( Q^\text{unobs}_h \) are the observable and the unobservable regions, respectively, from the perspective of \( h \). Then, we can partition the \( i \)-th robot’s input space \( Q^i(q_i) \) due to the \( h \)-th monitor’s visibility:

\[
Q^i(q_i) = Q^i(q_i) \cap (Q^\text{obs}_h \cup Q^\text{unobs}_h) = Q^\text{obs}_h \cup Q^\text{unobs}_h.
\]

The goal of the monitoring robot \( h \) is to establish whether the trajectory \( \hat{q}_i(t) \) of robot \( i \) is compliant with its partial knowledge of \( i \)'s neighborhood and the cooperation rules \( R \). From a mathematical point of view, we need to solve the following problem:

**Problem 1:** Consider the hybrid model \( \mathcal{H} \) of a robot \( i \), and a partition \( Q^i(q_i) = Q^\text{obs}_h \cup Q^\text{unobs}_h \) of its input space due to monitor \( h \). Given the trajectory \( \hat{q}_i(t) \), and \( n_o \) configurations \( q_1(t), \ldots, q_{n_o}(t) \) of known neighbors in \( Q^\text{obs}_h \), determine, if it exists, a choice of \( p - n_o \) configurations \( q_{n_o+1}, \ldots, q_p \) in \( Q^\text{unobs}_h \) such that the expected behavior

\[
\hat{q}_i = \mathcal{H}(\hat{q}_i, q_1, \ldots, q_{n_o}, q_{n_o+1}, \ldots, q_p)
\]

equals the given one, i.e. \( \hat{q}_i(t) = \hat{q}_i(t) \).

Solving this problem can be hard due to non-linearities and differential equations of the hybrid model \( \mathcal{H} \), and it would require the construction of an “unknown input observer” (UIO) \( \mathcal{H}^1 \) of the hybrid model itself, as we have discussed in [17]. Furthermore, a direct approach for the computation of such a UIO leads to find ad-hoc solutions for very specific cases. In contrast, we showed how this can be avoided and solutions can be found for the considered class of robotic multi-agent systems. The property that in our opinion makes our approach appealing is that all components of the proposed decentralized monitor can be automatically generated once the dynamics \( f \) of the plant, and the cooperation rules \( R \) are given. The reader may refer to our work [17] for a complete description of the method and can assume the existence of a procedure to build a UIO, \( \mathcal{H}^1 \), such that

\[
(q_{n_o+1}, \ldots, q_p) = \mathcal{H}^1(\hat{q}_i, q_1, \ldots, q_{n_o}),
\]

where \( \hat{q}_i \) for \( l = n_o + 1, \ldots, p \) are estimates of \( p - n_o \) configurations of robots in \( Q^\text{unobs}_h \) that can “explain” the behavior \( \hat{q}_i \) of the monitored robot \( i \).

In cases where the monitoring robot \( h \) has complete knowledge of robot \( i \)'s neighborhood, it will be able to distinguish a cooperative from a non-cooperative robot, and accordingly decide on its reputation \( r^h_i \). Whenever this is not true, the monitor tries to reconstruct any information on \( Q^\text{unobs}_h \) according to robot \( i \)'s behavior and the partial knowledge of its neighbors. In these cases, as long as a choice for \( \hat{q}_i \) exists, the reputation of robot \( i \) remains “uncertain” (indeed the robot may be correctly following the cooperation rules \( R \) or not). Otherwise, the reputation becomes “noncooperative”. In brief, the reputation \( r^h_i \) of robot \( i \) according to robot \( h \) is a discrete variable taking values in the set:

\[
R = \{ \text{cooperative}, \text{noncooperative}, \text{uncertain}, \text{unknown} \}.
\]

The introduction of the value “unknown” is instrumental for the purpose of communication. Indeed, whenever a monitor robot \( h \) does not see robot \( i \), but has to participate in an agreement on the value of its reputation, will initially exchange the value unknown.

We point out that the estimates \( \hat{q}_i \), for all \( l \), are evidences or unobservable explanations that the monitoring robot \( h \) has derived from the behavior of robot \( i \). Depending on the existence of such possible explanations, robot \( h \) assigns a neighboring robot \( i \) with a suitable reputation value. Fig. 3 shows a simulation run with a non–cooperative robot, vehicle 0 in the figure, that keeps traveling on the second lane, even though the lane on the right is free. The behavior of vehicle 0 is monitored by its neighbors that reconstruct different estimates \( \hat{q}_{n_o+1}, \ldots, \hat{q}_p \) of their unobservable regions. Such estimates are possibly non–convex regions where the presence of a robot is required (when reported in red) or is excluded (when reported in green).

**IV. OVERCOMING LOCAL MONITORING LIMITATION THROUGH COMMUNICATION**

The second “ingredient” of the proposed IDS is a distributed agreement mechanism by which monitors share locally collected information so as to reduce their uncertainty and eventually “converge” to a unique network decision. The communication among monitors is indeed necessary since they can not verify the actual correctness of the reconstructed hypotheses or explanations \( \hat{q}_{n_o+1}, \ldots, \hat{q}_p \) on \( Q^\text{unobs}_h \). Moreover, reaching an agreement is paramount before starting any emergency procedure whenever a non–cooperative robot is detected.

**A. Consensus algorithms and centralized decision**

Consider a piecewise–constant communication topology represented by the undirected graph \( G_c(V, E_c) \), where \( V \) is a set of nodes, and \( E_c \) is a set of edges. The presence of an edge \( e_{i,j} \) connecting \( v_i \) with \( v_j \) means that node \( v_i \) is able to share its knowledge with node \( v_j \). Now, we can recall from e.g. [20] the following...
Definition 2 (Consensus Algorithm): Given a set \( V = \{v_1,\ldots,v_n\} \) of nodes, and a communication graph \( G_c(V,E_c) \), a (distributed) consensus algorithm is an iterative interaction rule that specifies:

- which information \( d \in D \) is shared among neighbors,
- and how each node \( v_i \) updates its estimate \( d_i \) based on any received value \( d_j \), i.e. which update function \( \Omega : D \times D \to D \) is used to compute

\[
d_i^{t+1} = \Omega(d_i, d_j), \quad \text{for } i = 1,\ldots,n.
\]

Let us also define a centralized decision \( d^* \) as the value that would be chosen by a hypothetical monitor collecting all initial measures \( d_1(0),\ldots,d_n(0) \), and combining them according to \( \Omega \). The quantity \( d^* \) can be seen as a result limiting the performance of any distributed computation strategy as it represents the choice taken without any information loss. This motivates the effort that is often spent to design consensus algorithms converging to \( d^* \) (these algorithms are said to achieve the so-called \( f \)-consensus), irrespectively of the distributed nature of the computation.

B. Which Information To Share

In our application scenario, nodes in \( V \) are robots that are monitoring a common neighbor and that are supposed to communicate as in \( E_c \) in order to reach an agreement on the reputation of such neighbor. Consider vector

\[
r(k) = (r_1(k),\ldots,r_n(k))
\]

that is obtained by impiling all monitors’ decisions after \( k \) steps of a suitable consensus. Our objective here is to design a distributed consensus algorithm guaranteeing that, for any initial condition \( r(0) \), we have \( r(\infty) = r^* \), where \( r^* \) is the centralized decision.

A simple solution where the \( i \)-th monitor shares the locally established reputation \( r_i(k) \) is sufficient to reach an agreement. To achieve this, well-known consensus algorithms for scalar quantities can indeed be used (see e.g. [19]–[21]). However, in the majority of the cases, monitors are likely to have partial knowledge of the monitored robot’s neighborhood and remain uncertain about its actual behavior. Then, the whole network of robots will remain uncertain, except at the occurrence of fortunate cases where manifest faulty behaviors [24] that can trivially be detected.

For this reason, we propose a solution where monitors share any information that is directly measured or reconstructed by exploitation of \( \mathcal{H}^t \). Namely, each monitor \( h \) shares the following data related to a common neighbor \( i \):

\[
\xi_i^h = \{\hat{q}_i, q_1,\ldots,q_{\ell}, \mathcal{H}^t(q_1,\ldots,q_{\ell})\} = \{\hat{q}_i, q_1,\ldots,q_{\ell}, \hat{q}_{n_o+1},\ldots,\hat{q}_p\}.
\]

Theoretically, after having established the so-called “same context” for the value of such a neighborhood, they will use the same decision rule and hence decide for the same reputation value.

C. More General Consensus Algorithms

Well-known consensus algorithms are appealing since they are obtained through very simple combination rules, such as weighted average, or maximum occurrence value. However, they are applicable only with scalar quantities, whereas \( \hat{q}_{n_o+1},\ldots,\hat{q}_p \) are possibly non-convex sets or intervals (recall the example of Fig. 3).

Motivated by this fact, we introduce a more general class of consensus algorithms, partially inspired from the Computer Science literature (see e.g. Lynch’s works):

Definition 3 (General Consensus Algorithm): Given a set \( V = \{v_1,\ldots,v_n\} \) of nodes, and a communication graph \( G_c(V,E_c) \), a (distributed) general consensus algorithm is an iterative interaction rule that specifies:

- which information \( \xi \in \Xi \) is shared among neighbors,
- how each node \( v_i \) updates its knowledge \( \xi_i \) based on any received value \( \xi_j \), i.e. which update function \( \Omega : \Xi \times \Xi \to \Xi \) is used to compute

\[
\xi_i^{t+1} = \Omega(\xi_i, \xi_j), \quad \text{for } i = 1,\ldots,n,
\]

- and how each node \( v_i \) decides on the value \( d_i \in D \) of a common quantity of interest for which an agreement is desired, i.e. which decision function \( \Theta : \Xi \to D \) is used to compute

\[
d_i = \Theta(\xi_i), \quad \text{for } i = 1,\ldots,n.
\]

From a system theoretic point of view, the \( i \)-th node participating in the consensus is a discrete sub-system, where \( \xi_i \) is the state (a.k.a. the context), all \( \xi_j \)s are inputs, and \( d_i \) is the output (a.k.a. the decision) (see Fig. 4). Now the
centralized decision $d^*$ is the value that would be chosen by a hypothetical monitor collecting all initial measures $\xi_1(0), \ldots, \xi_n(0)$, combining them according to $\Omega$, and then applying $\Theta$.

V. ON THE ABSTRACT CONVERGENCE OF CONSensus ALGORITHMS WITH UNCERTAIN MEASURES

Let $\xi \in \mathbb{R}$ be a scalar quantity of interest for the network, and let $\xi_1, \ldots, \xi_n$ be $n$ elements on a $\sigma$-algebra $\Sigma$ over $\mathbb{R}$, representing uncertain estimates of a particular value $\bar{\xi}$ of $\xi$. Consider a consensus algorithm as in Def. 3, and assume that neighbors of a given communication graph $G_c(V, E_c)$ (as the one of Fig. 5) exchange the estimates $\xi_1, \ldots, \xi_n$, in order to reach an agreement on it.

It is worth noting that, even though the update function $\Omega : \Sigma \times \Sigma \to \Sigma$ in Def. 3 may be general, some essential properties are required to make it a legitimate update function for the distributed algorithm. In particular, we require that, for any $\xi_1, \xi_2,$ and $\xi_3$,  
\begin{itemize}
  \item $\Omega(\xi_1, \xi_2) = \Omega(\xi_2, \xi_1)$ (commutative);
  \item $\Omega(\xi_1, \Omega(\xi_2, \xi_3)) = \Omega(\Omega(\xi_1, \xi_2), \xi_3)$ (associative).
\end{itemize}
Indeed, without such assumptions, we have to specify further constraints concerning how each node updates its knowledge, and even how the centralized estimate is defined (the order by which estimates $\xi_j$s are considered is important).

In the remainder of this section, we will make a change in the notation of the update function $\Omega$ to make the exposition clearer. In particular, in place of the functional notation $\xi_i^+ = \Omega(\xi_i, \xi_j)$, we will use an equivalent form involving a binary operation $\otimes$: $\xi_i^+ = \xi_i \otimes \xi_j$. Accordingly, the iterative rule of the (distributed) consensus algorithm in Def. 3 can be written as:

$$\xi_i(k+1) = \otimes_{j \in V_i(1)} \xi_j(k) , \quad (1)$$

where $V_i(p) \triangleq \{ j \in V \mid d(i, j) \leq p \}$ is the communication neighborhood of order $p$ of the $i$-th node in $V$, and $d(i, j)$ is the geodesic distance, i.e. the shortest path length, between $i$ and $j$ (recall that $d(i, i) = 0$, $\forall i \in V$).

First, we give the following

**Definition 4**: A binary operator $\otimes$ is said to be idempotent if, and only if, for any $\xi \in \Xi$, it holds

$$\xi \otimes \xi = \xi . \quad (2)$$

**Lemma 1**: Consider $n$ initial estimates $\xi_1(0), \ldots, \xi_n(0)$ that are exchanged between neighbors of a given communication graph $G_c(V, E_c)$ according to a consensus algorithm as in Def. 3. If the binary operator $\otimes$ in Eq. 1 is commutative, associative, and idempotent, then it holds

$$\xi_i(k) = \otimes_{j \in V_i(k)} \xi_j(0) , \quad (3)$$

for all $i$ and all $k$.

**Proof**: Lemma 1 can be proved by logical induction. Consider the evolution of the $i$-th agent estimate, starting from the initial value $\xi_i(0)$. After one consensus step, we have

$$\xi_i(1) = \otimes_{j \in V_i(1)} \xi_j(0) , \quad \forall i \in V , \quad (4)$$

from Eq. 1.

Furthermore, assume that Eq. 3 holds for a certain value of $k$. Then, from Eq. 1 and Eq. 2, we obtain:

$$\xi_i(k+1) = \otimes_{j \in V_i(1)} \{ \otimes_{m \in V_j(k)} \xi_m(0) \} = \otimes_{m \in V_i(k+1)} \xi_m(0) , \quad (5)$$

where the commutative, associative, and idempotency properties of $\otimes$ have been exploited.

Observe that Eq. 3 holds also for $k = 1$, as it is shown in Eq. 4. Then, the general expression for $\xi_i(k)$ in Eq. 3 can be obtained by induction.

We are now ready to give the main result in the following

**Theorem 1 (Abstract convergence)**: Consider $n$ initial estimates $\xi_1(0), \ldots, \xi_n(0) \in \sigma(\mathbb{R})$ of a scalar $\xi \in \mathbb{R}$, a communication graph $G_c(V, E_c)$, and a legitimate update function $\Omega : \sigma(\mathbb{R}) \times \sigma(\mathbb{R}) \to \sigma(\mathbb{R})$ or the corresponding binary operator $\otimes$. The (distributed) general consensus algorithm

$$\xi_i(k+1) = \otimes_{j \in V_i(1)} \xi_j(k) \quad (6)$$

converges to a unique network decision on the centralized estimate

$$\xi^* = \otimes_{j \in V} \xi_j(0) , \quad (7)$$

i.e. $\xi(\infty) \to 1 \xi^*$, if

- $\otimes$ is idempotent, and
- $G_c$ is connected.

Furthermore, the convergence is guaranteed in a finite number of steps $\bar{n}$ given by:

$$\bar{n} \leq \max_{i,j \in V} d(i, j) = \text{diameter}(G_c) . \quad (8)$$

**Proof**: Sufficiency of the conditions on $\otimes$ can be proved by observing that, if $n = \max_{i,j \in V} d(i, j)$, then, since graph $G_c$ is connected,

$$V_i(k) = V , \quad \forall k \geq n , \quad (9)$$

and, for Lemma 1 and Eq. 7, we have

$$\xi_i(k) = \otimes_{j \in V_i(n)} \xi_j(0) = \otimes_{j \in V} \xi_j(0) = \xi^* , \quad (10)$$

for all $i \in V$, and for all $k \geq n$. Thus, we obtain the thesis.
In the example in study, the update function $\Omega$, or equivalently the operator $\otimes$, involved in the agreement mechanism is the set–intersection $\bigcap$, which satisfies the hypotheses of Theorem 1. Moreover, the decision function $\Theta$ is the decentralized monitoring mechanism based on the construction of the UIO $\mathcal{H}^\dagger$.

VI. APPLICATION

A. An Automated Highway

The case study considers a scenario where $n$ mobile robots are traveling along a highway with different maximum speed and may want to reach different desired positions. Robots are supposed to cooperate according to the common driving rules (the above set $\mathcal{R}$) in order to avoid collisions. More precisely, any robot is allowed to perform at any instant one of the following maneuvers based on logical conditions on its neighborhood (the associated events are in Table I and II):

- proceed at the maximum speed along the rightmost free lane when possible (fast maneuver);
- if a slower vehicle precedes in front on the same lane, then overtake the vehicle if the next lane on the left is free (left maneuver), or reduce the speed (slow maneuver) otherwise;
- as soon as the next lane on the right becomes free, change to that lane (right maneuver);
- overtaking any vehicle on the right is forbidden.

Our task is to detect misbehaving vehicles.

The physical state of the $i$–th robot is $q_i = (x_i, y_i, \theta_i, v_i)$ (refer to Fig. 6) and has the following continuous unicycle-like dynamics $f$:

$$
\begin{align*}
\dot{x}_i &= v_i \cos \theta_i, \\
\dot{y}_i &= v_i \sin \theta_i, \\
\dot{\theta}_i &= \omega_i, \\
\dot{v}_i &= a_i,
\end{align*}
$$

where $a_i$ and $\omega_i$ are linear acceleration and angular velocities, respectively. According to the set $\mathcal{R}$, the maneuver $\sigma_i$ of the $i$–th robot may take value on the set $\Sigma = \{\text{fast, left, right, slow}\}$ and has the discrete dynamics $\delta$ of the automaton in Fig. 7, where the low–level feedback controller $g$ ensures that the current maneuver $\sigma_i$ is performed.

1Observe that $x_i$ and $l_j$ are short–hands for $x_{ij}$ and $l_{ij}$, being related to the $j$–th neighbor of vehicle $i$.

![Fig. 6. A 2–lane automated highway with a set of common individual driving rules.](image)

![Fig. 7. Discrete dynamics $\delta$ of the automaton, and low–level feedback control $g$ ensuring that the plant $f$ behaves according to the rule set $\mathcal{R}$.](image)

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIST OF EVENTS FOR VEHICLES MOVING ALONG A 2–LANE HIGHWAY</strong></td>
</tr>
<tr>
<td>$e_i^{F \rightarrow L}$</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$e_i^{F \rightarrow S}$</td>
</tr>
<tr>
<td>$e_i^{R \rightarrow F}$</td>
</tr>
<tr>
<td>$e_i^{L \rightarrow F}$</td>
</tr>
<tr>
<td>$e_i^{R \rightarrow S}$</td>
</tr>
<tr>
<td>$e_i^{L \rightarrow S}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIST OF LITERALS FOR VEHICLES MOVING ALONG A 2–LANE HIGHWAY</strong></td>
</tr>
<tr>
<td>$l_1(q_i, q_j) = (x_j - x_i \leq d) \wedge (x_j \geq x_i) \wedge (</td>
</tr>
<tr>
<td>$l_2(q_i, q_j) = (</td>
</tr>
<tr>
<td>$l_3(q_i) =</td>
</tr>
<tr>
<td>$l_4(q_i) =</td>
</tr>
<tr>
<td>$l_5(q_i, q_j) = (</td>
</tr>
</tbody>
</table>

B. Consensus Simulation

Consider the following simulation run where robot 1 is non–cooperative since it remains in the second lane, whereas it should start a right maneuver as the next lane on its right is free (see Fig. 8–a). Furthermore, assume that the other robots, 2, 3, 4, and 5 in the figure, are acting as monitors of robot 1 and share their local estimates $\xi_i$ of vehicle 1’s neighborhood. Assume that communication occurs according to the following (undirected) graph $G_c(V, E_c)$, where is $V = \{2, 3, 4, 5\}$, and $E_c = \{e_{2,2}, e_{2,3}, e_{2,5}, e_{3,3}, e_{3,4}, e_{4,4}, e_{5,5}\}$.

Then, for the given communication graph $G_c$, we obtain the following instance of consensus algorithm:

$$
\begin{align*}
\xi_2(k + 1) &= \xi_2(k) \cap \xi_3(k) \cap \xi_5(k), \\
\xi_3(k + 1) &= \xi_2(k) \cap \xi_3(k) \cap \xi_4(k), \\
\xi_4(k + 1) &= \xi_3(k) \cap \xi_4(k), \\
\xi_5(k + 1) &= \xi_2(k) \cap \xi_5(k).
\end{align*}
$$

The first column of Fig. 9 is a graphical representation of the initial estimates $q_{n+1}, \ldots, q_p$ of robot 1’s neighborhood.
Fig. 9. Consensus run for the given communication graph $G_c$. Robot 1’s non–cooperation is detected, and an agreement is reached on its reputation.

reconstructed by all the monitors. The corresponding centralized estimate $\xi^* = \xi_2(0) \cap \xi_3(0) \cap \xi_4(0) \cap \xi_5(0)$ is illustrated in Fig. 8–b, where robot 1’s non–cooperation is detected (the centralized decision is indeed $d^* = \text{noncooperative}$). This observation along with the fact that the communication graph $G_c$ is connected ensure that the same decision can be reached by the distributed computation (see Theorem 1).

Simulation results are reported in Fig. 9, where the $k$–th column shows the monitors’ reconstructed neighborhood of vehicle 1, after $k$ steps of consensus. Moreover, we can define relative uncertainty measures of the monitors w.r.t. the desired centralized estimate $\xi^*$ reported in Fig. 8–b as

$$\mu_i(k) = \mu(\xi_i(k) \setminus \xi^*)$$

for $i = 2, 3, 4, 5$,

where $\mu$ is a function that computes the area of the set received as argument. Such uncertainties converge to 0 during the consensus run (see Fig. 10). Finally, robot 1’s non–cooperation is detected, and an agreement on $d^*$ is reached for its reputation in $\tilde{n} = 3$ steps as expected from theory (see Fig. 11).

Similar consensus runs can be shown for cooperative robots, and the agreement on the centralized decision for the reputation is always achieved. Notwithstanding, there are configurations for which it is not possible to distinguish a
cooperative from a non–cooperative robot (we omit examples for space reasons). However, this limitation is due to the instantaneous distribution of the sensors, and it is not due to the consensus algorithm.

Although results have been presented only from the same case study, the synthesis technique remains valid also for other multi–robot systems. Indeed, in Fig. 12, a snapshot from the simulation run of a system of vehicles travelling along crossing paths is reported. Vehicles are supposed to give way to vehicles coming from their right. In the figure, vehicle 0 is monitoring all vehicles that are in line–of–sight with it and reconstructs information about its unobservable regions. The reader may refer to the site http://www.piaggio.ccii.unipi.it/~fagiolini/icra2008 for some relevant videos.

VII. CONCLUSION

In this paper, we presented work aimed at developing a synthesis technique that makes it possible to build a distributed Intrusion Detection System (IDS) for securing a class of robotic multi–agents. The proposed IDS consists of a decentralized monitoring mechanism, by which every robot assigns all its neighbors with a direct reputation of their cooperativeness, and an agreement mechanism, by which all of such monitors sharing locally collected information can “converge” to a unique network decision. Many problems remain to be addressed, such as the presence of malicious monitors sharing false information and thus leading the system to incorrectly classify any monitored robot.

VIII. ACKNOWLEDGMENTS

Authors wish to thank Dr. L. Pallottino for useful discussions. The work was partially supported by EC Network of Excellence HYCON (Contract IST-2004-511368), and by Cassa di Risparmio di Pisa, Lucca e Livorno.

REFERENCES


