Consensus–based Distributed Intrusion Detection for Multi–Robot Systems

Adriano Fagiolini, Marco Pellinacci, Gianni Valenti, Gianluca Dini, and Antonio Bicchi

***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 $\mathcal{R}$, 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 florishing 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,

A. Fagiolini, G. Valenti, and A. Bicchi are with the Interdepartmental Research Center "E. Piaggio" of the Università di Pisa, Italy, {a.fagiolini, bicchi}@ing.unipi.it, posta@gianni.valenti.name.

M. Pellinacci, and G. Dini are with the Dipartimento dell'Informazione, Faculty of Engineering, Università of Pisa, Italy, marco.pellinacci@alice.it, gianluca.dini@ing.unipi.it.

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 particular, we assume that the set $\mathcal{R}$ defines κ possible *actions* $\Sigma = \{\sigma^1, \sigma^2, \ldots, \sigma^\kappa\}$ that robots can perform, and specifies ν *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 $g : \mathcal{Q} \times \Sigma \to \mathcal{U}$, i.e.

$$u_i = g(q_i, \sigma_i)\,,$$

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

$$\sigma_i{}^+ = \delta(\sigma_i, e)\,,$$

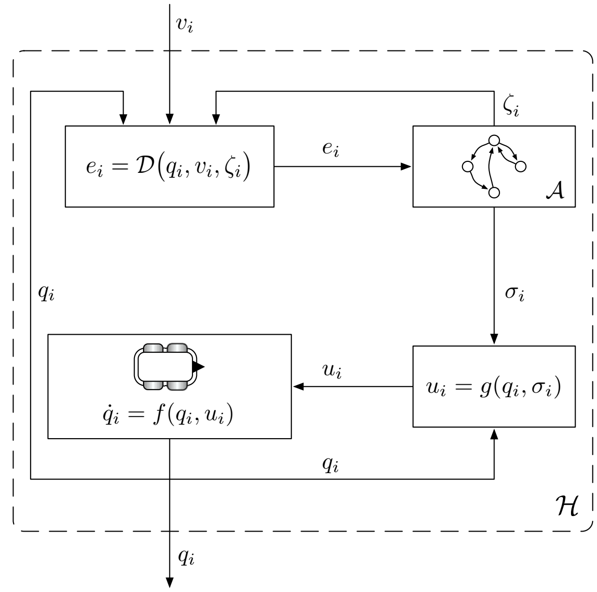

Fig. 1. Depiction of the hybrid model of robotic agents.

where e is an event requiring a change of maneuver from σ_i to $\sigma_i{}^+$. Event activation is detected by a static map $\mathcal{D} : \mathcal{Q} \times \mathcal{Q}^p \times Z \to E$, where p is the maximum number of neighbors whose configurations may affect the robot, and $\zeta_i \in Z$ 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 configurations 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 $\mathcal{H} : \mathcal{Q} \times \mathcal{Q}^p \to \mathcal{Q}$, 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 malicous 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 $\bar{q}_i(t)$ differs from the output $\tilde{q}_i(t)$ 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:

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

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

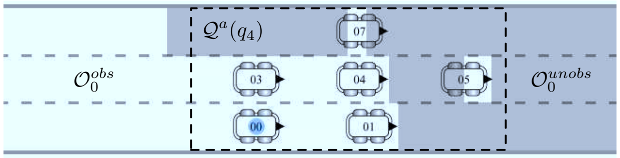

Fig. 2. Partition of the configuration space due to robot 0's visibility, and corresponding partition of the input space of robot 4.

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 $\mathcal{Q}$ according to the h–th monitor's visibility:

$$\mathcal{Q} = \mathcal{O}_h^{obs} \cup \mathcal{O}_h^{unobs} \,,$$

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

$$\begin{aligned}\mathcal{Q}^a(q_i) &= \mathcal{Q}^a(q_i) \cap \left(\mathcal{O}_h^{obs} \cup \mathcal{O}_h^{unobs}\right) = \\ &= \mathcal{Q}_h^{obs} \cup \mathcal{Q}_h^{unobs} \,.\end{aligned}$$

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

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

$$\tilde{q}_i = \mathcal{H}(\bar{q}_i, q_1, \ldots, q_{n_o}, q_{n_o+1}, \ldots, q_p)$$

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

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) $\mathcal{H}^\dagger$ 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 $\mathcal{R}$ 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, $\mathcal{H}^\dagger$, such that

$$(\hat{q}_{n_o+1}, \ldots, \hat{q}_p) = \mathcal{H}^\dagger(\bar{q}_i, q_1, \ldots, q_{n_o}) \,,$$

where $\hat{q}_l$ for $l = n_o + 1, \ldots, p$ are estimates of $p - n_o$ configurations of robots in $\mathcal{Q}_h^{unobs}$ that can "explain" the behavior $\bar{q}_i$ 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 ${r_i}^h$. Whenever this is not true, the monitor tries to reconstruct any information on $\mathcal{Q}_h^{unobs}$ according to robot i's behavior and the partial knowledge of its neighbors. In these cases, as long as a choice for $\hat{q}_l$ exists, the reputation of robot i remains "$\mathrm{uncertain}$" (indeed the robot may be correctly following the cooperation rules $\mathcal{R}$ or not). Otherwise, the reputation becomes "$\mathrm{noncooperative}$". In brief, the reputation ${r_i}^h$ of robot i according to robot h is a discrete variable taking values in the set:

$$R = \{\mathrm{cooperative}, \mathrm{noncooperative}, \mathrm{uncertain}, \mathrm{unknown}\} \,.$$

The introduction of the value "$\mathrm{unknown}$" 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 $\mathrm{unknown}$.

We point out that the estimates $\hat{q}_l$, 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 $\mathcal{Q}_h^{unobs}$. 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

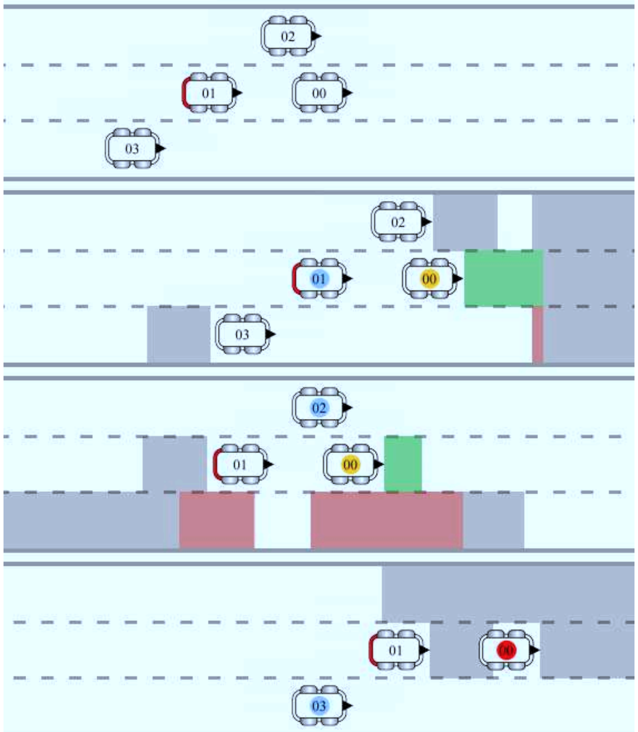

Fig. 3. Simulation run where robot 0 is non–cooperative as it keeps traveling on the second lane, even though the lane on the right is free (first picture). Monitor robots' point of views are reported in the other three pictures, where red and green colors indicate regions where the presence of a robot is required or is excluded, respectively.

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

$$d_i{}^+ = \Omega(d_i, d_j)\,, \text{ for } i = 1, \ldots, n\,.$$

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 Ω. 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 $r(\infty) = \mathbf{1}\, r^*$, 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 $\mathcal{H}^\dagger$. Namely, each monitor h shares the following data related to a common neighbor i:

$$\begin{aligned}\xi_i^h &= \{\bar{q}_i, q_1, \ldots, q_{n_o}, \mathcal{H}^\dagger(q_1, \ldots, q_{n_o})\} = \\ &= \{\bar{q}_i, q_1, \ldots, q_{n_o}, \hat{q}_{n_o+1}, \ldots, \hat{q}_p\}\,.\end{aligned}$$

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 ξ_i based on any received value ξ_j, i.e. which *update function* $\Omega : \Xi \times \Xi \to \Xi$ is used to compute

$$\xi_i{}^+ = \Omega(\xi_i, \xi_j)\,, \text{ for } i = 1, \ldots, n\,,$$

- 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

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

From a system theoretic point of view, the i–th node participarting in the consensus is a discrete sub–system, where ξ_i is the state (a.k.a. the *context*), all ξ_js are inputs, and d_i is the output (a.k.a. the decision) (see Fig. 4). Now the

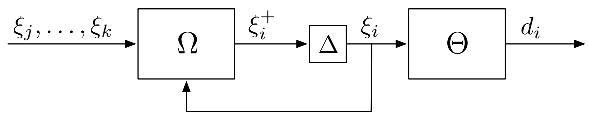

Fig. 4. Depiction of i–th node participating in the general consensus algorithm.

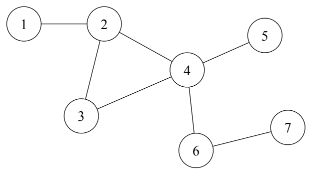

Fig. 5. A connected communication graph $G_c(V, E_c)$.

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 Ω, and then applying Θ.

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 σ–algebra Σ over $\mathbb{R}$, representing uncertain estimates of a particular value $\bar{\xi}$ of ξ. 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 $\bar{\xi}$.

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 *legittimate* update function for the distributed algorithm. In particular, we require that, for any ξ_1, ξ_2, and ξ_3,

- $\Omega(\xi_1, \xi_2) = \Omega(\xi_2, \xi_1)$ (commutative);
- $\Omega(\xi_1, \Omega(\xi_2, \xi_3)) = \Omega(\Omega(\xi_1, \xi_2), \xi_3)$ (associative).

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 ξ_js are considered is important).

In the remainder of this section, we will make a change in the notation of the update function Ω 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 *operator*: $\xi_i^+ = \xi_i \oslash \xi_j$. Accordingly, the iterative rule of the (distributed) consensus algorithm in Def. 3 can be written as:

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

where $V_i(p) \stackrel{\triangle}{=} \left\{ j \in V \mid d(i,j) \leq p \right\}$ 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 $\oslash$ is said to be *idempotent* if, and only if, for any $\xi \in \Xi$, it holds

$$\xi \oslash \xi = \xi\,. \tag{2}$$

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 $\oslash$ in Eq. 1 is commutative, associative, and idempotent, then it holds

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

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

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

from Eq. 1.

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

$$\begin{aligned} \xi_i(k+1) &= \oslash_{j \in V_i(1)} \left\{ \oslash_{m \in V_j(k)} \xi_m(0) \right\} = \\ &= \oslash_{m \in V_i(k+1)} \xi_m(0)\ , \end{aligned} \tag{5}$$

where the commutative, associative, and idempotency properties of $\oslash$ 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 legittimate update function $\Omega : \sigma(\mathbb{R}) \times \sigma(\mathbb{R}) \to \sigma(\mathbb{R})$ or the corresponding bynary operator $\oslash$. The (distributed) general consensus algorithm

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

converges to a unique network decision on the centralized estimate

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

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

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

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

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

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

$$V_i(k) = V\ ,\ \forall k \geq n\ , \tag{9}$$

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

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

for all $i \in V$, and for all $k \geq n$. Thus, we obtain the thesis. ■

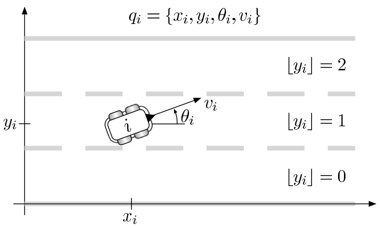

Fig. 6. A 2–lane automated highway with a set of common individual driving rules.

In the example in study, the update function Ω, or equivalently the operator $\oslash$, involved in the agreement mechanism is the set–intersection $\bigcap$, which satisfies the hypotheses of Theorem 1. Moreover, the decision function Θ 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[1]):

- proceed at the maximum speed along the rightmost free lane when possible (fast maneuver);
- if a slower vehicle proceeds 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:

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

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

[1]Observe that x_j and l_j are short–hands for x_{i_j} and l_{i_j}, being relating to the j–th neighbor of vehicle i.

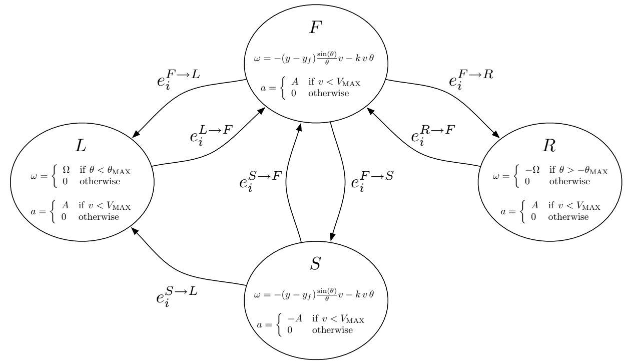

Fig. 7. Discrete dynamics δ of the automaton, and low–level feedback control g ensuring that the plant f behaves according to the rule set $\mathcal{R}$.

TABLE I

List of events for vehicles moving along a 2–lane highway

$$\begin{aligned}
e_i^{F\to L} &= (\exists\, j \in N_i \mid l_1(q_i, q_j)) \,\wedge \\
&\wedge \quad (\nexists\, k \neq j \in N_i \mid l_2(q_i, q_k)) \,\wedge \neg\, l_4(q_i) \\[1ex]
e_i^{F\to S} &= e_{i,1}^{F\to S} \vee e_{i,2}^{F\to S} \\
e_{i,1}^{F\to S} &= (\exists\, j \in N_i \mid l_1(q_i, q_j)) \wedge (\exists\, k \neq j \in N_i \mid l_2(q_i, q_k)) \\
e_{i,2}^{F\to S} &= (\exists\, j \in N_i \mid l_1(q_i, q_j)) \wedge l_4(q_i) \\[1ex]
e_i^{F\to R} &= (\nexists\, j \in N_i \mid l_5(q_i, q_j) \wedge \neg\, l_3(q_i) \\[1ex]
e_i^{L\to F} &= l_4(q_i)\,, \qquad e_i^{R\to F} = l_3(q_i) \\[1ex]
e_i^{S\to L} &= e_i^{F\to L}\,, \qquad e_i^{S\to F} = (\nexists\, j \in N_i \mid l_1(q_i, q_j))
\end{aligned}$$

TABLE II

List of literals for vehicles moving along a 2–lane highway

$$\begin{aligned}
l_1(q_i, q_j) &= (x_j - x_i \le d) \wedge (x_j \ge x_i) \wedge (\lfloor y_j \rfloor = \lfloor y_i \rfloor) \\
l_2(q_i, q_j) &= (|x_j - x_i| \le d) \wedge (\lfloor y_j \rfloor > \lfloor y_i \rfloor) \\
l_3(q_i) &= \lfloor y_i \rfloor = 1 \\
l_4(q_i) &= \lfloor y_i \rfloor = 2 \\
l_5(q_i, q_j) &= (|x_j - x_i| \le d) \wedge (\lfloor y_j \rfloor > \lfloor y_i \rfloor)
\end{aligned}$$

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 ξ_is 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:

$$\begin{cases} {\xi_2}^+(k+1) = \xi_2(k) \,\cap\, \xi_3(k) \,\cap\, \xi_5(k)\,, \\ {\xi_3}^+(k+1) = \xi_2(k) \,\cap\, \xi_3(k) \,\cap\, \xi_4(k)\,, \\ {\xi_4}^+(k+1) = \xi_3(k) \,\cap\, \xi_4(k)\,, \\ {\xi_5}^+(k+1) = \xi_2(k) \,\cap\, \xi_5(k)\,. \end{cases}$$

The first column of Fig. 9 is a graphical representation of the initial estimates $\hat{q}_{n_o+1}, \ldots, \hat{q}_p$ of robot 1's neighborhood

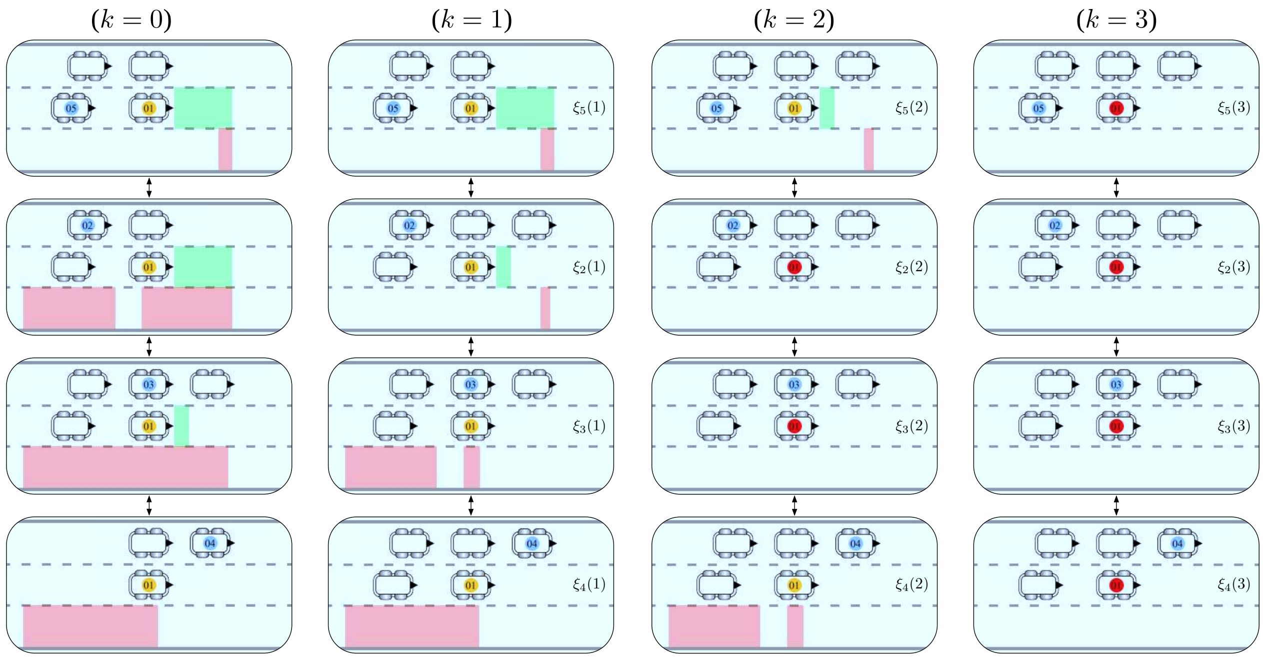

Fig. 9. Consensus run for the given communication graph G_c. Robot 1's non–cooperation is detected, and an agreement is reached on its reputation.

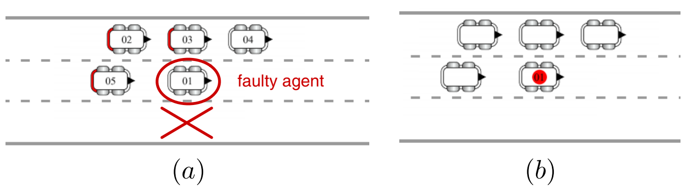

Fig. 8. Simulation run with a non–cooperative robot (a), and centralized decision d^* where the non–cooperation is detected (b).

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 $d^* = \mathrm{noncooperative}$). 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 ξ^* reported in Fig. 8–b as

$$\mu_i(k) = \mu(\xi_i(k) \setminus \xi^*)\,, \text{ for } i = 2, 3, 4, 5\,,$$

where μ 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

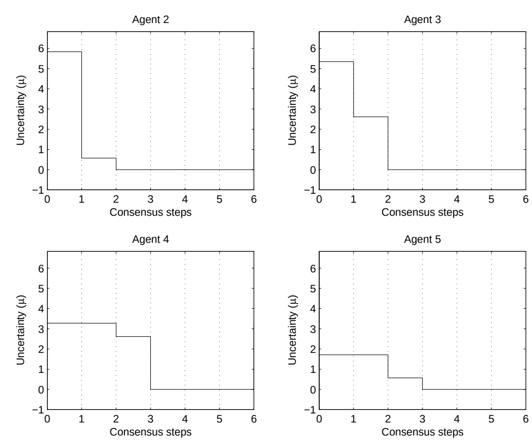

Fig. 10. Convergence of the uncertainty measures $\mu_i(k) = \mu(\xi_i(k) \setminus \xi^*)$, for $i = 2, 3, 4, 5$.

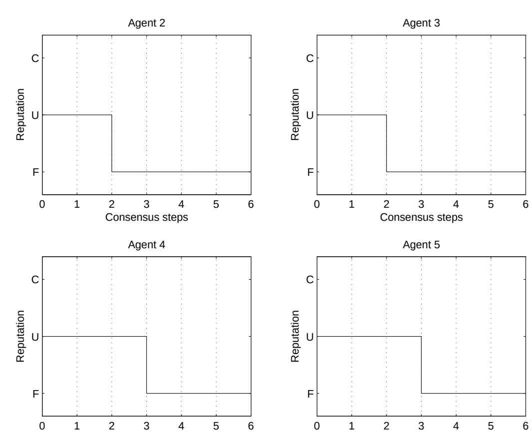

Fig. 11. Agreement on the centralized estimate d^* for robot 1's reputation.

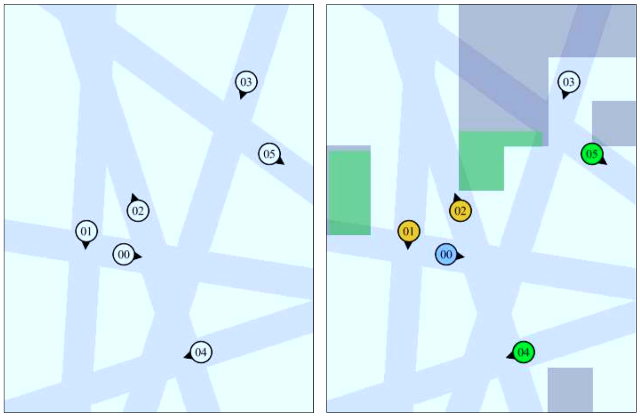

Fig. 12. Simulation run of a system of vehicles travelling along crossing paths. Vehicles are supposed to give way to vehicles coming from their right. In the figure, vehicle 0 is monitoring all vehicles that are in line–of–sight with it and reconstructs information about its unobservable regions.

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

[1] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: algorithms and theory,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.

[2] L. Figueiredo, I. Jesus, J. Machado, J. Ferreira, and J. Martins de Carvalho, “Towards the development of intelligent transportation systems,” *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 1206–1211, 2001.

[3] L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[4] C. Tomlin, G. J. Pappas, and S. Sastry, “Conflict resolution for air traffic management: A case study in multi-agent hybrid systems,” vol. 43, pp. 509–521, 1998.

[5] R. Ghosh and C. J. Tomlin, “Maneuver design for multiple aircraft conflict resolution,” Chicago, IL, 2000.

[6] L. Pallottino, V. Scordio, and A. Bicchi, “Decentralized cooperative conflict resolution among multiple autonomous mobile agents,” in *Proceedings of the Conference on Decision and Control*, vol. 5, Dec. 2004, pp. 4758–4763.

[7] T. Bass, “Intrusion detection systems and multisensor data fusion,” *Commun. ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[8] S. Snapp, J. Brentano, G. Dias, T. Goan, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, S. Smaha, T. Grance *et al.*, “DIDS (Distributed Intrusion Detection System)-Motivation, Architecture, and an Early Prototype,” *Proceedings of the 14th National Computer Security Conference*, pp. 167–176, 1991.

[9] S. Buchegger and J. Le Boudec, “Performance Analysis of the CONFIDANT Protocol: Cooperation Of NodesFairness In Dynamic Ad-hoc NeTworks,” *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, pp. 226–236, 2002.

[10] ——, “A Robust Reputation System for Mobile Ad-hoc Networks,” *Proceedings of P2PEcon, June*, 2004.

[11] T. Yoo and S. Lafortune, “Polynomial-time verification of diagnosability of partially observed discrete-event systems,” *Automatic Control, IEEE Transactions on*, vol. 47, no. 9, pp. 1491–1495, 2002.

[12] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Failure diagnosis using discrete-event models,” *Control Systems Technology, IEEE Transactions on*, vol. 4, no. 2, pp. 105–124, 1996.

[13] ——, “Diagnosability of discrete-event systems,” *Automatic Control, IEEE Transactions on*, vol. 40, no. 9, pp. 1555–1575, 1995.

[14] C. Özveren and A. Willsky, “Invertibility of Discrete-Event Dynamic Systems,” *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 5, no. 4, pp. 365–390, 1992.

[15] G. Fourlas, K. Kyriakopoulos, and N. Krikelis, “Diagnosability of Hybrid Systems,” *Proceedings of the 10th IEEE Mediterranean Conference on Control and Automation*, 2002.

[16] ——, “A Framework for Fault Detection of Hybrid Systems,” *Proceedings of the 9th IEEE Mediterranean Conference on Control and Automation*, 2001.

[17] A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, “Decentralized Intrusion Detection For Secure Cooperative Multi–Agent Systems,” *IEEE International Conference on Decision and Control*, 2007.

[18] ——, “Local Monitor Implementation for Decentralized Intrusion Detection in Secure Multi–Agent Systems,” *IEEE Conference on Automation, Science, and Engineering*, 2007.

[19] R. Olfati-Saber, J. A. Fax, and R. N. Murray, “Consensus and Cooperation in Networked Multi–Agent Systems,” *Proceedings of the IEEE*, 2007.

[20] R. Olfati-Saber, , and R. N. Murray, “Consensus Problems in Networks of Agents with Switching Topology and Time–Delays,” *IEEE Transactions on Automation and Control*, 2004.

[21] N. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, San Mateo, CA, 1996.

[22] M. Di Marco, A. Garulli, A. Giannitrapani, and A. Vicino, “Simultaneous localization and map building for a team of cooperating robots: a set membership approach,” *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 2, pp. 238–249, 2003.

[23] K. Marzullo, “Maintaining the time in a distributed system: An example of a loosely-coupled distributed service.” *Dissertation Abstracts International Part B: Science and Engineering[DISS. ABST. INT. PT. B- SCI. & ENG.]*,, vol. 46, no. 1, 1985.

[24] P. Lincoln and J. Rushby, “A Formally Verified Algorithm for Interactive Consistency Under a Hybrid Fault Model,” *Fault-Tolerant Computing, 1995,’Highlights from Twenty-Five Years’., Twenty-Fifth International Symposium on*, 1995.