



Distributed Fault-Tolerant Control for Networked Robots in the Presence of Recoverable/Unrecoverable Faults and Reactive Behaviors

Filippo Arrichiello^{1*}, Alessandro Marino² and Francesco Pierri³

¹ Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy,

² Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, Salerno, Italy,

³ School of Engineering, University of Basilicata, Potenza, Italy

OPEN ACCESS

Edited by:

Yongping Pan,
National University of Singapore,
Singapore

Reviewed by:

Qiuxuan Wu,
Hangzhou Dianzi University, China
Mohamed Hamdy,
Menoufia University, Egypt
Xiwang Dong,
Beihang University, China

*Correspondence:

Filippo Arrichiello
f.arrichiello@unicas.it

Specialty section:

This article was submitted to Robotic Control Systems, a section of the journal *Frontiers in Robotics and AI*

Received: 06 October 2016

Accepted: 17 January 2017

Published: 21 February 2017

Citation:

Arrichiello F, Marino A and Pierri F (2017) Distributed Fault-Tolerant Control for Networked Robots in the Presence of Recoverable/Unrecoverable Faults and Reactive Behaviors. *Front. Robot. AI* 4:2. doi: 10.3389/frobt.2017.00002

The paper presents an architecture for distributed control of multi-robot systems with an integrated fault detection, isolation, and recovery strategy. The proposed solution is based on a distributed observer-controller schema where each robot, by communicating only with its direct neighbors, is able to estimate the overall state of the system; such an estimate is then used by the controllers of each robot to achieve global missions as, for example, centroid and formation tracking. The information exchanged among the observers is also used to compute residual vectors that allow each robot to detect failures on anyone of the teammates, even if not in direct communication. The proposed strategy considers both recoverable and unrecoverable actuator faults as well as it properly manages the possible activation of reactive local control behaviors of the robots (e.g., the activation of obstacle avoidance strategy), which generate control inputs different from those required by the global mission control. In particular, when the robots are subject to recoverable faults, those are managed at a local level by computing a proper compensating control action. On the other side, when the robots are subject to unrecoverable faults, the faults are isolated from anyone of the teammates by means of a distributed fault detection and isolation strategy; then, the faulty robots are removed from the team and the mission is rearranged. The proposed strategy is validated via numerical simulations where the system properly identifies and manages the different cases of recoverable and unrecoverable actuator faults, as well as it manages the activation of local reactive control in an integrated case study.

Keywords: networked robots, multi-robot systems, fault-tolerant control, distributed control, fault detection

1. INTRODUCTION

Networked multi-robot systems pose challenging problems due to the need of integrating and controlling multiple entities to achieve a common objective. The integration among the robots, often performed via a communication network, allows the cooperation and the coordination of multiple, sometimes heterogeneous, nodes and it may allow to accomplish tasks otherwise impossible for a single robot.

Communication and control architectures for multi-robot systems may span from centralized to distributed architectures. In the first case, one node (e.g., one of the teammates) is able to

communicate with all the other nodes and to take decision for the overall team. However, limited communication capabilities, in term of ranges or bandwidth, and limited computational power could make impossible both the communication among all the robots and the adoption of a centralized control strategy. Moreover, the central unit may represent a weakness in terms of reliability since its failure may compromise the overall system.

In distributed systems [see, for example, Bullo et al. (2009) and Mesbahi and Egerstedt (2010)], each robot communicates only with a subset of its teammates and it must take its own decisions only on the base of the received information and of the data from local sensors. Distributed control strategies for multi-robot systems are often based on the so-called consensus task (Olfati-Saber et al., 2007) where, on the basis of only neighbor-to-neighbor information exchange, the robots are required to reach an agreement on a common value regarding a specific variable, exogenous or depending on the state of the single robot (Ren and Beard, 2008). The consensus problem has been mostly investigated for systems characterized by a first-order integrator (Olfati-Saber and Murray, 2004; Cao et al., 2008), or a second-order integrator (Hong et al., 2008; Yu et al., 2010), but some works involving systems with high-order dynamics are also present in the literature (Tuna, 2008; Wang et al., 2008). More in general, the problem of distributed control of multi-robot systems is aimed at achieving a global task (e.g., controlling the geometrical centroid and formation of a team of mobile robots) using only local information and interactions. In Belta and Kumar (2004), a partially decentralized algorithm is proposed to control the network centroid, the variance and the orientation of the system. Many approaches adopt decentralized observers in order to estimate the collective behavior, as in Smith and Hadaegh (2007) where a linear state-feedback control is proposed, or as in Yang et al. (2008) where such an estimate, in conjunction with a local controller, is adopted to minimize a cost function depending on the state of the whole system and aimed at controlling the robots' formation. An observer-controller scheme for the distributed tracking of time-varying global task variables has been developed in Antonelli et al. (2013, 2014). Time-varying formation control problems for general linear multiagent systems with switching directed topologies have been recently studied in Dong and Hu (2016).

The size of distributed robotic systems may be large and, consequently, the probability of failures of some of the teammates may increase. Despite distributed systems present a potential robustness to faults, the failure of one or more robots, if not properly handled, might jeopardize the task execution. Thus, fault detection and isolation (FDI) strategies become more and more important. Indeed, the detection of faulty robots may allow to exploit the intrinsic redundancy of multi-robot systems in such a way to accomplish the assigned tasks also in the case of failure of one or more units. To the aim, suitable fault accommodation strategies could be considered to correctly handle this situation. Several FDI strategies for single unit systems both in continuous (Zhang et al., 2002) or discrete-time (Caccavale et al., 2013) have been proposed in the literature, but only in the last years the fault diagnosis for multi-robot systems has been object of wider attention from the research community. In Wang et al. (2009), a centralized approach is presented where a central station,

aimed at detecting and isolating faults over the whole system, collects information about actuators and sensors coming from each robot. A comparison between a centralized (with a central unit collecting all the information), semi-decentralized (with information exchange only between neighboring robots), and fully decentralized (in which each robot only knows its own state) FDI approaches is presented in Meskin and Khorasani (2009). Observer-based decentralized solutions for detecting and isolating faults in interconnected subsystems are presented in Ferrari et al. (2009) and Zhang and Zhang (2012) where a bank of local adaptive observers, using only measurements and information from neighboring subsystems, are proposed. Observer-based approaches for overlapping linear systems can be found in Stankovic et al. (2010) where local observers are adopted to detect faults on the non-overlapping parts, while a consensus-like strategy is adopted for the overlapping parts.

Usually, distributed FDI schemes allow the detection of faults involving the same robot or its direct neighbors, namely, a direct communication between the faulty teammate and the detecting robot is required. For example, the work in Davoodi et al. (2014) presents a FDI approach for distributed and heterogeneous networked systems where each robot can detect and isolate not only its own faults but also the faults of its direct neighbors. Such limit is overcome in Arrichiello et al. (2015), in which we proposed a distributed FDI algorithm for networked robots where each robot is able to detect faults occurring on any other teammate, even if not in direct communication. In particular, the FDI solution in Arrichiello et al. (2015) builds on a distributed controller-observer schema presented in Antonelli et al. (2013, 2014) where, by means of a local observer, each robot, modeled with a continuous first-order integrator dynamics, estimates the overall state of the team and uses such an estimate to compute its local control input to achieve global tasks. Such approach is extended in Arrichiello et al. (2015) where the same information exchanged by the local observers is used to compute residual vectors whose aim is to allow the detection and the isolation of actuator faults occurring on any robot of the team. A recovery strategy aimed at removing the faulty robot from the team, based on the estimate of the maximum detection time, was developed and presented in Arrichiello et al. (2014a,b), while in Arrichiello et al. (2014c) we extended the strategy also to manage local reactive controls on board each robot to handle possible emerging situations as, for example, the presence of unexpected obstacles. In Marino and Pierri (2015) and Marino et al. (2015), we consider a multi-robot system characterized by a more general and discrete-time linear dynamics.

In this work, we extend the conference papers (Arrichiello et al., 2014a,b,c) by presenting a unified distributed fault-tolerant control (DFTC) approach composed of a distributed observer-controller strategy to achieve global mission, of a fault detection, identification, and recovery (FDIR) strategy to manage unrecoverable faults, of a local fault estimator to accommodate recoverable faults; moreover, the proposed architecture results robust to the activation of local behaviors not included in the global control strategy. Furthermore, with respect to those conference papers, here we provide further analytical details, we consider the case of noisy state measurements, and we improve the numerical validation by providing an integrated case study concerning

multi-robot formation control where recoverable/unrecoverable faults and activation of local reactive behavior occur.

1.1. Distributed Fault-Tolerant Control Architecture

The architecture of the Distributed Fault-Tolerant Control approach is depicted in **Figure 1**, and it consists of the following blocks:

- **Observer Block:** Each robot runs a consensus-based observer that uses only local information and a suitable vector received from the direct neighbors to estimate the state of the overall system.
- **Controller Block:** Each robot computes the control input to achieve a global mission on the base of a state-feedback control law that uses the local estimate of the overall system state performed by the observer block.
- **Local Supervisor Block:** A local supervisor might activate a reactive control to manage emergency situations, as obstacle avoidance, that are not taken into consideration by the global mission controller.
- **FDIR Block:** Using the same information exchanged by the observer block for the overall state estimation, each robot computes a set of residual vectors that are sensitive to faults affecting anyone of the robots, even if not in direct communication. The residual vectors are used to detect and isolate faulty robots by considering a set of adaptive thresholds, defined on the basis of the residual dynamics. After the faulty robots have been detected and isolated, they are removed by the team referring to a decentralized recovery policy based on the estimate of the maximum detection time.
- **Local Fault Estimator Block:** Each robot runs a local estimator to detect faults and, possibly, to try accommodate them at a local level.

The paper is organized as follows. Section 2 summarizes some basic notation of the vectors used in the rest of the paper. Section 3 introduces the modeling of the single robot, of the collective system, and of the intra-robots communication. Section 4 presents

the decentralized observer-controller scheme, and it discusses the convergence analysis of the observer errors. Section 5 describes the fault detection and isolation algorithm, and the recovery strategy. Section 6 shows the results of numerical simulations and, finally, Section 7 presents some conclusions and future works.

2. NOTATION

Let us introduce the some basic notation adopted throughout the rest of the paper. The generic variable relative to the i -th robot will be denoted as v_i , while the corresponding collective variable, i.e., a stacked vector collecting the variables v_i for all the robots, will be denoted without any subscript/superscript; namely, $v = [v_1^T v_2^T \dots v_N^T]^T$ where N is the number of robots. The estimate of collective variable v locally made by the i -th robot is denoted as ${}^i\hat{v}$, while the collection of the estimation vectors of the different robots is denoted as $\hat{v}^* = [{}^1\hat{v}^T {}^2\hat{v}^T \dots {}^N\hat{v}^T]^T$. The estimation error performed by i -th robot is denoted as ${}^i\tilde{v} = v - {}^i\hat{v}$, while the collection of the estimation errors of the different robots is expressed as $\tilde{v}^* = [{}^1\tilde{v}^T {}^2\tilde{v}^T \dots {}^N\tilde{v}^T]^T$. Finally, the vector v^* represents $v^* = \mathbf{1}_N \otimes v$ where $\mathbf{1}_N$ is the $(N \times 1)$ vector of ones and \otimes denotes the Kroneker product operator.

3. MODELING

Let us consider a multi-robot system composed of N robots, each of them characterized by the following dynamics ($i = 1, 2, \dots, N$)

$$\dot{x}_i = u_i + \phi_i(t) + \xi_i(t, x_i), \tag{1}$$

where $x_i \in \mathfrak{R}^n$ and $u_i \in \mathfrak{R}^n$ denote, respectively, the i -th robot's state and input, $\phi_i \in \mathfrak{R}^n$ is an additive actuator fault term that is zero in healthy conditions and $\xi_i(t, x_i) \in \mathfrak{R}^n$ is a term collecting the model uncertainties and disturbances. The following assumptions on the single robot's model hold:

ASSUMPTION 1. *The model uncertainties $\xi_i(t, x_i)$ are assumed to be norm bounded by a known constant term $\bar{\xi}$, assumed equal for each robot, i.e.,*

$$\|\xi_i(t, x_i)\| \leq \bar{\xi}, \forall t, \forall i = 1, 2, \dots, N. \tag{2}$$

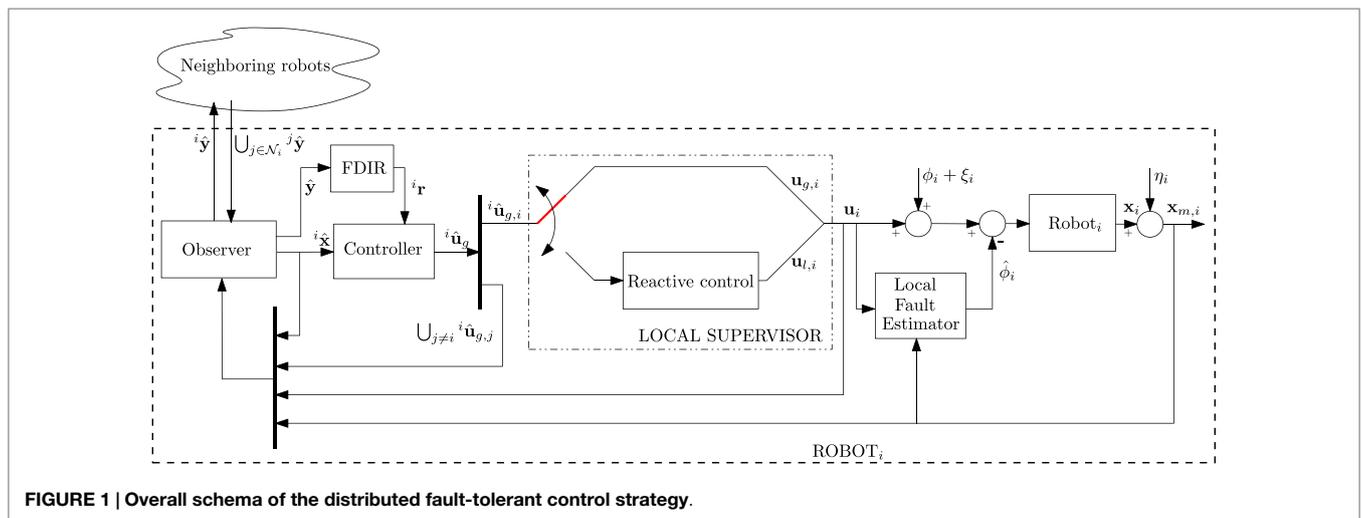


FIGURE 1 | Overall schema of the distributed fault-tolerant control strategy.

ASSUMPTION 2. Each robot has direct access only to a noisy measure $x_{m,i}$ of its own state, i.e.,

$$x_{m,i} = x_i + \eta_i < +\infty, \quad (3)$$

where $\eta_i \in \mathbb{R}^n$ is the measurement additive noise supposed to be norm bounded by a known positive scalar $\bar{\eta}$, assumed equal for each robot, i.e.,

$$\|\eta_i(t)\| \leq \bar{\eta} < +\infty \quad \forall t, \quad \forall i = 1, 2, \dots, N. \quad (4)$$

Let us define the collective variables of the system as the vectors collecting the variables of all the robots of the network, as

$$\begin{aligned} x &= [x_1^T \ x_2^T \ \dots \ x_N^T]^T \in \mathbb{R}^{Nn} \\ u &= [u_1^T \ u_2^T \ \dots \ u_N^T]^T \in \mathbb{R}^{Nn} \\ \phi &= [\phi_1^T \ \phi_2^T \ \dots \ \phi_N^T]^T \in \mathbb{R}^{Nn} \\ \xi &= [\xi_1^T \ \xi_2^T \ \dots \ \xi_N^T]^T \in \mathbb{R}^{Nn} \\ \eta &= [\eta_1^T \ \eta_2^T \ \dots \ \eta_N^T]^T \in \mathbb{R}^{Nn}. \end{aligned} \quad (5)$$

On the basis of the single robot dynamics [equation (1)], the collective dynamics is expressed as

$$\dot{x} = u + \phi + \xi, \quad (6)$$

while the collective noisy measurement of the system state is

$$x_m = x + \eta. \quad (7)$$

3.1. Communication Modeling

As common in multi-robot systems [see, e.g., Godsil and Royle (2001), Fax and Murray (2004), and Olfati-Saber and Murray (2004)], the information exchange among the robots can be modeled referring to graph theory approaches. Indeed, the robot network can be described as a graph $\mathcal{G}(\mathcal{E}, \mathcal{V})$ characterized by its topology, where \mathcal{V} is a set of indexes labeling the N vertices (nodes) representing the robots, and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is the set of edges (arcs) representing the communications between robots; i.e., robot i and robot j can communicate only if there is an arc between node i and node j . Thus, the i -th robot receives information only from its neighbors $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. The cardinality of \mathcal{N}_i is the in-degree of node i , i.e., $d_i = |\mathcal{N}_i| = \sum_{j=1}^N a_{ij}$.

Moreover, the cardinality of the set of nodes receiving information from node i represents the out-degree of node i , i.e., $D_i = \sum_{k=1}^N a_{ki}$.

If all the communication links among the robots are bi-directional, the graph is called *undirected* (i.e., $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$) otherwise, the graph is called *directed*. An undirected graph is called *connected* if there is an undirected path between every pair of distinct nodes. A directed graph is called *strongly connected* if any two distinct nodes of the graph can be connected via a directed path, i.e., a path that follows the direction of the edges of the graph. A node of a directed graph is named *balanced* if its in-degree and its out-degree are equal; a directed graph is called *balanced*

if each node of the graph is balanced. Any undirected graph is balanced.

From the mathematical point of view, the graph topology can be represented by the $(N \times N)$ adjacency matrix

$$A = \{a_{ij}\} : a_{ii} = 0, a_{ij} = \begin{cases} 1 & \text{if } (j, i) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases}$$

where $a_{ij} = 1$ if there exists an arc from vertex j to vertex i . Adjacency matrix is symmetric in the case of undirect graph. The communication topology can be also characterized by the $(N \times N)$ Laplacian matrix (Ren and Beard, 2008; Mesbahi and Egerstedt, 2010) defined as

$$L = \{l_{ij}\} : l_{ii} = \sum_{j=1, j \neq i}^N a_{ij}, \quad l_{ij} = -a_{ij}, \quad i \neq j.$$

The Laplacian matrix has some useful properties that make it very popular in the field of multi-robot systems and that will be exploited in the following paper. More in detail, all the eigenvalues of the matrix L have real parts equal to or greater than zero (i.e., $\text{Re}(\lambda_i) \geq 0, \forall i$); moreover, L exhibits at least one zero eigenvalue with the $N \times 1$ vector of all ones $\mathbf{1}_N$ as the corresponding right eigenvector. It holds that $\text{rank}(L) \leq N - 1$, in particular it results $\text{rank}(L) = N - 1$ if and only if the directed graph is strongly connected. The N eigenvalues of L can be ordered such that

$$0 = \lambda_1 \leq \text{Re}(\lambda_2) \leq \dots \leq \text{Re}(\lambda_N);$$

thus, $\text{Re}(\lambda_2)$ is greater than zero if and only if the graph is strongly connected; for this reason, λ_2 is defined as the algebraic connectivity of the graph. For a balanced directed graph (and, thus, for an undirected graph), $\mathbf{1}_N$ is also a left eigenvector of L , i.e., $\mathbf{1}_N^T L = \mathbf{0}_N^T$. Finally, if the graph is undirected, the Laplacian is symmetric and positive semidefinite.

The following assumptions about the communication hold:

- each robot does not know the topology of the overall communication graph but it only knows who are its neighbors.
- the graph topology is assumed fixed, i.e., there are no communication links that can appear or disappear over the time.

4. DECENTRALIZED OBSERVER-CONTROLLER SCHEME

The robots are required to achieve a *global* mission, i.e., a mission depending on the collective state x of the overall team. However, since the collective state measurement is not available to the robots, a distributed observer-controller scheme is adopted in which, for each robot, a local observer provides a local estimate of the collective state, \hat{x} ; then, the controller computes the robot input in a pseudo-centralized manner on the base of the estimated collective state (Antonelli et al., 2013, 2014; Arrichiello et al., 2015).

In addition to the global mission, each robot must achieve also *local* (reactive) tasks to take into account unexpected events such as the presence of obstacles along its path. The switching between global behavior and reactive behavior is handled by a *Local Supervisor* according to the current scenario.

4.1. Nominal Control Input

The control input of the i -th robot has the following form

$$\mathbf{u}_i = \alpha_{i,0} \mathbf{u}_{g,i} + \sum_{k=1}^{m_i} \alpha_{i,k} \mathbf{u}_{l,i,k} = \alpha_{i,0} \mathbf{u}_{g,i} + \mathbf{u}_{l,i}, \quad (8)$$

where $\mathbf{u}_{g,i}(\hat{\mathbf{x}})$ is the control input that allows to achieve the global task, depending on the estimate of the collective input, while $\mathbf{u}_{l,i,k}(\mathbf{x}_i, \hat{\mathbf{x}})$ ($k=1, \dots, m_i$) are a set of m_i local reactive control inputs, and $\mathbf{u}_{l,i} = \sum_{k=1}^{m_i} \alpha_{i,k} \mathbf{u}_{l,i,k}$. The parameters $\alpha_{i,h}$ ($h=0, \dots, m_i$) are scalar gains such that:

- $\alpha_{i,k} = 0$ or $1, \forall t > 0, \forall k = 0, 1, \dots, m_i$;
- $\sum_{k=0}^{m_i} \alpha_{i,k} = 1$;

in other words, at each time instant only one of them is 1 and all the others are equal to 0. Thus, the input terms on the right side of equation (8) are mutually exclusive control inputs, and each robot selects the value of gains $\alpha_{i,h}$ on the basis of its own state and of the external environment.

4.1.1. The Global Mission Controller

In the proposed approach, the global objective for the team of robots is expressed via a task function $\boldsymbol{\sigma} \in \mathbb{R}^p$ depending on the collective state as

$$\boldsymbol{\sigma} = \mathbf{J}\mathbf{x}, \quad (9)$$

where $\mathbf{J} \in \mathbb{R}^{p \times Nn}$ is a constant matrix; thus,

$$\dot{\boldsymbol{\sigma}} = \mathbf{J}\dot{\mathbf{x}}. \quad (10)$$

A common example of global tasks of this form is that of assigning desired time-varying references to the team centroid and the relative formation. In this case, the task function becomes

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1 \\ \boldsymbol{\sigma}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix} \mathbf{x} = \mathbf{J}\mathbf{x}, \quad \dot{\boldsymbol{\sigma}} = \mathbf{J}\dot{\mathbf{x}}, \quad (11)$$

where $\mathbf{J} \in \mathbb{R}^{Nn \times Nn}$, $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are the centroid and formation task functions defined as:

$$\boldsymbol{\sigma}_1(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i = \mathbf{J}_1 \mathbf{x}, \quad (12)$$

with $\mathbf{J}_1 \in \mathbb{R}^{n \times Nn}$ the corresponding Jacobian matrix (Antonelli et al., 2013);

$$\boldsymbol{\sigma}_2(\mathbf{x}) = \left[(\mathbf{x}_2 - \mathbf{x}_1)^T \dots (\mathbf{x}_N - \mathbf{x}_{N-1})^T \right]^T = \mathbf{J}_2 \mathbf{x}, \quad (13)$$

with $\mathbf{J}_2 \in \mathbb{R}^{(N-1)n \times Nn}$ the corresponding Jacobian matrix (Antonelli et al., 2013). A centralized solution for such kind of global task can be obtained by computing the global collective input \mathbf{u}_g as

$$\mathbf{u}_g = \mathbf{u}(t, \mathbf{x}) = \mathbf{J}^\dagger [\dot{\boldsymbol{\sigma}}_d + k_c \tilde{\boldsymbol{\sigma}}(\mathbf{x})], \quad (14)$$

where $\boldsymbol{\sigma}_d$ is the desired value of the task function, $\tilde{\boldsymbol{\sigma}}(\mathbf{x}) = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}(\mathbf{x})$ is the task error, \mathbf{J}^\dagger is the pseudo-inverse of the Jacobian matrix, and k_c is a constant positive gain.

In a distributed scenario, since the single robots do not know the collective state of the system, each of them uses its collective state estimate ${}^i \hat{\mathbf{x}}$ in order to compute an estimate of the collective input as

$${}^i \hat{\mathbf{u}}_g = \mathbf{u}(t, {}^i \hat{\mathbf{x}}) = \mathbf{J}^\dagger \left[\dot{\boldsymbol{\sigma}}_d + k_c {}^i \tilde{\boldsymbol{\sigma}}({}^i \hat{\mathbf{x}}) \right], \quad (15)$$

where ${}^i \tilde{\boldsymbol{\sigma}}({}^i \hat{\mathbf{x}}) = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}({}^i \hat{\mathbf{x}})$ is the estimate of the task error.

Then, the robot i computes its own input $\mathbf{u}_{g,i}({}^i \hat{\mathbf{x}})$ in equation (8) selecting the relative component from ${}^i \hat{\mathbf{u}}_g$, i.e.,

$$\mathbf{u}_{g,i} = \boldsymbol{\Gamma}_i {}^i \hat{\mathbf{u}}_g, \quad (16)$$

where $\boldsymbol{\Gamma}_i$ is the $(n \times Nn)$ selection matrix that extracts the components of the i -th robot from a collective vector

$$\boldsymbol{\Gamma}_i = \left\{ \begin{array}{cccc} \mathbf{O}_n & \cdots & \underbrace{\mathbf{I}_n}_{i\text{th node}} & \cdots & \mathbf{O}_n \end{array} \right\}.$$

4.1.2. The Local Reactive Behavior

A part from collaborating with the teammates to achieve the global mission, each robot is responsible of its own safety. This means that it can activate, if necessary, reactive behaviors, e.g., to prevent obstacles along its path. Reactive behaviors are local control inputs that are activated and deactivated by a local supervisor depending only on the robot own state, \mathbf{x}_i , and on information from onboard sensors (e.g., cameras, rangefinders, etc.). As an example, in the presence of an obstacle in the advancing direction, the goal of the obstacle avoidance reactive task is that of computing an input $\mathbf{u}_{l,1}$ aimed at keeping the robot at a safe distance, $d > 0$, from the obstacle, whose position, $\mathbf{x}_o \in \mathbb{R}^n$, can be obtained by onboard sensors. Therefore, the task function can be written as

$$\sigma_{l,1} = \|\mathbf{x}_i - \mathbf{x}_o\| \in \mathbb{R}, \quad \sigma_{l,1,d} = d, \quad \mathbf{J}_{l,1} = \hat{\mathbf{r}}^T \in \mathbb{R}^{1 \times n},$$

where $\mathbf{J}_{l,1}$ is the task Jacobian matrix, and

$$\hat{\mathbf{r}} = \frac{\mathbf{x}_i - \mathbf{x}_o}{\|\mathbf{x}_i - \mathbf{x}_o\|}$$

is the unit vector aligned with the obstacle-to-vehicle direction. According to null space-based behavioral control approach presented in Antonelli et al. (2008), the obstacle avoidance is obtained by means of the following expression of $\mathbf{u}_{l,1}$

$$\mathbf{u}_{l,1}(\mathbf{x}_i, \mathbf{x}_o) = \mathbf{J}_{l,1}^\dagger \theta_1 (d - \|\mathbf{x}_i - \mathbf{x}_o\|), \quad (17)$$

where θ_1 is a positive scalar gain. Thus, when an obstacle is detected by robot i and it is closer than a threshold value, the local supervisor activates the obstacle avoidance behavior (i.e., according to equation (8), it sets $\alpha_{i,1} = 1$ and $\alpha_{i,0} = 0$). Hence, the input to the robot i would become $\mathbf{u}_i = \mathbf{u}_{l,1}$. However, since by adopting the control input [equation (17)], the global task would be totally neglected, the local input can be improved by adding the global task as secondary task, in such a way the robot still tries

to achieve it while avoiding the obstacle (primary task) (Antonelli et al., 2008). Thus, equation (17) can be replaced by

$$\mathbf{u}_{l,1}(\mathbf{x}_i, \hat{\mathbf{x}}, \mathbf{x}_o) = J_{l,1}^\dagger \theta_1 (d - \|\mathbf{x}_i - \mathbf{x}_o\|) + N(J_{l,1}) \mathbf{u}_{g,i} \quad (18)$$

where the global input vector $\mathbf{u}_{g,1}(\hat{\mathbf{x}})$ is projected into the null space of matrix $J_{l,1}$ by means of matrix

$$N(J_{l,1}) = \mathbf{I}_n - J_{l,1}^\dagger J_{l,1} = \mathbf{I}_n - \hat{\mathbf{r}} \hat{\mathbf{r}}^\top.$$

4.2. State Observer

As seen in Section 4.1.1, to compute the global input $\mathbf{u}_{g,i}$, each robot needs an estimate of the collective state. To this aim, the decentralized observer first proposed in Arrichiello et al. (2015) and then modified in Arrichiello et al. (2014c) can be adopted. In particular, the observer on board the generic robot i -th computes the estimate of the collective state \mathbf{x} as follows

$$\dot{\hat{\mathbf{x}}} = k_o \left(\sum_{j \in \mathcal{N}_i} ({}^j \hat{\mathbf{y}} - {}^i \hat{\mathbf{y}}) + \mathbf{\Pi}_i (\mathbf{y}_m - {}^i \hat{\mathbf{y}}) \right) + {}^i \hat{\mathbf{u}}_g + \mathbf{\Gamma}_i^\top \Delta \mathbf{u}_i, \quad (19)$$

where \mathbf{y} is an auxiliary variable defined as

$$\mathbf{y} = \mathbf{x} - \int_{t_0}^t \mathbf{u}(\tau) d\tau, \quad (20)$$

that results useful both for state estimation and for diagnosis purposes discussed in the following section; k_o is a positive scalar gain; $\mathbf{\Pi}_i$ is the $(Nn \times Nn)$ matrix defined as $\mathbf{\Pi}_i = \mathbf{\Gamma}_i^\top \mathbf{\Gamma}_i$, that selects only the components of the i -th robot;

$$\begin{cases} \mathbf{y}_m &= \mathbf{x}_m - \int_{t_0}^t \mathbf{u}(\tau) d\tau \\ {}^i \hat{\mathbf{y}} &= {}^i \hat{\mathbf{x}} - \int_{t_0}^t ({}^i \hat{\mathbf{u}}_g(\tau) + \mathbf{\Gamma}_i^\top \Delta \mathbf{u}_i) d\tau \end{cases} \quad (21)$$

where t_0 is the initial time instant; $\Delta \mathbf{u}_i \in \mathfrak{R}^n$ takes into account the possibility to activate the reactive local control and it is given by

$$\Delta \mathbf{u}_i = (\alpha_{i,0} - 1) \mathbf{u}_{g,i} + \sum_{k=1}^{m_i} \alpha_{i,k} \mathbf{u}_{l,k,i}. \quad (22)$$

REMARK 4.1. It is worth highlighting that the definition of \mathbf{y}_m makes use of the real input [equation (16)] (global or reactive), while the estimate ${}^i \hat{\mathbf{y}}$ takes into account only the estimated global input ${}^i \hat{\mathbf{u}}_g$ for robots different from i -th and, thanks to the additive term $\Delta \mathbf{u}_i$, the real input for the i -th robot. In this way, the reactive behaviors cannot be confused with faults by other robots in the team, despite they affect the global mission.

Moreover, the selection matrix $\mathbf{\Pi}_i$ allows to select only the i -th components from global variables as \mathbf{y} . This ensures that equation (19) can be implemented on a distributed system since each robot only uses variable components locally available.

Let us define the collective estimated variables

$$\begin{aligned} \hat{\mathbf{x}}^* &= [{}^1 \hat{\mathbf{x}}^\top \ {}^2 \hat{\mathbf{x}}^\top \ \dots \ {}^N \hat{\mathbf{x}}^\top]^\top \in \mathfrak{R}^{N^2 n} \\ \hat{\mathbf{y}}^* &= [{}^1 \hat{\mathbf{y}}^\top \ {}^2 \hat{\mathbf{y}}^\top \ \dots \ {}^N \hat{\mathbf{y}}^\top]^\top \in \mathfrak{R}^{N^2 n} \\ \hat{\mathbf{u}}_g^* &= [{}^1 \hat{\mathbf{u}}_g(t, {}^1 \hat{\mathbf{x}}) \ {}^1 \hat{\mathbf{u}}_g(t, {}^2 \hat{\mathbf{x}}) \ \dots \ {}^N \hat{\mathbf{u}}_g(t, {}^N \hat{\mathbf{x}})] \in \mathfrak{R}^{N^2 n}, \end{aligned} \quad (23)$$

and the collective vectors

$$\begin{aligned} \mathbf{x}^* &= \mathbf{1}_N \otimes \mathbf{x} \in \mathfrak{R}^{N^2 n} \\ \mathbf{y}^* &= \mathbf{1}_N \otimes \mathbf{y} \in \mathfrak{R}^{N^2 n} \\ \mathbf{u}^* &= \mathbf{1}_N \otimes \mathbf{u} \in \mathfrak{R}^{N^2 n} \\ \boldsymbol{\eta}^* &= \mathbf{1}_N \otimes \boldsymbol{\eta} \in \mathfrak{R}^{N^2 n} \\ \boldsymbol{\xi}^* &= \mathbf{1}_N \otimes \boldsymbol{\xi} \in \mathfrak{R}^{N^2 n} \\ \boldsymbol{\phi}^* &= \mathbf{1}_N \otimes \boldsymbol{\phi} \in \mathfrak{R}^{N^2 n} \end{aligned}$$

$$\Delta \mathbf{u}^* = [(\mathbf{\Gamma}_1^\top \Delta \mathbf{u}_1)^\top \ (\mathbf{\Gamma}_2^\top \Delta \mathbf{u}_2)^\top \ \dots \ (\mathbf{\Gamma}_N^\top \Delta \mathbf{u}_N)^\top]^\top \in \mathfrak{R}^{N^2 n}, \quad (24)$$

where \otimes denotes the Kronecker product operator. By taking into account equations (19), (23), and (24), the collective state estimation dynamics is

$$\dot{\tilde{\mathbf{x}}}^* = -k_o L^* \tilde{\mathbf{y}}^* + k_o \mathbf{\Pi}^* \tilde{\mathbf{y}}^* + k_o \mathbf{\Pi}^* \boldsymbol{\eta}^* + \hat{\mathbf{u}}_g^* + \Delta \mathbf{u}^* + \boldsymbol{\xi}^* + \boldsymbol{\phi}^*, \quad (25)$$

where $L^* = L \otimes \mathbf{I}_{Nn}$, $\tilde{\mathbf{y}}^* = \mathbf{y}^* - \hat{\mathbf{y}}^*$ and

$$\mathbf{\Pi}^* = \text{diag} \{[\mathbf{\Pi}_1 \ \dots \ \mathbf{\Pi}_N]\}. \quad (26)$$

The dynamics of the collective state estimation error $\tilde{\mathbf{x}}^* = \mathbf{x}^* - \hat{\mathbf{x}}^*$ is given by

$$\dot{\tilde{\mathbf{x}}}^* = -k_o L^* \tilde{\mathbf{y}}^* + (\mathbf{u}^* - \mathbf{u}_g^*) + \boldsymbol{\xi}^* + \boldsymbol{\phi}^* - \Delta \mathbf{u}^*. \quad (27)$$

In the previous works (Arrichiello et al., 2014c, 2015), it has been proven that with the observer [equation (25)], in the absence of faults, measurement noise, and model uncertainties, the collective state estimation error $\tilde{\mathbf{x}}^*$ exponentially converges to the origin if no robots are adopting a reactive control (i.e., $\alpha_{i,k} = 0, \forall k \neq 0$ in equation (8) and $\forall i$).

To prove the convergence of $\tilde{\mathbf{x}}^*$, the exponential convergence to the origin of $\tilde{\mathbf{y}}^*$ is first proven by the following theorem

THEOREM 1. In the presence of a strongly connected directed communication graph (or connected undirected graph) and in the absence of faults ($\boldsymbol{\phi} = \mathbf{0}$), measurement noise, and model uncertainties ($\boldsymbol{\eta} = \boldsymbol{\xi} = \mathbf{0}$), $\tilde{\mathbf{y}}^*$ is exponentially convergent to the origin with the observer [equation (25)], independent from the local control input, \mathbf{u}_i ($i = 1, \dots, N$), applied by the robots. Furthermore, in the presence of bounded measurement noise and model uncertainty, $\tilde{\mathbf{y}}^*$ is globally uniformly ultimately bounded.

Then, based on the previous theorem, also the convergence of $\tilde{\mathbf{x}}^*$ in the absence of reactive behaviors is stated by the following.

THEOREM 2. In the presence of a strongly connected directed communication graph (or connected undirected graph) and in the absence of faults ($\boldsymbol{\phi} = \mathbf{0}$), measurement noise, and model uncertainties ($\boldsymbol{\eta} = \boldsymbol{\xi} = \mathbf{0}$), with the observer in [equation (25)], the stacked vector of the collective state estimation errors $\tilde{\mathbf{x}}^*$ is exponentially convergent to the origin. Moreover, in the presence of bounded measurement noise and model uncertainty, $\tilde{\mathbf{x}}^*$ is globally uniformly ultimately bounded.

The complete proofs of both the theorems can be found in Arrichiello et al. (2014c) and Arrichiello et al. (2015) and they are not reported here for the sake of brevity.

REMARK 4.2. Different from $\tilde{\mathbf{y}}^*$, in the presence of reactive behaviors, the convergence of $\tilde{\mathbf{x}}^*$ to zero cannot be guaranteed even in the absence of faults and disturbances, since $(\mathbf{u}^* - \mathbf{u}_g^*) - \Delta \mathbf{u}^*$ does not converge to zero.

5. FAULT TOLERANT STRATEGY

In this section, we present the proposed strategy to handle recoverable and unrecoverable actuator faults occurring on any robot of the team. In the presence of a fault, the first step is to decide if the robot objective can be achieved in spite of its occurrence by opportunely reconfiguring the control input. In this case, it is possible to classify the fault as *recoverable* (Staroswiecki, 2008) (e.g., a bias between the commanded input and the actual one), and active fault-tolerant strategies can be *locally* activated by single robots, that locally modifies the nominal control law [equation (16)] so as to accommodate the fault by compensating the effects of the fault on its dynamics. Regarding *non-recoverable* faults (e.g., permanent actuator failure) active strategies are not available, but in multi-robot system the intrinsic redundancy can be exploited in order to define a passive fault tolerance strategy aimed at removing the faulty vehicle from the team and rescheduling the mission needs.

5.1. Recoverable Faults: Local Fault Estimate and Accommodation

In the presence of recoverable faults, an active fault tolerance approach is pursued, i.e., the nominal control law [equation (8)] is reconfigured by adding a compensation term, $\hat{\phi}_i$. Thus, the new input vector \bar{u}_i , given by

$$\bar{u}_i = u_i - \hat{\phi}_i, \quad (28)$$

is adopted instead of equation (8). In equation (28), u_i has the same form than in equation (8), while the adaptive term $\hat{\phi}_i$ provides an estimate of the fault, in such a way to compensate its effect on the robot dynamics.

In order to estimate the fault ϕ_i , a local state observer is designed as in Arrichiello et al. (2014b) and Zhang et al. (2013)

$$\dot{z}_i = u_i + K(x_{m,i} - z_i), \quad (29)$$

where $K \in \mathbb{R}^{n \times n}$ is a symmetric and positive definite gain matrix. The update law for the fault estimation is given by

$$\dot{\hat{\phi}}_i(t) = G^{-1} (\dot{e}_i(t) + Ke_i(t)), \quad (30)$$

where $e_i(t) = x_i - z_i$ and $G \in \mathbb{R}^{n \times n}$ is a symmetric positive matrix gain to be designed.

REMARK 5.1. From equation (30), it can be recognized that the on-line fault estimator has the following expression

$$\hat{\phi}_i(t) = G^{-1} \left(e_i(t) + K \int_{t_{f,i}}^t e_i(\tau) d\tau \right), \quad (31)$$

where $t_{f,i}$ is the fault occurrence instant. The integral term allows to achieve an asymptotic estimate of a constant fault while, in the presence of time-varying faults, it can be proven that the estimation error is bounded.

From the robot dynamics [equation (1)] and the new control law [equation (28)] with the update law [equation (30)], it holds

$$\dot{e}_i(t) = -Ke_i + \phi_i - \dot{\hat{\phi}}_i - K\eta_i + \xi_i = -Ke_i + \tilde{\phi}_i - K\eta_i + \xi_i, \quad (32)$$

where $\tilde{\phi}_i = \phi_i - \hat{\phi}_i$ is the fault estimation error. In the absence of faults, it can be easily seen that, in the absence of measurement noise η_i and disturbances ξ_i , the estimate $\hat{\phi}_i$ is zero, provided that the initial observer error is zero. On the other side, in the presence of measurement noise and disturbances, the estimate $\hat{\phi}_i$ is different from zero even in the absence of faults and converges to $K\eta_i - \xi_i$.

By introducing the collective vectors of errors $e = [e_1^T, \dots, e_N^T]^T \in \mathbb{R}^{Nn}$ and fault estimations $\tilde{\phi} = [\tilde{\phi}_1^T, \dots, \tilde{\phi}_N^T]^T \in \mathbb{R}^{Nn}$, from equations (30) and (32) it holds

$$\dot{e} = -(I_N \otimes K)e + \tilde{\phi} - (I_N \otimes K)\eta + I_N \otimes \xi, \quad (33)$$

$$\dot{\tilde{\phi}} = \dot{\hat{\phi}} - I_N \otimes G^{-1} (\dot{e} + e). \quad (34)$$

As reported in Section 4.2, the estimation error of the collective state coming from observer [equation (25)], without local fault estimation [equation (30)], is exponential convergent to zero in the absence of faults, disturbances, and measurement noise and under the assumption of a connected undirected graph, while in the presence of bounded disturbances and measurement noise, \tilde{x}^* is globally uniformly ultimately bounded (Arrichiello et al., 2014b, 2015).

In the presence of a recoverable fault and the adaptive fault observer [equation (29)] with the update law [equation (30)], it has been proven in Arrichiello et al. (2014b) that, under the assumptions of connected undirected communication graph and fault velocity bounded with a known bound, the estimation errors \tilde{y}^* , \tilde{x}^* , and $\tilde{\phi}$ are all uniformly ultimately bounded, as well as the task-tracking errors $\tilde{\sigma}_l$ ($l=1, 2$). Here, the proof is not reported for the sake of brevity.

5.2. Non-Recoverable Faults

In the presence of non-recoverable faults, the intrinsic redundancy of a multi-robot system is exploited in order to accomplish the mission in spite of the faulty teammates. To this aim, a passive fault tolerance strategy is designed aimed at removing the faulty robots from the team, and at allowing the remaining vehicles to reconfigure themselves. To this purpose, a two-stage procedure is designed: at first, a fault detection and isolation strategy, developed as in Arrichiello et al. (2015) and reported in the following for the sake of completeness, is adopted to make each robot able to detect the occurrence of a fault and to recognize the faulty vehicle even if not in direct communication; then, the faulty vehicle is removed from the team by suitably resizing the exchanged vectors.

5.2.1. Fault Detection and Isolation

In order to detect the occurrence of a fault and isolate the faulty vehicle, let us define the following residual vector for the i -th robot (Arrichiello et al., 2015)

$${}^i r = \sum_{j \in \mathcal{N}_i} ({}^j \hat{y} - {}^i \hat{y}) + \Pi_i(y_m - \hat{y}_i); \quad (35)$$

it is worth noticing that the computation of the above quantity does not require additional information exchange since it

makes use of the same terms used by the state observer [equation (19)]. The vector ${}^i r$ can be seen as a stacked vector, i.e., ${}^i r = [{}^i r_1^T \ {}^i r_2^T \ \dots \ {}^i r_N^T]^T \in \mathfrak{R}^{Nn}$, where each component ${}^i r_k \in \mathfrak{R}^n$ represents the residual computed by robot i relative to robot k and allows the robot i to monitor the healthy state of robot k . After some algebraic manipulations [see Arrichiello et al. (2015) for details], it is possible to rewrite the residual component ${}^i r_k$ as the sum of two contributions: the first term, ${}^i h_k$, depending on the observer dynamics and on the presence of noise and disturbances; the second term, ${}^i f_k$, depending on the fault ϕ_k , i.e.,

$${}^i r_k = {}^i h_k + {}^i f_k, \quad (36)$$

with

$${}^i h_k = \Gamma_i \Pi_k \eta + \Gamma_i \tilde{L}_k^* \left[e^{-k_o \tilde{L}_k^* t} \tilde{y}_k^*(0) + \int_0^t e^{-k_o \tilde{L}_k^* (t-\tau)} (\mathbf{1}_N \otimes \xi_k(\tau) - k_o \Pi_k \eta(\tau)) d\tau \right] \quad (37)$$

$${}^i f_k = \Gamma_i \tilde{L}_k^* \int_0^t e^{-k_o \tilde{L}_k^* (t-\tau)} (\mathbf{1}_N \otimes \phi_k(\tau)) d\tau, \quad (38)$$

where vector $\tilde{y}_k^* = [{}^1 \tilde{y}_k^T \ \dots \ {}^N \tilde{y}_k^T]^T \in \mathfrak{R}^{Nn}$ collects the estimation errors ${}^i \tilde{y}_k$ of all the observers, and $\tilde{L}_k^* = L \otimes I_n + \Pi_k$.

From equations (37) to (38), it can be argued that each residual component ${}^i r_k$, $i = 1, 2, \dots, N$, is sensitive only to fault occurring on robot k and is insensitive to all the other faults [see Arrichiello et al. (2015) for details]. Thus, a fault ϕ_k affecting the robot k is detected and isolated by the robot i if

$$\left\{ \begin{array}{l} \exists t > t_f : \quad ||{}^i r_k(t)|| > {}^i \mu_k(t) \\ \forall l \in (1, 2, \dots, N), l \neq k, \forall t > 0, \quad ||{}^i r_l(t)|| \leq {}^i \mu_l(t) \end{array} \right. \quad (39)$$

where $t_f > 0$ is the instant at which the fault occurs and ${}^i \mu_j(t)$ ($\forall i, j$) are suitable adaptive thresholds. The use of thresholds is necessary to avoid false alarms since, due to the measurement noise and model uncertainties, the residuals are always different from zero even in the absence of faults. In Arrichiello et al. (2015), the following expression for ${}^i \mu_k$ has been derived

$${}^i \mu_k(t) = \left(\sqrt{n} d_i + {}^i \delta_k \right) \left(||\tilde{y}_k^*(0)|| \kappa e^{-\lambda t} + \frac{\kappa (k_o \bar{\eta} + \sqrt{N} \bar{\xi})}{\lambda} (1 - e^{-\lambda t}) \right) + {}^i \delta_k \bar{\eta}, \quad (40)$$

where ${}^i \delta_k$ is 1 if $i = k$ and 0 otherwise, $\bar{\eta}$ and $\bar{\xi}$ are the bounds defined in equations (2) and (4), while κ and λ are positive scalars such that

$$||e^{-k_o \tilde{L}_k^* t}|| \leq \kappa e^{-\lambda t}, \quad (41)$$

that always exist since $-k_o \tilde{L}_k^*$ is Hurwitz.

REMARK 5.2. It is worth noticing that, in the presence of a recoverable fault on robot l , residuals ${}^i r_l$, $i = 1, 2, \dots, N$, can exceed their threshold during the transient phase needed to the adaptive term ϕ_l in equation (30) to converge to the fault. To avoid these false alarms, the decision about a fault can be taken only if the thresholds is exceeded for a certain time window. This is a quite standard approach in decision-making scheme design, see, e.g., Pierri et al. (2008) for further details.

5.2.2. Fault Recovery via Exclusion of the Faulty Vehicle

In order to accomplish the team mission in spite of the presence of a non-recoverable fault, the faulty vehicle must be excluded from the team and the task function references must be replanned. The adopted detection and isolation strategy allows each robot to isolate the faulty teammate but, since each robot detects the fault asynchronously from each other, the problem is to determine the time instant from which the faulty vehicle must be excluded.

More in detail, the exclusion of a faulty vehicle requires each robot to change the dimension of all the involved variables; that is, each healthy robot must resize the estimate of the collective state vector (${}^i \hat{x}$), the input vector (${}^i \hat{u}$), and the auxiliary variable vector (${}^i \hat{y}$) that is exchanged with the neighbors. The vectors' dimensions become $(N-1)n$ instead of Nn .

Since the detection time instant t in equation (39) is different for each robot, problems can arise if the vector resizing is not synchronized. For example, let us suppose that the i th vehicle has detected a fault on robot l th (i.e., $||{}^i r_l||$ exceeds the thresholds ${}^i \mu_l$) at time t_i , and it resizes the vector ${}^i \hat{y}$ to be sent to its neighbors. Supposing that one or more of these neighbors has not yet detected the fault, they would receive a reduced size vector from the neighbor i but they have not any information about the faulty teammate, so they do not know which components to remove from their collective state estimate. To avoid this issue, all the robots must have already detected and isolated the faulty vehicle before starting to exchange the reduced size vector. To this purpose, let us define the time ${}^i t_d$ as the first instant at which $||{}^i r_l|| > {}^i \mu_l$ in equation (39), i.e.,

$${}^i t_d = \min_t \{t > 0\} : ||{}^i r_l(t)|| > {}^i \mu_l(t).$$

and the detection delay of robot i , ${}^i \Delta t_d = {}^i t_d - t_f$, as the time occurring from the fault occurrence and its detection/isolation by vehicle i th.

Based on the previous considerations, the following steps for removing the faulty robot have to be developed:

- each robot estimates the maximum detection delay, i.e.,

$$\Delta t_{d,\max} = \max_{i=1, \dots, N} {}^i \Delta t_d, \quad (42)$$

- the first robot that detects the fault occurrence sends the reduced size vectors starting from time ${}^j t_d + \Delta t_{d,\max}$,
- the other robots, once received the reduced-order vector ${}^j \hat{y}$, have already complete knowledge about the faulty robot and can resize correctly their own estimates.

In sum, the resize of the estimates of the collective variables is possible after the time instant

$$t_r = \min_i {}^i t_d + \Delta t_{d,\max}. \quad (43)$$

Thus, the problem is how to get a reliable estimate of $\Delta t_{d,\max}$.

5.2.3. Estimate of the Maximum Detection Delay

In order to estimate $\Delta t_{d,\max}$, a detection time analysis should be performed. In general, the detection time depends both on the fault dynamics and on the residuals dynamics. First of all, let

us consider, on the base of equation (36), that the part of the residual not depending by the fault is below its threshold, i.e. $\|i\mathbf{h}_k\| \leq i\mu_k$, and thus, in the presence of a fault, the following chain of inequalities can be written

$$\|i\mathbf{r}_k\| = \|i\mathbf{f}_k + i\mathbf{h}_k\| \geq \|i\mathbf{f}_k\| - \|i\mathbf{h}_k\| \geq \|i\mathbf{f}_k\| - i\mu_k(t). \quad (44)$$

The fault detection by the robot i occurs if $\|i\mathbf{r}_k(t)\| > i\mu_k(t)$, hence, from equation (44) a sufficient detectability condition can be defined as

$$\exists t > t_f: \|i\mathbf{f}_k\| - i\mu_k(t) \geq i\mu_k(t), \quad (45)$$

and thus

$$\exists t > t_f: \|i\mathbf{f}_k\| \geq 2i\mu_k(t). \quad (46)$$

Equation (46) could be adopted to compute an estimate of the maximum detection time and, thus, of $\Delta t_{d,max}$. Such an estimate will be quite conservative since it is based on a sufficient detectability condition and on thresholds also computed from conservative bounds. However, since the fault dynamics is included in equation (46), it is not possible to have an estimate of $\Delta t_{d,max}$ valid for any possible fault, but a time detection analysis can be made only for limited class of faults. More in detail, a possible solution for this problem has been developed in Arrichiello et al. (2014a), where the maximum detection delay was computed for constant faults. By using the same arguments as in Arrichiello et al. (2014a) and the results in Keliris et al. (2013), the time detection analysis can be easily extended to other faults ϕ_i , provided that $\|\phi_i\| > \bar{\phi}$ for a time sufficient for the fault detection, with $\bar{\phi}$ opportunely large.

6. NUMERICAL SIMULATIONS

In this section, we present a numerical simulation case study where a team of 4 robots ($N = 4$), using the proposed distributed fault-tolerant control approach, moves in a plane ($n = 2$) keeping an assigned formation. During the execution of the mission some of the robots are subject to recoverable faults, unrecoverable faults, as well as reactive behavior activations due to obstacle avoidance.

Gains k_c , k_o , K , and G in equations (15), (19), (29), and (30) were, respectively, set to $k_c = 4$, $k_o = 6$, $K = 3I_2$, and $G = 0.01I_2$. The measurement noise $\boldsymbol{\eta}_i$ in equation (3) is assumed to be bounded by 0.001 m ($\bar{\boldsymbol{\eta}} = 0.001$), while the process noise in equation (1) is bounded by 0.005 m/s ($\bar{\boldsymbol{\xi}} = 0.005$ in equation (2)). The directed network topology shown in **Figure 2** is assumed for the information exchange.

The desired path of the centroid $\boldsymbol{\sigma}_{1,d}(t)$ is reported in **Figure 3** (dashed blue line), while the desired formation $\boldsymbol{\sigma}_{2,d}(t)$ is constant and it corresponds to a regular polygon formation of side 0.5 m around the desired centroid. During the simulation, two faults of different nature are considered. In detail, a recoverable fault occurs on robot 2 at $t_{f,2} = 19 \text{ s}$ and it has the following expression:

$$\phi_2 = \begin{cases} [0, 0]^T, & \text{for } t < t_{f,2} \\ [0.3, -0.35]^T (1 - e^{-\frac{t-t_{f,2}}{0.45}}), & \text{for } t \geq t_{f,2}. \end{cases}$$

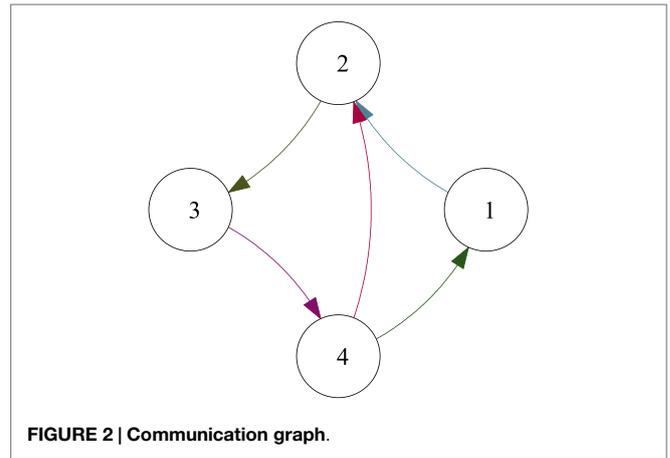


FIGURE 2 | Communication graph.

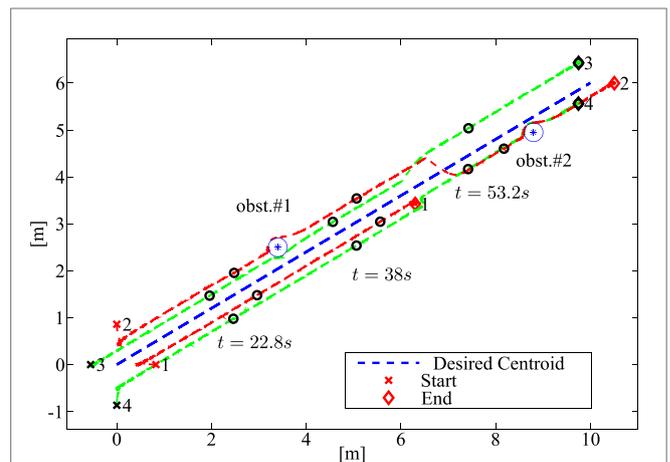


FIGURE 3 | Path followed by the robots in the numerical case study.

The dashed green and red lines, respectively, show the paths of healthy vehicles and vehicles subject to faults; the blue line represents the desired centroid path. The team configuration at three intermediate instants 22.8, 38, and 53.2 s is shown (black circles) together with the initial (crosses) and final (diamonds) configurations.

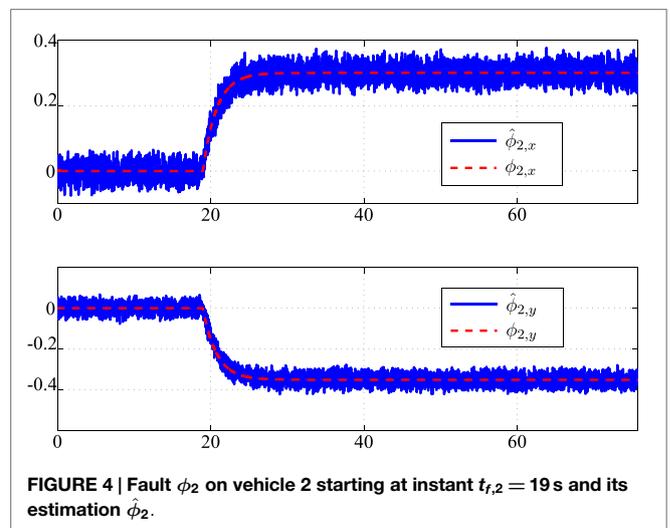


FIGURE 4 | Fault ϕ_2 on vehicle 2 starting at instant $t_{f,2} = 19 \text{ s}$ and its estimation $\hat{\phi}_2$.

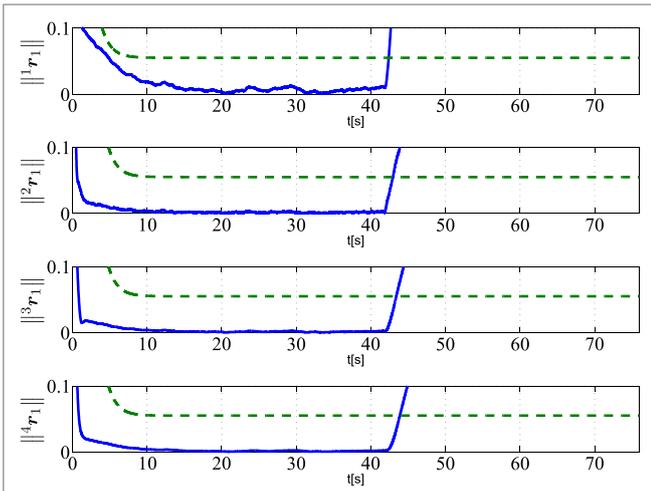


FIGURE 5 | Norm of residuals ${}^i r_1$ ($i = 1, 2, 3, 4$) (solid lines) as calculated by vehicle i and relative to vehicle 1 subject to unrecoverable fault at $t_{f,1} = 41.8$ s. Dashed green lines are the corresponding thresholds used for FDIR strategy.

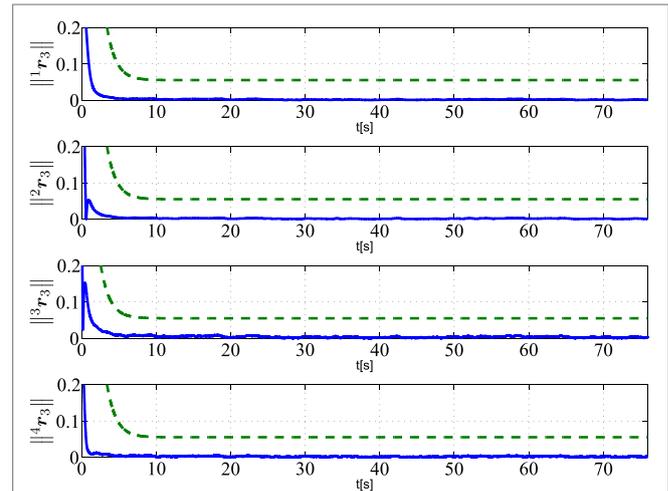


FIGURE 7 | Norm of residuals ${}^i r_3$ ($i = 1, 2, 3, 4$) (solid lines) as calculated by vehicle i and relative to the healthy vehicle 3. Dashed green lines are the corresponding thresholds used for FDIR strategy.

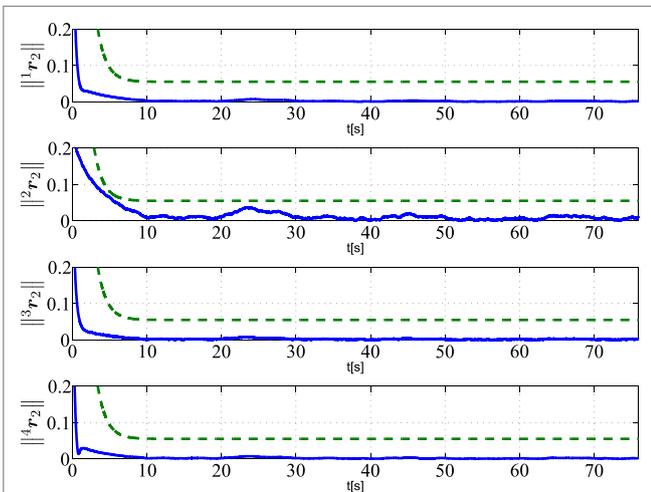


FIGURE 6 | Norm of residuals ${}^i r_2$ ($i = 1, 2, 3, 4$) (solid lines) as calculated by vehicle i and relative to the vehicle 2 subject to a recoverable fault at $t_{f,2} = 30.4$ s. Dashed green lines are the corresponding thresholds used for FDIR strategy.

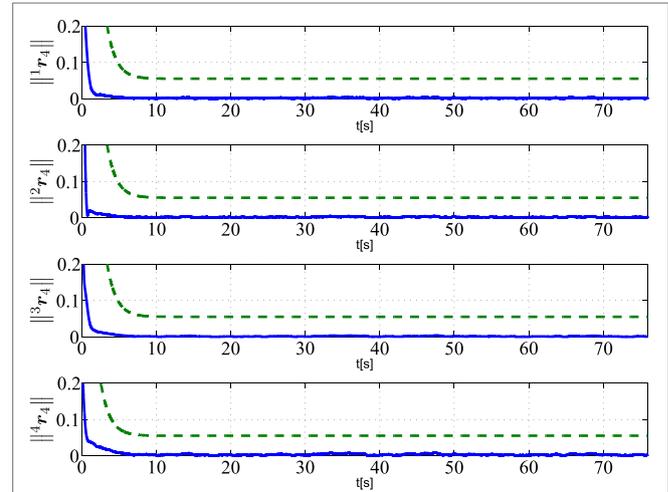


FIGURE 8 | Norm of residuals ${}^i r_4$ ($i = 1, 2, 3, 4$) (solid lines) as calculated by vehicle i and relative to the healthy vehicle 4. Dashed green lines are the corresponding thresholds used for FDIR strategy.

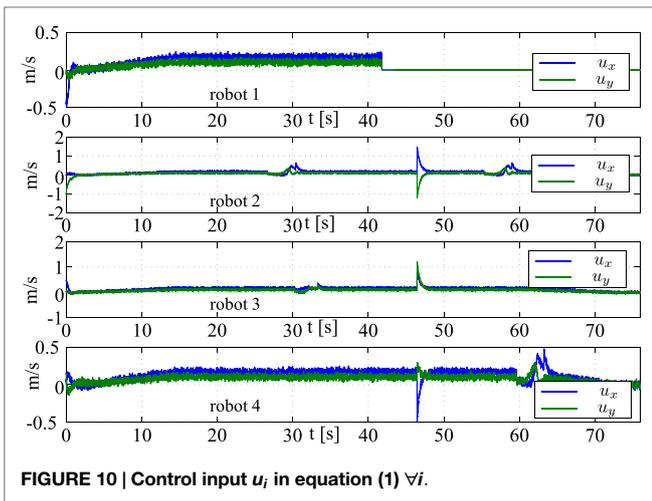
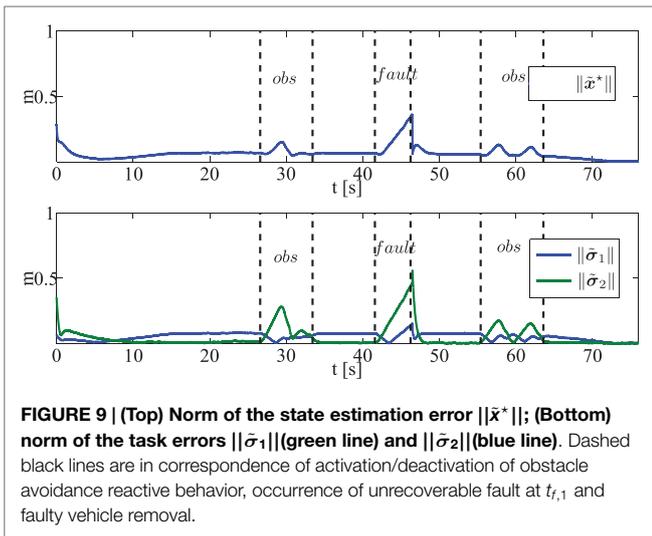
A non-recoverable fault on robot 1, consisting in a complete stop of the robot, is assumed occurring at $t_{f,1} = 41.8$ s. The vehicle paths are shown in **Figure 3** (green dashed lines for healthy robots 3 and 4, and red lines for robots 1 and 2 subject to faults). **Figure 4** shows the components of $\phi_2 = [\phi_{2,x}, \phi_{2,y}]^T$ and its estimate $\hat{\phi}_2 = [\hat{\phi}_{2,x}, \hat{\phi}_{2,y}]^T$ performed by the local fault estimator on board robot 2; the figure also shows that, depending on the local observer gains, the measurement and process noise affects the fault estimates. In this sense, based on equation (1) the process noise is seen as a (bounded) fault affecting the robot dynamics.

Figure 5 shows the residuals calculated by each robot of the team and relative to the faulty robot 1. In detail, the residual norms $\|{}^i r_1\|$ ($i = 1, 2, 3, 4$) (continuous blue lines) and the corresponding thresholds (dashed green lines) are plotted. The plots also show

that, being the fault unrecoverable, all the residuals are kept below the corresponding threshold as long as the fault is not affecting the robot, while the residuals computed by each robot overcome the thresholds after the fault occurrence even if the faulty robot has not a direct communication link with all the teammates.

Figures 6–8 show that the residuals relative to other robots, namely, $\|{}^i r_j\|$ ($i = 1, 2, 3, 4$ and $j \neq 1$), do not overcome the corresponding thresholds; this means that no fault is detected on robots 2, 3, 4. In particular, with regard to robot 2, that is affected by a recoverable fault after $t_{f,2} = 19$ s, the residuals do not overcome the threshold since the fault is *locally* compensated by the controller according to the strategy in Section 5.1, and it does not need to be *globally* detected and recovered.

It is worth noticing from **Figure 3** that the robots are also able to avoid obstacles along their paths via the activation of local reactive



behaviors, and that, as shown in **Figures 5–8**, the activation of the local behaviors do not effect the residual components. Concerning the FDIR strategy described in Section 5.2.2, **Figure 3** shows that,

REFERENCES

- Antonelli, G., Arrichiello, F., Caccavale, F., and Marino, A. (2013). A decentralized controller-observer scheme for multi-agent weighted centroid tracking. *IEEE Trans. Aut. Control* 58, 1310–1316. doi:10.1109/TAC.2012.2220032
- Antonelli, G., Arrichiello, F., Caccavale, F., and Marino, A. (2014). Decentralized time-varying formation control for multi-robot systems. *Int. J. Robot. Res.* 33, 1029–1043. doi:10.1177/0278364913519149
- Antonelli, G., Arrichiello, F., and Chiaverini, S. (2008). The null-space-based behavioral control for autonomous robotic systems. *J. Intell. Service Robot.* 1, 27–39. doi:10.1007/s11370-007-0002-3
- Arrichiello, F., Marino, A., and Pierri, F. (2014a). “A decentralized fault tolerant control strategy for multi-robot systems,” in *Proceedings 19th World Congress of the International Federation of Automatic Control* (Cape Town, South Africa), 6642–6647.
- Arrichiello, F., Marino, A., and Pierri, F. (2014b). “Distributed fault detection and recovery for networked robots,” in *IEEE/RSJ International Conf. on Intelligent Robots and Systems* (Chicago, IL), 3734–3739.
- Arrichiello, F., Marino, A., and Pierri, F. (2014c). “Distributed fault-tolerant strategy for networked robots with both cooperative and reactive controls,” in *2014 IEEE International Conference on Information and Automation (ICIA)* (Harbin, China: IEEE), 677–682.
- Arrichiello, F., Marino, A., and Pierri, F. (2015). Observer-based decentralized fault detection and isolation strategy for networked multirobot systems. *IEEE Trans. Control Syst. Technol.* 23, 1465–1476. doi:10.1109/TCST.2014.2377175
- Belta, C., and Kumar, V. (2004). Abstraction and control of groups of robots. *IEEE Trans. Robot.* 20, 865–875. doi:10.1109/TRO.2004.829498
- Bullo, F., Cortés, J., and Martínez, S. (2009). *Distributed Control of Robotic Networks. Applied Mathematics Series*. Princeton, NJ: Princeton University Press.
- Caccavale, F., Marino, A., Muscio, G., and Pierri, F. (2013). Discrete-time framework for fault diagnosis in robotic manipulators. *IEEE Trans. Control Syst. Technol.* 21, 1858–1873. doi:10.1109/TCST.2012.2212196
- Cao, M., Morse, A. S., and Anderson, B. D. (2008). Reaching a consensus in a dynamically changing environment: a graphical approach. *SIAM J. Control Optim.* 47, 575–600. doi:10.1137/060657005

after the exclusion of robot 1, the formation is rearranged according to a regular triangular formation. The maximum detection time $\Delta t_{d,\max}$ in equation (42) is set to 5 s.

In **Figure 9**, the norms of the state estimation error $\|\tilde{x}^*\|$ (top) and the task errors $\|\tilde{\sigma}_1\|$ and $\|\tilde{\sigma}_2\|$ (bottom) are shown. The figure shows that these errors grow during the obstacle avoidance and the fault detection phases and decreases toward a neighborhood of the origin in normal conditions. Finally, **Figure 10** reports the control input u_i ($\forall i$) in equation (1). The top plot is relative to robot 1 that is affected by a non-recoverable fault; thus, the input becomes zero after the fault occurrence. Concerning the other plots, the discontinuities on the inputs is due to the faulty robot exclusion and formation rearrangement.

7. CONCLUSION

In this paper, a distributed fault-tolerant control strategy for networked robots was presented. The proposed approach builds on a distributed controller-observer architecture, a fault detection, isolation and recovery strategy for unrecoverable faults, and a local fault estimation and compensation for recoverable faults. The proposed approach also results robust to the activation of reactive behaviors acting at local levels and not included in the global control strategy.

The proposed architecture allows to manage in an integrated manner the different issues related to distributed formation control and fault management, while keeping limited the communication burden; i.e., the information exchanged among neighbors for state estimation purposes are also used for fault management without further requirements. The approach was successfully tested via numerical simulations, and the results of a single integrated case study involving the occurrence of the different kinds of faults as well as the activation of reactive behaviors was presented to validate the approach.

AUTHOR CONTRIBUTIONS

All authors listed have made substantial, direct, and intellectual contribution to the work and approved it for publication.

- Davoodi, M., Khorasani, K., Talebi, H., and Momeni, H. (2014). Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems. *IEEE Trans. Control Syst. Technol.* 22, 1061–1069. doi:10.1109/TCST.2013.2264507
- Dong, X., and Hu, G. (2016). Time-varying formation control for general linear multi-agent systems with switching directed topologies. *Automatica* 73, 47–55. doi:10.1016/j.automatica.2016.06.024
- Fax, J., and Murray, R. (2004). Information flow and cooperative control of vehicle formations. *IEEE Trans. Automatic Control* 49, 1465–1476. doi:10.1109/TAC.2004.834433
- Ferrari, R., Parisini, T., and Polycarpou, M. (2009). Distributed fault diagnosis with overlapping decomposition: an adaptive approximation approach. *IEEE Trans. Automatic Control* 54, 794–799. doi:10.1109/TAC.2008.2009591
- Godsil, C., and Royle, G. (2001). *Algebraic Graph Theory. Graduate Texts in Mathematics*. New York: Springer.
- Hong, Y., Chen, G., and Bushnell, L. (2008). Distributed observers design for leader-following control of multi-agent networks. *Automatica* 44, 846–850. doi:10.1016/j.automatica.2007.07.004
- Keliris, C., Polycarpou, M. M., and Parisini, T. (2013). A distributed fault detection filtering approach for a class of interconnected continuous-time nonlinear systems. *IEEE Trans. Automat. Contr.* 58, 2032–2047. doi:10.1109/TAC.2013.2253231
- Marino, A., and Pierri, F. (2015). “Discrete-time distributed control and fault diagnosis for a class of linear systems,” in *IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)* (Hamburg, Germany), 2974–2979.
- Marino, A., Pierri, F., Chiacchio, P., and Chiaverini, S. (2015). “Distributed fault detection and accommodation for a class of discrete-time linear systems,” in *Proceeding of the IEEE International Conf. on Information and Automation*, (Lijiang, China), 469–474.
- Mesbahi, M., and Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton University Press.
- Meskin, N., and Khorasani, K. (2009). Actuator fault detection and isolation for a network of unmanned vehicles. *IEEE Trans. Automatic Control* 54, 835–840. doi:10.1109/TAC.2008.2009675
- Olfati-Saber, R., Fax, J., and Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* 95, 215–233. doi:10.1109/JPROC.2006.887293
- Olfati-Saber, R., and Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Automatic Control* 49, 1520–1533. doi:10.1109/TAC.2004.834113
- Pierri, F., Paviglianiti, G., Caccavale, F., and Mattei, M. (2008). Observer-based sensor fault detection and isolation for chemical batch reactors. *Eng. Appl. Artif. Intell.* 21, 1204–1216. doi:10.1016/j.engappai.2008.02.002
- Ren, W., and Beard, R. (2008). *Distributed Consensus in Multi-vehicle Cooperative Control. Communications and Control Engineering*. Berlin: Springer.
- Smith, R., and Hadaegh, F. (2007). Closed-loop dynamics of cooperative vehicle formations with parallel estimators and communication. *IEEE Trans. Automatic Control* 52, 1404–1414. doi:10.1109/TAC.2007.902735
- Stankovic, S., Ilic, N., Djurovic, Z., Stankovic, M., and Johansson, K. (2010). “Consensus based overlapping decentralized fault detection and isolation,” in *Conf. on Control and Fault-Tolerant Systems* (Nice, France), 570–575.
- Staroswiecki, M. (2008). On fault handling in control systems. *Int. J. Control Automation Syst.* 6, 296–305.
- Tuna, S. E. (2008). Synchronizing linear systems via partial-state coupling. *Automatica* 44, 2179–2184. doi:10.1016/j.automatica.2008.01.004
- Wang, J., Cheng, D., and Hu, X. (2008). Consensus of multi-agent linear dynamic systems. *Asian J. Control* 10, 144–155. doi:10.1002/asjc.15
- Wang, Y., Ye, H., Ding, S., Cheng, Y., Zhang, P., and Wang, G. (2009). Fault detection of networked control systems with limited communication. *Int. J. Control* 82, 1344–1356. doi:10.1080/00207170802558967
- Yang, P., Freeman, R., and Lynch, K. (2008). Multi-agent coordination by decentralized estimation and control. *IEEE Trans. Automatic Control* 53, 2480–2496. doi:10.1109/TAC.2008.2006925
- Yu, W., Chen, G., Cao, M., and Kurths, J. (2010). Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics. *IEEE Trans. Syst. Man Cybern. B Cybern.* 40, 881–891. doi:10.1109/TSMCB.2009.2031624
- Zhang, K., Jiang, B., and Shi, P. (2013). *Observer-Based Fault Estimation and Accommodation for Dynamic Systems*. Berlin, Heidelberg: Springer.
- Zhang, X., Polycarpou, M., and Parisini, T. (2002). A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Trans. Automatic Control* 47, 576–593. doi:10.1109/9.995036
- Zhang, X., and Zhang, Q. (2012). Distributed fault diagnosis in a class of interconnected nonlinear uncertain systems. *Int. J. Control* 85, 1644–1662. doi:10.1080/00207179.2012.696146

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Arrichiello, Marino and Pierri. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.