International Journal of Advanced Robotic Systems

Distributed Fault Detection and Isolation for Flocking in a Multi-robot System with Imperfect Communication

Regular Paper

Shao Shiliang[1,*], Wang Ting[1], Yao Chen[1], Li Xiaofan[1] and Zhao Hai[2]

1 State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China
2 Northeastern University, China
* Corresponding author E-mail: shaoshiliangswu@gmail.com

Received 10 Feb 2014; Accepted 27 Mar 2014

DOI: 10.5772/58601

Abstract In this paper, we focus on distributed fault detection and isolation (FDI) for a multi-robot system where multiple robots execute a flocking task. Firstly, we propose a fault detection method based on the local-information-exchange and sensor-measurement technologies to cover cases of both perfect communication and imperfect communication. The two detection technologies can be adaptively selected according to the packet loss rate (PLR). Secondly, we design a fault isolation method, considering a situation in which faulty robots still influence the behaviours of other robots. Finally, a complete FDI scheme, based on the proposed detection and isolation methods, is simulated in various scenarios. The results demonstrate that our FDI scheme is effective.

Keywords Multi-Robot, Flocking, Fault Detection, Isolation

1. Introduction

Flocking behaviours exist widely in nature in the form of the flocking of birds, the schooling of fish and the swarming of bacteria [1, 2]. Motivated by observations of these flocking behaviours in their environments, multi-robot systems are expected to be applied in a wide set of situations, such as exploration, search-and-rescue, unmanned aerial vehicles (UAV), and so on [3]. Recently, the flocking of multi-robot systems has attracted much attention [4, 5]. As the size and complexity of multi-robot systems rapidly increase, the fault detection and isolation (FDI) method in flocking is becoming more and more important. For example, a faulty UAV in a UAV formation can significantly jeopardize the performance of its nearby UAVs, or even destroy the whole formation.

There exist many works on the design of FDI schemes [6-16]. Some of these works concentrate on FDI schemes with a centralized control structure [6-10]. Robust model-based fault diagnosis methods for dynamic systems are proposed in [6], but they focus on fundamental issues, such as basic definitions, residual generation methods and the importance of robustness in model-based fault diagnosis approaches. A Hinf-based structured fault detection and isolation (Hinf-SFDI) method is introduced in [7]. For Hinf-SFDI, all the information is sent to a central FDI unit through wireless channels, which increases the communication burden on the FDI unit. Based on the centralized structure, many FDI approaches that utilize observers have been proposed in [8-10]. However, observers increase the computational burden on robots. Thus, it is not wise to solve the FDI problem with a centralized structure due to the limitations of the communication and computation capabilities of FDI units and robots [11]. Distributed FDI approaches for multi-robot systems have been studied in [12-16]. In [12], by using a bank of decentralized observers, a scheme for fault tolerant distributed network control systems is proposed. In [13], the FDI problem in multi-robot systems is considered as a double integrator dynamics system and each robot uses decentralized observers to estimate system input (e.g., control command signal, noise, interference, etc.). The FDI schemes proposed by [12] and [13] belong to the schemes of state estimation. However, the computational complexity of the state-estimation-based schemes will increase with the increase of the number of robots in a system. The research into distributed FDI schemes is just beginning and there still exist many problems. When performing fault *detection*, most existing works assume that the communication between robots is always perfect [14-16]. In practice, it is common to find that some robots cannot communicate with other robots. Although in [7] the author has studied an FDI scheme with an imperfect communication channel, it is based on a centralized control structure. When performing fault *isolation*, the existing works simply remove the robots that present misbehaviour from their connected-graph models. However, these robots still exist in practice and may influence the performance of the system.

In this paper, to tackle these problems, we focus on developing a distributed FDI scheme for the flocking of multi-robot systems. In the proposed scheme, we design a detection method that consists of the local-information-exchange-based detection technology and the sensor-measurement-based detection technology to cover both cases of perfect and imperfect communication, respectively. According to the PLR, the two detection technologies are adaptively selected. By considering a situation in which faulty robots still influence the behaviours of other robots, a fault isolation solution for faulty robots is proposed. Finally, a complete FDI scheme based on the proposed detection and isolation methods is simulated in various scenarios. The results demonstrate that our FDI scheme is effective.

The remainder of this paper is organized as follows. The system model is described in Section 2. The detection and isolation methods are discussed in Section 3 and Section 4, respectively. The simulation results are presented in Section 5. The conclusion is introduced in Section 6.

2. System Model

In this section, we firstly model a multi-robot system as a graph. Then, we introduce the residual generator of the multi-robot system.

2.1 *Topology of Flocks: Proximity Nets*

The topology of a multi-robot system can be regarded as a graph $G(V,E)$, which consists of a set of vertices $V=\{1,2,...,n\}$ and edges $E\subseteq\{(i,j):i,j\in V, j\neq i\}$. The graph G is undirected, i.e., the edges of satisfying the condition of $(i,j)\in V \Leftrightarrow (j,i)\in V$.

The adjacency matrix $\mathbf{A}=[a_{ij}]$ of the graph is a nonzero matrix satisfying the property $a_{ij}\neq 0 \Leftrightarrow (i,j)\in E$. Here, graph G is a weighted graph with position dependent adjacency elements. For undirected graph G, the adjacency matrix $\mathbf{A}$ is symmetric (or $\mathbf{A}^T=\mathbf{A}$). The set of neighbours of node i is defined by

$$N_i=\{j\in V: a_{ij}\neq 0\}=\{j\in V:(i,j)\in E\}\,. \tag{1}$$

We consider a group of dynamic robots with the motion equation

$$\begin{cases}\mathbf{p}_i(k+1)=\mathbf{p}_i(k)+\mathbf{v}_i(k)\\ \mathbf{v}_i(k+1)=\mathbf{v}_i(k)+\mathbf{u}_i(k)\end{cases} \tag{2}$$

where $\mathbf{p}_i(k),\mathbf{v}_i(k),\mathbf{u}_i(k)\in\mathbb{R}^m$ $k\geq 0$, m is the dimension of the system. Additionally, $\mathbf{p}_i(k)$ denotes the position of robot i. $\mathbf{v}_i(k)$ and $\mathbf{u}_i(k)$ are the velocity of robot i and the excitation input of robot i, respectively. We set that the sample time T is between $k+1$ and k.

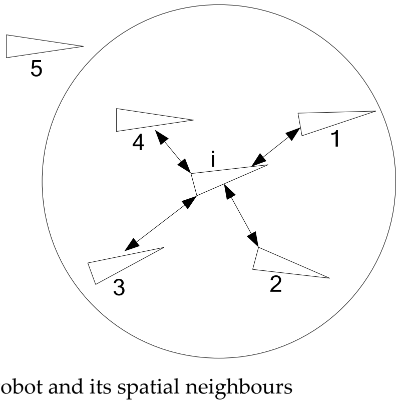

Figure 1. A robot and its spatial neighbours

Let $r>0$ denote the communication range between two robots. An open ball with radius r determines the set of the spatial neighbours of robot i, which is denoted by

$$N_i=\{j\in V:\|\mathbf{p}_j(k)-\mathbf{p}_i(k)\|<r\} \tag{3}$$

where $\|\bullet\|$ is the Euclidean norm in $\mathbb{R}^m$. A robot with its spatial neighbours is shown in Figure 1. Let $d > 0$ denote the measured range between two robots, which is twice that of the communication range r , i.e. $d = 2r$.

Flocks: A group of robots is called a *flock* if all the robots are connected over the time interval $t \in [t_0, t_f]$, $t_f \geq 0$, have the same velocity and keep the lattice or shape.

$\sigma - norm$: The non-negative map is called $\sigma - norm$, which is used to construct a smooth collective potential of a flock and the spatial adjacency matrix of the proximity net.

The $\sigma - norm$ of a vector is a map $\mathbb{R}^m \to \mathbb{R}_{\geq 0}$ that can be defined as

$$\|z\|_\sigma = \frac{1}{\varepsilon}\left[\sqrt{1+\varepsilon\|z\|^2} - 1\right] \tag{4}$$

With a parameter $\varepsilon > 0$, the map $\|z\|_\sigma$ is differentiable everywhere, while $\|z\|$ is not differentiable at $z = 0$. Thus, $\|z\|_\sigma$ will be used in this paper.

According to [17], each robot in the free flocking applies a control input that consists of two terms

$$\begin{aligned} \mathbf{u}_i^\alpha(k) = c_1^\alpha \sum_{j\in N_i^\alpha} \varphi_\alpha(\| \mathbf{p}_j(k) - \mathbf{p}_i(k) \|_\sigma)\mathbf{n}_{i,j} \\ + c_2^\alpha \sum_{j\in N_i^\alpha} a_{ij}(\mathbf{p}_i(k))(\mathbf{v}_j(k) - \mathbf{v}_i(k)) \end{aligned} \tag{5}$$

$$\mathbf{u}_i^\gamma(k) = -c_1^\gamma\sigma_1(\mathbf{p}_i(k) - \mathbf{p}_\gamma(k)) - c_2^\gamma(\mathbf{v}_i(k) - \mathbf{v}_r(k)) \tag{6}$$

where $\mathbf{u}_i^\alpha(k)$ and $\mathbf{u}_i^\gamma(k)$ are the inputs based on neighbours and the virtual leader of robot i . Thus, the control input of each robot is

$$\mathbf{u}_i^t(k) = \mathbf{u}_i^\alpha(k) + \mathbf{u}_i^\gamma(k) \tag{7}$$

Based on the control input of the multi-robot systems, we can generate the residual for the FDI.

2.2 Residual Generator of the Multi-Robot System

In multi-robot systems, every robot has its own individual FDI system. And the residual of every robot is defined as

$$r_i(k) = \| \mathbf{u}_i^t(k) - \mathbf{u}_i^m(k) \| \tag{8}$$

where $r_i(k)$ is the residual signal, $\mathbf{u}_i^t(k)$ is the theoretical value and $\mathbf{u}_i^m(k)$ is the measurement value of the control input. In this paper, we assume that $\mathbf{u}_i^\gamma(k)$ in $\mathbf{u}_i^t(k)$ and $\mathbf{u}_i^m(k)$ are equal and do not participate in the calculation process.

Since an attempt to evaluate the residual signal over the entire time would usually be unrealistic, the evaluation function in this article is computed as the average energy of the residual signal over a given time interval (k_1, k_τ)

$$J_r(\tau) = \mathrm{E}\left\{\sqrt{\frac{1}{\tau}\sum_{k=k_1}^{k_\tau} r^T(k)r(k)}\right\} \tag{9}$$

where k_1 denotes the initial evaluation time instant and k_τ stands for the evaluation time. The detection logic unit is based on the works proposed in [18].

Thus, the threshold used to detect faults is defined as

$$J_{th} = \sup_{d(k)\in \mathrm{L}_{E2}, f=0} J_r \tag{10}$$

Based on equation (10), the occurrence of faults can be detected by comparing $J_r(\tau)$ and J_{th}

$$\begin{aligned} J_r(\tau) > J_{th} \Rightarrow alarm, \\ J_r(\tau) \leq J_{th} \Rightarrow no\ fault. \end{aligned} \tag{11}$$

By comparing the residual with its threshold, we can obtain the definition of the faulty robots.

Faulty robot: the robot i is faulty if $J_r(\tau)$ satisfies the constraint that

$$J_r(\tau) > J_{th} \tag{12}$$

3. Fault Detection

When performing fault detection, most existing works assume that the communication between robots is always perfect. In practice, it is a common for some of the robots not to be able to communicate with the other robots. So it is very relevant to research distribute fault detection methods, considering whether communication between robots is perfect or imperfect.

3.1 Perfect Communication

If communication between different robots is perfect, we propose a fault detection method, which requires that robot j broadcast its own position information $\mathbf{p}_j(k)$ and its neighbours' position information $\mathbf{p}_{s\in N_j(k)}(k)$. In this case, robot j firstly collects $\mathbf{p}_{s\in N_j(k)}(k)$. Then, robot j puts every short data packet $\mathbf{p}_{s\in N_j(k)}(k)$ into a length data packet. We call this procedure data packet reconstruction [19]. The data packet reconstruction increases the transmission efficiency compared with transmitting multiple short packets.

When robot i receives $\mathbf{p}_j(k)$ and $\mathbf{p}_{s\in N_j(k)}(k)$ from j, the robot i can construct $\mathbf{u}^t_{j\in N_i(k)}(k)$ precisely using equations (5)-(7).

The measurement value of $\mathbf{u}^m_j(k)$ according to equation (2) can be calculated as follows

$$\begin{aligned}\mathbf{u}^m_j(k) &= f(\mathbf{p}_j(k+1), \mathbf{p}_j(k), \mathbf{p}_j(k-1)) \\ &= \frac{\left(\mathbf{p}_j(k+1)-\mathbf{p}_j(k)\right)}{T} - \frac{\left(\mathbf{p}_j(k)-\mathbf{p}_j(k-1)\right)}{T}\end{aligned} \tag{13}$$

Then, according to equation (8), robot i can generate the residual signal of robot $j \in N_i(k)$ using the follow equation

$$\begin{aligned}r_j(k) &= \| \mathbf{u}^t_j(k) - \mathbf{u}^m_j(k) \| \\ &= \| \mathbf{u}^t_j(k) - f(\mathbf{p}_j(k+1), \mathbf{p}_j(k), \mathbf{p}_j(k-1)) \|\end{aligned} \tag{14}$$

Based on the threshold J_{th}, robot i can identify whether the robot $j \in N_i(k)$ is faulty or not.

If robot i cannot receive the data packet correctly, it means that the communication channel is imperfect. In this case, sensor-measurement-based detection technology is used.

3.2 Imperfect Communication

When the communication between the different robots is imperfect, we consider using on-board sensors (e.g., laser scanning rangefinder) to measure the distance $d_{ij}(k)$ between two robots and the relative angle $\beta_{ij}(k)$ of these two robots' direction at the time step *k*.

For the sensor, we assume that the measurable range of the distance is twice the range of communication. Thus, if the robot i has a neighbour j at the time step *k*, the robot i can measure all the neighbours of robot j because they are within the communication range of robot j. To generate the residual signal, we need to calculate $\mathbf{u}^t_j(k)$ and $\mathbf{u}^m_j(k)$.

$\mathbf{u}^t_j(k)$ is calculated using equations (5)-(7), where

$$\mathbf{d}_{ij}(k) = \mathbf{p}_i(k) - \mathbf{p}_j(k) = \begin{bmatrix} d_{ij}(k)\cos\beta_{ij}(k) \\ d_{ij}(k)\sin\beta_{ij}(k) \end{bmatrix} \tag{15}$$

and

$$\begin{aligned}\mathbf{v}_j(k) &= \frac{\mathbf{p}_j(k+1) - \mathbf{p}_i(k+1) - (\mathbf{p}_j(k) - \mathbf{p}_i(k))}{T} \\ &= \frac{\mathbf{d}_{ij}(k+1) - \mathbf{d}_{ij}(k)}{T}\end{aligned} \tag{16}$$

Then, we calculate $\mathbf{u}^m_j(k)$ according to (13), measuring all the neighbours of robot *j*. The position information $\mathbf{p}_j(k+1)$ can be calculated by

$$\begin{aligned}\mathbf{p}_j(k+1) &= \mathbf{p}_i(k+1) - \left(\mathbf{p}_i(k+1) - \mathbf{p}_j(k+1)\right) \\ &= \mathbf{p}_i(k+1) - \mathbf{d}_{ij}(k+1)\end{aligned} \tag{17}$$

After predicting $\mathbf{u}^t_j(k)$ and measuring $\mathbf{u}^m_j(k)$, we can generate the residual signal $r_j(k)$ for each $j \in N_i(k)$. Then, the threshold J_{th} used to indicate faulty robots is calculated according to $r_j(k)$.

3.3 Estimating Communication Quality

The Gaussian Mixture Model (GMM) can be used to approximate any distribution [20]. In this paper, we utilize the GMM to build the distribution of PLR.

Essentially, the GMM is a kind of multidimensional probability density function and can be expressed by a linear combination of the Gaussian density function. Therefore, the distribution of PLR can expressed as follows

$$p(\mathbf{x} \mid \mathbf{\Theta}) = \sum_{i=1}^{M} \rho_i p_i(\mathbf{x} \mid \mathbf{\Theta}) \tag{18}$$

where $\mathbf{x}$ is the real variable set of PLR, M is the mixture degree of GMM. The variable $\mathbf{\Theta}$ is the parameter set and can be represented by $\mathbf{\Theta} = \{\rho_i, \boldsymbol{\mu}_i, \boldsymbol{\lambda}_i\}\ \forall\, i = 1,\ldots,M$. The variable ρ_i is the weight of the *i*-th Gaussian component which satisfies the condition that $\sum_{i=1}^{k} \rho_i = 1$. The variables $\boldsymbol{\mu}_i$ and $\boldsymbol{\lambda}_i$ represent mean and covariance, respectively. $p_i(\mathbf{x} \mid \mathbf{\Theta})$ represents the *i*-th Gaussian density of the GMM. It is given as $p_i(\mathbf{x} \mid \mathbf{\Theta}) = 2\pi \,|\, \boldsymbol{\lambda}_i \,|^{-1/2} \exp\{-\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\lambda}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\}$.

We utilize the Expectation Maximization (EM) algorithm to estimate the parameter set of GMM [21, 22]. It is an iterative method for finding parameters in statistical models. The EM algorithm involves two steps: the E step and M step. It repeats the E step and M step to obtain the parameter set $\mathbf{\Theta}$.

Let $\mathbf{x} = \{x_1, x_2, \ldots x_j \ldots x_D\}\ \forall\, j = 1 \ldots D$ be the data set of the measurements of PLR with the a total number of D. The detailed process of the E step and M step is as follows.

The E step computes the conditional probabilities

$$p(i \mid x_j, \mathbf{\Theta}) = \frac{\rho_i p(x_j \mid \boldsymbol{\varepsilon}_i, \boldsymbol{\lambda}_i)}{\sum_{i=1}^{M} \rho_i p(x_j \mid \boldsymbol{\varepsilon}_i, \boldsymbol{\lambda}_i)}. \tag{19}$$

The M step updates the parameter set $\boldsymbol{\Theta}$

$$\hat{\rho}_i = \frac{1}{D}\sum\nolimits_{j=1}^{D} p(i \mid x_j, \boldsymbol{\Theta}^p), \tag{20}$$

$$\hat{\boldsymbol{\varepsilon}}_i = \frac{\sum\nolimits_{j=1}^{D} x_j p(i \mid x_j, \boldsymbol{\Theta}^p)}{\sum\nolimits_{j=1}^{D} p(i \mid x_j, \boldsymbol{\Theta}^p)}, \tag{21}$$

$$\hat{\boldsymbol{\lambda}}_i = \frac{\sum\nolimits_{j=1}^{D} p(i \mid x_j, \boldsymbol{\Theta}^p)(x_j - \hat{\boldsymbol{\varepsilon}}_i)(x_j - \hat{\boldsymbol{\varepsilon}}_i)^T}{\sum\nolimits_{j=1}^{D} p(i \mid x_j, \boldsymbol{\Theta}^p)}, \tag{22}$$

where $\boldsymbol{\Theta}^p$ is the parameter set acquired from the previous iteration.

According to the GMM and EM algorithm, we can obtain the prior distribution of the PLR. To estimate the PLR, a few probe packets are sent. The details of the estimation process are described as follows.

Consider a scenario where robot A sends packets to robot B. If A sends D_{send} packets and B receives D_{recv} packets during time $t, t \in \{k, k+1\}$, we can calculate the PLR of the communication link from robot A to robot B by $x(t) = 1 - \frac{D_{recv}}{D_{send}}$. Then, we can use the empirical data of the PLR to estimate $x(k+1)$ at time step $k+1$ by the method of maximum a posteriori (MAP) probability estimation.

We want to estimate unobserved PLR $x(k+1)$ on the basis of observations $x(t)$. Let p be the sampling distribution of $x(t)$. Assuming that a prior distribution g over $x(k+1)$ exists, the posterior distribution of $x(k+1)$ is

$$p(x(t) \mid x(k+1)) = \frac{p(x(t) \mid x(k+1))g(x(k+1))}{\int p(x(t) \mid x(k+1)')g(x(k+1)')dx(k+1)'} \tag{23}$$

The estimated value of $x(k+1)$ is

$$x(k+1)_{MAP}(x(t)) = \underset{x(k+1)}{\arg\max} \frac{p(x(t))g(x(k+1))}{\sum\limits_{j=1}^{D} p(x(t) \mid x(k+1))g(x(k+1))} \tag{24}$$

The threshold of PLR determines whether the communication is perfect or not and guides robots to select the suitable detection method. We define the threshold of PLR as m. If the estimated value of PLR is larger than the threshold m, it means that the communication is imperfect.

4. Fault Isolation

In this section, we propose a fault isolation scheme by considering a situation in which faulty robots still influence the behaviours of other robots.

We define the dangerous distance of a non-faulty robot as r_{d}. We define the security distance between a non-faulty robot and a faulty robot as r_{s}. We assume that the dangerous distance r_{d} is smaller than the communication range r and the sensing distance d, i.e., $0 < r_{\mathrm{d}} < r < d$. The ball with the faulty robot's position as a spherical centre and r_{d} as the radius is called the dangerous zone. Figure 2 shows how a robot isolates the dangerous zone of a faulty robot. In Figure 2, the vector from the non-faulty robot to the faulty robot is defined as $\mathbf{dp}$, and $\mathbf{dt}_1$ and $\mathbf{dt}_2$ are defined as the tangent vectors from the non-faulty robot to the dangerous zone. The vector from the non-faulty robot to the lead robot is $\mathbf{d}_\gamma$.

After a non-faulty robot detects the faulty robot, the non-faulty robot will judge whether the orientation of its velocity is between $\mathbf{dt}_1$ and $\mathbf{dt}_2$, and will start to perform the faulty robot isolation. The judgment rule is

$$\det[\mathbf{dt}_1, \mathbf{v}_i(k)] < 0 \wedge \det[\mathbf{dt}_2, \mathbf{v}_i(k)] > 0 \tag{25}$$

If the orientation of the velocity does not satisfy the rule, the non-faulty robot does not need to change its trajectory. The reason is that the faulty robot cannot influence the non-faulty robot in this situation. If the orientation of the velocity satisfies the above rule, then we start to perform the isolation solution. The solution is designed to cover cases whether flocks are formed or not.

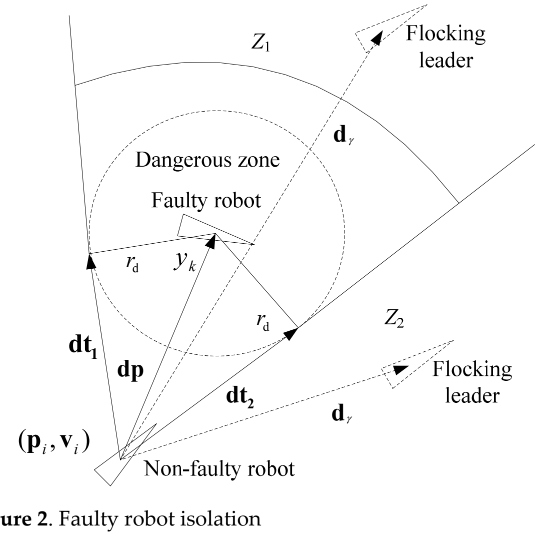

Figure 2. Faulty robot isolation

4.1. Not Forming Flocks

If the flocks have not been formed, every robot does not have the same velocity as its neighbours. All the non-faulty robots' final movement goals are the leader's position. As a result, the faulty robot isolation should regard the leader as the reference.

As shown in Figure 2, when the leader's position is in the zone Z_1 (i.e., behind the faulty robots) and the non-faulty robot satisfies the following rule

$$\det[\mathbf{dt}_1, \mathbf{d}_\gamma(k)] < 0 \wedge \det[\mathbf{dp}, \mathbf{d}_\gamma(k)] > 0 \tag{26}$$

Then, the non-faulty robot i adjusts the orientation of its velocity to the tangent of the dangerous zone to isolate the faulty robot and make the least deviation possible from the leader at the same time. To achieve this, the new velocity of this non-faulty robot can be defined as

$$\mathbf{v}_i(k+1) = \begin{cases} \|\mathbf{v}_i(k)\|_\sigma \dfrac{\mathbf{dt}_1}{\|\mathbf{dt}_1\|_\sigma}, \\ \{\|\mathbf{dp}\|_\sigma \le r_s\} \cap \{\det[\mathbf{dt}_1, \mathbf{v}_i(k)] < 0 \\ \qquad \wedge \det[\mathbf{dp}, \mathbf{v}_i(k)] > 0\} \\ \|\mathbf{v}_i(k)\|_\sigma \dfrac{\mathbf{dt}_2}{\|\mathbf{dt}_2\|_\sigma}, \\ \{\|\mathbf{dp}\|_\sigma \le r_s\} \cap \{\det[\mathbf{dp}, \mathbf{v}_i(k)] < 0 \\ \qquad \wedge \det[\mathbf{dt}_2, \mathbf{v}_i(k)] > 0\} \end{cases} \tag{27}$$

Sometimes, the lead robot is in Z_2 (i.e., it is not behind the faulty robot) and the non-faulty robot satisfies the rule

$$\det[\mathbf{dt}_1, \mathbf{d}_\gamma(k)] > 0 \wedge \det[\mathbf{dp}, \mathbf{d}_\gamma(k)] < 0 \tag{28}$$

In such a case, the faulty robot will not influence the non-faulty robot. The robot i will turn the orientation of its velocity to the leader's orientation and the new velocity of this non-faulty robot is defined as

$$\mathbf{v}_i(k+1) = \begin{cases} \|\mathbf{v}_i(k)\|_\sigma \dfrac{(\mathbf{p}_i(k) - \mathbf{p}_\gamma(k))}{\|(\mathbf{p}_i(k) - \mathbf{p}_\gamma(k))\|_\sigma}, \\ \{\|\mathbf{dp}\|_\sigma \le r_s\} \cap \{\det[\mathbf{dt}_1, \mathbf{v}_i(k)] < 0 \\ \qquad \wedge \det[\mathbf{dt}_2, \mathbf{v}_i(k)] > 0\} \end{cases} \tag{29}$$

4.2. Having Formed Flocks

If flocks have been formed, the non-faulty robot i has the same velocity as all its neighbours. In this case, the non-faulty robot should guarantee the same velocity as all its neighbours as much as it can. It does not need to consider the position of the leader. When the faulty robot is detected by the non-faulty robot, the non-faulty robot i adjusts the orientation of its velocity to the tangent of the dangerous zone. The new velocity of this non-faulty robot is the same as (27).

5. Experimental Results

In this section, we apply our FDI scheme to the multi-robot system that performs the flocking task. The system has double integrator dynamics. Robots move to flock in the 2D space with the aforementioned control law as depicted in (5)-(7). The initial x and y axes of the robots are randomly generated between [0, 50]. Furthermore, their initial velocity is set to zero. The leader of these robots always moves at a speed of 16m/s in the simulation procedures. Each robot has a communication range of 8.4m, and correspondingly, their measureable distance is 16.8m. When the robots form the flocks, the distance between any two neighbour robots is set to 7m. The dangerous distance is set to 6m. Unless specified, we consider practical conditions in our simulation situation, i.e., that the communication may be perfect or imperfect.

5.1 FDI in Multi-robot System

In this subsection, we show the effectiveness of the proposed FDI scheme by observing the flocking of the system. We configure 10 robots in our simulations.

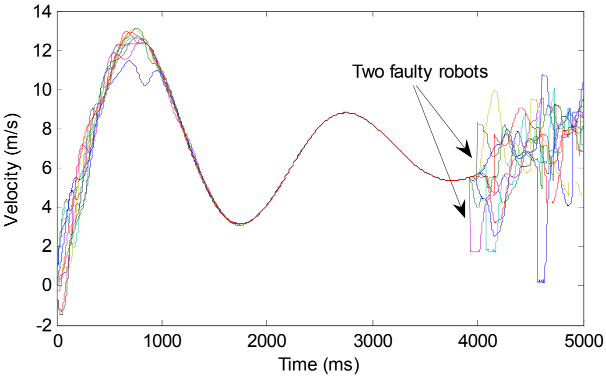

Figure 3. Velocities of the robots *without* FDI

Figure 3 demonstrates a scenario where two faulty robots occur at time 3750ms in the multi-robot system that has formed the flocking. Each curve in this figure shows the velocity of the robot with time. We assume that the communication between the robots is perfect and no FDI scheme is used. We can observe that the velocities of all the robots are not identical anymore after 4000ms, i.e., the flocking has been broken. The reason is that the faulty robots that are not isolated have a bad influence on the behaviours of the other robots in the system.

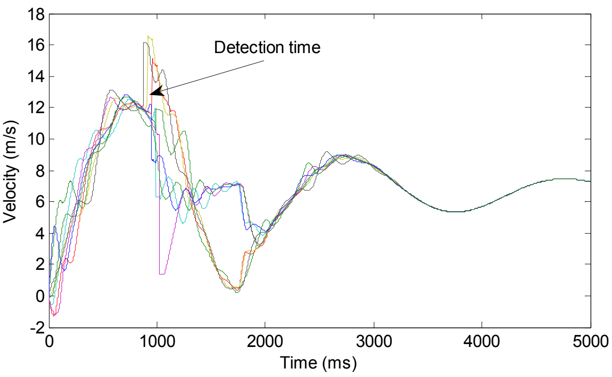

Figure 4. Velocities of the robots *with* FDI

Figure 4 demonstrates the velocities of the robots when the faulty robots are detected and isolated by the proposed FDI scheme. In this figure, two faulty robots occur at time 750ms. It can be observed that the robots reach a similar velocity again after a short confusion time caused by the faulty robots, i.e., the multi-robot system forms the flocking again. This phenomenon indicates that the proposed FDI scheme, which isolates the faulty robots, is effective. It is necessary to avoid faulty robots instead of just ignoring them as in the conventional isolation schemes (e.g., the schemes in [7] and [8]).

5.2 Benefit of the Proposed FDI Scheme

The benefits of using our FDI scheme for multi-robot systems are investigated in this subsection. We compare our FDI scheme with two typical methods in [7] and [13]. A Hinf-based structured FDI method (Hinf-SFDI) is introduced in [7]. Hinf-SFDI is a centralized FDI architecture. For the method introduced in [13], a scheme based on the unknown input observer (UIO) was used to detect faulty robots. Each robot has one observer corresponding to a neighbour. As in other conventional works, the isolation methods in [7] and [13] simply ignore the faulty robots.

In the simulation, we configure $n \in \{2, \cdots, 10\}$ robots and one faulty robot. All the robots have the same initial x and y axes, and their initial velocity is set to zero. Each simulation runs 20 times to determine the mean and standard deviation of the time of the FDI.

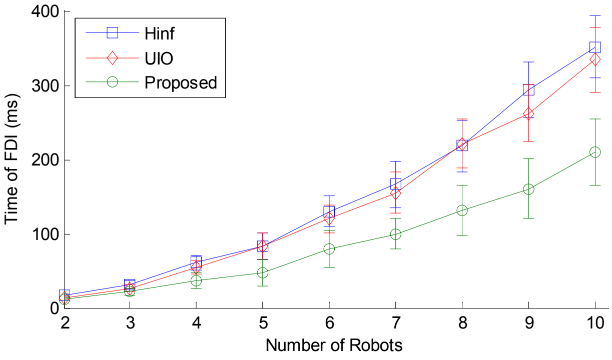

Figure 5. Time efficiency vs. the number of robots

In Figure 5, we present the time efficiency of the fault detection varying with the number of robots involved in the task of flocking in a multi-robot system. From this figure, we find that the proposed FDI method based on a local information exchange takes less time to implement the fault detection than the other two approaches based on a large number of statistics, the results of which are above. It indicates that the proposed FDI scheme is more efficient than the other two schemes.

Then we configure 10 robots and $q \in \{1, \cdots, 3\}$ faulty robots. Each simulation will run 50 times to determine the correct rate of fault detection.

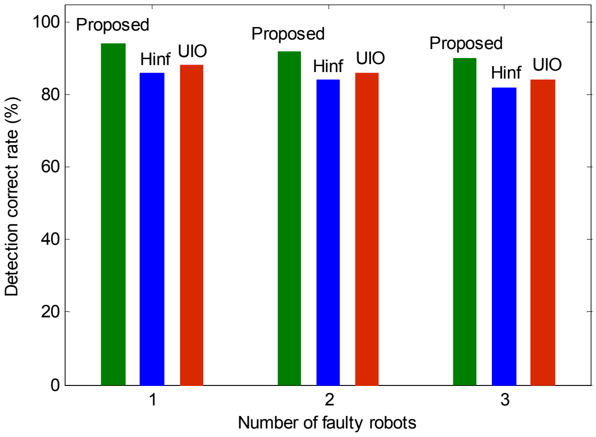

Figure 6. Rate of correction detection vs. the number of robots

Figure 6 shows the correct rate of fault detection varying with the number of faulty robots involved in the task of flocking in a multi-robot system. The correct rate of detection is defined as $r_{correct} = \frac{f_s}{f_t} \times 100\%$. Where f_s is defined as the number of successful fault detections and f_t is the total number of fault detections, including successful and failed. In this figure, it can be observed that the rate of correct detections of the proposed detection method is larger than the two other approaches. It indicates that the proposed fault detection scheme is more effective than the other two schemes.

6. Conclusions

In this paper, we considered the problem of faulty robots in multi-robot systems with double integrator dynamics. We proposed a distributed FDI scheme to resolve this problem. A detection method based on the local-information-exchange and sensor-measurement technologies was proposed to cover cases of both perfect communication and imperfect communication. Considering the fact that faulty robots still exert a bad influence on the system, we designed an isolation method that allows the non-faulty robots to avoid the faulty robots. The simulation results demonstrate that the application of the proposed FDI scheme is effective.

7. References

[1] Toner J. and Tu Y. (1998) Flocks, herds, and schools: A quantitative theory of flocking. Physical Review E.58:4828–4858.

[2] Cao Y.C., Yu W.W. and Ren W. (2013) An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. IEEE Transactions on Industrial Informatics. 9:427-438.

[3] Lee G. and Chong N.Y. (2008) Adaptive Flocking of Robot Swarms: Algorithms and Properties. IEICE Transactions on Communications. E91-B: 2848-2855.

[4] Yang Y., Xiong N., Chong N.Y. and Defago X. (2008) A Decentralized and Adaptive Flocking Algorithm for Autonomous Mobile Robots. In Proceedings of 3rd International Conference on Grid and Pervasive Computing. 2008 May 25-28; Kunming, China. United States: Inst. of Elec. and Elec. Eng. Computer Society. pp. 262-268.

[5] Lee G. and Chong N.Y. (2011) Low-Cost Dual Rotating Infrared Sensor for Mobile Robot Swarm Applications. IEEE Transactions on Industrial Informatics. 7: 277-286.

[6] Chen J. and Patton R.J. (1999) Robust model-based fault diagnose is for dynamic systems. Berlin. Springer.

[7] Meskin N. and Khorasani K. (2009) Fault Detection and Isolation of discrete-time Markovian jump linear systems with application to a network of multi-agent systems having imperfect communication channels. Automatica. 45:2032-2040.

[8] Scholtz E. and Lesieutre B.C. (2008) Graphical observer design suitable for large-scale DAE power systems. IEEE Conference on Decision and Control. 2008 Dec 09-11; Cancun, Mexico. New York: IEEE. pp. 2955–2960.

[9] Aldeen M. and Crusca F. (2006) Observer-based fault detection and identification scheme for power systems. IEEE Proceedings Generation Transmission and Distribution. 153:71–79.

[10] Ferrari R.M.G., Parisini T. and Polycarpou M.M. (2012) Distributed Fault Detection and Isolation of Large-scale Nonlinear Systems: an Adaptive Approximation Approach. IEEE Transactions on Automatic Control. 57:275-290.

[11] Zhang X. (2010) Decentralized fault detection for a class of large-scale nonlinear uncertain systems. American Control Conference. 2010 June 30-July 02; Baltimore, America. America: American Control Conference (ACC).pp. 5650-5655.

[12] Ding S.X. Zhang P., Chihaia C., Li W., Wang Y. and Ding E.L. (2008) Advanced design scheme for fault tolerant distributed networked control systems. In Proceedings of the 17th IFAC world congress. 2008 July 6-11; Seoul.

[13] Shames I., Teixeira A.H., Sandberg H. and Johansson K.H. (2011) Distributed fault detection for interconnected second-order systems. Automatica. 47:2757-2764.

[14] Ferrari R.M.G., Parisini T. and Polycarpou M.M. (2006) Dynamic distributed intrusion detection for secure multi-robot systems. Proceedings of the 2006 IEEE International Symposium on Intelligent Control. 2006 October 4-6, Munich, Germany. IEEE. pp.2742-2747.

[15] Ferrari R.M.G., Parisini T. and Polycarpou M.M. (2007) Distributed fault diagnosis with overlapping decompositions and consensus filters. Proceedings of American Control Conference. 2007 July 11-13; New York, USA. IEEE. pp. 693-698.

[16] Ferrari R.M.G., Parisini T. and Polycarpou M.M. (2009) Distributed fault diagnosis with overlapping decompositions: An adaptive approximation approach. IEEE Transactions on Automatic Control. 54:794-799.

[17] Olfati-Saber R. (2006) Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Transactions on Automatic Control. 51:401-420.

[18] Long Y. and Yang G.H. (2013) Fault detection for networked control systems subject to quantization and packet dropout. International Journal of Systems Science. 44: 1150-1159.

[19] Shao S.L., Zhao H., Zhu J. and Li D.Z. (2012) Design and Implementation of Prolonging Life Cycle of WSN in Data Collection. Journal of Northeastern University (Natural Science). 33:344-347.

[20] He J.X. (2013) Mixture model based multivariate statistical analysis of multiply censored environmental data. Advances in Water Resources. 59:15-24.

[21] Lee G. and Scott C. (2012) EM algorithms for multivariate Gaussian mixture models with truncated and censored data. Computational Statistics and Data Analysis. 56:2816-2829.

[22] Perlovsky L.I. and Deming R.W. (2013) Maximum Likelihood Joint Tracking and Association in Strong Clutter. Int. j. adv. robot. syst. 10:35. doi: 10.5772/52859.