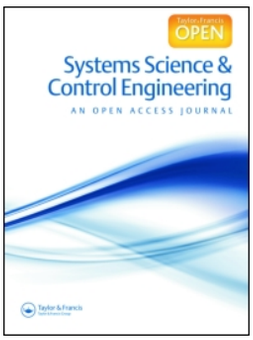


Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) 2164-2583 (Online) Journal homepage: http://www.tandfonline.com/loi/tssc20

Improved sensor fault detection, isolation, and mitigation using multiple observers approach

Zheng Wang, D. M. Anand, J. Moyne & D. M. Tilbury

To cite this article: Zheng Wang, D. M. Anand, J. Moyne & D. M. Tilbury (2017) Improved sensor fault detection, isolation, and mitigation using multiple observers approach, Systems Science & Control Engineering, 5:1, 70-96, DOI: 10.1080/21642583.2016.1278410

To link to this article: http://dx.doi.org/10.1080/21642583.2016.1278410

Published online: 23 Jan 2017.

SYSTEMS SCIENCE & CONTROL ENGINEERING: AN OPEN ACCESS JOURNAL, 2017
VOL. 5, 70–96
http://dx.doi.org/10.1080/21642583.2016.1278410

Taylor & Francis
Taylor & Francis Group

Improved sensor fault detection, isolation, and mitigation using multiple observers approach

Zheng Wang[a], D. M. Anand[b], J. Moyne[a] and D. M. Tilbury[a]

[a]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA; [b]Software and Systems Division, National Institute of Standards and Technology, Gaithersburg, MD, USA

ABSTRACT

Traditional fault detection and isolation (FDI) methods analyze a residual signal to detect and isolate sensor faults. The residual signal is the difference between the sensor measurements and the estimated outputs of the system based on an observer. The traditional residual-based FDI methods, however, have some limitations. First, they require that the observer has reached its steady state. In addition, residual-based methods may not detect some sensor faults, such as faults on critical sensors that result in an unobservable system. Furthermore, the system may be in jeopardy if actions required for mitigating the impact of the faulty sensors are not taken before the faulty sensors are identified. The contribution of this paper is to propose three new methods to address these limitations. Faults that occur during the observers' transient state can be detected by analysing the convergence rate of the estimation error. Open-loop observers, which do not rely on sensor information, are used to detect faults on critical sensors. By switching among different observers, we can potentially mitigate the impact of the faulty sensor during the FDI process. These three methods are systematically integrated with a previously developed residual-based method to provide an improved FDI and mitigation capability framework. The overall approach is validated mathematically, and the effectiveness of the overall approach is demonstrated through simulation on a five-state suspension system.

ARTICLE HISTORY
Received 14 August 2016
Accepted 29 December 2016

KEYWORDS
Fault detection and isolation; fault mitigation; fault diagnosis; dedicated observer scheme

1. Introduction

Sensors are considered to be the weak link in a system, especially when they transmit data through a vulnerable public network (e.g. the Internet) (Cardenas, Amin, & Sastry, 2008; Silva, Saxena, Balaban, & Goebel, 2012). A sensor fault in a physical system can be a major problem that may degrade the system performance, and even put the system in jeopardy in severe cases. The International Federation of Automatic Control (IFAC) SAFEPROCESS Technical Committee defines a *fault* as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition (Isermann, 1997; Schrick, 1997).

Fault detection and isolation (FDI) and fault mitigation mechanisms are crucial for protecting a system that is susceptible to sensor faults. Fault detection makes a binary decision on whether a fault has occurred or not. Fault isolation determines the location, and assesses the extent of the fault (Willsky, 1976). Fault mitigation reduces the effect of the fault (Dubey et al., 2007). Fault mitigation differs from Fault Tolerant Control, which aims at controlling the faulty system in the presence of the fault. In this paper, we propose three new methods to improve the performance of the traditional sensor fault detection, isolation and mitigation method.

1.1. Literature review

A significant amount of research has been carried out to detect and isolate sensor faults using observer-based methods due to their cost efficiency. The most common approach is to calculate residuals based on the difference between the measured outputs of the system and the estimated outputs of the observer, and compare residuals with certain thresholds to detect a sensor fault (Hwang, Kim, Kim, & Seah, 2010). For fault detection, a single observer or Kalman filter is sufficient (Clark, 1978). Fault isolation is usually addressed with a bank of observers, called a dedicated observer scheme (DOS) (Frank and Ding, 1997). In the DOS proposed by Clark (1978), each observer uses only one sensor for state estimation based on the assumption that the system is observable with any one of the sensors. Similarly, Bouibed, Seddiki, Guelton, and Akdag (2014) design multiple robust sliding mode observers with different subsets of sensor measurements and actuator inputs to generate residuals for both sensor

CONTACT Zheng Wang zhengwa@umich.edu

and actuator faults detection. Each sliding mode observer excluding a particular sensor or actuator is designed so that the residual generated by this observer is sensitive to a fault on this sensor or actuator, but insensitive to faults on other sensors and actuators.

In addition to observers designed using different inputs and outputs of the physical system, some DOSs consist of unknown input observers. Chadli, Akhenak, Maquin, and Ragot (2008) use a sliding mode observer to detect and isolate faults for nonlinear systems represented by multiple local linear models. The sliding mode observer is a linear combination of several local unknown input observers which can isolate the unknown disturbances to achieve robust FDI. Instead of isolating unknown disturbances, Methnani, Lafont, Gauthier, Damak, and Toumi (2013) consider a single additive fault as an unknown input, and attempt to reconstruct the fault with a bank of unknown input observers for each sensor and actuator.

An observer-based method can also be integrated with other methods for FDI. Rios, Edwards, Davila, and Fridman (2015) propose an approach that combines a high-order-sliding-mode multiple-observer technique and a multiple-model technique. This combined methodology has the advantages of both sliding mode observers and multiple models. The equivalent output injection of a sliding mode observer, which is a function of estimation error, can be used as a residual to detect faults in the system. Multiple models can be designed based on different fault scenarios to isolate faults.

Note that all of the methods mentioned above assume that the observers have reached their steady state so that the effect of the uncertain initial condition on a residual has died out. Otherwise, the methods may generate false alarms or missed alarms.

Some types of sensor faults may not be detected by traditional fault detection methods based on closed-loop observers. These include sensor faults caused by certain types of cyber attacks on a networked control system. Liu, Ning, and Reiter (2011) propose a cyber attack that injects false data in the sensor measurements and show that this attack cannot be detected by a static residual-based fault detector. To detect this type of sensor fault with a static residual-based fault detector, Bobba et al. (2010) propose to protect the subset of sensor measurements which are needed to ensure the system observability. Mo and Sinopoli (2010) and Mo and Sinopoli (2015) propose another kind of cyber attack which can bypass not only a static fault detector, but also one utilizing the system dynamics, such as a χ^2 fault detector. Theorem 2 in Mo and Sinopoli (2010) indicates that the system is not detectable when removing the faulty sensor, and as a result, the attacker could impose arbitrary large errors between the faulty sensor measurements and the actual system outputs. The faulty sensors in Mo and Sinopoli (2010) are a subset of the critical sensors that are indispensable for system observability. Instead of closed-loop observers, a method using open-loop observers is needed to detect critical sensor faults.

After a fault is detected and isolated, a control scheme is reconfigured (Edwards and Tan, 2006; Choy and Weyer, 2008). Although the diagnosis of a fault can lead to appropriate maintenance, the physical system may be in jeopardy during the diagnosis time. A timely mitigation technique during the FDI process may help maintain acceptable performance of the physical system. To the best of our knowledge, fault mitigation techniques that can be applied during the FDI process have not been developed for sensor faults (Lefebvre, 2014).

Based on our literature review, three research gaps are identified:

(1) how to detect a sensor fault during the observers' transient state;
(2) how to detect a sensor fault that can bypass a closed-loop observer-based method; and
(3) how to potentially mitigate the impact of a sensor fault during the FDI process.

1.2. Contribution

Given a linear time-invariant discrete-time system with multiple sensors, assuming only one sensor is faulty at a time, the general goals of this research are to

- determine the occurrence of a sensor fault;
- identify the faulty sensor and estimate the fault signal; and
- mitigate the impact of the sensor fault.

With respect to the previously mentioned research gaps, our contribution is to propose three new methods that respectively

(1) enable sensor fault detection and reduce false alarms during the observers' transient state;
(2) detect faults on critical sensors; and
(3) potentially mitigate the impact of the faulty sensor during the FDI process.

These three methods are then systematically integrated with a previously developed residual-based method to create a new FDI and mitigation framework. The first two contributions are shown in Figure 1.

The rest of the paper is organized as follows. In Section 2, an overview of problem statement and solution

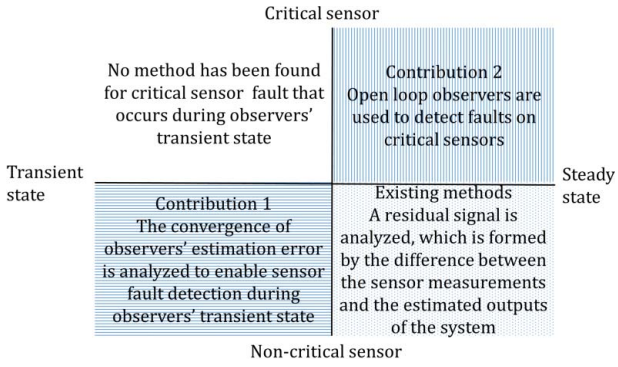


Figure 1. The first two contributions positioned in the space of critical vs. non-critical sensors, and observers' transient vs. steady state.

is provided. In Section 3, the mathematical description of the system is given. In Section 4, we introduce three new methods to address the research gaps, and the proposed methods are integrated with a previously developed method. In Section 5, an illustrative example validates the proposed algorithm. Conclusion and future work are given in Section 6.

2. Problem/solution overview

Given a linear time-invariant discrete-time system with multiple sensors, multiple observers, a state feedback controller, a residual-based fault detector, and the following assumption.

Assumption 2.1: Only one sensor is faulty at a time.

The specific goals of this paper are to

- propose a non-residual based method for sensor fault detection during the observers' transient state (Contribution 1);
- propose a method for critical sensor FDI (Contribution 2);
- propose a method to potentially mitigate the impact of the faulty sensor during the FDI process (Contribution 3); and
- systematically integrate the three new methods with a previously developed residual-based method for FDI and mitigation.

Based on the one faulty sensor assumption, the sensors can be divided into two sets. In one set, the sensors are indispensable for system observability. They are called *critical sensors* in this paper. In another set, the system is still observable with one sensor removed. These sensors are called *non-critical sensors*. Faults on non-critical sensors can be detected and isolated using a closed-loop observer which is designed excluding the faulty sensor. Since some sensor faults caused by certain types of cyber attacks (Mo and Sinopoli, 2010) on critical sensors are disguised as sensor noise, we use a bank of open-loop observers, which are artificial copies of the system fed with the same input signal (Bemporad, 2010). Two methods are running in parallel to determine which sensor is faulty. One method is based on closed-loop observers, while the other is based on open-loop observers.

To detect faults on non-critical sensors, we design one closed-loop observer with all of the sensor measurements, and multiple closed-loop observers each with one non-critical sensor excluded. Each observer is compared with all other observers, and the difference of estimated states between two observers is decoupled to calculate the estimation errors of these two observers. Thus, each observer has multiple calculated estimation errors. These calculated estimation errors are combined to determine the overall estimation error of the observer. The convergence ratio (CR) of the estimation error of an observer should be related to the designed state matrix of the observer, and not affected by the uncertain initial condition. But a sensor fault or a disturbance can change the CR of the estimation error. Based on this property, we propose the CR method for fault detection to reduce the false alarms during the observers' transient state. Bias analysis based on the calculated estimation errors is developed to distinguish a sensor fault from a disturbance. In the ideal case, the biases calculated based on the estimation errors of all observers should be the same when the system is under disturbance, but should be different under sensor fault. With bounded system noise, the bound of the difference between the calculated bias and the actual disturbance signal can be determined. Therefore, a threshold can be selected and compared with the difference between any two calculated biases. The threshold is specific for each pair of biases. If any one pair of them exceeds their threshold, the system is under sensor fault. Otherwise, the system is under disturbance.

To detect and isolate faults on critical sensors, we design multiple open-loop observers (MOLO), and analyze the residuals formed based on the difference between the measured outputs of the system and the estimated outputs. This method is only applicable to an open-loop stable or marginally stable system. If the system is open-loop unstable, the estimation error of an open-loop observer could diverge exponentially. To increase the estimation accuracy, we periodically update the states of multiple open-loop observers with the state estimated by the closed-loop observer using all of the sensor measurements when no fault is detected. There is a trade-off between estimation performance and the ability to detect a sensor fault. Therefore, we divide

the multiple open-loop observers into several groups. The observers within the same group are updated with the same update frequency. To mitigate the impact of noise, the update time steps of the observers in the same group are distributed evenly within one update period, and the residuals generated by the observers within the same group are averaged. The averaged residual is compared with a threshold, which is related to the known upper bound of noise and the update frequency. If the residual is larger than the threshold, then an alarm is triggered and the states of the open-loop observers of that group are not updated with the estimated state of the closed-loop observer until the alarm is cleared. Logic is provided to determine whether the system is under normal operation or under sensor fault based on which groups of open-loop observers trigger alarms. Then the residuals of the groups that trigger alarms are analysed to determine which sensor is faulty.

For fault mitigation, we also need to consider two cases: faults on critical sensors and faults on non-critical sensors. For faults on non-critical sensors, a closed-loop observer without the faulty sensor provides a better state estimation, based on which a state feedback controller can give the control input closest to the ideal control input. Thus, pinpointing this observer during the FDI process is the key for fault mitigation. Based on this property, we propose the calculated control input (CCI) method to switch among different observers, and potentially mitigate the impact of the fault on a non-critical sensor during the FDI process. For faults on critical sensors, none of the closed-loop observers can provide a good state estimation. If the system is open-loop stable, we can use an open-loop observer for state estimation to mitigate the impact of the sensor fault. If the system is marginally stable, the only way is to replace the faulty sensor.

We also need a residual-based method based on closed-loop observers for non-critical sensor fault isolation. In this paper, we use a method adopted from Bouibed et al. (2014), and call it the calculated outputs (CO) method. The method in Bouibed et al. (2014) consists of several sliding mode observers, each excluding a particular sensor or actuator. The sliding mode observer without the faulty sensor generates a significant residual signal. In contrast, we use a bank of Luenberger observers (or Kalman filters)[1] for the CO method. In this case, the observers with the faulty sensor generate significant residuals, and the CO method is not robust to disturbance in the system.

Table 1 shows the abilities of the CO, CR, MOLO, and CCI methods. Figure 2(a) shows when to use those four methods based on their abilities. We systematically integrate them as shown in Figure 2. During the observers' transient state, we use the CR method for non-critical sensor fault detection. If a sensor fault is detected and the observers have already reached their steady state, then we use the CO method for fault isolation. The CCI method is used for non-critical sensor fault mitigation during both the observers' transient state and steady state. Suppose a fault on a non-critical sensor starts at t_f, and it is detected and isolated at t_d. During the detection delay $t_d - t_f$, the CCI method may have already switched to the observer without the faulty sensor, providing estimated state to the controller. The MOLO method is running in parallel with the CR, CO, and CCI methods to detect and isolate a critical sensor fault.

Table 1. Abilities of the CO, CR, MOLO and CCI methods.

	Fault detection			
	Observers transient state	Observers steady state	Fault isolation	Fault mitigation
S_{nc}	CR	CR, CO MOLO	CO MOLO	CCI
S_c				

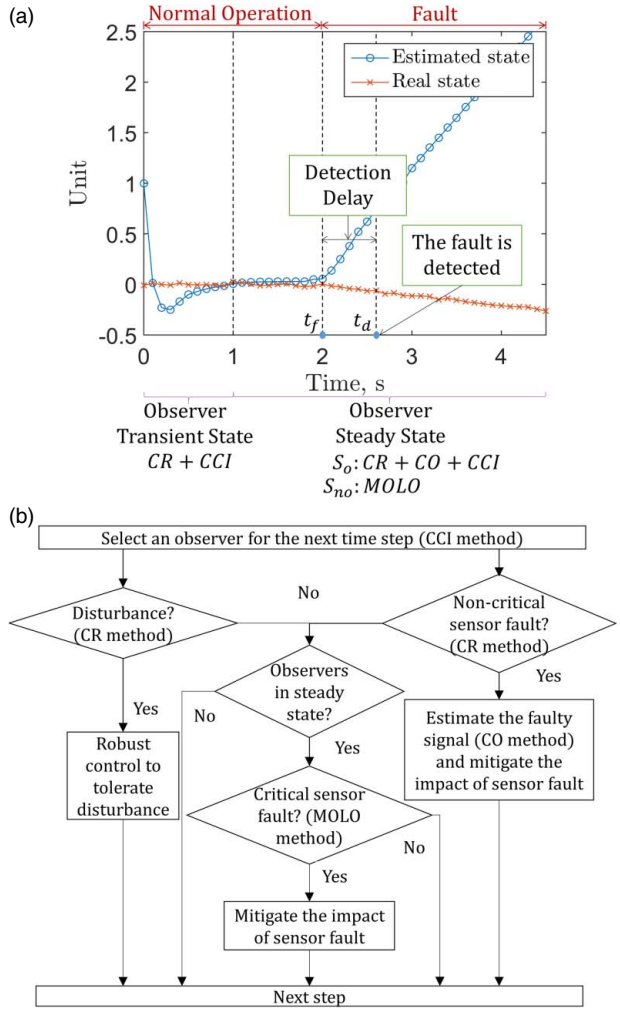


Figure 2. (a) Integration of the four methods: CO, CR, MOLO and CCI; (b) Flow chart of the integration.

3. Mathematical formulation of the problem

The analysis is carried out based on a linear time-invariant discrete-time system equipped with multiple observers, a state feedback controller and a residual-based fault detector.

3.1. Physical system

We model the physical system as a linear time-invariant discrete-time system. It has the following form:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Dd(k) + w(k), \\ y(k) &= Cx(k) + Ff(k) + v(k), \end{aligned} \tag{1}$$

where $x(k) \in \mathbb{R}^{n\times 1}$ is the system state, $y(k) \in \mathbb{R}^{m\times 1}$ is the sensor measurement, $u(k) \in \mathbb{R}^{p\times 1}$ is the control input, $d(k) \in \mathbb{R}^{s\times 1}$ is the unknown disturbance, $f(k) \in \mathbb{R}^{1\times 1}$ is the fault signal added to the sensor measurements, the process noise $w(k) \in \mathbb{R}^{n\times 1}$ and the sensor noise $v(k) \in \mathbb{R}^{m\times 1}$ are zero mean random vectors with bounds $\|w(k)\| \leq \omega$ and $\|v(k)\| \leq \upsilon$ (in this paper, we use $\|\cdot\|$ to denote $\|\cdot\|_\infty$), respectively, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{n\times s}$ are real constant matrices, and $F = [0 \cdots 1_{i_f} \cdots 0]^{\mathrm{T}} \in \mathbb{R}^{m\times 1}$ is a fault vector, with 0 corresponding to the faultless sensor, and 1_{i_f} corresponding to the faulty sensor, and i_f is the index for the faulty sensor. Based on Assumption 2.1, F has at most one non-zero element.

3.2. Closed-loop observers and open-loop observers

At each time step, all of the sensor measurements $y(k)$ and the control inputs $u(k)$ are gathered for state estimation. Two different kinds of observers can be utilized: closed-loop observers and open-loop observers.

3.2.1. Closed-loop observers

A closed-loop observer corrects the estimation with a feedback from the sensor measurements as shown in Figure 3.

Based on Assumption 2.1, sensor measurements can be divided into two sets: S_{nc} and S_c. S_{nc} contains m_o non-critical sensors. S_c contains critical sensors. In order to design multiple closed-loop observers, we need the following assumption:

Assumption 3.1: Set S_{nc} contains at least one non-critical sensor, i.e. $m_o > 0$.

We assume without loss of generality that the rows of the output matrix C are ordered such that the first m_o sensors are non-critical sensors. Thus, $m_o + 1$ closed-loop observers can be designed. Observer 0 uses all of the sensor measurements. Observer i uses all but sensor i ($i = 1, 2, \ldots, m_o$). For the closed-loop observers, we use Luenberger observers with the following form:

$$\begin{aligned} \tilde{x}_i(k+1) &= E_i\tilde{x}_i(k) + L_i y_i(k) + Bu(k) \\ &= E_i\tilde{x}_i(k) + L_i(C_i x(k) + v_i(k) + F_i f(k)) + Bu(k), \end{aligned} \tag{2}$$

where $\tilde{x}_i(k) \in \mathbb{R}^{n\times 1}$ is the state estimated by the closed-loop observer i ($i = 0, 1, 2, \ldots, m_o$), $y_i(k) \in \mathbb{R}^{(m-1)\times 1}$ is the sensor measurements used by observer i which does not contain the ith element of $y(k)$, $v_i(k)$ does not contain the ith sensor noise, $E_i = A - L_i C_i$, $L_i \in \mathbb{R}^{n\times(m-1)}$ is the observer gain, placing the eigenvalues of E_i in the unit circle, $C_i \in \mathbb{R}^{(m-1)\times n}$ is the output matrix for observer i and it does not contain the ith row of C, and $F_i \in \mathbb{R}^{(m-1)\times 1}$ is the fault vector of observer i which does not contain the ith element of F. If $i = i_f$, then $F_i = 0^{(m-1)\times 1}$. This means that observer i_f does not use the faulty sensor i_f for state estimation. The corresponding observer state matrix and observer gain that do not use the faulty sensor are E_{i_f} and L_{i_f}, respectively.

Remark: Our assumption indicates that the system is detectable without one of the sensors in S_{nc}. If the system is detectable and the noise is truncated Gaussian, the time varying gain of a Kalman filter converges in a few steps. Therefore, for the closed-loop observers, we can also use Kalman filters with the steady-state Kalman gains (Mo and Sinopoli, 2010).

3.2.2. Open-loop observers

An open-loop observer is running in parallel with the physical system, reproducing the behaviour of the system as shown in Figure 4.

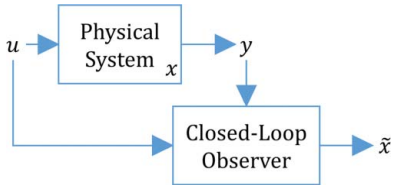


Figure 3. Structure of a closed-loop observer.

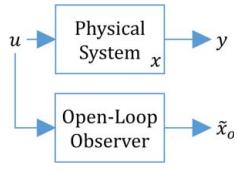


Figure 4. Structure of an open-loop observer.

Due to the lack of guaranteed estimation error convergence, the state of the open-loop observer is updated periodically by the closed-loop observer 0 which uses all of the sensor measurements. As mentioned in Section 2, we design M groups of open-loop observers, each group with N observers. The observers in the same group have the same update period. Then, an open-loop observer has the following form after one update period

$$\hat{x}_{g,i}(k+\kappa_{f,g}) = A^{\kappa_{f,g}}\tilde{x}_0(k) + \Sigma_{j=0}^{\kappa_{f,g}-1} A^j B u(k+\kappa_{f,g}-1-j), \tag{3}$$

where $\hat{x}_{g,i}(k) \in \mathbb{R}^{n\times 1}$ is the state estimated by the open-loop observer i in group g ($i = 1,\ldots,N, g = 1,\ldots,M$), and $\kappa_{f,g}$ is the update period of group g.

3.3. State feedback controller

A state feedback controller calculates a control command based on the system state, and applies it to the input of the system. The following assumption enables the utilization of a state feedback controller.

Assumption 3.2: The system is controllable.

Since the real state of the system is unknown, the controller can only use the state estimated by a closed-loop observer with the following form (Phillips and Nagle, 1994):

$$u(k) = K\tilde{x}_i(k), \tag{4}$$

where $K \in \mathbb{R}^{p\times n}$ is the controller gain placing the eigenvalues of $A + BK$ in the unit circle. Notice that an open-loop observer cannot provide as good of an estimation of performance as a closed-loop observer due to system noise. Therefore, we use a closed-loop observer for the state feedback controller if the system is under normal operation or under non-critical sensor fault. If an open-loop stable system is under critical sensor fault, then we can switch to an open-loop observer to help mitigate the impact of the sensor fault.

3.4. Residual-based fault detector

In this paper, the residual-based fault detector uses the CO method, which is adopted from Bouibed et al. (2014). In contrast to the method in Bouibed et al. (2014), the CO method consists of multiple Luenberger observers as shown in Equation (2), and generates the residuals based on the subtraction between the sensor measurements y_i (without the ith output) and the estimated outputs $C_i\tilde{x}_i$ as shown in Equation (5)

$$r_i(k) = (y_i(k) - C_i\tilde{x}_i(k))^{\mathsf{T}} Q_i (y_i(k) - C_i\tilde{x}_i(k)) \tag{5}$$

where Q_i is a real constant weighting matrix for observer i[2], and $r_i(k) \in \mathbb{R}$ is the residual generated based on observer i.

The residual generated based on observer 0 is compared with a selected threshold θ_{CO} to determine the occurrence of a sensor fault. When a sensor fault occurs, the closed-loop observer i_f, which does not use the faulty sensor, is not affected by the sensor fault, and thus provides a better state estimation compared to other observers .[3] Then the residual generated by observer i_f is smaller than the residuals generated by other observers ($i \neq i_f$). Therefore, we can locate the faulty sensor by finding the smallest residual among the observers from 1 to m_o. After the faulty sensor is located, the estimated fault signal is given by

$$\tilde{f}(k) = \{y(k) - C\tilde{x}_{i_f}(k)\}_{i_f} \tag{6}$$

where $\tilde{f}(k) \in \mathbb{R}^{1\times 1}$ is the estimated fault signal, and $\{y(k) - C\tilde{x}_{i_f}(k)\}_{i_f}$ is the i_fth element of the vector $y(k) - C\tilde{x}_{i_f}(k)$

Algorithm 1 gives the procedure of the CO method. First, we calculate the residuals based on different observers. Then, we use the residual of observer 0 for fault detection, and compare the rest of the residuals for fault isolation. Notice that the CO method cannot distinguish a disturbance from a sensor fault since Luenberger observer is not robust to disturbance. This issue is addressed by complementing the CO method with the CR

Algorithm 1: CO method for sensor FDI

```
function CO;
Input  : y(k), x̃_i(k) (i = 0, 1, ..., m_o)
Output: l_F, i_f
//Residual generation for all observers;
for i = 0 to m_o do
    r_i(k) = (y_i(k) − C_i x̃_i(k))^T Q_i (y_i(k) − C_i x̃_i(k));
end
//Fault detection;
if r_0(k) ≥ θ_CO then
    l_F = 1;
    //Fault isolation;
    i_f = min_i r_i(k);
    l_FB = i_f;
    f̃(k) = {y(k) − C x̃_{i_f}(k)}_{i_f};
else
    l_FB = 0;
end
```

method introduced in Section 4.3 which has the ability to distinguish a disturbance from a sensor fault.

3.5. Notations

Main notations are summarized here. x is the real system state. $\tilde{x}$ is the estimated state by a closed-loop observer. $\hat{x}$ is the estimated state by an open-loop observer. e is the estimation error between observer state and system real state. $e_{\mu,\nu}$ is the difference of estimated states between two closed-loop observers μ and ν. $\tilde{e}_{\mu(\nu)}$ is the calculated estimation error of closed-loop observer μ, and the calculation is based on $e_{\mu,\nu}$. Detailed notations are shown in Table A.7

4. Framework components description and integration

Throughout this section, a simple system of a moving object is utilized as an illustration. First, we simulate sensor faults on the moving object system equipped with the CO method-based fault detector to understand its limitations. Then, three new methods are introduced and analysed in the deterministic case (noise free). The impact of random system noise is discussed for each method thereafter. The simulation result shows the improvements of the proposed methods compared to the CO method. Finally, we provide an algorithm to integrate the CO method and the three new methods.

4.1. Moving object system

The moving object system is a 1 kg mass moving along a horizontal line. Two sensors are measuring the two outputs: the velocity y_v and the position y_p, respectively. A state feedback controller applies a horizontal force on the mass. The sampling time is 0.1 s. The system has initial state $(0, 0)$, process noise with bound 0.001 (m/s or m), and sensor noise with bound 0.01 (m/s or m). The initial states of the observers are chosen as $(1, 0.5)$ [4]. The state space representation of the moving object system is shown as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k), \\ y(k) &= Cx(k) + v(k), \end{aligned} \tag{7}$$

where $x = \begin{bmatrix} x_v \\ x_p \end{bmatrix}$, $y = \begin{bmatrix} y_v \\ y_p \end{bmatrix}$, $A = \begin{bmatrix} 1 & 0 \\ 0.1 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0.1 \\ 0.005 \end{bmatrix}$, and $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

By checking the rank of observability matrix, the moving object system is observable with y_v and y_p or only y_p, but unobservable with only y_v. Therefore, $y_v \in S_{nc}$, and $y_p \in S_c$. Two observers can be designed with observer poles placed at [0.1 0.11]. Observer 0 uses both sensor measurements y_v and y_p. Observer 1 uses only y_p.

Two fault scenarios are considered:

(1) fault α: a ramp signal with slope 0.05 m/s^2 (0.005 m/s per time step) added to the velocity sensor y_v, saturating at 1 m/s;
(2) fault β: a ramp signal with slope 0.001 m/s (0.0001 m per time step) added to the position sensor y_p, saturating at 1 m.

Both faults start at 10 s and run until the end of the simulation. Here, we consider ramp faults with small slopes because they are hard to detect compared to ramp faults with large slopes or step faults with large magnitudes.

4.2. The impact of sensor faults

Two fault cases are run on the moving object system equipped with the CO method-based fault detector to show its limitations. Based on each limitation, a new method is discussed and proposed.

Figures 5–7 show the estimated position states of both observers $\tilde{x}_0, \tilde{x}_1$, the real state x, and the sensor measurement y of the system equipped with the CO method-based fault detector under fault α, β and normal operation, respectively. In both Figures 5 and 6, false alarms are generated by the CO method during the observers' transient state, which is about 0.2 s, when the system is actually under normal operation. From Figure 7, it can be seen that the imperfect initial state of the observers causes the CO method to generate false alarms. According to Equation (5), the residual $r_i(k)$ of the CO method is a function of the observer's estimation error under normal operation. A large estimation error makes the residual exceed the threshold, causing false alarms during the observers' transient state. To enable fault detection during observers' transient state, the CR method, described in Section 4.3, which utilizes the CR of observers' estimation error, will be applied.

As shown in Figure 6(b), when the system is under fault β, no alarm is generated since the fault is not detected by the CO method-based fault detector. The reason behind this behaviour is that the system is not detectable when the position sensor y_p is removed, and the fault signal is changing slightly at each time step to avoid significant change in the residuals. An open-loop observer (3) does not use any sensor for state estimation. Thus, this issue can be potentially addressed by the MOLO method introduced in Section 4.4.

As shown in Figure 5(b, c), although the CO method successfully locates the faulty sensor and then the system switches to observer 1 for state estimation after

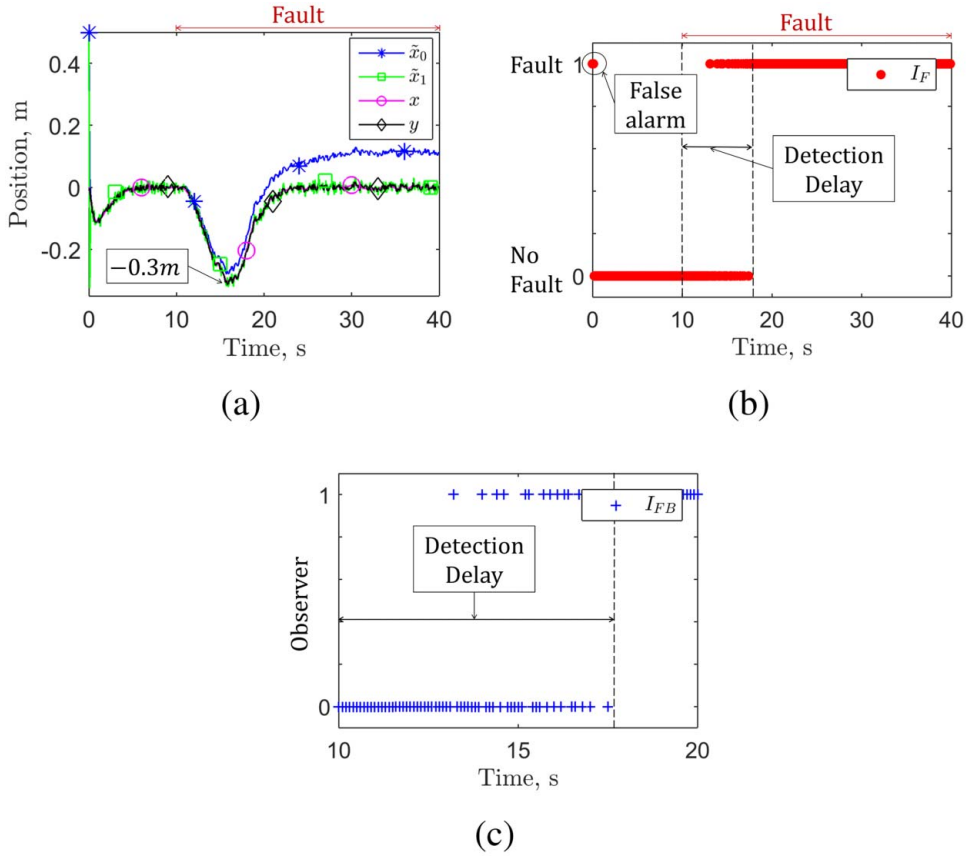


Figure 5. (a) The estimated position states of both observers $\tilde{x}_0, \tilde{x}_1$, the real state x, and the sensor measurement y of the system under fault α; (b) fault alarms I_F of the CO method under fault α; and (c) observer index I_{FB} selected for the state feedback controller under fault α.

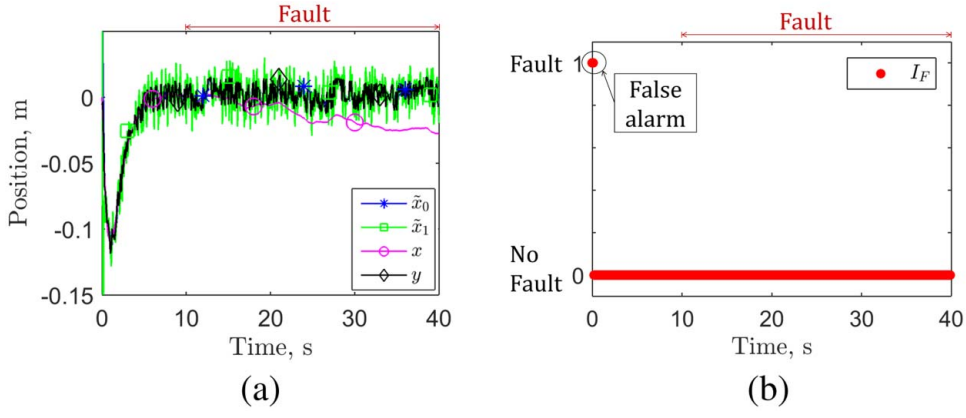


Figure 6. (a) The estimated position states of both observers $\tilde{x}_0, \tilde{x}_1$, the real state x, and the sensor measurement y of the system under fault β; (b) fault alarms I_F of the CO method under fault β.

18 s, there is 8 s detection delay and the system switches between the two observers during 13 s to 18 s. This is caused by the relatively small fault signal compared to the system noise and the threshold. Thus, the faulty sensor cannot be located immediately. This detection delay makes the maximum absolute value of the position of the mass reach 30 cm as shown in Figure 5(a). The direct reason for this divergence is the discrepancy of the control input provided by the observer-based state feedback controller. To address this issue, we need to switch to the closed-loop observer without the faulty sensor as soon as possible and continue using that observer during the FDI

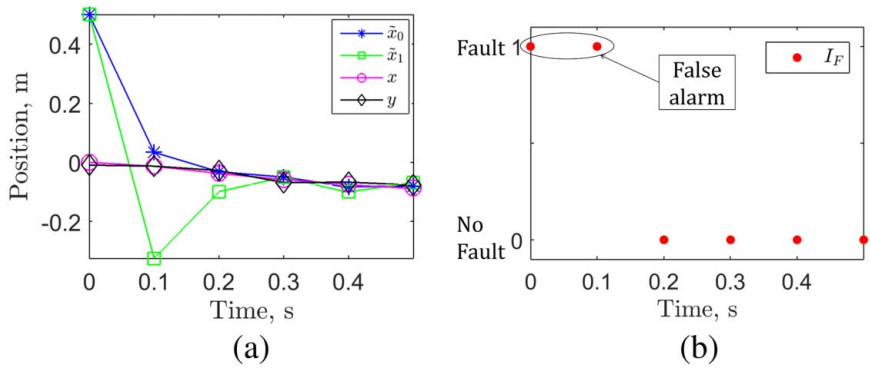


Figure 7. (a) The estimated position states of both observers $\tilde{x}_0, \tilde{x}_1$, the real state x, and the sensor measurement y of the system during the observers' transient state under normal operation; (b) fault alarms I_F during the observers' transient state.

process. Thus, we propose the CCI method to compare the control input calculated based on the state estimated by each closed-loop observer with an 'ideal' control input calculated based on the state estimated by an open-loop observer, and to switch to the observer which gives the smallest difference between the CCI and the ideal control input. This method has the potential to mitigate the impact of a non-critical faulty sensor during the FDI process. The maximum absolute value of the position of the system under the CO method will be compared with that under the CCI method in Section 4.5.

4.3. CR method for fault detection during transient and steady state

This method is proposed to detect the occurrence of an anomaly based on the convergence of estimation error. It enables fault detection during the observers' transient state. To achieve robust fault detection, a disturbance in the system is distinguished from a sensor fault by analysing the bias of the estimation error. First, this method is introduced on an ideal control system. Then the impact of the process noise and the sensor noise are discussed.

4.3.1. Ideal system case

Three different situations are considered for this method: normal operation, disturbance, and sensor fault. Estimation error e_i of closed-loop observer i, and the difference of estimated states $e_{\mu,\nu}$ between two closed-loop observers μ and ν under three situations are shown in Equation (8) through Equation (13).

Under normal operation

$$e_i(k+1) = x(k) - \tilde{x}_i(k) = E_i e_i(k), \tag{8}$$

$$\begin{aligned} e_{\mu,\nu}(k+1) &= \tilde{x}_\mu(k+1) - \tilde{x}_\nu(k+1) \\ &= E_\nu e_\nu(k) - E_\mu e_\mu(k). \end{aligned} \tag{9}$$

Under disturbance

$$e_i(k+1) = E_i e_i(k) + Dd(k), \tag{10}$$

$$e_{\mu,\nu}(k+1) = E_\nu e_\nu(k) - E_\mu e_\mu(k). \tag{11}$$

Notice that Equations (9) and (11) are the same.

Under sensor fault

$$e_i(k+1) = E_i e_i(k) - L_i F_i f(k), \tag{12}$$

$$e_{\mu,\nu}(k+1) = E_\nu e_\nu(k) - E_\mu e_\mu(k) - (L_\nu F_\nu - L_\mu F_\mu) f(k). \tag{13}$$

The first step of the CR method is to calculate the estimation error of each closed-loop observer. The dynamics of $e_{\mu,\nu}$ under both normal operation and disturbance are the evolution of the estimation errors of the two closed-loop observers e_μ and e_ν. Therefore, the estimation errors of both observers can be decoupled over two time steps. However, the dynamics of $e_{\mu,\nu}$ under sensor fault involves two unknown fault vectors F_μ and F_ν, and the unknown fault signal $f(k)$. Thus, the estimation errors cannot be correctly decoupled under sensor fault. Lemma 4.1 gives the formulas for estimation error decoupling of any two different observers.

Lemma 4.1: *Given an ideal control system* (1) *with* $w(k) = 0$ *and* $v(k) = 0$, *the calculated estimation error* $\tilde{e}_{\mu(\nu)}$ *and* $\tilde{e}_{\nu(\mu)}$ *are derived based on Equations* (14) *and* (15) *respectively, with the following results:*

(1) *When the system is under normal operation or under disturbance,* $\tilde{e}_{\mu(\nu)} = e_\mu$ *and* $\tilde{e}_{\nu(\mu)} = e_\nu$;
(2) *When the system is under sensor fault,* $\tilde{e}_{\mu(\nu)} \neq e_\mu$ *and* $\tilde{e}_{\nu(\mu)} \neq e_\nu$ *if* $L_\nu F_\nu \neq L_\mu F_\mu$.

$$\tilde{e}_{\mu(\nu)}(k) = (E_\nu - E_\mu)^{-1}(e_{\mu,\nu}(k+1) - E_\nu e_{\mu,\nu}(k)) \tag{14}$$

$$\tilde{e}_{\nu(\mu)}(k) = (E_\nu - E_\mu)^{-1}(e_{\mu,\nu}(k+1) - E_\mu e_{\mu,\nu}(k)), \tag{15}$$

where $E_\nu - E_\mu = A - L_\nu C_\nu - A + L_\mu C_\mu = L_\mu C_\mu - L_\nu C_\nu$.

Remark: We design L_μ and L_ν to make $E_\nu - E_\mu$ invertible.

Proof: (1) Under normal operation or under disturbance, the evolution of $e_{\mu,\nu}$ (9) and $e_{\mu,\nu}(k) = e_\nu(k) - e_\mu(k)$ are substituted to Equation (14),

$$\tilde{e}_{\mu(\nu)}(k) = (E_\nu - E_\mu)^{-1}(E_\nu e_\nu(k) - E_\mu e_\mu(k) - E_\nu(e_\nu(k) - e_\mu(k))) = e_\mu(k). \quad (16)$$

Similarly, $\tilde{e}_{\nu(\mu)}(k) = e_\nu(k)$.

(2) Under sensor fault, the evolution of $e_{\mu,\nu}$ (13) and $e_{\mu,\nu}(k) = e_\nu(k) - e_\mu(k)$ are substituted to Equation (14),

$$\begin{aligned}\tilde{e}_{\mu(\nu)}(k) &= (E_\nu - E_\mu)^{-1}(E_\nu e_\nu(k) - E_\mu e_\mu(k) \\ &\quad - (L_\nu F_\nu - L_\mu F_\mu)f(k) - E_\nu(e_\nu(k) - e_\mu(k))) \\ &= e_\mu(k) - (E_\nu - E_\mu)^{-1}(L_\nu F_\nu - L_\mu F_\mu)f(k) \quad (17)\end{aligned}$$

if $L_\nu F_\nu \neq L_\mu F_\mu$, then $\tilde{e}_\mu(k) \neq e_\mu(k)$.

Similarly, $\tilde{e}_{\nu(\mu)}(k) \neq e_\nu(k)$ if $L_\nu F_\nu \neq L_\mu F_\mu$. ■

Based on Lemma 4.1, m_o estimation errors can be calculated for each observer. In ideal system case, these m_o estimation errors are averaged to be the estimation error $\tilde{e}_i$ of each observer. The combination of m_o estimation errors for a noisy system is introduced in Section 4.3.2.

After getting the estimation errors of all of the observers, the next step is to analyze the convergence behaviour of the estimation error of each observer. For each observer, $\tilde{e}_i \in \mathbb{R}^{n\times 1}$ contains n states. The evolution matrix E_i of the estimation error of observer i may not be a diagonal matrix. This causes the coupling of estimation errors between different states, which makes the ratio of estimation error of each state non-constant. Therefore, instead of using the estimation errors directly, we diagonalize the evolution matrix E_i using a basis of eigenvectors V_i. The diagonal elements in the diagonalized matrix $E_{\Lambda,i}$ (eigenvalues of E_i), where $E_{\Lambda,i} = (V_i)^{-1}E_iV_i$, are the same as the time-invariant observer poles. Then, we can define the CR to specify the convergence of the estimation error for each state.

Definition 4.2 (CR): Ratio of the absolute value of estimation error along with time step k (18) is called the CR.

$$\{cr_i\}_j(k) = \frac{1}{\kappa_{CR}}\left[\left|\frac{\{\tilde{e}_{\Lambda,i}(k)\}_j}{\{\tilde{e}_{\Lambda,i}(k-1)\}_j}\right| + \sum_{k_i=2}^{\kappa_{CR}}\sqrt[k_i]{\left|\frac{\{\tilde{e}_{\Lambda,i}(k)\}_j}{\{\tilde{e}_{\Lambda,i}(k-k_i)\}_j}\right|}\right], \quad (18)$$

where $\tilde{e}_{\Lambda,i}(k) = (V_i)^{-1}\tilde{e}_i(k)$, $\{\tilde{e}_{\Lambda,i}\}_j(k)$ is the jth element in $\tilde{e}_{\Lambda,i}(k)$, and κ_{CR} is a selected integer to average the CRs over κ_{CR} time steps.

Based on the above definition, the CR of each estimation error $\{cr_i\}_j$ is actually the same as the corresponding jth observer pole under normal operation. This is also indicated by

$$\left|\frac{\{\tilde{e}_{\Lambda,i}(k)\}_j}{\{\tilde{e}_{\Lambda,i}(k-k_i)\}_j}\right| = \left|\frac{\{(E_{\Lambda,i})^{k_i}\tilde{e}_{\Lambda,i}(k-k_i)\}_j}{\{\tilde{e}_{\Lambda,i}(k-k_i)\}_j}\right| = \left|(\{E_{\Lambda,i}\}_j)^{k_i}\right|, \quad (19)$$

where $\{E_{\Lambda,i}\}_j$ is the jth diagonal element of matrix $E_{\Lambda,i}$. Therefore,

$$\{cr_i\}_j(k) = |\{E_{\Lambda,i}\}_j|, \quad \forall k \geq 0. \quad (20)$$

An anomaly (a disturbance or a sensor fault) can change the CR of the estimation error in two possible cases. One case is that an anomaly makes the estimation error converge faster to zero. The other case is that an anomaly makes the estimation error converge slower or diverge to some other non-zero value. In ideal system case, the anomalies in both cases can be detected by comparing the CRs with observer poles. If a CR is larger or smaller than its corresponding observer pole, then this CR indicates the occurrence of an anomaly. Definition 4.2 shows that $(m_o + 1) \times n$ CRs are calculated at each time step. Because of the system noise, it is possible that some of the CRs indicate an anomaly even though there is no anomaly. So we define the system as an anomalous system if as least half of the CRs indicate anomaly. A threshold is selected for noisy system as discussed in Section 4.3.2.

To achieve robust fault detection, a disturbance should be distinguished from a sensor fault (Hwang et al., 2010). For this purpose, bias is defined

Definition 4.3 (Bias): The term $b(k)$ in an affine function $x(k+1) = Ax(k) + b(k)$ is called bias.

Under disturbance, the bias is $Dd(k)$, which is the same for all observers. Under sensor fault, the bias is $-L_iF_if(k)$, which is different for different observers. The disturbance signal $d(k)$ can be correctly determined when the system is under disturbance because of the correct decoupled estimation error. In contrast, the fault signal cannot be correctly determined because of the incorrect decoupled estimation error and unknown F_i. Based on this analysis, the bias is calculated based on each observer according to Equation (22) in Theorem 4.4.

Theorem 4.4: *Given an ideal control system* (1) *with* $w(k) = 0$ *and* $v(k) = 0$, *the biases* $\tilde{d}_{\mu(\nu)}(k)$ *and* $\tilde{d}_{\Lambda,\mu(\nu)}(k)$ *are calculated according to Equations* (21) *and* (22) *respectively, with the following results:*

(1) *When the system is under disturbance,*

$$\forall \mu, \nu = 0, 1, \ldots, m_o \wedge \mu \neq \nu,$$
$$\tilde{d}_{\mu(\nu)}(k) = \tilde{d}_{\Lambda,\mu(\nu)}(k) = d(k).$$

(2) *When the system is under sensor fault,*

$$\begin{aligned} &\forall \mu, \nu = 0, 1, \ldots, m_o \wedge \mu \neq \nu, \\ &\tilde{d}_{\mu(\nu)}(k) = \tilde{d}_{\nu(\mu)}(k), \\ &\tilde{d}_{\Lambda,\mu(\nu)}(k) \neq \tilde{d}_{\Lambda,\nu(\mu)}(k) \quad \text{if} \quad V_\mu \neq V_\nu, \end{aligned} \tag{21}$$

$$\begin{aligned} \tilde{d}_{\mu(\nu)}(k) &= (D^{\mathsf{T}}D)^{-1}D^{\mathsf{T}}[\tilde{e}_{\mu(\nu)}(k+1) - E_\mu \tilde{e}_{\mu(\nu)}(k)], \\ \tilde{d}_{\Lambda,\mu(\nu)}(k) &= ((D_{\Lambda,\mu})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}} \\ &\quad [\tilde{e}_{\Lambda,\mu(\nu)}(k+1) - E_{\Lambda,\mu}\tilde{e}_{\Lambda,\mu(\nu)}(k)], \end{aligned} \tag{22}$$

where $D_{\Lambda,\mu} = (V_\mu)^{-1}D$, *and* $E_{\Lambda,\mu} = (V_\mu)^{-1}E_\mu V_\mu$.

Proof: See Appendix 2 ■

Theorem 4.4 shows that $(m_o + 1) \times m_o$ biases are calculated at each time step. Each bias is compared with other biases. If any two biases disagree with each other, then the system is under sensor fault. If the bias analysis indicates that the system is under disturbance, then we can determine the disturbance signal by averaging all of the biases. The combination of all of the biases for a noisy system is introduced in Section 4.3.2.

4.3.2. Noisy system case

Lemma 4.1 and Theorem 4.4 in Section 4.3.1 show the effectiveness of the CR method in fault detection when the system is ideal. In practice, we also need to consider system noise: process noise and sensor noise. When only process noise exists in the system, the output of the system can still be correctly measured, which means the state of the system can be exactly known. Therefore, process noise does not affect the accuracy of the estimation error calculation. However, when sensor noise contaminates the sensor measurements, the estimation error cannot be correctly calculated. The boundedness of sensor noise ensures the boundedness of the error of estimation error $\|\tilde{e}_{\mu(\nu)} - e_\mu\|$. Lemmas 4.5 and 4.6 give the impact of process noise and the impact of sensor noise on the estimation error calculation, respectively.

Lemma 4.5: *Given a control system* (1) *with bounded process noise and* $v(k) = 0$, $\tilde{e}_{\mu(\nu)}(k) = e_\mu(k)$ *still holds when the system is under normal operation or under disturbance.*

Proof: When the system is subject to the process noise $w(k)$, the estimation error evolution becomes

$$e_\mu(k+1) = E_\mu e_\mu(k) + w(k), \tag{23}$$

Then the difference of the estimated states between two observers μ and ν is the same as Equation (9). By substituting Equation (9) into Equation (14), the calculated estimation error becomes

$$\begin{aligned} \tilde{e}_{\mu(\nu)}(k) = (E_\nu - E_\mu)^{-1}[&E_\nu e_\nu(k) - E_\mu e_\mu(k) \\ &- E_\nu(e_\nu(k) - e_\mu(k))] = e_\mu(k). \end{aligned} \tag{24}$$

■

Lemma 4.6: *Given a control system* (1) *with bounded sensor noise and* $w(k) = 0$, $\|\tilde{e}_{\mu(\nu)}(k) - e_\mu(k)\|$ *is bounded by* $\|(E_\nu - E_\mu)^{-1}\|(\|L_\nu\| + \|L_\mu\|)\upsilon$.

Proof: See Appendix 3. ■

Lemma 4.6 shows that the impact of sensor noise is different for estimation errors calculated based on different pairs of observers. Thus, when calculating the estimation error of each observer, we combine its m_o decoupled estimation errors with different weighting ratios. The weighting ratio is determined based on the bound of $\|\tilde{e}_{\mu(\nu)} - e_\mu\|$. If the bound of $\|\tilde{e}_{\mu(\nu)} - e_\mu\|$ is larger, then the corresponding weighting ratio is smaller. The combined estimation error and the weighting ratio are shown as follows:

$$\begin{aligned} \tilde{e}_\mu(k) &= \Sigma_{\nu=0,\nu\neq\mu}^{m_o} \phi_\nu \tilde{e}_{\mu(\nu)}, \\ \Sigma_{\nu=0,\nu\neq\mu}^{m_o} \phi_\nu &= 1, \\ \phi_\nu &= \frac{1}{m_o - 1} \frac{\Sigma_{j=0,j\neq\mu,j\neq\nu}^{m_o} \bar{e}_{\mu(j)}}{\Sigma_{j=0,j\neq\mu}^{m_o} \bar{e}_{\mu(j)}}, \\ \bar{e}_{\mu(\nu)} &= \|(E_\nu - E_\mu)^{-1}\|(\|L_\nu\| + \|L_\mu\|)\upsilon. \end{aligned} \tag{25}$$

The sensor noise affects the accuracy of estimation error decoupling, thus affecting the CRs and anomaly detection. Lemma 4.6 indicates that the impact of sensor noise can be mitigated by choosing the observer gains L_μ and L_ν with smaller norms. An observer gain with a smaller norm, however, may reduce the convergence speed of the estimation error. Thus, there is a trade-off in choosing observer gains. The impact of sensor noise on the CRs can also be mitigated via averaging over κ_{CR} time steps as shown in Definition 4.2. In addition to techniques for mitigating the impact of sensor noise, a threshold θ_{CR} for CRs should be selected to balance the tolerance of system noise and the ability to detect an anomaly. As discussed in Section 4.3.1, the CRs are the same as the observer

poles under normal operation but they are different from observer poles under anomaly in ideal case. However, the observer poles are usually selected to be close to 0 to ensure fast observer's estimation error convergence and noise exists on the system. So we select a upper threshold θ_{CR}, which is larger than the largest observer pole but less than one. Then the sensor fault, which makes the estimation error converge faster, cannot be detected by the CR method. With the threshold θ_{CR}, the lower bound of the fault signal that can be detected is ($\kappa_{CR} = 1$)

$$\begin{aligned}\|f(k)\| \geq\ & \|\{(E_\nu - E_\mu)^{-1}\}_j(L_\nu F_\nu - L_\mu F_\mu)\|^{-1} \\ & (\theta_{CR}\|\{e_\mu(k-1)\}_j\|(1+\theta_{CR})\|\{(E_\nu - E_\mu)^{-1}\}_j\| \\ & (\|L_\nu\| + \|L_\mu\|)\upsilon + \|\{e_\mu(k)\}_j\|. \end{aligned} \tag{26}$$

This lower bound is proportional to the threshold θ_{CR} and the bound of the sensor noise υ.

Both the process noise and the sensor noise affect the accuracy of the bias calculation, thus affecting the ability to distinguish a disturbance from a sensor noise. Based on the boundedness of the process noise and the sensor noise, the error of the bias calculation $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ is also bounded when the system is under disturbance. Lemmas 4.7 and 4.8 give the bound of $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ under disturbance when the system is subject to either the process noise or the sensor noise, respectively.

Lemma 4.7: *Given a control system* (1) *with bounded process noise and* $\upsilon(k) = 0$, $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ *is bounded by* $\|((D_{\Lambda,\mu})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}}(V_\mu)^{-1}\|\omega$.

Proof: See Appendix 4. ■

Lemma 4.8: *Given a control system* (1) *with bounded sensor noise and* $w(k) = 0$, $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ *is bounded by* $\|((D_{\Lambda,\mu})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}}(V_\mu)^{-1}\|(1 + \|E_\mu\|)$ $\|(E_\nu - E_\mu)^{-1}\|(\|L_\nu\| + \|L_\mu\|)\upsilon$.

Proof: See Appendix 5. ■

Combining Lemmas 4.7 and 4.8, the bound of the error of the bias calculation is

$$\begin{aligned}\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\| \leq\ & \|((D_{\Lambda,\mu})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}}(V_\mu)^{-1}\| \\ & (\omega + (1+\|E_\mu\|)\|(E_\nu - E_\mu)^{-1}\|(\|L_\mu\| + \|L_\nu\|)\upsilon). \end{aligned} \tag{27}$$

Notice that the bounds are different for biases calculated based on different pairs of observers, and that they are all zero-mean. Based on the bounds, one specific threshold $\theta_{d,\mu(\nu),\zeta(\eta)}$ ($\mu, \nu, \zeta, \eta = 0, .., m_o \wedge \mu \neq \nu \wedge \zeta \neq \eta$) can be selected to compare with the difference between any two biases averaged over κ_{CR} time steps, thus determining whether the system is under disturbance or sensor fault. If any one pair of the biases exceeds the corresponding threshold, then the system is under sensor noise. Otherwise, the system is under disturbance.

If the system is under disturbance, the combination of the weighted biases is considered as the disturbance signal. The weighting ratio of each bias is determined based on the bound of $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$. If the bound is larger, then the corresponding weighting ratio is smaller. The combined bias and the weighting ratio are shown as follows:

$$\begin{aligned}\tilde{d}(k) &= \Sigma_{\nu=0,\nu\neq\mu}^{m_o}\Sigma_{\mu=0}^{m_o}\psi_{\mu(\nu)}\tilde{d}_{\Lambda,\mu(\nu)}(k) \\ &\Sigma_{\nu=0,\nu\neq\mu}^{m_o}\Sigma_{\mu=0}^{m_o}\psi_{\mu(\nu)} = 1, \\ \psi_{\mu(\nu)} &= \frac{1}{(m_o+1)m_o - 1} \\ &\frac{\Sigma_{j=0,j\neq i}^{m_o}\Sigma_{i=0,i\neq\mu}^{m_o}\bar{d}_{i(j)} + \Sigma_{j=0,j\neq\nu}^{m_o}\bar{d}_{\mu(j)}}{\Sigma_{j=0,j\neq i}^{m_o}\Sigma_{i=0}^{m_o}\bar{d}_{i(j)}}, \\ \bar{d}_{\mu(\nu)} &= \|((D_{\Lambda,\mu})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}}(V_\mu)^{-1}\| \\ &\times (\omega + (1+\|E_\mu\|)\|(E_\nu - E_\mu)^{-1}\|(\|L_\mu\| + \|L_\nu\|)\upsilon). \end{aligned} \tag{28}$$

Algorithm 2 shows the procedure of the CR method. The CR method contains three steps. The first step is to calculate the estimation error for each observer. Then the CRs of the estimation errors are used to detect the occurrence of an anomaly. If an anomaly is detected, biases are calculated and analysed to determine whether the anomaly is a disturbance or a sensor fault.

Figure 8 shows the fault alarms generated by the CR method under fault α. During the observers' transient state, false alarms are eliminated compared to Figures 5(b), 6(b) and 7(b). When the system is under sensor fault α, there is about 2 s detection delay, which is caused by κ_{CR} for averaging the CR and the threshold θ_{CR}. The detection delay is decreased compared to the 8 s detection delay in Figure 5.

4.4. MOLO method for critical sensor FDI

The MOLO method has the potential to detect and isolate faults on critical sensors. It consists of multiple groups of open-loop observers. The states of the open-loop observers are updated periodically by the estimated state of the closed-loop observer using all of the sensor measurements. The open-loop observers in different groups have different update frequencies. Residuals are formed based on the difference between the measured outputs of the system and the estimated outputs of the open-loop observers. Then the averaged residual is analysed to determine the occurrence of a critical sensor fault, and to isolate the faulty sensor.

Algorithm 2: CR method for sensor fault detection

function CR;
Input : $\tilde{x}_i(k-\kappa_{CR}:k+1)(i=0,1,\ldots,m_o)$ from time step $k-\kappa_{CR}$ to $k+1$
Output: $I_A, I_F, I_D, \tilde{d}(k-1)$
//Estimation error calculation;
for $\mu = 0$ *to* m_o **do**
  for $\nu = 0$ *to* m_o **do**
    if $\mu \neq \nu$ **then**
      $e_{\mu,\nu}(k) = \tilde{x}_\mu(k) - \tilde{x}_\nu(k)$;
      $e_{\mu,\nu}(k+1) = \tilde{x}_\mu(k+1) - \tilde{x}_\nu(k+1)$;
      $\tilde{e}_{\mu(\nu)}(k) =$
      $(E_\nu - E_\mu)^{-1}(e_{\mu,\nu}(k+1) - E_\nu e_{\mu,\nu}(k))$;
      $\tilde{e}_{\Lambda,\mu(\nu)}(k) = (V_\mu)^{-1}\tilde{e}_{\mu(\nu)}(k)$;
    end
  end
  $\tilde{e}_\mu(k) = \Sigma_{\nu=0,\nu\neq\mu}^{m_o}\phi_\nu\tilde{e}_{\mu(\nu)}$;
  $\tilde{e}_{\Lambda,\mu}(k) = (V_\mu)^{-1}\tilde{e}_\mu(k)$;
  //Convergence ratio calculation;
  for $j = 1$ *to* n **do**
    $\{cr_\mu\}_j(k) =$
    $\frac{1}{\kappa_{CR}}\left[\left|\frac{\{\tilde{e}_{\Lambda,\mu}(k)\}_j}{\{\tilde{e}_{\Lambda,\mu}(k-1)\}_j}\right| + \sum_{k_i=2}^{\kappa_{CR}}\sqrt[k_i]{\left|\frac{\{\tilde{e}_{\Lambda,\mu}(k)\}_j}{\{\tilde{e}_{\Lambda,\mu}(k-k_i)\}_j}\right|}\right]$;
    //Anomaly detection;
    if $\{cr_\mu\}_j(k) > \theta_{CR}$ **then**
      $I_A = I_A + 1$;
    end
  end
end
//Determine whether it is a sensor fault or a disturbance;
if $I_A \geq \frac{(m_o+1)\times n}{2}$ **then**
  for $i = 1$ *to* m_o **do**
    $\tilde{d}_{\Lambda,\mu(\nu)}(k-1) =$
    $((D_{\Lambda,\mu})^T D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^T[\tilde{e}_{\Lambda,\mu(\nu)}(k) -$
    $E_{\Lambda,\mu}\tilde{e}_{\Lambda,\mu(\nu)}(k-1)]$;
  end
  if
  $Any\, avg(\tilde{d}_{\Lambda,\mu(\nu)}(k-1-\kappa_{CR}:k-1)$
  $-\tilde{d}_{\Lambda,\zeta(\eta)}(k-1-\kappa_{CR}:k-1)) > \theta_{d,\mu(\nu),\zeta(\eta)}$
  then
    $I_F = 1$;
  else
    $I_D = 1$;
    $\tilde{d}(k-1) =$
    $\Sigma_{\nu=0,\nu\neq\mu}^{m_o}\Sigma_{\mu=0}^{m_o}\psi_{\mu(\nu)}\tilde{d}_{\Lambda,\mu(\nu)}(k-1)$;
  end
end

In noise-free case ($w(k) = 0, v(k) = 0$), the MOLO method only works if the open-loop system is stable or marginally stable. This is due to the fact that the estimation error of open-loop observer will diverge if the system is unstable, i.e. the eigenvalues of A lie outside of the unit circle, according to Equation (29).

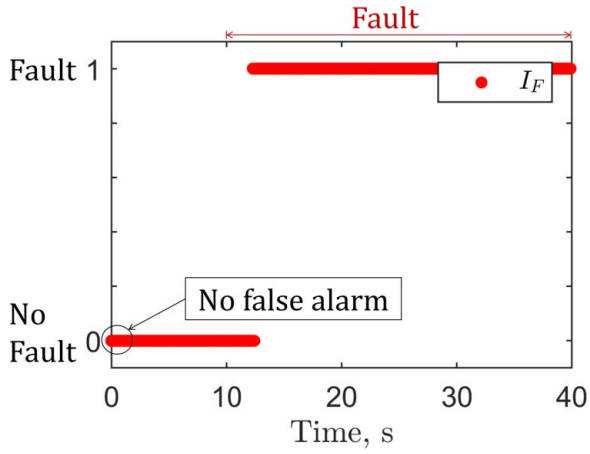


Figure 8. Fault alarms I_F of the CR method under fault α.

$$e_o(k) = x(k) - \hat{x}(k) = A^k e_o(0), \tag{29}$$

where $e_o(0)$ is the initial estimation error. After introducing system noise, the condition for the estimation error of an open-loop observer to be bounded is given in Proposition 4.9.

Proposition 4.9: *Given a control system* (1)*, and an open-loop observer* (3) *the following results can be drawn:*

(1) *If all of the eigenvalues of A lie inside the unit circle, then the estimation error of an open-loop observer is bounded;*
(2) *If one or more of the eigenvalues of A lie on the unit circle and* $\|A\| = 1$*, then the estimation error of an open-loop observer is bounded.*

Proof: See Appendix 6. ■

For systems that do not satisfy the conditions in Proposition 4.9, we need to periodically update the state of the open-loop observer with the state estimated by the closed-loop observer 0 which uses all of the sensor measurements when no fault is detected. The initial estimation error of the open-loop observer is then the same as the estimation error of the closed-loop observer.

There is a trade-off between the estimation performance and the ability to detect a critical sensor fault. If the update frequency is fast, then the state estimated by the open-loop observer can track the state estimated by the closed-loop observer well, which is indicated by

$$e_o(k) = A^k e(0) + \Sigma_{i=0}^{k-1} A^i w(k-1-i) \tag{30}$$

where $e(0)$ is the estimation error of the closed-loop observer 0. If k is smaller, then the divergence of $\Sigma_{i=0}^{k-1}A^i w(k-1-i)$ is smaller, which means a better estimation under normal operation. However, fast update frequency can degrade the ability to detect a sensor fault, which is indicated by

$$\begin{aligned} r(k) &= y(k) - C\hat{x}(k) \\ &= C(A^k e(0) + \Sigma_{i=0}^{k-1}A^i w(k-1-i)) + v(k) + Ff(k). \end{aligned} \tag{31}$$

The ramp fault signal $f(k)$ is increasing with the time step k. At the time step that $f(k)$ is significant, the fault can be detected.

The above discussion on the trade-off shows the necessity to have multiple open-loop observers for a marginally stable system with $\|A\| > 1$. In this paper, we divide the multiple open-loop observers into M groups. Group 1 has the slowest update frequency and group M has the fastest update frequency. Each group has N observers with the same update frequency. Based on the trade-off, if one group triggers an alarm, then the groups with slower update frequencies generate alarms as well, but the groups with faster update frequencies may not generate alarms. So if all of the groups detect a sensor fault, then we can say that the fault signal has a large slope. If only some of the groups detect a sensor fault, then we can say that the fault signal has a small slope.

Although the estimated state under the case that $\|A\| > 1$ may diverge for a marginally stable system, we can mitigate the impact of the process noise via averaging because the process noise has zero mean. To average the residuals, we need to find the time steps that the open-loop observers have similar divergence caused by system noise. Taking one open-loop observer for example, the state of the open-loop observer is updated every $\kappa_{f,g}$ time steps and has been updated for j_N times. At time step $k + (j_N - 1)\kappa_{f,g}$, we need to average the residual at time steps $k + (j_N - j)\kappa_{f,g}$ ($j = 1, \ldots, j_N$) to mitigate the impact of system noise. Proposition 4.10 validates the effectiveness of averaging.

Proposition 4.10: *Given a control system* (1) *an open-loop observer is updated every $\kappa_{f,g}$ time steps. The impact of the system noise on the averaged residual* (32) *is mitigated.*

$$r_{avg,g}(k + (j_N - 1)\kappa_{f,g}) = \frac{1}{j_N}\Sigma_{j=1}^{j_N} r_g(k + (j_N - j)\kappa_{f,g}), \tag{32}$$

where j_N is a positive integer.

Proof: See Appendix 7. ■

Proposition 4.10 shows the averaging method if we only have one open-loop observer in each group. Then the time steps that are needed for averaging is about $j_N \cdot \kappa_{f,g}$, which is large. To reduce the time steps for averaging, we have N ($N \leq \kappa_{f,g}$) open-loop observers in each group. We evenly distribute the time steps to update the states of the open-loop observers within the same group during one update period and we have

$$N = \lfloor \frac{\kappa_{f,g}}{\kappa_{\Delta,g}} \rfloor, \tag{33}$$

where $\kappa_{\Delta,g}$ is the update time step interval between two adjacent open-loop observers i and $i+1$ in group g. Then we calculate the average of the residuals generated by the open-loop observers in the same group.

In order to average the residuals of N observers, we need the following definition

Definition 4.11 (Leading observer): The leading observer is the open-loop observer which has not been updated for the longest time steps among all of the observers in the same group during the time steps $(j-1)\cdot\kappa_{\Delta,g}$ and $j\cdot\kappa_{\Delta,g}$, where j is a positive integer. The leading observer could be found according to the following formula:

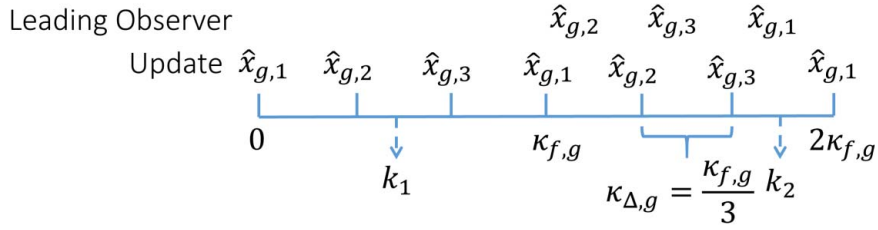
$$H_g = \left\lceil \frac{k - \kappa_{f,g}\lfloor \frac{k}{\kappa_{f,g}} \rfloor}{\kappa_{\Delta,g}} \right\rceil + 1. \tag{34}$$

Note that if $\lceil (k - \kappa_{f,g}\lfloor k/\kappa_{f,g})\rfloor/\kappa_{\Delta,g}\rceil$ equals N, then set $H_g = 1$.

To average the residuals, the first step is to find the leading observer during the time steps $(j-1)\cdot\kappa_{\Delta,g}$ and $j\cdot\kappa_{\Delta,g}$. Figure 9 helps explain how we average the residuals generated by a group of three observers. Suppose we are at time step k_1, which is during the first update period $\kappa_{f,g}$. We simply average all the estimated states at time step k_1. Suppose we are at time step k_2. Observer $(g,1)$ has not been updated for $k_2 - \kappa_{f,g}$ time steps, which is larger than that of observer $(g,2)$ ($k_2 - \kappa_{f,g} - \kappa_{\Delta,g}$) and that of observer $(g,3)$ ($k_2 - \kappa_{f,g} - 2\kappa_{\Delta,g}$). Therefore, observer $(g,1)$ is the leading observer at time step k_2. Based on this leading observer, we find the corresponding time steps when the divergence is similar for the other two observers. After getting the three estimated states, we can calculate the averaged residual at time step k_2. It can been seen that the averaged residual is generated over $2\kappa_{f,g}$ time steps. The following formula shows the averaged residual at time step k:

$$\begin{aligned} r_{avg,g}(k) = \frac{1}{N}(&\Sigma_{i=1}^{H_g} r_{g,i}(k - (H_g - i)\kappa_{\Delta,g}) \\ &+ \Sigma_{i=H_g+1}^{N} r_{g,i}(k - \kappa_{f,g} + (i - H_g)\kappa_{\Delta,g})). \end{aligned} \tag{35}$$

The average of the finite zero-mean random vector ($N < \infty$) does not exactly equal the zero vector. Based on



$$r_{avg,g}(k_1) = \frac{1}{3}\sum_{i=1}^{3} r_{g,i}(k_1)$$

$$r_{avg,g}(k_2) = \frac{1}{3}[\sum_{i=2}^{3} r_{g,i}(k_2 - \kappa_{f,g} + (i-1)\kappa_{\Delta,g}) + r_{g,1}(k_2)]$$

Figure 9. Residuals averaging.

the bounds of the system noise and update period $\kappa_{f,g}$, a threshold $\theta_{MOLO,g}$ can be set for each group to compare with the averaged residual $r_{avg,g}$. Notice that $\theta_{MOLO,g} \in \mathbb{R}^{m\times 1}$ is a vector. We compare each element $\{r_{avg,g}\}_j(k)$ in $r_{avg,g}(k)$ with the corresponding element $\{\theta_{MOLO,g}\}_j$ in $\theta_{MOLO,g}$. If $\{r_{avg,g}\}_j(k) \geq \{\theta_{MOLO,g}\}_j$, then group g triggers a fault alarm. Once the fault alarm is triggered, the states of the group of the open-loop observers are not updated by the closed-loop observer until the alarm is cleared.

Logic is applied to determine whether the system is under sensor fault or under normal operation based on which groups trigger fault alarms. Based on the discussion about the trade-off, if a group triggers an alarm, the groups with slower update frequencies should also trigger alarms theoretically. Therefore, we find the group g' which has the fastest update frequency among the groups that trigger fault alarms. If the majority of groups from 1 to g' trigger fault alarms, i.e. the inequality (36) holds, then the system is under sensor fault. Otherwise, it could be false alarms and the system is under normal operation.

$$\frac{1}{g'}\Sigma_{g=1}^{g'} I_{F,g}(k) \geq \theta_f, \tag{36}$$

where θ_f is a selected value with range 0.5–1. The sensor j, which makes the most of the groups that trigger alarms have $\{r_{avg,g}\}_j(k) \geq \{\theta_{MOLO,g}\}_j (g = 1, 2, \ldots, g')$, is identified as the faulty sensor.

When the system is subject to a sensor fault on a critical sensor, the averaged residual is

$$\begin{aligned} r_{avg,g}(k) &= \frac{1}{N}(\Sigma_{i=1}^{N} r_{g,i}(k - (N-i)\kappa_{\Delta,g}) \\ &= \frac{1}{N}\Sigma_{i=1}^{N} CA^{k_3} e(k - k_3 - (N-i)\kappa_{\Delta,g}) \\ &\quad + \frac{1}{N}\Sigma_{i=1}^{N} Ff(k - (N-i)k_{\Delta,g}). \end{aligned} \tag{37}$$

The above equation is drawn based on the assumption that observer N is the leading observer at time step k and it is updated at time step $k - k_3$. Suppose the sensor fault starts between time step $k - k_3$ and k. Theorem 1 in Mo and Sinopoli (2010) indicates that $\|e(k - k_3 - (N - i)\kappa_{\Delta,g})\| < \epsilon, \forall N$, where ϵ is a small positive number and it is related to system noise and initial estimation error. Therefore, the fault signal could increase the averaged residual generated by multiple open-loop observers, thus detected by the MOLO method. If the slope of the ramp fault signal is arbitrarily small, then the fault signal can still bypass the MOLO method.

Remark: The fault signal could be designed to make $\Sigma_{i=1}^{N} Ff(k - (N-i)k_{\Delta,g}) = 0$ in order to bypass the multiple open-loop observers. That means, however, the fault signal is changing around zero every $\kappa_{\Delta,g}$ time steps. If the change is small, then the impact of the fault is insignificant. If the change is large, then the fault signal can cause a significant change in the residual generated by a closed-loop observer.

Although this approach cannot guarantee the detection of a sensor fault with arbitrarily small slope, a sensor fault with a small slope would take a long time to disrupt the performance of the system. In addition, if the sensor fault is caused by a cyber attack, this long time increases the cost of the attack implementation. During this time, other techniques, such as sensor fusion, may have already detected the sensor fault.

Algorithm 3 shows the procedure of the MOLO method. At each time step, we first find the leading observer. Then we average the residuals for each group. The averaged residual is analysed to determine the occurrence of a critical sensor fault, and isolate the faulty sensor. After the faulty sensor is detected, if the system

Algorithm 3: MOLO method for critical sensor FDI

function MOLO;
Input : $y(k), u(k), \tilde{x}_0(k), I_{F,g}(k-1), \hat{x}_{g,i}$
Output: $I_F, I_{F,g}(k), i_f, \hat{x}_{g,i}(k+1)$
for $g = 1$ *to* M **do**
  $H_g = \lceil \frac{k-\kappa_{f,g}\lfloor \frac{k}{\kappa_{f,g}} \rfloor}{\kappa_{\Delta,g}} \rceil + 1$;
  if $H_g > N$ **then**
    $H_g = 1$;
  end
  for $i = 1$ *to* N **do**
    if *time to update* $\hat{x}_{g,i}$ **then**
      $\hat{x}_{g,i}(k) = (1 - I_{F,g}(k-1))\tilde{x}_0(k) + I_{F,g}(k-1)\hat{x}_{g,i}(k)$;
      $\hat{x}_{g,i}(k+1) = A\hat{x}_{g,i}(k) + Bu(k)$;
    else
      $\hat{x}_{g,i}(k+1) = A\hat{x}_{g,i}(k) + Bu(k)$;
    end
    //Residuals generation;
    $r_{g,i}(k) = y(k) - C\hat{x}_{g,i}(k)$
  end
  //Averaged residual;
  if $k \le \kappa_{f,g}$ **then**
    $r_{avg,g}(k) = \frac{1}{N}\Sigma_{i=1}^{N} r_{g,i}(k)$;
  else
    $r_{avg,g}(k) = \frac{1}{N}(\Sigma_{i=1}^{H_g} r_{g,i}(k - (H_g - i)\kappa_{\Delta,g}) + \Sigma_{i=H_g+1}^{N} r_{g,i}(k - \kappa_{f,g} + (i - H_g)\kappa_{\Delta,g}))$;
  end
end
//Fault detection and isolation;
$tmp = 0$; //The number of groups that trigger fault alarms;
$tmp_{sensor,j} = 0$; //The sensor that each group thinks it is faulty;
for $g = 1$ *to* M **do**
  for $j = 1$ *to* n **do**
    if $\{r_{avg,g}(k)\}_j \ge \{\theta_{MOLO,g}\}_j$ **then**
      $I_{F,g}(k) = 1$;
      $g' = g$;
      $tmp = tmp + 1$;
      $tmp_{sensor,j} = tmp_{sensor,j} + 1$;
    end
  end
end
if $\frac{1}{g'}\Sigma_{g=1}^{g'} I_{F,g}(k) \ge \theta_f$ **then**
  $I_F = 1$;
  $i_f = \max_j tmp_{sensor,j}$;
end

is stable or marginally stable with $\|A\| = 1$, then we can directly use the state estimated by an open-loop observer for the state feedback controller as indicated in Proposition 4.9. Otherwise, we need to replace the faulty sensor.

Figure 10 shows the performance of the MOLO method under fault β. In this example, we have two groups of open-loop observers. Group 1 has update period 8 s and group 2 has update period 2 s. There are 20 observers in each group and the update time steps are distributed evenly within one update period. Figure 10(a) shows the averaged residuals and Figure 10(b) shows the fault alarms of the two groups. After the first update period, the averaged residual is less noisy and the threshold of each group could be smaller. It can also be seen that the fault is successfully detected by Group 1 at about 27 s but bypasses Group 2. This is because the update period of Group 2 is too short compared to the slope of the fault signal. Overall, fault β is successfully detected by the MOLO method compared to Figure 6.

4.5. CCI method for non-critical sensor fault mitigation

The CCI method can potentially mitigate the impact of a fault on a non-critical sensor during the FDI process. At each time step, this method selects the closed-loop observer, based on which the state feedback controller gives the smallest divergence of the control input. This divergence is defined as follows:

Definition 4.12 (Divergence of the control input): Divergence of the control input $\|\Delta u_i\|$ is the absolute difference between the CCI based on the closed-loop observer and that based on an open-loop observer.

$$\|\Delta u_i(k)\| = \|K\tilde{x}_i(k) - K\hat{x}(k)\|. \tag{38}$$

The open-loop observer in the CCI method is slightly different from those used in the MOLO method. Since the CCI method switches among several closed-loop observers from time-to-time, the state of the open-loop observer should be updated to be the estimated state by the closed-loop observer which is used for feedback at time step k. For example, if closed-loop observer i is used for feedback at time step k, then we need to calculate the estimated state $\hat{x}(k+1)$ of the open-loop observer with the initial state $\tilde{x}_i(k)$.

First, we analyze this method in ideal system, and give the lower bound of the fault signal that the CCI method can switch to the observer without the faulty sensor during the FDI process. Then, we analyze the impact of system noise on the lower bound of the fault signal.

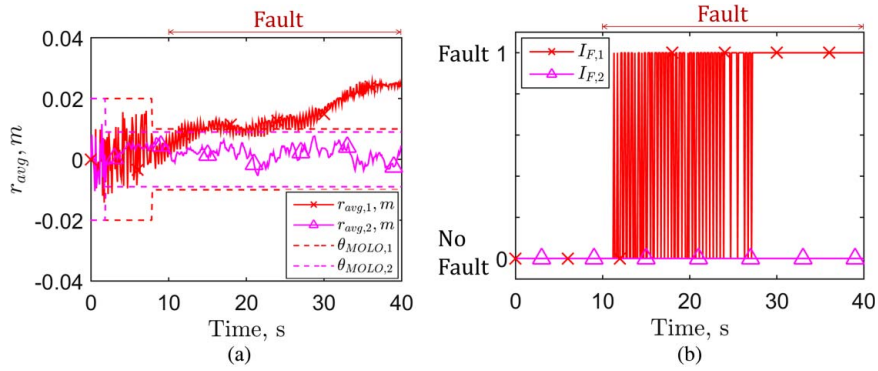


Figure 10. (a) The averaged residuals of the two groups of observers: Group 1 has update period 8 s and Group 2 has update period 2 s; (b) fault alarms of the two groups $I_{F,1}$ and $I_{F,2}$.

4.5.1. Ideal system case

Under normal operation, the divergence of the control input calculated based on a closed-loop observer is a function of its estimation error. Under sensor fault, the closed-loop observer without faulty sensor gives the best state estimation, thus the smallest divergence. Theorem 4.13 demonstrates that the divergence of the control input $\|\Delta u_{i_f}(k+1)\|$ based on the closed-loop observer i_f without the faulty sensor i_f is smaller than that based on other closed-loop observers with the faulty sensor.

Theorem 4.13: *Given an ideal control system* (1) *with* $w(k) = 0$ *and* $v(k) = 0$, *and a sensor fault starting at time step k on sensor* i_f, *observer* i_f *gives the smallest divergence of the control input* $\|\Delta u_{i_f}(k+1)\|$ *if the lower bound of the fault signal satisfies Equation (39).*

$$\forall i = 0, 1, \ldots, m_o, \quad i \neq i_f$$
$$\|f(k)\| \geq \|KL_iF_i\|^{-1}[\|KL_{i_f}C_{i_f}e_{i_f}(k)\| + \|KL_iC_ie_i(k)\| + \|KAe_{i_f,i}(k)\|]. \tag{39}$$

Proof: With faulty sensor i_f starting at time step k, observer i_f is not affected by the faulty sensor. The estimated state $\tilde{x}_i(k+1)$ of observer i ($i \neq i_f$) containing the faulty sensor and the estimated state $\tilde{x}_{i_f}(k+1)$ observer i_f are

$$\tilde{x}_i(k+1) = E_i\tilde{x}_i(k) + L_i(C_ix(k) + F_if(k)) + Bu(k),$$
$$\tilde{x}_{i_f}(k+1) = E_{i_f}\tilde{x}_{i_f}(k) + L_{i_f}C_{i_f}x(k) + Bu(k). \tag{40}$$

Since the initial state of the open-loop observer is the same as the estimated state of the observer which is used for feedback at time step k, two cases should be considered:

(1) At time step k, observer i ($i \neq i_f$) is used for feedback,

$$\hat{x}(k+1) = A\tilde{x}_i(k) + Bu(k). \tag{41}$$

(2) At time step k, observer i_f is used for feedback,

$$\hat{x}(k+1) = A\tilde{x}_{i_f}(k) + Bu(k). \tag{42}$$

Under case (1), the divergence of the control input of observer i_f and observer i ($i \neq i_f$) are shown in Equations (43) and (44), respectively.

$$\|\Delta u_{i_f}(k+1)\| = \|KAe_{i_f,i}(k) + KL_{i_f}C_{i_f}e_{i_f}(k)\|, \tag{43}$$
$$\|\Delta u_i(k+1)\| = \|KL_iC_ie_i(k) + KL_iF_if(k)\|. \tag{44}$$

So when the lower bound of the fault signal satisfies Equation (39), observer i_f gives the smallest divergence of the control input, and is selected to provide feedback for the state feedback controller at time step k+1. The same result is also drawn for case (2). ■

Based on Theorem 4.13, when the system is under non-critical sensor fault and the fault signal satisfies Equation (39), the CCI method can switch to the observer without the faulty sensor before the faulty sensor is identified. If the magnitude or the slope of the fault signal is too small, then the CCI method may not be able to select the observer without the faulty sensor to mitigate the impact of sensor fault; and the lower bound of the fault signal during the observers' transient state is larger than that during steady state because of the relatively large estimation error. In order to reduce the lower bound of the fault signal, horizon size κ_{CCI} is introduced to calculate the divergence of the control input to consider the impact of the integral of the fault signal over κ_{CCI} steps. Therefore, at each time step k, we need to recalculate the state of the open-loop observer with initial state same as the estimated state $\tilde{x}_i(k+1-\kappa_{CCI})$ of the selected closed-loop

observer at time step $k+1-\kappa_{CCI}$. Then, the divergence of the control input of observer i_f and i are

$$\begin{aligned}\|\Delta u_{i_f}(k+1)\| &= \|K(-(E_{i_f})^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI}) \\ &\quad + A^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI}))\|, \qquad (45)\\ \|\Delta u_i(k+1)\| &= \|K(-(E_i)^{\kappa_{CCI}} e_i(k+1-\kappa_{CCI}) \\ &\quad - \Sigma_{j=0}^{\kappa_{CCI}-1}(E_i)^j L_i F_i f(k-j) \\ &\quad + A^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI}))\|, \qquad (46)\end{aligned}$$

Thus, the lower bound of the integral of the fault signal is

$$\begin{aligned}&\|\Sigma_{j=0}^{\kappa_{CCI}-1} K(E_i)^j L_i F_i f(k-j)\| \\ &\geq 2\|KA^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI})\| \\ &\quad + \|K(E_{i_f})^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI})\| \\ &\quad + \|K(E_i)^{\kappa_{CCI}} e_i(k+1-\kappa_{CCI})\|. \qquad (47)\end{aligned}$$

If the fault starts between time steps $k+1-\kappa_{CCI}$ and k, $e_{i_f}(k+1-\kappa_{CCI})$ and $e_i(k+1-\kappa_{CCI})$ are very small. In addition, the absolute value of the eigenvalues of E_{i_f} and E_i are smaller than 1. Increasing the horizon step κ_{CCI} and placing the observer poles closer to the origin can reduce both $\|K(E_{i_f})^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI})\|$ and $\|K(E_i)^{\kappa_{CCI}} e_i(k+1-\kappa_{CCI})\|$. For the term $\|KA^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI})\|$, however, we need to consider three conditions: A is stable, marginally stable and unstable. If the open-loop system is stable or marginally stable, i.e. the eigenvalues of A lie inside or on the unit circle, the term $\|KA^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI})\|$ is bounded. Thus, increasing κ_{CCI} can reduce the lower bound of the fault signal and increase the ability of the CCI method to select the observer without the faulty sensor. If the open-loop system is unstable, i.e. the one or more eigenvalues of A lie inside the unit circle, the term $\|KA^{\kappa_{CCI}} e_{i_f}(k+1-\kappa_{CCI})\|$ is diverging, which reduces the ability of the CCI method. Therefore, the selection of the optimal horizon step κ_{CCI} depends on the property of the physical system.

4.5.2. Noisy system case

With system noise, the lower bound of the fault signal is increased as shown in Lemma 4.14 (the horizon step κ_{CCI} is not considered in Lemma 4.14).

Lemma 4.14: *Given a control system* (1), *and a sensor fault starting at time step k on sensor i_f, observer i_f gives the smallest divergence of the control input if the lower bound of the fault signal satisfies Equation (48).*

$$\begin{aligned}&\forall i = 0, 1, \ldots, m_o, i \neq i_f \\ &\|f(k)\| \geq \|KL_iF_i\|^{-1}(\|KL_{i_f}C_{i_f}e_{i_f}(k)\| + \|KL_iC_ie_i(k)\| \\ &\quad + \|KAe_{i_f,i}(k)\| + \|KL_{i_f}\|\upsilon_{i_f} + \|KL_i\|\upsilon_i). \qquad (48)\end{aligned}$$

The proof is similar to Theorem 4.13.

The transient dynamics caused by switching among observers may degrade the performance of the control system (Liberzon and Morse, 1999). To avoid frequently switching, a threshold θ_{CCI} is used to decide when to enable or disable the switching. θ_{CCI} should be selected to balance the frequency of switching and the ability to mitigate the impact of the sensor fault.

Algorithm 4 gives the procedure of the CCI method. At each time step, the CCI method calculates the estimated state of an open-loop observer with the initial state the same as the selected observer at time step $k+1-\kappa_{CCI}$. Then it switches to the observer which gives the smallest divergence of the control input if the switching is enabled.

Figure 11 shows the system with the CCI method under sensor fault α. The maximum absolute value of position under sensor fault is 4 cm, which is smaller than that with the CO method as shown in Figure 5(a). During the detection delay (2 s), the CCI method has already switched to observer 1 for state estimation at 13 s, thus mitigating the impact of the sensor fault.

4.6. Integration of CO, CR, MOLO and CCI methods

In this section, the three new methods, CR, MOLO, and CCI methods are introduced and compared with the

Algorithm 4: CCI method for non-critical sensor fault mitigation

function CCI;
Input : $k, \tilde{x}_i(k+1), \tilde{x}_i(k+1-\kappa_{CCI})$ $(i = 0, \ldots, m_o)$, $l_{FB}(k+1-\kappa_{CCI})$
Output: $l_{FB}(k+1)$
`//Open-loop observer state estimation;`
if $k > \kappa_{CCI}$ **then**
  $\hat{x}(k+1) = A^{\kappa_{CCI}}\tilde{x}_{l_{FB}(k+1-\kappa_{CCI})}(k+1-\kappa_{CCI}) + \Sigma_{j=0}^{\kappa_{CCI}-1}A^jBu(k-j)$
else
  $\hat{x}(k+1) = A^{k+1}\tilde{x}_0(0) + \Sigma_{j=0}^{k}A^jBu(k-j)$
end
`//Control input calculation;`
$u_o(k+1) = K\hat{x}(k+1)$;
$u_i(k+1) = K\tilde{x}_i(k+1)$;
$\|\Delta u_i(k+1)\| = \|u_i(k+1) - u_o(k+1)\|$;
if $\|\Delta u_i(k+1)\| \geq \theta_{CCI}$ *for all i* **then**
  $l_{FB}(k+1) = \min_i \|\Delta u_i(k+1)\|$;
else
  $l_{FB}(k+1) = l_{FB}(k)$;
end

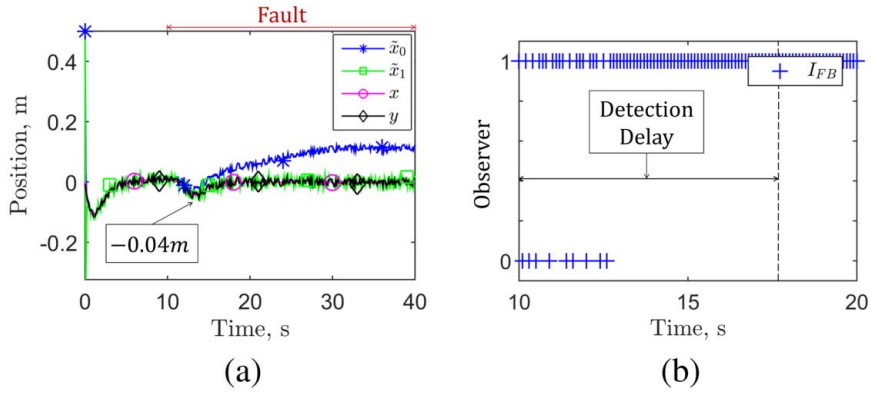


Figure 11. (a) Estimated states of both observers $\tilde{x}_0, \tilde{x}_1$, real state x, and sensor measurement y of the system; (b) Selected observer index I_{FB} with the CCI method during time 10 s and 20 s.

CO method through simulation to show the improvements.

- The CR method enables fault detection during the observers' transient state, and no false alarms generated compared to CO method;
- The MOLO method successfully detects the critical sensor fault, while CO method fails;
- The CCI method switches to the observer without the faulty sensor during the FDI process, and the position of the object during sensor fault is reduced to 0.04 m compared to 0.3 m with CO method.

We systematically integrate all of the above methods to utilize their advantages, improving the overall performance of FDI and fault mitigation. Algorithm 5 shows the integration of the CO, CR, MOLO, and CCI methods. At each time step, the CCI method is used to mitigate the impact of a potential sensor fault. Then the CR method determines whether there is a faulty sensor on the system. If the CR method flags an alarm, and if the system observers have reached their steady state under normal operation ($k > k_{ss}$, where k_{ss} is the number of time steps that is needed for observers to reach their steady state), the CO method is used to isolate the faulty sensor, and the system switches to the observer that can mitigate the impact of the sensor fault after the faulty sensor is isolated. Meanwhile, the MOLO method detects whether there is a fault on a critical sensor. Robust control design in the presence of a disturbance is not within the scope of this paper.

Algorithm 5: Integration of four methods

for $k = 0$ *to the end of simulation* **do**
  //Estimated state of closed-loop observers;
  $\tilde{x}_i(k+1) = E_i\tilde{x}_i(k) + L_i y_i(k) + Bu(k)$;
  //FDI and Mitigation begins;
  $I_{FB}(k+1) = CCI(k, \tilde{x}_i(k+1), \tilde{x}_i(k+1-\kappa_{CCI}), I_{FB}(k+1-\kappa_{CCI}))$;
  $u(k+1) = K\tilde{x}_{I_{FB}(k+1)}(k+1)$;
  $[I_A, I_F, I_D, \tilde{d}(k-1)] = CR(\tilde{x}_i(k-k_{CR}:k+1))$;
  if $I_D = 0$ *and* $k \geq k_{ss}$ **then**
    $[I_F, I_{F,g}(k), i_f, \hat{x}_{g,i}(k+1)] = MOLO(y(k), u(k), \tilde{x}_0(k), I_{F,g}(k-1), \hat{x}_{g,i})$;
  end
  if $I_F = 1$ *and* $k \geq k_{ss}$ **then**
    $[I_F, i_f] = CO(y(k), \tilde{x}_i)$;
    $I_{FB}(k+1) = i_f$;
    $u(k+1) = K\tilde{x}_{I_{FB}(k+1)}$;
  else if $I_{F,g} = 1$ *for any* g **then**
    if *A is stable or* (*A is marginally stable and* $\|A\| \leq 1$) **then**
      $u(k+1) = -K\hat{x}_{g,1}(k+1)$;
    else
      Replace the faulty sensor i_f
    end
  else
    Robust control to tolerate disturbance
  end
end

5. Illustrative example

A simplified suspension system (a two-mass-two-spring system) (Control Tutorial for Matlab & Simulink) is used to test the proposed algorithm with four methods. The system shown in Figure 12 has five states: position h_1 of mass 1, velocity $\dot{h}_1$ of mass 1, distance between two mass h, velocity $\dot{h}$, and integral of h, which is used to achieve zero steady-state error. The five states are measured by five sensors directly, as shown in Table 2. A controller controls the system through u. Potential disturbance comes

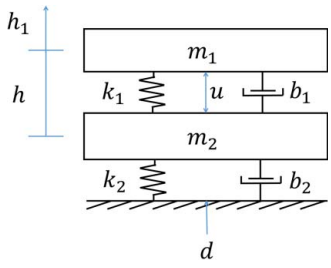


Figure 12. Simplified suspension system.

Table 2. Sensors of the simplified suspension system.

	Variable	Set
Sensor 1	h_1	S_{nc}
Sensor 2	$\dot{h}_1$	S_{nc}
Sensor 3	h	S_{nc}
Sensor 4	$\dot{h}$	S_{nc}
Sensor 5	Σh	S_c

from the ground. We want to maintain h to stay at 0 m, which is also the reference signal of this system.

The system has sampling time 0.01 s, process noise bound 0.001 (m or m/s) and sensor noise bound 0.01 (m or m/s). The observers' transient state is about 0.1 s (10 time steps). The initial state of the system is $(0, 0, 0, 0, 0)$. The initial state of the observers is $(0.02, 0.01, 1, 0, 0)$. Table 3 shows part of parameters of the four methods.

Four scenarios are considered as examples:

- Scenario 1: A ramp fault signal with slope 1 m/s (0.01 m per time step) added to sensor 3, saturating at 10 m;
- Scenario 2: A ramp fault signal with slope 1 m/s (0.01 m per time step) added to sensor 3, saturating at 10 m;
- Scenario 3: A ramp fault signal with slope 0.01 m/s (0.0001 m per time step)added to sensor 5, saturating at 10 m;
- Scenario 4: A step disturbance from the ground with magnitude 0.2 m, starting at $t = 30$ s.

The faults in Scenario 1 and 3 start at $t = 30$ s. The fault in Scenario 2 starts at $t = 0.05$ s.

Table 3. Part of parameters of the four methods.

CO	θ_{CO}	0.012
CR	κ_{CR}	0.1 s (10 time steps)
	θ_{CR}	0.9
MOLO	M	2
	N	20
	$k_{f,1}$	10 s (1000 time steps)
	$k_{f,2}$	0.4 s (40 time steps)
	$\{\theta_{MOLO,1}\}_5$	0.025 m
	$\{\theta_{MOLO,2}\}_5$	0.015 m
CCI	κ_{CCI}	10 s (1000 time steps)
	θ_{CCI}	0.001 N

Figure 13 shows the system under a non-critical sensor fault happening during the steady state of the system. During the observers' transient state, the CR method eliminates false alarms as shown in Figure 13(b). At the time the sensor fault occurs, the CCI method switches to observer 3 for feedback as shown in Figure 13(c), allowing more time for FDI. The CR method triggers an alarm after detecting the sensor fault. The CO method isolates the faulty sensor, and calculates the fault signal as shown in Figure 13(d). The proposed algorithm integrating the four methods successfully protects the system from a non-critical sensor fault happening during the observers' steady state.

Figure 14 shows the system under a non-critical sensor fault happening during the observers' transient state. The CR method successfully detects the occurrence of the sensor fault with about 0.06 s time delay as shown in Figure 14(b), which is caused by relatively large θ_{CR} (0.9) compared to the observer poles (about 0.1). The CCI method switches to the observer without the faulty sensor later than the time step that CR method detects the sensor fault. This is because the observers cannot provide good state estimations during the observers' transient state, thus the observer without the faulty sensor may not give the smallest divergence of the CCI. This scenario shows the effectiveness of the CR method for fault detection during the observers' transient state.

Figure 15 shows the system is subject to a critical sensor fault. In Figure 15(b), the averaged residuals are less noisy after the first update period. Group 1 successfully detects the occurrence of the sensor fault, while group 2 does not. This scenario shows the effectiveness of the MOLO method for a non-critical sensor FDI.

Figure 16 shows the system under disturbance from the ground. The CR method successfully distinguishes a disturbance from a sensor fault, and correctly estimates the disturbance signal.

6. Conclusions and future work

In this paper, the CO method and three new methods, the CR, MOLO, and CCI methods, are integrated to solve the FDI and mitigation problem using multiple closed-loop and open-loop observers. The closed-loop observers include one that uses all of the sensor measurements for state estimation, and others that exclude a non-critical sensor. Based on the two different types of observers, new methods are proposed and integrated to solve various problems:

- the CR method can detect non-critical sensor faults during the observers' transient state;

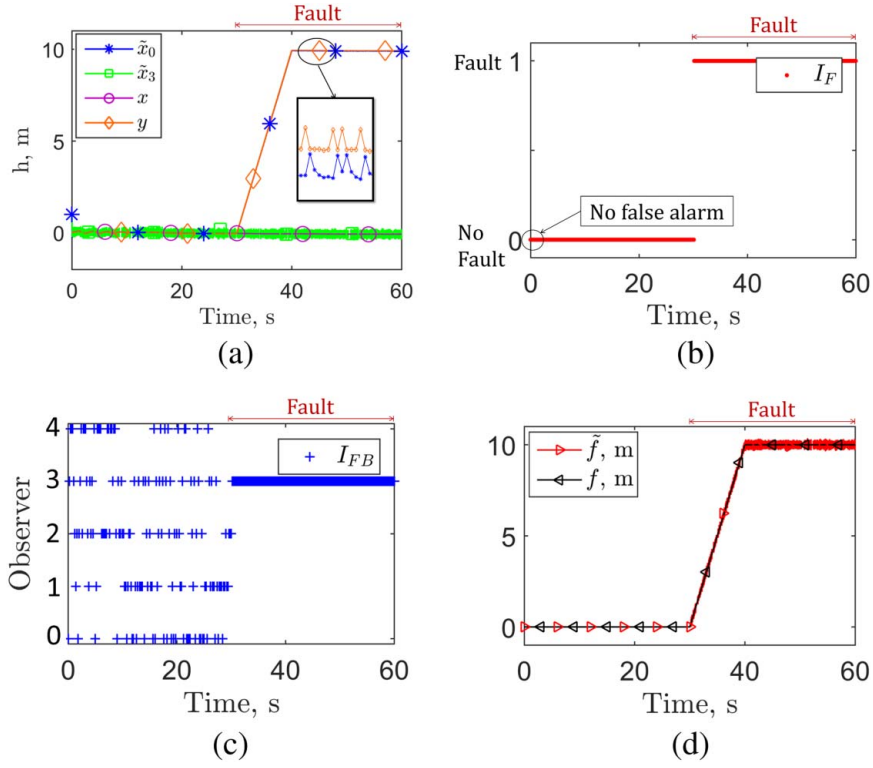


Figure 13. Scenario 1: (a) the estimated states of observer 0 and observer 3 $\tilde{x}_0, \tilde{x}_3$, the real state x, and the sensor measurement y of the system under the non-critical sensor fault; (b) fault alarms I_F; (c) Observer index I_{FB} selected for the state feedback controller; and (d) estimated fault signal $\tilde{f}$ and the real fault signal f.

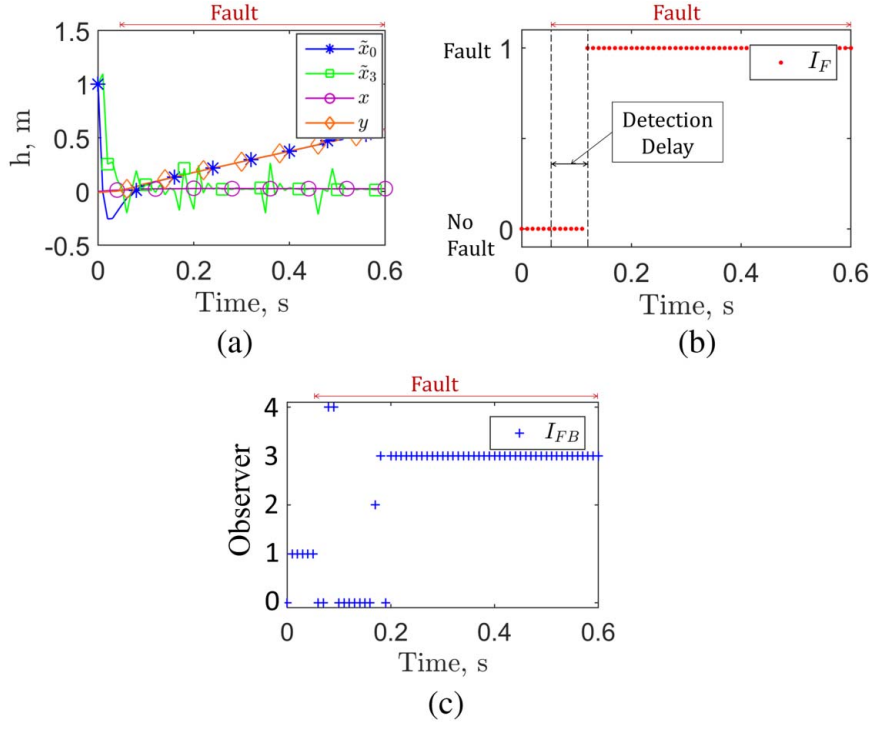


Figure 14. Scenario 2: (a) the estimated states of observer 0 and observer 3 $\tilde{x}_0, \tilde{x}_3$, the real state x, and the sensor measurement y of the system under the non-critical sensor fault; (b) fault alarms I_F; and (c) observer index I_{FB} selected for the state feedback controller.

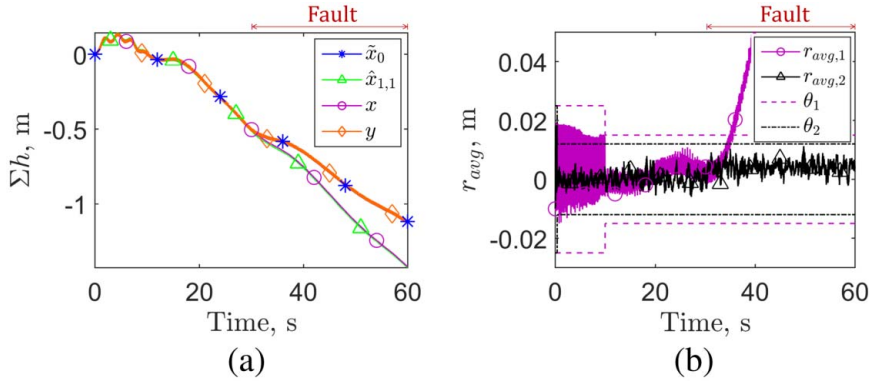


Figure 15. Scenario 3: (a) the estimated states of closed-loop observer 0 and the open-loop observer (1,1), $\tilde{x}_0, \hat{x}_{1,1}$ the real state x, and the sensor measurement y of the system; (b) averaged residuals of both groups of open-loop observers $r_{avg,1}, r_{avg,2}$.

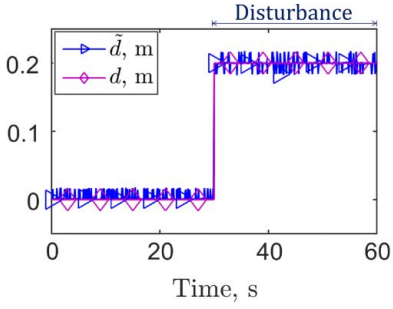


Figure 16. Scenario 4: The calculated disturbance $\tilde{d}$ and the real disturbance d of the system.

- the MOLO method can detect and isolate critical sensor faults; and
- the CCI method can mitigate the impact of non-critical sensor faults during the FDI process.

The CR method uses the CRs of the observers' estimation errors to determine whether or not there is a non-critical sensor fault. The CRs of the observers' estimation errors are not affected by the uncertain initial condition. Therefore, the CR method can reduce false alarms during the observers' transient state. To achieve robust FDI, bias analysis is used to distinguish a sensor fault from a disturbance.

The MOLO method utilizes a bank of open-loop observers, which do not use sensor measurements for state estimation, to detect and isolate critical sensor faults. The state of the open-loop observers are updated periodically by the closed-loop observer which uses all of the sensor measurements. Because of the trade-off between estimation performance and the ability to detect a sensor fault, the open-loop observers are divided into several groups. In the same group, the open-loop observers are updated with the same update frequency, but the time steps to update them are evenly distributed in one update period. The residuals generated by observers in the same group are averaged. Then the averaged residuals of different groups are analysed to determine the occurrence of a sensor fault and to locate the faulty sensor.

The CCI method switches among different closed-loop observers to potentially mitigate the impact of non-critical sensor faults during the FDI process. This method selects the closed-loop observer which gives the smallest divergence of the control input, for state estimation at the next time step.

The three new methods are integrated with a previously developed residual-based method (CO method) to collaboratively address the FDI and mitigation problem in this paper. The collaboration of the methods is illustrated in Figure 2(a) and Table 1. The proposed algorithm allows any residual-based method to be integrated besides the CO method. Simulation results show the effectiveness of our proposed framework.

This multi-observer approach can be easily extended to the multiple sensor faults case as long as the system observability still holds without the faulty sensors. However, at a high level, the framework we propose has some limitations. There is not currently a method to detect a critical sensor fault during the observers' transient state. Also, no method can potentially mitigate the impact of a critical sensor fault.

Other limitations of the framework proposed herein include an inability to detect a ramp sensor fault with arbitrary slope, and the requirement of a lower bound on the magnitude of the fault signal for detection. In some cases, our framework cannot distinguish a sensor fault from sensor noise in some cases. Addressing this issue is a topic of future work. Sensor fusion, statistic analysis, and machine learning methods are potential solutions to this problem.

Each of the methods we propose presents opportunities for future work. The CR method is very sensitive to sensor noise for fault detection. The threshold θ_{CR}, which is used to compare with the CR, is selected to be much

larger than the observer poles to reduce false alarms during the observers' steady state. Therefore, the CR method cannot detect the sensor faults that make the CR smaller or slightly larger than the observer poles. If we can make the CR method robust to sensor noise, upper and lower bounds for the CRs can be set to address more sensor faults. The bias analysis of the CR method is also sensitive to system noise. Our future work could use several robust observers for the CR method, which requires that we decouple estimation errors based on the estimated states of the observers.

The MOLO method, used for critical sensor FDI, does not work for open-loop unstable systems. Techniques such as sensor fusion could be exploited to protect unstable systems from critical sensor faults.

The CCI method does not perform well if the sensor fault occurs during the observers' transient state, as shown in the suspension system example in Section 5, because of the relatively large estimation error. This issue could be potentially addressed by combining the CR method and the CCI method together because the CR method can decouple the estimation errors of the observers. Finally, the optimal horizon step, which could reduce the lower bound of the fault signal, is unknown. A cost function should be proposed to determine the optimal horizon step in the future.

It may be impossible in general to detect every kind of sensor fault. The aim of our sensor FDI and mitigation method is to decrease the lower bound of sensor fault that can be detected, and to allow more time for other techniques to protect the system before it runs into some severe condition.

Notes

1. The reason we use the Luenberger observers (or Kalman filters) is that we can decouple observers' estimation errors for the CR method.
2. Q_i should be designed to make the element $\{y_i(k) - C_i\tilde{x}_i(k)\}_j (j \in S_{nc})$, where j corresponds to the critical sensors, have larger weighting ratios than the element corresponding to non-critical sensors.
3. The demonstration is shown in Theorem A.1.
4. The initial estimation errors of the observers are large to help us understand the limitations of a residual-based method using closed-loop observers during the observers' transient state.

Acknowledgements

We thank Richard Candell, Electronics Engineer of NIST, for his inputs to this research. We thank Dr Hossein Rastgoftar, a Postdoctoral fellow at University of Michigan, Ilya Kovalenko, Xingye Da and Miguel Saez, Ph.D. students at University of Michigan, and Dr Larissa Sano in Sweetland Writing Center at University of Michigan, for their inputs to the review and editing of this paper.

Disclosure statement

Official contribution of the National Institute of Standards and Technology under Award No. 60NANB13D166. Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Funding

This work was supported by National Institute of Standards and Technology (NIST) Award No. 60NANB13D166

References

Bemporad, A. (2010). *State estimation and linear observers* [*Automatic control lecture notes*]. Retrieved from http://cse.lab.imtlucca.it/ ~ bemporad/teaching/ac/pdf/06b-estimator.pdf

Bobba, R., Rogers, K., Wang, Q., Khurana, H., Nahrstedt, K., & Overbye, T. (2010). *Detecting false data injection attacks on DC state estimation*. Stockholm: Workshop on Secure Control Systems (SCS).

Bouibed, K., Seddiki, L., Guelton, K., & Akdag, H. (2014). Actuator and sensor fault detection and isolation of an actuated seat via nonlinear multi-observers. *Systems Science & Control Engineering: an Open Access Journal*, *2*(1), 150–160. doi:10.1080/21642583.2014.888525

Cardenas, A., Amin, S., & Sastry, S. (2008). *Secure control: towards survivable cyber-physical systems*. Paper presented at the 28th international conference on distributed computing systems workshops. doi:10.1109/ICDCS.Workshops.2008.40

Chadli, M., Akhenak, A., Maquin, D., & Ragot, J. (2008). Fuzzy observer for fault detection and reconstruction of unknown input fuzzy models. *International Journal of Modelling, Identification and Control*, *3*(2), 193–200. doi:10.1504/IJMIC.2008.019358

Choy, S., & Weyer, E. (2008). Reconfiguration schemes to mitigate faults in automated irrigation. *Control Engineering Practice*, *16*(10), 1184–1194. doi:10.1016/j.conengprac.2008.01.003

Clark, R. N. (1978). Instrument fault detection. *IEEE Transaction on Aerospace Electronic Systems*, *AES-14*(3), 456–465. doi:10.1109/TAES.1978.308607

Clark, R. N. (1978). A simplified instrument failure detection scheme. *IEEE Transactions on Aerospace and Electronic Systems*, *AES-14*(4), 558–563. doi:10.1109/TAES.1978.308680

Control Tutorial for Matlab & Simulink. Retrieved from http://ctms.engin.umich.edu/CTMS/index.php?example = Suspension &section = SystemModeling

Dubey, A., Nordstrom, S., Keskinpala, T., Neema, S., Bapty, T., & Karsai, G. (2007). Towards a verifiable real-time, autonomic, fault mitigation framework for large scale real-time systems. *Innovations in Systems and Software Engineering*, *3*(1), 33–52. doi:10.1007/s11334-006-0015-7

Edwards, C., & Tan, C. (2006). Sensor fault tolerant control using sliding mode observers. *Control Engineering Practice*, *14*(8), 897–908. doi:10.1016/j.conengprac.2005.05.002

Frank, P. M., & Ding, X. (1997). Survey of robust residual generation and evaluation methods in observer-based fault

detection systems. *Journal of Process Control*, *7*(6), 403–424. doi:10.1016/S0959-1524(97)00016-4

Hwang, I., Kim, S., Kim, Y., & Seah, C. E. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, *18*(3), 636–653. doi:10.1109/TCST.2009.2026285

Isermann, R. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, *5*(5), 709–719. doi:10.1016/S0967-0661(97)00053-1

Lambers, J. (2009). MAT 610 Summer Session 2009-10 [Lecture notes]. Retrieved from http://www.math.usm.edu/lambers/mat610/sum10/lecture2.pdf

Lefebvre, D. (2014). Fault diagnosis and prognosis with partially observed petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(10). doi:10.1109/TSMC.2014.2311760

Liberzon, D., & Morse, A. S. (1999). Basic problems in stability and design of switched systems. *IEEE Control Systems*, *19*(5), 59–70. doi:10.1109/37.793443

Liu, Y., Ning, P., & Reiter, M. (2011). False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security*, *14*(1). doi:10.1145/1952982.1952995

Methnani, S., Lafont, F., Gauthier, J., Damak, T., & Toumi, A. (2013). Actuator and sensor fault detection, isolation and identification in nonlinear dynamical systems, with an application to a waste water treatment plant. *Journal of Computer Engineering and Informatics*, *1*(4), 112–125. Retrieved from http://www.academicpub.org/jcei/

Mishra, S., Karamchandani, N., Tabuada, P., & Diggavi, S. (2014). *Secure state estimation and control using multiple (insecure) observers*. Paper presented at the 53rd IEEE conference on decision and control. Los Angeles, CA, United States. doi:10.1109/CDC.2014.7039631

Mo, Y., & Sinopoli, B. (2010). *False data injection attacks in control systems*. Workshop on Secure Control Systems (SCS), Stockholm, Sweden.

Mo, Y., & Sinopoli, B. (2015). On the performance degradation of cyber-physical systems under stealthy integrity attacks. *IEEE Transactions on Automatic Control*, *PP*(99), 1. doi:10.1109/TAC.2015.2498708

Phillips, C., & Nagle, H. (1994). *Digital control system analysis and design*. 3rd ed. Eaglewood Cliffs, NJ: Prentice-Hall.

Rios, H., Edwards, C., Davila, J., & Fridman, L. (2015). Fault detection and isolation for nonlinear systems via high-order-sliding-mode multiple-observer. *International Journal of Robust and Nonlinear Control*, *25*(16), 2871–2893. doi:10.1002/rnc.3232

Schrick, D. V. (1997). Remarks on terminology in the field of supervision, fault detection and diagnosis. *Proceeding of IFAC Symposium on Fault Detection, Supervision Safety for Technical Processes (SAFEPROCESS'97)*. Kingston Upon Hull: Pergamon.

Silva, J., Saxena, A., Balaban, E., & Goebel, K. (2012). A knowledge-based system approach for sensor fault modeling, detection and mitigation. *Journal of Expert Systems with Applications*, *39*(12), 10977–10989. doi:10.1016/j.eswa.2012.03.026

Willsky, A. S. (1976). A survey of design methods for failure detection in dynamic systems. *Automatica*, *12*(6), 601–611. doi:10.1016/0005-1098(76)90041-8

Appendix

A.1 Proof of CO method

Theorem A.1: *Given an ideal control system* (1) *with* $w(k) = 0$ *and* $v(k) = 0$, *when sensor* i_f *is faulty at time step k, the observer* i_f *gives the smallest norm estimation error if the fault signal satisfies*

$$\|f(k)\| > \|L_i F_i\|^{-1}(\|E_{i_f} e_{i_f}(k)\| + \|E_i e_i(k)\|), \quad i \neq i_f. \tag{A1}$$

Proof: When sensor i_f is faulty, $F_{i_f} = 0^{(m-1)\times 1}$. Then the estimation error of observer i_f is

$$e_{i_f}(k+1) = x(k+1) - \tilde{x}_{i_f}(k+1) = E_{i_f} e_{i_f}(k). \tag{A2}$$

In contrast, the estimation error of observer i ($i \neq i_f$) is

$$e_i(k+1) = x(k+1) - \tilde{x}_i(k+1) = E_i e_i(k) - L_i F_i f(k). \tag{A3}$$

Therefore, if Equation (A1) holds, the following is true

$$\|e_{i_f}(k+1)\| < \|e_i(k+1)\| \quad \forall i = 0, 1 \ldots, m_o \wedge i \neq i_f \tag{A4}$$

■

Remark A.2: There is no physical meaning for $\|f(k)\|$. Theorem A.1 gives a lower bound of $f(k)$ that the residual-based detection method could be used to select observer i_f, which is the one without the faulty sensor i_f.

A.2 Proof of Theorem 4.4

Theorem 4.4: *Given an ideal control system* (1) *with* $w(k) = 0$ *and* $v(k) = 0$, *the biases* $\tilde{d}_{\mu(v)}(k)$ *and* $\tilde{d}_{\Lambda,\mu(v)}(k)$ *are calculated according to Equations* (21) *and* (22), *respectively, with the following results:*

(1) *When the system is under disturbance,*

$$\forall \mu, v = 0, 1, \ldots, m_o \wedge \mu \neq v,$$
$$\tilde{d}_{\mu(v)}(k) = \tilde{d}_{\Lambda,\mu(v)}(k) = d(k).$$

(2) *When the system is under sensor fault,*

$$\forall \mu, v = 0, 1, \ldots, m_o \wedge \mu \neq v,$$
$$\tilde{d}_{\mu(v)}(k) = \tilde{d}_{v(\mu)}(k),$$
$$\tilde{d}_{\Lambda,\mu(v)}(k) \neq \tilde{d}_{\Lambda,v(\mu)}(k) \quad \text{if} \quad V_\mu \neq V_v. \tag{A5}$$

$$\tilde{d}_{\mu(v)}(k) = (D^{\mathsf{T}}D)^{-1}D^{\mathsf{T}}[\tilde{e}_{\mu(v)}(k+1) - E_\mu \tilde{e}_{\mu(v)}(k)],$$
$$\tilde{d}_{\Lambda,\mu(v)}(k) = ((D_{\Lambda,\mu})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}}$$
$$[\tilde{e}_{\Lambda,\mu(v)}(k+1) - E_{\Lambda,\mu}\tilde{e}_{\Lambda,\mu(v)}(k)], \tag{A6}$$

where $D_{\Lambda,\mu} = (V_\mu)^{-1}D$, *and* $E_{\Lambda,\mu} = (V_\mu)^{-1}E_\mu V_\mu$.

Proof: (1) According to Lemma 4.1, $\tilde{e}_{\mu(v)} = e_\mu$ if a disturbance exists. By substituting Equation (10) into Equation (21), the calculated bias becomes

$$\tilde{d}_{\mu(v)}(k) = (D^{\mathsf{T}}D)^{-1}D^{\mathsf{T}}[E_\mu e_\mu(k) + Dd(k) - E_\mu e_\mu(k)] = d(k). \tag{A7}$$

Similarly,

$$\tilde{d}_{\Lambda,\mu(v)}(k) = ((D_{\Lambda,\mu,})^{\mathsf{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathsf{T}}(V_\mu)^{-1}[E_\mu e_\mu(k) + Dd(k) - E_\mu e_\mu(k)]$$
$$= d(k). \tag{A8}$$

(2) Under sensor fault, the estimation error cannot be correctly calculated. Therefore, $\tilde{e}_{\mu(\nu)}$ in Equation (17) and e_μ in Equation (12) are substituted to Equation (21) to calculate the difference between two biases based on two observers,

$$\begin{aligned}\tilde{d}_{\mu(\nu)}(k) - \tilde{d}_{\nu(\mu)}(k) = (D^{\mathrm{T}}D)^{-1}D^{\mathrm{T}}[&e_\mu(k+1) - E_\mu e^{\mu}(k)\\ &+ E_\mu(E_\nu - E_\mu)^{-1}(L_\nu F_\nu - L_\mu F_\mu)f(k)\\ &- e_\nu(k+1) + E_\nu e_\nu(k)\\ &- E_\nu(E_\nu - E_\mu)^{-1}(L_\nu F_\nu - L_\mu F_\mu)f(k)]\\ &= 0. \end{aligned} \tag{A9}$$

When the biases are calculated based on Equation (22), then

$$\begin{aligned}\tilde{d}_{\Lambda,\mu(\nu)}(k) = ((D_{\Lambda,\mu})^{\mathrm{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathrm{T}}(V_\mu)^{-1}&[\tilde{e}_{\mu(\nu)}(k+1)\\ &- E_\mu\tilde{e}_{\mu(\nu)}(k)]\end{aligned} \tag{A10}$$

$$\begin{aligned}\tilde{d}_{\Lambda,\nu(\mu)}(k) = ((D_{\Lambda,\nu})^{\mathrm{T}}D_{\Lambda,\nu})^{-1}(D_{\Lambda,\nu})^{\mathrm{T}}(V_\nu)^{-1}&[\tilde{e}_{\nu(\mu)}(k+1)\\ &- E_\nu\tilde{e}_{\nu(\mu)}(k)]\end{aligned} \tag{A11}$$

are obtained for observer μ and ν, respectively. Based on Equation (A9), the following is true:

$$\tilde{e}_{\mu(\nu)}(k+1) - E_\mu\tilde{e}_{\mu(\nu)}(k) = \tilde{e}_{\nu(\mu)}(k+1) - E_\nu\tilde{e}_{\nu(\mu)}(k). \tag{A12}$$

So if $V_\mu \neq V_\nu$, then $((D_{\Lambda,\mu})^{\mathrm{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathrm{T}}(V_\mu)^{-1} \neq ((D_{\Lambda,\nu})^{\mathrm{T}}D_{\Lambda,\nu})^{-1}(D_{\Lambda,\nu})^{\mathrm{T}}(V_\nu)^{-1}$. Thus $\tilde{d}_{\Lambda,\mu(\nu)}(k) \neq \tilde{d}_{\Lambda,\nu(\mu)}(k)$. ■

A.3 Proof of Lemma 4.6

Lemma 4.6: *Given a control system* (1) *with bounded sensor noise and* $w(k) = 0$, $\|\tilde{e}_{\mu(\nu)}(k) - e_\mu(k)\|$ *is bounded by* $\|(E_\nu - E_\mu)^{-1}\|(\|L_\nu\| + \|L_\mu\|)\upsilon$.

Proof: When sensor noise exists in the system, the estimation error evolution becomes

$$e_\mu(k+1) = E_\mu e_\mu(k) - L_\mu v_\mu(k). \tag{A13}$$

Then, the difference of the estimated states between two observers μ and ν becomes

$$e_{\mu,\nu}(k+1) = E_\nu e_\nu(k) - E_\mu e_\mu(k) - L_\nu v_\nu(k) + L_\mu v_\mu(k). \tag{A14}$$

Therefore, the calculated estimation error becomes

$$\tilde{e}_{\mu(\nu)}(k) = e_\mu(k) - (E_\nu - E_\mu)^{-1}(L_\nu v_\nu(k) - L_\mu v_\mu(k)). \tag{A15}$$

So, $\|\tilde{e}_{\mu(\nu)}(k) - e_\mu(k)\|$ is bounded by $\|(E_\nu - E_\mu)^{-1}\|(\|L_\nu\| + \|L_\mu\|)\upsilon$. ■

A.4 Proof of Lemma 4.7

Lemma 4.7: *Given a control system* (1) *with bounded process noise and* $\upsilon(k) = 0$, $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ *is bounded by* $||((D_{\Lambda,\mu})^T D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^T(V_\mu)^{-1}||\omega$.

Proof: Estimation error can still be correctly calculated when the system is subject to process noise as proved in Lemma 5.

Then the bias calculated based on Equation (22) becomes

$$\tilde{d}_{\Lambda,\mu(\nu)}(k) = d(k) + ((D_{\Lambda,\mu})^{\mathrm{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathrm{T}}(V_\mu)^{-1}w(k). \tag{A16}$$

Therefore, $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ is bounded by $\|((D_{\Lambda,\mu})^{\mathrm{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathrm{T}}(V_\mu)^{-1}\|\omega$. ■

A.5 Proof of Lemma 4.8

Lemma 4.8: *Given a control system* (1) *with bounded sensor noise and* $w(k) = 0$, $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ *is bounded by* $||((D_{\Lambda,\mu})^T D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^T(V_\mu)^{-1}||(1 + ||E_\mu||)||(E_\nu - E_\mu)^{-1}||(||L_\nu|| + ||L_\mu||)\upsilon$.

Proof: When the system has sensor noise, by substituting Equation (A15) into Equation (22),

$$\begin{aligned}\tilde{d}_{\Lambda,\mu(\nu)}(k) = d(k) - ((D_{\Lambda,\mu})^{\mathrm{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathrm{T}}(V_\mu)^{-1}&\\ [(E_\nu - E_\mu)^{-1}(L_\nu v_\nu(k+1) - L_\mu v_\mu(k+1))&\\ - E_\mu(E_\nu - E_\mu)^{-1}(L_\nu v_\nu(k) - L_\mu v_\mu(k))].&\end{aligned} \tag{A17}$$

Therefore, $\|\tilde{d}_{\Lambda,\mu(\nu)}(k) - d(k)\|$ is bounded by $\|((D_{\Lambda,\mu})^{\mathrm{T}}D_{\Lambda,\mu})^{-1}(D_{\Lambda,\mu})^{\mathrm{T}}(V_\mu)^{-1}\|(1 + \|E_\mu\|)\|(E_\nu - E_\mu)^{-1}\|(\|L_\nu\| + \|L_\mu\|)\upsilon$. ■

A.6 Proof of Proposition 4.9

Proposition 4.9: *Given a control system* (1)*, and an open-loop observer* (3) *the following results can be drawn:*

(1) *If all of the eigenvalues of A lie inside the unit circle, then the estimation error of an open-loop observer is bounded;*
(2) *If one or more of the eigenvalues of A lie on the unit circle and* $\|A\| = 1$*, then the estimation error of an open-loop observer is bounded.*

Proof: The real state of the system is

$$x(k) = A^k x(0) + \Sigma_{i=0}^{k-1}A^i Bu(k-1-i) + \Sigma_{i=0}^{k-1}A^i w(k-1-i). \tag{A18}$$

The state estimated by the open-loop observer is

$$\hat{x}(k) = A^k\hat{x}(0) + \Sigma_{i=0}^{k-1}A^i Bu(k-1-i). \tag{A19}$$

Then, the estimation error of the open-loop observer is

$$e_o(k) = A^k e_o(0) + \Sigma_{i=0}^{k-1}A^i w(k-1-i). \tag{A20}$$

(1) If all of the eigenvalues of A lie inside the unit circle, then $A^k e_o(0)$ is converging and according to Lambers (2009)

$$\lim_{i\to\infty}\{A^i\}_{j_1 j_2} = 0 \quad j_1, j_2 = 1, \ldots, n. \tag{A21}$$

Let $\{\bar{A}\}_{j_1 j_2} = max(\{A^i\}_{j_1 j_2})$, where $i = 0, 1, \ldots, k-1$ and $\bar{A}$ is formed by $\{\bar{A}\}_{j_1 j_2}$. Then,

$$\Sigma_{i=0}^{k-1}A^i w(k-1-i) \leq \bar{A}\Sigma_{i=0}^{k-1}w(k-1-i). \tag{A22}$$

Since the random process noise w has zero-mean and bound ω, $\Sigma_{i=0}^{k-1}A^i w(k-1-i)$ is bounded as well.

(2) If one or more of the eigenvalues of A lie on the unit circle, then $A^k e_o(0)$ is bounded. The other term $\Sigma_{i=0}^{k-1}A^i w(k-1-i)$ is a linear combination of the random vector $A^i w(k-1-i)$. For a vector $Aw(k)$, each element is a linear combination of zero-mean random variables in vector $w(k)$ with the elements in the same row of A as coefficients

$$\{Aw(k)\}_j = \Sigma_{i=1}^{n}\{A\}_{j,i}\{w\}_j(k). \tag{A23}$$

Since $\|A\| = 1$, i.e., $\Sigma_{i=1}^{n}|\{A\}_{j,i}| \leq 1$, $Aw(k)$ is a zero-mean random vector with bound ω. Thus, $A^i w(k-1-i)$ is also a zero-mean random vector with bound ω. Therefore, $\Sigma_{i=0}^{k-1}A^i w(k-1-i)$ is bounded. ■

A.7 Proof of Proposition 4.10

Proposition 4.10: *Given a control system* (1), *an open-loop observer is updated every* $\kappa_{f,g}$ *time steps. The impact of the system noise on the averaged residual (A24) is mitigated.*

$$r_{avg,g}(k+(j_N-1)\kappa_{f,g}) = \frac{1}{j_N}\Sigma_{j=1}^{j_N} r_g(k+(j_N-j)\kappa_{f,g}), \quad \text{(A24)}$$

where j_N *is a positive integer.*

Proof: Since the process noise and sensor noise are zero-mean vectors,

$$\begin{aligned}\Sigma_{i=0}^{\infty} w(i) &= 0^{n\times 1},\\ \Sigma_{i=0}^{\infty} v(i) &= 0^{m\times 1}.\end{aligned} \quad \text{(A25)}$$

The residual generated by a single open-loop observer over one update period is

$$\begin{aligned} r_g(k+(j_N-j)\kappa_{f,g}) &= y(k+(j_N-j)\kappa_{f,g})\\ &\quad - C\hat{x}_g(k+(j_N-j)\kappa_{f,g})\\ &= Cx(k+(j_N-j)\kappa_{f,g}) + v(k+(j_N-j)\kappa_{f,g})\\ &\quad - C\hat{x}_g(k+(j_N-j)\kappa_{f,g})\\ &= CA^k e((j_N-j)\kappa_{f,g}) + v(k+(j_N-j)\kappa_{f,g})\\ &\quad + \Sigma_{i=0}^{k-1} CA^i w(k-1+(j_N-j)\kappa_{f,g}-i). \end{aligned} \quad \text{(A26)}$$

Then the averaged residual is

$$\begin{aligned} &r_{avg,g}(k+(j_N-1)\kappa_{f,g})\\ &\quad = \frac{1}{j_N}\Sigma_{j=1}^{j_N} r_g(k+(j_N-j)\kappa_{f,g})\\ &\quad = \frac{1}{j_N}\Sigma_{j=1}^{j_N}(CA^k e((j_N-j)\kappa_{f,g})\\ &\qquad + v(k+(j_N-j)\kappa_{f,g})\\ &\qquad + \Sigma_{i=0}^{k-1} CA^i w(k-1+(j_N-j)\kappa_{f,g}-i)). \end{aligned} \quad \text{(A27)}$$

If $j_N \to \infty$, then

$$r_{avg,g}(k+(j_N-1)\kappa_{f,g}) = \frac{1}{j_N}\Sigma_{j=1}^{j_N} CA^k e((j_N-j)\kappa_{f,g}) \quad \text{(A28)}$$

Therefore, the impact of system noise is mitigated. ■

A.8 Table of notations in the paper

Table A1. Notation

	Matrices
A,B,C,D,K,F	System matrices, controller gain, and fault vector
C_i, L_i, F_i, E_i	Output matrix, observer gain, fault vector, state matrix for observer i, $E_i = A - L_iC_i$
V_i	A collection of Eigenvectors of matrix E_i
$E_{\Lambda,i}, B_{\Lambda,i}, D_{\Lambda,i}$	Transformed matrices for observer i
	Variables
x,y,u,w,v,d,f	System state, output, input, process noise, sensor noise, disturbance, and fault signal
ω, υ	Bounds of the process noise and the sensor noise
m_o	Number of the non-critical sensors
n,m,p,s	Dimensions of system state, output, input, and disturbance
S_{nc}, S_c	Sets of non-critical sensors and critical sensors
y_i, v_i	Output and sensor noise for observer i
e_i, e_o	Estimation error of closed-loop observer i and an open-loop observer
$\bar{x}_i$	Estimated state by closed-loop observer i
$\hat{x}_{g,i}$	Estimated state by open-loop observer i in group g
k_{ss}	Time steps for a closed-loop observer to reach its steady state
	Indicators, index
I_A, I_F, I_D	Alarms for anomaly, sensor fault, and disturbance
i_f	Faulty sensor index
I_{FB}	Index of the closed-loop observer for feedback
	CO Method
Q_i	Weighting matrix for observer i
θ_{CO}	Threshold for the CO method
r_j	The residual generated by closed-loop observer i
$\tilde{f}$	Calculated fault signal
	CR Method
$e_{\mu,\nu}$	The difference of estimated states of two observers
$\tilde{e}_{\mu(\nu)}, \bar{e}_{\mu(\nu)}$	Estimation error of observer μ calculated based on observers μ and ν and its upper bound
$\tilde{e}_{\Lambda,\mu(\nu)}$	The calculated estimation error of observer μ after changing the coordinates
$\tilde{e}_{\mu}$	Overall estimation error of observer μ, which is a function of $\tilde{e}_{\mu(\nu)}, \nu = 0, 1, \ldots, m_o \wedge \nu \neq \mu$
$\tilde{e}_{\Lambda,\mu}$	Overall estimation error of observer μ after changing the coordinates
$\{cr_i\}_j$	CR of the jth state estimation error of observer i
$\tilde{d}_{\mu(\nu)}$	The bias based on the calculated estimation error $\tilde{e}_{\mu(\nu)}$
$\tilde{d}_{\Lambda,\mu(\nu)}, \bar{d}_{\mu(\nu)}$	The bias based on the calculated estimation error $\tilde{e}_{\Lambda,\mu(\nu)}$ and its upper bound
κ_{CR}	Time steps for the CR method
θ_{CR}	Threshold to determine the occurrence of an anomaly
$\theta_{d,\mu(\nu),\zeta(\eta)}$	Threshold to distinguish a sensor fault from a disturbance
ϕ_ν	Weighting ratio of calculated estimation error $\tilde{e}_{\mu(\nu)}$
$\psi_{\mu(\nu)}$	Weighting ratio of calculated bias $\tilde{d}_{\Lambda,\mu(\nu)}$
	MOLO Method
M,N	The number of open-loop observers groups and the number of open-loop observers in one group
$\kappa_{f,g}, \kappa_{\Delta,g}$	Update period, update interval between two adjacent open-loop observers for group g
$r_{g,i}$	Residual signal of observer i in group g
$H_g, r_{avg,g}$	Leading observer, averaged residual in group g
$\theta_{MOLO,g}$	Threshold for the MOLO
	CCI Method
$\kappa_{CCI}, \theta_{CCI}$	Horizontal window, threshold for the CCI method
Δu_i	Control input difference of closed-loop observer i
	Others
$\{\cdot\}_j$	The jth element of a vector, the jth row of a matrix, the jth diagonal element of a diagonal matrix
$\{\cdot\}_{j_1,j_2}$	The element at the j_1th row and the j_2th column of a matrix
$\|\| \cdot \|\|$	The infinity norm $\|\| \cdot \|\|_\infty$