Set-valued Observer-based Active Fault-tolerant Model Predictive Control

Feng Xu[1,2], Vicenç Puig[1], Carlos Ocampo-Martinez[1] and Xueqian Wang[2]

[1]Institut de Robòtica i Informàtica Industrial (CSIC-UPC),Technical University of Catalonia (UPC)
Llorens i Artigas, 4-6, 08028 Barcelona, Spain
[2]Center of Intelligent Control and Telescience, Graduate School at Shenzhen
Tsinghua University, 518055 Shenzhen, Guangdong, P.R.China

Abstract

This paper proposes an integrated actuator and sensor active fault-tolerant model predictive control (FTMPC) scheme. In this scheme, fault detection (FD) is implemented by using a set-valued observer, fault isolation (FI) is done by set manipulations and fault-tolerant control (FTC) is carried out through the design of a robust model predictive control (MPC) law. In this paper, a set-valued observer is used to passively complete the FD task while FI is actively performed by making use of the constraint-handling capability of robust MPC. The set-valued observer is chosen to implement fault detection and isolation (FDI) due to its simple mathematical structure that is not affected by the type of faults such as sensor, actuator and system-structural faults. This means that only one set-valued observer is needed to monitor all considered actuator and sensor statuses (health and fault) and to carry out the FDI task instead of using a bank of observers (each observer matching a health/fault status). Furthermore, in the proposed scheme, the advantage of robust MPC is that it can effectively deal with system constraints, disturbances and noises and allow to implement an active FI strategy, which can improve FI sensitivity when compared with the passive FI methods. Finally, a case study based on the well-known two-tank system is used to illustrate the effectiveness of the proposed FTMPC scheme.

Keywords: Actuator and Sensor Faults, Fault Detection and Isolation, Fault-tolerant Control, Model Predictive Control, Set-valued Observer

1 Introduction

For engineering systems, there exist two different ways to implement fault-tolerant control (FTC) (i.e., passive FTC (PFTC) and active FTC (AFTC)) [2, 27]. PFTC relies on the controller robustness, which means that the controller is designed to deal with all possible faults as system uncertainties. In general, the PFTC scheme is relatively easy to be implemented but only has limited fault-tolerant capability and suboptimal performance. In order to improve the FTC performance, the other alternative is to turn to the AFTC approach. The difference between PFTC and AFTC mainly consists in the uses of an extra module named FDI and some reconfiguration/accomodation strategy. The FDI module can capture the fault information in real time, which is further used for the FTC system to apply fault-tolerant strategies to cope with faults. The approach proposed in this paper involves both AFTC and PFTC by using the inherent PFTC ability of MPC.

FTC has been implemented with different control strategies such as pole placement, adaptive control, fuzzy control, among others [2, 27]. In this paper, the objective is to implement an FTMPC scheme, which aims to make full use of the advantages of MPC (i.e., constraint handling, multi-variable control, etc.). It is known that MPC is an optimization-based control strategy that can effectively deal with system constraints but relies on the accuracy of system model, while the occurrence of faults implies changes of the system model [4]. Considering the wide application of MPC in industry, it is necessary to endow MPC with active fault-tolerant ability by using FDI techniques, which can further facilitate its application [13].

In the literature, there already exist some works dealing with the topics on FTMPC and set-based FDI. In [18], an integrated actuator and sensor FTC scheme can be found, which was implemented by using virtual actuators and sensors. However, this paper does not consider system constraints on inputs, states and outputs and is based on a passive FDI method using invariant sets. In [25], a multi-sensor FTMPC scheme was specially proposed to handle sensor faults and is based on the same FDI method as in [18]. In [14], the proposed method uses an active set-based FDI with on-line computation of separating inputs. Comparing with the passive methods, the method in [14] can effectively reduce the FI conservatism but has to compute separating inputs on-line able to separate the output sets corresponding to all considered healthy and faulty statuses. Although the active method has lower FI conservatism, it requires higher computational resources than the passive method used in [18, 25]. In [21], the authors previously proposed an FTMPC scheme to handle sensor faults, which uses an active FI strategy based on interval observers and MPC. The proposed active FI strategy uses the ultimate sets (i.e., invariant sets) to establish guaranteed FI conditions off-line but implement FI on-line with off-line computed input sets, which can provide FI guarantees and have lower FI conservatism than the method in [18, 25] and lower computational complexity than that in [14]. In [20], an FTMPC scheme using the two-stage Kalman filter to implement FDI was proposed to deal with partial actuator failures, where the results showed the feasibility to integrate MPC with FDI but without considering key features such as the feasibility after fault occurrence of the optimization problem related to the MPC design. Moreover, the scheme reported in [20] neither provide FI guarantees nor use active FI strategies when compared with the approach proposed in this paper. In [24, 6], FTMPC strategies were used for wind turbine and aerospace benchmarks, respectively. In [15], an FDI method based on set-valued observers was proposed, where instead of identifying a fault status, the proposed method discards candidate faulty statuses to assure no occurrence of false alarms. More references can be found in [3, 5, 22] for set-based FDI and FTC including approaches using set-membership estimation and interval observers.

The contribution of this paper consists in the following three aspects, which have been already treated by the authors in a preliminary work [21]:

- use the transient sets induced by status changing to establish guaranteed FI conditions instead of the ultimate sets, which can reduce FI conservatism.
- isolate faults within a time window, which can be calculated off-line with the transient sets.
- integrate actuator and sensor FDI and FTMPC into one scheme with a relative simple structure and lower computational complexity.

In order to reach these objectives, the set-valued observer-based method is used to implement FDI instead of interval observers and invariant sets (see [17, 23]). The main reason is that the set-valued observer has a key advantage over the others. That is, the set-valued observer always has the same mathematical form for both actuator and sensor faults [1, 3]. In particular, for a set-valued observer, only the state/output equation is changed for modeling actuator/sensor faults instead of changing the whole internal model of the observer as in techniques based on interval/conventional observers. This means that the FDI structure can be simplified with the use of a single set-valued observer when both actuator and sensor faults are considered together into an FTC scheme.

The remainder of the paper is organized as follows. Section II introduces the proposed FTC scheme. Section III shows the FD principle based on the set-valued observer and proposes FDI methods to detect and isolate actuators and sensors. In Section IV, the proposed FTMPC strategy and related issues are presented. In Section V, a case study using the two-tank system is used to show the effectiveness of the proposed scheme. Section VI draws the conclusions of the paper. Finally, the definitions of several types of sets used throughout the paper are given and some important properties of zonotopes used to implement the proposed approach are summarized in Appendix A.

2 System Description

This section introduces the proposed FTMPC scheme, which includes the plant, control objective and robust MPC controller.

2.1 Plant Models

In this paper, we consider a group of actuator/sensor faults critical for the system safety/performance. Under the effect of actuator/sensor faults, the system can be modelled as a piece-wise discrete-time linear system:

$$x_{k+1} = Ax_k + BFu_k + \omega_k, \tag{1a}$$

$$y_k = GCx_k + \eta_k, \tag{1b}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{q\times n}$ are time-invariant matrices, $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^p$ and $y_k \in \mathbb{R}^q$ are state, input and output vectors at time instant k, respectively, $\omega_k \in \mathbb{R}^n$ and $\eta_k \in \mathbb{R}^q$ denote process disturbances and measurement noises, respectively, and $F \in \mathbb{R}^{p\times p}$ and $G \in \mathbb{R}^{q\times q}$ are diagonal matrices modelling actuator and sensor state (healthy or faulty) and take values from a finite set, each value corresponding to an actuator/sensor status, respectively. Note the switching nature of the system in (1) when faults occur and make matrices F and G vary along the time.

Remark 2.1 *The proposed FTMPC scheme can deal with both single and multiple actuator/sensor faults. In order to simplify the exposition, only the case of single fault is taken into account (i.e., at each time instant, only one actuator/sensor becomes faulty) in the remaining of the paper. However, in the same way, it is straightforward to extend the proposed scheme to handle multiple faults by considering that several elements in the matrices F/G are null simultaneously.*

Thus, in this paper, a collection of actuator and sensor faults are considered. Under Remark 2.1, F can take $p+1$ values (i.e., $F = F_{i_a}$ ($i_a \in \mathbb{I}_a = \{0, 1, 2, \cdots, p\}$), each one corresponding to one actuator status), where F_0 is the identity matrix denoting the healthy actuator status and F_{i_a} ($i_a \neq 0$) modeling the i_a-th actuator-fault status is denoted as

$$F_{i_a} = \mathrm{diag}(1, \cdots, 1, f_{i_a}, 1, \cdots, 1),\ i_a \in \mathbb{I}_a \setminus \{0\}, \tag{2}$$

where diag$(\cdot)$ denotes the diagonal matrix and f_{i_a}, a scalar inside the interval $[0,1)$, denotes the actuator-fault magnitude in the i_a-th actuator. Similarly, G can take $q+1$ values, (i.e., $G = G_{i_s}$ ($i_s \in \mathbb{I}_s = \{0, 1, 2, \cdots, q\}$), each one corresponding to one sensor status), where G_0 is the identity matrix denoting the healthy sensor status and G_{i_s} ($i_s \neq 0$) modeling the i_s-th sensor-fault status is represented as

$$G_{i_s} = \text{diag}(1, \cdots, 1, g_{i_s}, 1, \cdots, 1),\ i_s \in \mathbb{I}_s \setminus \{0\}, \tag{3}$$

where a scalar g_{i_s} inside the interval $[0,1)$ denotes the sensor-fault magnitude in the i_s-th sensor.

The process disturbance and measurement noise vectors ω_k and η_k are assumed to be bounded by known sets W and V, respectively, which are denoted as

$$W = \{\omega \in \mathbb{R}^n :\ |\omega - \omega^c| \leq \bar{\omega}, \omega^c \in \mathbb{R}^n, \bar{\omega} \in \mathbb{R}^n\}, \tag{4a}$$
$$V = \{\eta \in \mathbb{R}^q :\ |\eta - \eta^c| \leq \bar{\eta}, \eta^c \in \mathbb{R}^q, \bar{\eta} \in \mathbb{R}^q\}, \tag{4b}$$

where ω^c, η^c, $\bar{\omega}$ and $\bar{\eta}$ are constant and known vectors. Additionally, system states and input constraints are taken into acount and denoted as

$$X = \{x \in \mathbb{R}^n :\ |x - x^c| \leq \bar{x}, x^c \in \mathbb{R}^n, \bar{x} \in \mathbb{R}^n\}, \tag{5a}$$
$$U = \{u \in \mathbb{R}^p :\ |u - u^c| \leq \bar{u}, u^c \in \mathbb{R}^p, \bar{u} \in \mathbb{R}^p\}, \tag{5b}$$

respectively, where the vectors x^c, u^c, $\bar{x}$ and $\bar{u}$ are assumed to be constant and known.

Assumption 2.1 *The pairs (A, BF_{i_a}) and $(A, G_{i_s}C)$, for all $i_a \in \mathbb{I}_a$ and $i_s \in \mathbb{I}_s$, are stabilizable and detectable, respectively.*

2.2 Control Objective

In the proposed FTMPC scheme, the control objective is to regulate the system output to reach the given output setpoints. That is, in a healthy, actuator-fault or sensor-fault status, it is always expected to keep the system output close to a given setpoint. However, due to the potential effect of faults, the system may have to face a degree of performance degradation during faulty operation. But, practically, if the system can be kept in safe operation, a degree of performance degradation is acceptable in the faulty statuses. In this FTMPC scheme, for different statuses, different output setpoints may be defined. When the system is in the i_a-th actuator and i_s-th sensor system status[1], in the absence of disturbances and noises, the control objective can be described by

$$\lim_{k \to \infty} (y_k - y^*_{i_a, i_s}) \to 0, \tag{6}$$

where $y^*_{i_a,i_s}$ is the given output setpoint corresponding to the i_a-th actuator and i_s-th sensor status.

Both input and output vectors are p-dimensional and q-dimensional, respectively. Under Remark 2.1, there are totally $p+1$ actuator statuses (healthy or faulty) and $q+1$ sensor statuses (healthy or faulty) considered. This means that there are $(p+1) \times (q+1)$ system statuses[2].

[1] In the following, *in the i_a-th actuator and i_s-th sensor status* means that, in the current system, the actuators are in the i_a-th actuator status and the sensors are in the i_s-th sensor status.

[2] The terms *actuator status*, *sensor status* and *system status* have different meanings, where *actuator status* only focuses on changes in actuators, *sensor status* only considers changes in sensors while *system status* takes changes in both actuators and sensors into account.

Thus, $(p+1)\times(q+1)$ output setpoints should be given, each one corresponding to one system status[3]. With respect to (1), the plant model in the i_a-th actuator and i_s-th sensor status is obtained by neglecting ω_k and η_k in (1) as

$$\bar{x}_{k+1}^{i_a,i_s} = A\bar{x}_k^{i_a,i_s} + BF_{i_a}\bar{u}_k^{i_a,i_s}, \tag{7a}$$

$$\bar{y}_k^{i_a,i_s} = G_{i_s}C\bar{x}_k^{i_a,i_s}, \tag{7b}$$

where $\bar{u}_k^{i_a,i_s}$, $\bar{x}_k^{i_a,i_s}$ and $\bar{y}_k^{i_a,i_s}$ denote the nominal input, state and output vectors corresponding to the i_a-th actuator and i_s-th sensor status, respectively. With (7), a state-input setpoint pair $(x^*_{i_a,i_s}, u^*_{i_a,i_s})$ matching the given output setpoint $y^*_{i_a,i_s}$ can be obtained at steady state by solving

$$\begin{bmatrix} A-I & BF_{i_a} \\ G_{i_s}C & O \end{bmatrix}\begin{bmatrix} x^*_{i_a,i_s} \\ u^*_{i_a,i_s} \end{bmatrix} = \begin{bmatrix} O \\ y^*_{i_a,i_s} \end{bmatrix}, \tag{8}$$

which is obtained by considering the behavior of (7) at steady state.

Assumption 2.2 *Under the constraints (5), the equation (8) has at least one solution for all $i_a \in \mathbb{I}_a$ and $i_s \in \mathbb{I}_s$.*

Remark 2.2 *Without loss of generality, this paper only considers the regulation problem. However, the scheme can be extended to the output-tracking problem as long as extra changes are made.*

2.3 Robust MPC

In the proposed FTMPC scheme, due to the disturbances and noises, robust MPC is used to implement the control objective. Moreover, an active FI strategy to cope with both actuator and sensor faults is proposed, which is based on adjusting the input constraint of MPC controller (i.e., by means of the direct input constraint-handling capability of MPC.) Based on these two considerations, the min-max MPC technique is chosen as the control strategy [4, 8].

According to [4, 8], when the plant is in the i_a-th actuator and i_s-th sensor status, the corresponding min-max MPC controller is designed from the following open-loop finite-horizon optimization problem:

$$J_k = \min_{\mathbf{u}}\max_{\mathbf{w}} \sum_{j=0}^{N-1} \|(x_{k+j|k} - x^*_{i_a,i_s})\|^2_{Q_{i_a}} + \|(u_{k+j|k} - u^*_{i_a,i_s})\|^2_{R_{i_a}} + \|(x_{k+N|k} - x^*_{i_a,i_s})\|^2_{P_{i_a}}$$

$$\text{subject to} \quad \left.\begin{aligned} x_{k+j|k} &\in X, \\ u_{k+j|k} &\in U, \\ x_{k+N|k} &\in X_M^{i_a}, \\ x_{k|k} &= \hat{x}_k, \end{aligned}\right\} \forall \omega_{k+j|k} \in W, \tag{9}$$

with an MPC internal model

$$x_{k+j+1|k} = Ax_{k+j|k} + BF_{i_a}u_{k+j|k} + \omega_{k+j|k}, \tag{10}$$

[3] Although for each system status, an output setpoint is given (i.e., totally $(p+1)\times(q+1)$ output setpoints), this is just a theoretical result and some of the output setpoints may be the same. Practically, if the output setpoints can be the same under all the considered system statuses, this will be the best situation in light of the system performance.

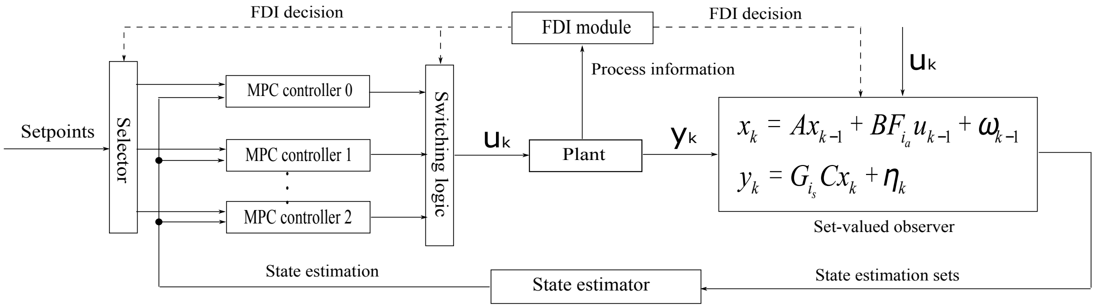

Figure 1: The proposed FTMPC scheme

where N denotes the prediction horizon, $\hat{x}_k$ is the system state estimation, $X_M^{i_a}$ is the terminal state-constraint that is the *maximal robust control invariant* (MRCI) set of the i_a-th dynamics (10) corresponding to the i_a-th actuator status under the constraints (5), $\mathbf{u} = [u_{k|k}, u_{k+1|k}, \cdots, u_{k+N-1|k}]$, $\mathbf{w} = [\omega_{k|k}, \omega_{k+1|k}, \cdots, \omega_{k+N-1|k}]$, and Q_{i_a}, R_{i_a} and P_{i_a} are the positive-definite matrices corresponding to the i_a-th MPC controller (see Appendix for the definition of *robust control invariant* (RCI) and MRCI sets). For the sake of understanding, the proposed FTMPC scheme is shown in Figure 1.

Remark 2.3 *In this scheme, a bank of MPC controllers are used, each corresponding to one actuator status. Thus, if the actuator status changes (i.e., the MPC internal model changes), another MPC controller should be activated. But, if the sensor status changes, only the state-input setpoint pair of the MPC controller matching a new sensor status needs to be used provided that this new sensor status uses a different output setpoint. That is, it is switched among a bank of robust MPC controllers for actuator-fault tolerance or a collection of state-input setpoint pairs for sensor-fault tolerance.*

3 Fault Detection and Isolation

This section introduces the set-valued observer and the corresponding FD and FI strategies used in the proposed FTMPC scheme, respectively.

3.1 Set-valued Observer

In order to introduce the set-valued observer, the healthy system status is taken as an example (i.e., the model (1) with both F and G as identity matrices).

Definition 3.1 *Given the system* (1) *in the healthy status and a measured output* y_k*, the measurement-consistent set at time instant* k *is defined as* $X_{y_k} = \{x_k \in \mathbb{R}^n : Cx_k - y_k \in -V\}$.

Definition 3.2 *Given the system* (1) *in the healthy status, the exact state estimation set at time instant* k *is obtained as* $X_k = (AX_{k-1} \oplus \{Bu_k\} \oplus W) \cap X_{y_k}$ $(k \geq 1)$*, which is the state set consistent with the output* y_k *and the initial state set* X_0 *satisfying* $x_0 \in X_0$.

Generally, it is difficult to obtain the exact state estimation set because of the complex geometric structure of sets. In order to reduce computational complexity, an outer approximation with a simple geometric structure (zonotopes) is used to bound the exact state estimation set.

Assume that a zonotopic outer-approximation $\hat{X}_{k-1}$ of the exact set X_{k-1} and the output measurement y_k are available. According to [1], the set-membership estimation algorithm to obtain a zonotope outer-approximation $\hat{X}_k$ can be divided into three steps:

- prediction step: a zonotope $\bar{X}_k = A\hat{X}_{k-1} \oplus \{Bu_k\} \oplus W$ is obtained to bound all possible values of the states at time instant k.
- measurement step: the output y_k is obtained and the current measurement-consistent set X_{y_k} is computed.
- correction step: the measurement-consistent set X_{y_k} is used to correct the predicted set $\bar{X}_k$ and an intersection $X_{ek} = \bar{X}_k \cap X_{y_k}$ can be obtained. Furthermore, a zonotopic outer-approximation $\hat{X}_k$ to bound X_{ek} is built and used for the next-step prediction.

Properties A.1, A.2 and A.3 in Appendix are used to implement the set-valued observer. Specially note that, as k increases, the order of the estimated state zonotopes also increases. In order to control the order of the estimated state zonotopes, Property A.3 is used in the implementation of the set-valued observer. Furthermore, in [9], an algorithm to compute a zonotopic outer-approximation of the intersection of a zonotope and a polytope is presented in Property A.4.

3.2 Fault Detection

In order to implement FDI, a set-valued observer is designed to monitor the system. When the system is in the i_a-th actuator and i_s-th sensor status, the set-valued observer should use the model

$$x_{k+1}^{i_a,i_s} = Ax_k^{i_a,i_s} + BF_{i_a}u_k + \omega_k, \tag{11a}$$

$$y_k^{i_a,i_s} = G_{i_s}Cx_k^{i_a,i_s} + \eta_k. \tag{11b}$$

At time instant k, a predicted state set $\bar{X}_k^{i_a,i_s}$ can be obtained by propagating (11a):

$$\bar{X}_k^{i_a,i_s} = A\hat{X}_{k-1}^{i_a,i_s} \oplus \{BF_{i_a}u_{k-1}\} \oplus W.$$

Furthermore, a predicted output set corresponding to $\bar{X}_k^{i_a,i_s}$ can be computed by

$$\bar{Y}_k^{i_a,i_s} = G_{i_s}C\bar{X}_k^{i_a,i_s} \oplus V. \tag{12}$$

Remark 3.1 *When the system is in the i_a-th actuator and i_s-th sensor status, if the state x_{k^*} is contained in the set $\hat{X}_{k^*}^{i_a,i_s}$ (i.e., $x_{k^*} \in \hat{X}_{k^*}^{i_a,i_s}$), $x_k \in X_{ek}^{i_a,i_s}$, $x_k \in \bar{X}_k^{i_a,i_s}$, $x_k \in \hat{X}_k^{i_a,i_s}$ and $y_k \in \bar{Y}_k^{i_a,i_s}$ will hold for all $k > k^*$. Note that, in this paper, it is assumed that the initial state of the plant is bounded by the initial set of the set-valued observer.*

It can be seen that the set-valued observer introduced in this subsection, which predicts the output set $\bar{Y}_k^{i_a,i_s}$, is a little different from that in Section 3.1. Particularly, for estimating state sets, it is better to compute $X_{ek}^{i_a,i_s}$ that has a smaller size than $\bar{X}_k^{i_a,i_s}$, while for FD, $\bar{X}_k^{i_a,i_s}$ and $\bar{Y}_k^{i_a,i_s}$ are more interesting. In this paper, the set-valued observer is used for both FD and state estimation. Thus, both output and state sets can be computed and both forms of the set-valued observer are used. According to [15], the FD task can be performed by checking whether or not $X_{ek}^{i_a,i_s}$ is empty. Moreover, according to previous results, Proposition 3.1 is stated.

Proposition 3.1 *When the system is in the i_a-th actuator and i_s-th sensor status, FD based on the set-valued observer consists in testing whether or not $X_{e_k}^{i_a,i_s}$ is empty or $y_k \in \bar{Y}_k^{i_a,i_s}$ is violated. Moreover, these two FD criteria are equivalent.*

Proof : The FD principle is to test the consistency between predictions and measurements. At time instant k, if $X_{e_k}^{i_a,i_s}$ is empty or $y_k \in \bar{Y}_k^{i_a,i_s}$ is violated, it implies that the consistency is lost, which indicates fault occurrence. Particularly, first, at time instant k, $X_{y_k}^{i_a,i_s}$ contains all possible values of states, which can explain the measured output y_k. $\bar{X}_k^{i_a,i_s}$ includes all possible values of states from prediction based on the state estimation set at the previous step. Thus, if $X_{e_k}^{i_a,i_s}$ is empty, it implies inconsistency between the measurements and predictions. Second, $\bar{Y}_k^{i_a,i_s}$ contains all possible values of outputs from prediction, $y_k \notin \bar{Y}_k^{i_a,i_s}$ also means inconsistency. Thus, if $X_{e_k}^{i_a,i_s}$ is empty, it implies that no values inside $\bar{X}_k^{i_a,i_s}$ can explain y_k (i.e., $y_k \notin \bar{Y}_k^{i_a,i_s}$). Similarly, if $y_k \notin \bar{Y}_k^{i_a,i_s}$, it means that $X_{y_k}^{i_a,i_s}$ cannot intersect with $\bar{X}_k^{i_a,i_s}$ (i.e., $X_{e_k}^{i_a,i_s}$ is empty). Thus, these two FD criteria are equivalent and can be used to detect a fault occurrence. □

3.3 Fault Isolation

3.3.1 Actuator Faults.

It is assumed that the system is in the i_a-th actuator and i_s-th sensor status and the i_a-th MPC controller with the i_s-th sensor state-input setpoint pair is used to control the system. If no faults occur, it implies that $u_k \in U$ always holds (as long as the corresponding MPC controller activated is always feasible). Moreover, the dynamics (1a) can be rewritten as

$$x_{k+1} = Ax_k + \begin{bmatrix} BF_{i_a} & I \end{bmatrix} \begin{bmatrix} u_k \\ \omega_k \end{bmatrix}. \tag{13}$$

If u_k is treated as disturbances, by taking $u_k \in U$ and $\omega_k \in W$ into account, a robust positively invariant (RPI) set (denoted as $X_s^{i_a}$) can be constructed for (13) (see [7] for the notion and computation of RPI sets). According to the definition of RPI sets, in the i_a-th actuator status, the system states keep staying inside $X_s^{i_a}$ if the previous states are inside $X_s^{i_a}$. Thus, for each actuator status, an RPI set can always be constructed. Note that, in order to assure the existence of RPI sets, Assumption 3.1 is made as follows:

Assumption 3.1 *The matrix A in (1) is a Schur matrix.*

For explaining actuator FI, it is assumed that an actuator status switching[4] is detected at time instant k_d while the system always keeps the same sensor status (i.e., the i_s-th sensor status). This implies that the actuator status of the system changes from the i_a-th one to another one. In this case, the system operation can be divided into three phases:

- for $k < k_d$, $x_k \in \bar{X}_k^{i_a,i_s}$ may hold and $y_k \in \bar{Y}_k^{i_a,i_s}$ must hold.
- for $k = k_d$, $x_{k_d} \notin \bar{X}_{k_d}^{i_a,i_s}$ and $y_{k_d} \notin \bar{Y}_{k_d}^{i_a,i_s}$ must hold.
- for $k > k_d$, it is not known whether $x_k \notin \bar{X}_k^{i_a,i_s}$ or $y_k \notin \bar{Y}_k^{i_a,i_s}$ can always hold.

[4] A status switching is from the health status to a fault status, from a fault status to another fault status or from a fault status to the health status.

Proposition 3.2 *In the i_a-th actuator and i_s-th sensor status, when an actuator fault is detected at time instant k_d, if no outer-approximations are introduced into the computational implementation of set-valued observer, the estimated state set sequence $\hat{X}_k$ should converge to and stay inside the minimal robust positively invariant (mRPI) set of the dynamics (13) for $k < k_d$.*

Proof : As long as the activated MPC controller is feasible, $u_k \in U$ should always hold. The construction of the corresponding RPI sets is based on the set-based dynamics $X_{k+1}^{i_a,i_s} = AX_k^{i_a,i_s} \oplus BF_{i_a}U \oplus W$ and the iteration of this set-based dynamics should converge to the mRPI set (denoted as $X_m^{i_a}$) of (13). For the set-valued observer, the prediction set $\bar{X}_k^{i_a,i_s}$ is computed by $\bar{X}_k^{i_a,i_s} = A\hat{X}_{k-1}^{i_a,i_s} \oplus BF_{i_a}u_{k-1} \oplus W$. According to the principle of set-valued observer, the estimated state set $X_{e_k}^{i_a,i_s}$ is over-bounded by $\hat{X}_k^{i_a,i_s}$ based on zonotopes. If no zonotopic outer-approximation is used in the implementation of set-valued observer (i.e., directly using accurate polytopic sets), $\hat{X}_k^{i_a,i_s} = X_{e_k}^{i_a,i_s}$ should hold. Thus, as k increases (sufficiently large), $\hat{X}_k^{i_a,i_s}$ and $X_{e_k}^{i_a,i_s}$ will converge to the mRPI set. □

However, the situation presented in Property 3.2 is ideal. Practically, in order to reduce the computational complexity of set-valued observer implementation, zonotopic outer-approximations are introduced into the set-valued observer as aforementioned. Thus, by using Proposition A.4 to implement the set-valued observer, the obtained sets are more conservative than the corresponding accurate sets due to errors. Thus, $\hat{X}_k^{i_a,i_s}$ may not converge to the mRPI set $X_m^{i_a}$. But, it is guaranteed that $\hat{X}_k^{i_a,i_s}$ will converge to an RPI set

$$X_s^{i_a} = \alpha X_m^{i_a} \tag{14}$$

for a value $\alpha \geq 1$ due to the fact that the scaling of an RPI set is also RPI for the linear discrete time-invariant dynamics, which can be used as the bounding set of $\hat{X}_k^{i_a,i_s}$ as k increases sufficiently.

For the proposed actuator FI strategy, the key is to describe the system behaviors one step before FD (i.e., $k = k_d - 1$). Thus, the aforementioned FD criterion yields

$$y_{k_d-1} = G_{i_s}Cx_{k_d-1} + \eta_{k_d-1} \in \bar{Y}_{k_d-1}^{i_a,i_s}, \tag{15a}$$

$$y_{k_d} = G_{i_s}Cx_{k_d} + \eta_{k_d} \notin \bar{Y}_{k_d}^{i_a,i_s}. \tag{15b}$$

By considering (15a), at time instant $k_d - 1$, the following results can be further obtained:

$$G_{i_s}Cx_{k_d-1} \in \bar{Y}_{k_d-1}^{i_a,i_s} \oplus \{-\eta_{k_d-1}\} \subseteq G_{i_s}C\bar{X}_{k_d-1}^{i_a,i_s} \oplus V \oplus (-V). \tag{16}$$

In steady-state operation, $\bar{X}_{k_d-1}^{i_a,i_s}$ should be inside the RPI set $X_s^{i_a}$ (note that, by choosing α, one can always find a set $X_s^{i_a}$ that satisfies this condition). By using (16), the following results can be obtained:

$$G_{i_s}Cx_{k_d-1} \in G_{i_s}CX_s^{i_a} \oplus V \oplus (-V). \tag{17}$$

By using (17), a fixed set (free from the effect of actuator status switching) can be constructed by means of Proposition A.4 to contain $x_{k_{d-1}}$ one step before FD, which is denoted as $\hat{X}_{k_{d-1}}^{i_a,i_s}$, i.e.,

$$x_{k_d-1} \in \hat{X}_{k_d-1}^{i_a,i_s}. \tag{18}$$

Although, at time instant $k_d - 1$, it is not known which status the system is in, (18) can always hold. This is the key of the proposed actuator FI approach. Based on $\hat{X}_{k_{d-1}}^{i_a,i_s}$ at the FD time $k_d - 1$, one step later, p possible sets can be constructed where x_{k_d} may enter, each of which corresponds to one possible actuator status. For example, due to $u_{k_d-1} \in U$, if the l_a-th ($l_a \in \mathbb{I} \setminus \{i_a\}$) actuator status occurs, a set containing x_{k_d} is

$$\hat{X}_{k_d}^{l_a,i_s} = A\hat{X}_{k_d-1}^{i_a,i_s} \oplus BF_{l_a}U \oplus W. \tag{19}$$

In this FTMPC scheme, FD is passive based on the set-valued observer, while FI is active by changing the input constraint of the corresponding MPC controller at the FD time k_d online. According to (13), it can be observed that the size and center of RPI sets of (13) can be manipulated by adjusting the input set. Thus, according to the idea of the proposed actuator FI method, the input-constraint set of the corresponding MPC controller should be switched from U to $U_{k_d+1}^{f,i_a,i_s}$ for the time instant k_d ($U_{k_d+1}^{f,i_a,i_s}$ is specially designed for FI, but for u_{k_d}, it is still generated based on U). Moreover, as long as the MPC controller is feasible at time instant k_d, $u_{k_d} \in U$ can be obtained and a set containing x_{k_d+1} can further be constructed as

$$\hat{X}_{k_d+1}^{l_a,i_s} = A\hat{X}_{k_d}^{l_a,i_s} \oplus BF_{l_a}U \oplus W. \tag{20}$$

Furthermore, by using $\hat{X}_{k_d+1}^{l_a,i_s}$, the p possible output sets to contain y_{k_d+1} can be obtained as

$$\hat{Y}_{k_d+1}^{l_a,i_s} = G_{i_s}C\hat{X}_{k_d+1}^{l_a,i_s} \oplus V. \tag{21}$$

As a result, for the l_a-th actuator status, the state and output sets after FD can be computed by

$$\hat{X}_{k+1}^{l_a,i_s} = A\hat{X}_{k}^{l_a,i_s} \oplus BF_{l_a}U_k^{f,i_a,i_s} \oplus W, \tag{22}$$

$$\hat{Y}_{k+1}^{l_a,i_s} = G_{i_s}C\hat{X}_{k+1}^{l_a,i_s} \oplus V,\ k \geq k_d + 1 \text{ and } l_a \in \mathbb{I}_a \setminus \{i_a\}, \tag{23}$$

where $\hat{X}_{k+1}^{l_a,i_s}$ and $\hat{Y}_{k+1}^{l_a,i_s}$ are the estimated state and output sets at time instant $k+1$ and U_k^{f,i_a,i_s} is the input set specially designed for active FI at time instant k.

3.3.2 Sensor Faults

It is assumed that the system is in the i_a-th actuator and i_s-th sensor status and that the actuator status always keeps being in the i_a-th one while the sensor status changes due to sensor faults. It is assumed that a sensor status switching is detected at time instant k_d. Then, the whole system operation can be divided into three phases. When a sensor-status switching is detected at time instant k_d, l_s ($l_s \in \mathbb{I}_s \setminus \{i_s\}$) is used to denote this new sensor status. Totally, there are q sensor-candidate statuses and q output candidate sets are constructed at time instant $k_d + 1$, each one corresponding to one sensor-candidate status. It can be known that, at time instant $k_d + 1$, the outputs will enter into at least one of these q output sets.

Assumption 3.2 *Before FD, the MPC controller is feasible (i.e., $u_k \in U$ holds for $k \leq k_d$).*

Remark 3.2 *Since before FD, it is not known whether the sensor status has changed, the system structure keeps the same. In order to guarantee the nominal (healthy) operation, Assumption 3.2 is stated.*

Under Assumption 3.2, for the l_s-th sensor status, the system state should always be inside the RPI set $X_s^{i_s}$ for $k \leq k_d$. Similar with actuator FI, at the FD time instant k_d, the input-constraint set of the MPC controller is adjusted from U to another input set $U_{k_d}^{f,i_a,i_s}$. If the

MPC controller is feasible at time instant k_d, $u_{k_d+1} \in U^{f,i_a,i_s}_{k_d+1}$ for time instant k_d+1 and x_{k_d+1} will enter into a set

$$\hat{X}^{i_a,l_s}_{k_d+1} = AX^{i_s}_s \oplus BF_{i_a}U \oplus W, \tag{24}$$

Thus, the output set can be computed as

$$\hat{Y}^{i_a,l_s}_{k_d+1} = G_{l_s}C\hat{X}^{i_a,l_s}_{k_d+1} \oplus V. \tag{25}$$

Similarly, for the l_s-th sensor status, the state and output sets after FD can be computed by

$$\hat{X}^{i_a,l_s}_{k+1} = A\hat{X}^{i_a,l_s}_{k} \oplus BF_{i_a}U^{f,i_a,i_s}_{k} \oplus W, \tag{26}$$

$$\hat{Y}^{i_a,l_s}_{k+1} = G_{i_s}C\hat{X}^{i_a,l_s}_{k+1} \oplus V,\ k \geq k_d+1 \text{ and } l_s \in \mathbb{I}_s \setminus \{i_s\}, \tag{27}$$

where $\hat{X}^{i_a,l_s}_{k+1}$ and $\hat{Y}^{i_a,l_s}_{k+1}$ are the estimated state and output sets corresponding to the l_s-th sensor status at time instant $k+1$.

Remark 3.3 *In Sections 3.3.1 and 3.3.2, the system behaviors before FD and after FD have been described. This implies that, for each candidate actuator/sensor status, one output set should be constructed at a time instant. Thus, for $k \geq k_d+1$, output sets for all considered candidate statuses are constructed at each time instant. This means that at a time instant, p output sets for actuator statuses and q output sets for sensor statuses should be considered. Note that at any time instant after FD and before FI, it is guaranteed that the system outputs will enter into one out of the $p+q$ output sets corresponding to this time instant, considering the single fault assumption. Thus, when actuator and sensor faults are considered together, there always exist $p+q$ candidate system statuses (p actuator and q sensor statuses) after fault occurrence.*

3.3.3 Integrated Actuator and Sensor Fault Isolation.

Since both actuator and sensor faults have different features and their effects on the system are also different, the observer-based methods generally use a bank of observers. This paper proposes an integrated actuator and sensor FI approach.

Under the framework of set-based FI, the objectives of the proposed integrated actuator and sensor FI method are to reduce the difference of the effect of actuator and sensor faults and lower the conservatism of FI conditions as much as possible. The former objective can be reached by using the set-valued observer. In this case, the same observer structure can be used to deal with the two types of faults, while the latter objective will be achieved by using MPC-based active FI, which is the goal of this subsection.

In order to explain the proposed FI strategy, it is assumed that the system is in the i_a-th actuator and i_s-th sensor status and a status changing is detected at time instant k_d. Thus, p after-fault output sets can be constructed for p actuator-candidate statuses and q after-fault output sets for q sensor-candidate statuses at any time instant $k \geq k_d+1$ (i.e., $\hat{Y}^{l_a,i_s}_{k}$ ($l_a \in \mathbb{I}_a \setminus \{i_a\}$) and $\hat{Y}^{i_a,l_s}_{k}$ ($l_s \in \mathbb{I}_s \setminus \{i_s\}$)). Moreover, with the results in Sections 3.3.1 and 3.3.2, it is known that the sizes and positions of those $p+q$ output sets can be adjusted by the specially designed input sets U^{f,i_a,i_s}_{k} ($k \geq\ k_d+1$). This feature is the basic principle of the proposed active FI method, which achieves FI by designing a sequence of input sets and using them as the input-constraint sets of MPC controller during the whole FI phase. Comparing with methods able to isolate faults in one step after FD, in this paper, the conservatism of FI conditions is reduced by properly augmenting the FI time.

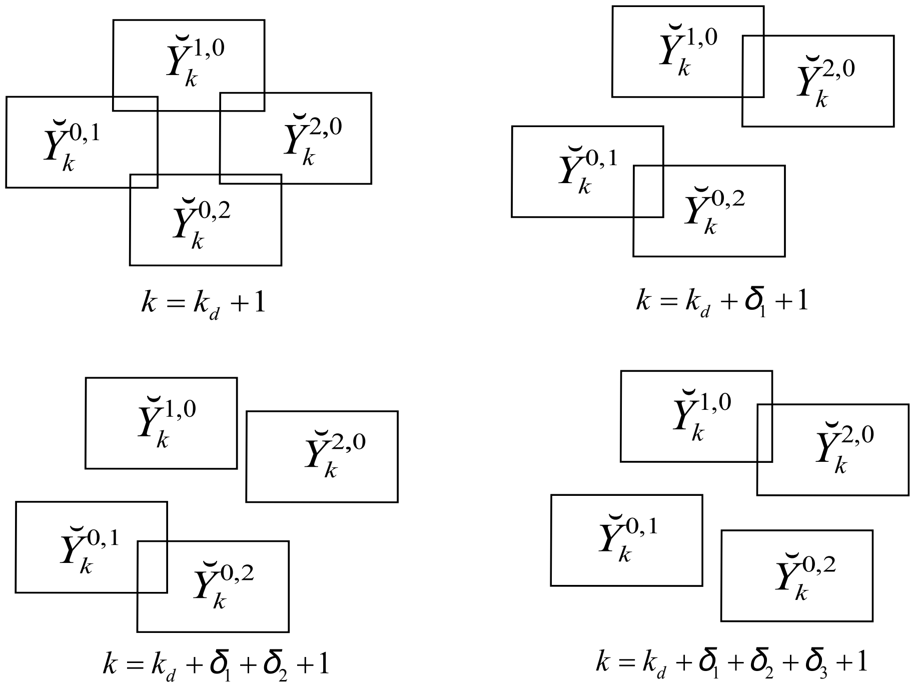

Figure 2: An example for FI

According to the results in Section 3.3.2, it is known that all estimated output sets $\hat{Y}_k^{l_a,i_s}$ and $\hat{Y}_k^{i_a,l_s}$ can be constructed off-line. This implies that guaranteed FI conditions off-line based on those output sets can be established. Since there always exist $p+q$ candidate statuses after FD, they should be able to be distinguished from each other for the sake of FI. In order to reduce the conservatism of FI conditions, it is not required to separate all $p+q$ sets at one time instant. Instead, it is considered to distinguish faulty statuses among them during a given time slot. For the sake of simplicity, an example used to show the procedure on how to separate all candidate statuses is shown in Figure 2. In this example, it is assumed that the system is healthy at the beginning and two actuator and two sensor faults are considered. Thus, the estimated output sets after FD are $\hat{Y}_k^{1,0}$, $\hat{Y}_k^{2,0}$, $\hat{Y}_k^{0,1}$ and $\hat{Y}_k^{0,2}$ and the procedure is shown as follows:

- at $k = k_d + 1$, it is observed that all the four sets intersect at least with another set, which means that it cannot be guaranteed to distinguish the four candidate faulty statuses at this time instant.

- at $k = k_d + \delta_1 + 1$, it is observed that, although it cannot be guaranteed to distinguish all the four statuses or several statuses out of the four, it can be guaranteed to differentiate the actuator-fault statuses 1 and 2 from the sensor-fault statuses 1 and 2 by using δ_1 extra time instants.

- at $k = k_d + \delta_1 + \delta_2 + 1$, furthermore, it is shown that it can be guaranteed to distinguish the actuator-fault status 1 from 2 with another δ_2 time instants.

- at $k = k_d + \delta_1 + \delta_2 + \delta_3 + 1$, it can be guaranteed to distinguish the remaining two sensor-fault statuses 1 and 2 with another δ_3 time instants. It does not matter that $\hat{Y}_k^{1,0}$

and $\hat{Y}_k^{2,0}$ intersect again at this time instant, because these two statuses have already been distinguished at $k = k_d + \delta_1 + \delta_2 + 1$.

Remark 3.4 *Here, $\delta_i \in \mathbb{N}_+$, for $i = 1, 2, 3$, which denotes the time needed to distinguish the corresponding candidate statuses, respectively. Theoretically, the smaller δ_1, δ_2 and δ_3 are, the better it is for FI and FTC. In practice, their values are decided by the system dynamics, faults and designed input sets at each time instant. Additionally, note that sets and intersections in Figure 2 are elements of a scheme used to illustrate the proposed FI idea.*

The example in Figure 2 is used to illustrate the way of implementing FI. From the example, it can be seen that the requirement for the designed input sets is that they should be able to separate all candidate output sets in a time period. Since all estimated output sets can be computed off-line, the input set sequence can be designed off-line as well. In this example, the output sets at time instant $k_d + 1$ are decided by U at time instant $k = k_d$. An input set $U_{k_d+1}^{f,i_a,i_s}$ is designed to separate the output sets at time instant $k_d + 2$. But, because $U_{k_d+1}^{f,i_a,i_s}$ cannot distinguish all the statuses at time instant $k_d + 2$, an input set sequence $\{U_{k_d+2}^{f,i_a,i_s}, U_{k_d+3}^{f,i_a,i_s}, \ldots, U_{k_d+\delta_1}^{f,i_a,i_s}\}$ can be further designed, which can distinguish the actuator-fault statuses 1 and 2 from the sensor-fault statuses 1 and 2. Furthermore, input sequences

$$\{U_{k_d+\delta_1+1}^f, U_{k_d+\delta_1+2}^f, \ldots, U_{k_d+\delta_1+\delta_2}^f\}$$

and

$$\{U_{k_d+\delta_1+\delta_2+1}^f, U_{k_d+\delta_1+\delta_2+2}^f, \ldots, U_{k_d+\delta_1+\delta_2+\delta_3}^f\}$$

are designed to distinguish the two actuator-fault and two sensor-fault statuses, respectively. As observed, the design of the input set sequence in this example is divided into four phases. Thus, the whole input set sequence for the isolation of the four faulty status is denoted as

$$\begin{aligned} \mathbb{U}^{f,i_a,i_s} = &\{U_{k_d+1}^{f,i_a,i_s}, U_{k_d+2}^{f,i_a,i_s}, \ldots, U_{k_d+\delta_1}^{f,i_a,i_s}, U_{k_d+\delta_1+1}^{f,i_a,i_s}, \\ &\ldots, U_{k_d+\delta_1+\delta_2}^{f,i_a,i_s}, U_{k_d+\delta_1+\delta_2+1}^{f,i_a,i_s}, \ldots, U_{k_d+\delta_1+\delta_2+\delta_3}^{f}\}. \end{aligned} \tag{28}$$

Remark 3.5 *Although the designed input set sequence includes four parts, they together represent the whole design procedure, which finally obtains the complete input set sequence $\mathbb{U}^{f,i_a,i_s}$ able to isolate all actuator and sensor faults. The most important point is that, because all output sets can be constructed off-line, designing and verifying a proper input set sequence can be completely done off-line. Furthermore, notice that the input sets should be designed one by one, because the estimated output sets at later time instants are decided by the estimated output sets and designed input sets at the previous time instants (i.e., first designing $U_{k_d+1}^{f,i_a,i_s}$, then checking whether the estimated output sets are separated to further design $U_{k_d+2}^{f,i_a,i_s}$, $U_{k_d+3}^{f,i_a,i_s}$, ..., step by step). Actually, it does not matter whether an input set can separate several estimated output sets at the next step. Instead, the whole input set sequence composed of all input sets must be able to isolate all faults during the whole time span covered by the input set sequence (see Figure 2).*

Note that, since system constraints are taken into account in the proposed scheme, the designed input sets must always satisfy the input constraints:

$$U_k^{f,i_a,i_s} \subseteq U, \text{ for } k \geq k_d + 1. \tag{29}$$

If the designed input set sequence spans a time length of N steps, it actually forms a tube $\mathbb{U}^{f,i_a,i_s}$ along the time axis. Moreover, when another tube $\mathbb{U}$ with the same time length is considered, which is composed of the input-constraint set U at each time instant, it can be observed that

$$\mathbb{U}^{f,i_a,i_s} \subseteq \mathbb{U}. \tag{30}$$

Remark 3.6 *Based on the above discussion, the design of the input set sequence for FI can be formulated as the problem of finding a subtube that can completely differentiate all considered statuses inside the input-constraint tube $\mathbb{U}$ during the time span of the subtube.*

The existence of such an input-set sequence is determined by the considered faults and the features of the system such as constraints, dynamics, etc. In the current paper, the input sequence is designed by trial and error since all relevant sets can be computed and the whole design procedure can be performed off-line. However, for theoretical integrity, the following assumption is made.

Assumption 3.3 *For all considered statuses, there exists an input set sequence of length N that can both satisfy (30) and differentiate all considered statuses described by the estimated output sets from (23) and (27).*

Remark 3.7 *Under Assumption 3.3, an input set sequence for on-line FI can be designed. In this paper, such an input set sequence is designed by performing a large number of simulations to implement the proposed active FI strategy. It should be emphasized that computing an optimal input set sequence to achieve guaranteed FI and control performance during the transition is still an open issue. However, methods proposed in [10, 16] to compute inputs separating different modes are already reported in the literature.*

In the system operation, once a faulty situation is detected at time instant k_d, it is started to apply the designed input-set sequence during the FI phase and the FI strategy is to test whether the following conditions are satisfied or not:

$$y_k \in \hat{Y}_k^{l_a,i_s},\ k \geq k_d + 1, \text{ for all } l_a \in \mathbb{I}_a \setminus \{i_a\}, \tag{31a}$$

$$y_k \in \hat{Y}_k^{i_a,l_s},\ k \geq k_d + 1, \text{ for all } l_s \in \mathbb{I}_s \setminus \{i_s\}, \tag{31b}$$

where the output sets are computed off-line in advance and used on-line for FI.

Note that, if an estimated output set does not contain y_k, it implies that the status corresponding to this output set should be excluded from the set of candidate statuses. Moreover, in the next time instants, the estimated output sets corresponding to this status will not be tested again for saving computational resources. Finally, till a time instant when there exists one and only one output set that can contain the current outputs, it means that the FI task is completed at this time instant.

Remark 3.8 *The FI conditions in Assumption 3.3 are sufficient but unnecessary. This means that if Assumption 3.3 holds, it is guaranteed that an input set sequence can be designed to isolate all considered statuses. However, even though Assumption 3.3 cannot be satisfied, it is still possible to isolate those considered statuses by making some extra efforts. Besides, the length of the input set sequence just says that within the time length, all the considered statuses can be distinguished. But in real time, a status can perhaps be isolated in a shorter time than the time length of the input sequence that corresponds to the worst case.*

4 Fault-tolerant Control

In this paper, the control strategy should satisfy several conditions:

- it should be able to control the system to reach its expected performance,
- it should be robust to process disturbances and measurement noises,
- it should be able to assist the FDI module to actively achieve FI,
- it should be able to deal with constraints.

Considering these conditions, robust MPC is chosen as the control strategy in this section.

4.1 Constraint Satisfaction

In Section 2.3, min-max MPC is briefly introduced. The details of the min-max MPC technique can be found in [8]. For each MPC controller, the plant model corresponding to an actuator status is used as its internal model. In order to deal with all actuator statuses, this paper proposes to switch among a bank of pre-designed MPC controllers, each corresponding to one actuator status, while for dealing with sensor statuses, only the state-input setpoint pair of the corresponding MPC controller is adjusted instead of switching the MPC controller. This means that an MPC controller is used for an actuator status and all sensor statuses corresponding to this actuator status. For example, in the i_a-th actuator and i_s-th sensor status, the i_a-th MPC controller is the activated controller and the state dynamics of the i_a-th actuator status is used as the internal model of this MPC controller. Moreover, corresponding to the i_s-th sensor status, the state-input setpoint pair $(x^*_{i_a,i_s}, u^*_{i_a,i_s})$ is used in the i_a-th MPC controller as in (9) to achieve the expected performance.

Generally, if no process disturbances and measurement noises are considered and the states are directly measurable, the MPC controller can directly be updated with accurate states and the system constraints can be satisfied by generated control actions. However, since process disturbances and measurement noises are taken into account, it is impossible to have the real states. Instead, only state estimations can be used to update the MPC controller to generate control actions (see (9)). Due to state estimation errors, the MPC controller cannot always guarantee constraint satisfaction. However, as long as the MPC controller is feasible, the generated control actions should always satisfy the input constraints. Thus, if the feasibility of the MPC controller can be guaranteed, the remaining main problem consists in how to guarantee state-constraint satisfaction with the generated control actions based on state estimations. In order to overcome this problem, the notion of RPI sets is used and Assumption 4.1 is stated.

Assumption 4.1 *The mRPI set $X_m^{i_a}$ of the dynamics (13) in the i_a-th actuator status corresponding to $u_k \in U$ and $\omega_k \in W$ is contained inside X, i.e., $X_m^{i_a} \subseteq X$ for $i_a \in \mathbb{I}_a$.*

Remark 4.1 *Considering that the scaling sets of an RPI set of the linear time-invariant dynamics are also the RPI sets of the dynamics, under Assumption 4.1, in the i_a-th actuator status, there always exists a scalar $\beta \geq 1$ such that the RPI set $X_s^{i_a} = \beta X_m^{i_a}$ satisfies $X_s^{i_a} \subseteq X$. Furthermore, the RPI set contained inside X should also be included inside the MRCI set $X_M^{i_a}$ based on the notions of the MRCI and RPI sets (i.e., $X_s^{i_a} \subseteq X_M^{i_a}$).*

Based on the state dynamics, Assumption 4.1 is only involved in actuator statuses and does not have any requirements on sensor statuses. Thus, if all considered actuator statuses satisfy

Assumption 4.1, the state and input constraints can be guaranteed by Assumption 4.1 and the feasibility of MPC open-loop optimization problem, respectively. Moreover, even though the state estimation errors exist, as long as the MPC controller is feasible, the generated control actions will always satisfy $u_k \in U$. Moreover, the states will always remain inside its RPI set (i.e., in the state-constraint set X) as long as the initial state belongs to the RPI set or the initial state does not belong to the RPI set but the RPI set is attractive for the state trajectory.

4.2 State Estimation

As observed in Figure 1, there is a module named *State Estimator*. Although the set-valued observer can estimate the sets of states at each time instant, specific state-estimation values has to be obtained for the MPC controller to update control actions since states are not measurable in the proposed FTMPC scheme. Thus, for a state estimator, it should have at least two conditions. First, the estimator should generate as good state estimations as possible for the MPC controller. Second, the state estimator should be able to generate state estimations that can guarantee the feasibility of MPC open-loop optimization problem. Thus, in this scheme, without loss of generality, Assumption 4.2 is made.

Assumption 4.2 *The system is healthy after initialization and the initial state belongs to the RPI set of the dynamics corresponding to the healthy status.*

Additionally, a fault occurrence always implies a status changing (i.e., the system changes from one status to another). The transition of a status changing can be described by the settling time of the plant dynamics. In this paper, the settling time is chosen to guarantee that, after it, the states have entered into the RPI set corresponding to a new system status. That is, if the actuator status does not change, the states finally return back to the same RPI set, while if the actuator status changes, the states enter into the RPI set corresponding to the new actuator status. Since a collection of system statuses are considered, without loss of generality, only one transient time is defined to describe all transitions for simplicity of analysis. Moreover, this transient time should also be able to describe the transition of the set-valued observer, which is also based on the settling time.

Definition 4.1 *The transient time T is defined as the maximal settling time of the dynamics of all considered system statuses and the set-valued observer such that the system has entered into the steady state of a new system status after a period of T elapses.*

Definition 4.2 *The steady-state operation of a system status is defined as its dynamic behaviors after the system operates in the status for at least one period of T.*

Assumption 4.3 *The persistent time of each status is at least longer than the transient time T.*

Under Assumptions 4.1 and 4.2 , system constraints should always be satisfied at steady state of the corresponding system status as long as no status changing occurs. Furthermore, it is assumed that the system is in the i_a-th actuator and i_s-th sensor status. Correspondingly, the current internal model used in the set-valued observer is

$$x_k^{i_a,i_s} = Ax_{k-1}^{i_a,i_s} + BF_{i_a}u_{k-1} + \omega_{k-1}, \tag{32a}$$

$$y_k = G_{i_s}Cx_k + \eta_k. \tag{32b}$$

Thus, at time instant k, a state estimation set $X_{ek}^{i_a,i_s}$ from the set-valued observer can be obtained and, according to the principle of the set-valued observer,

$$x_k \in X_{ek}^{i_a,i_s} \tag{33}$$

should hold. Moreover, due to the MPC controller and Assumption 4.1, at steady state, x_k should also belong to the terminal state-constraint set $X_M^{i_a}$:

$$x_k \in X_M^{i_a}. \tag{34}$$

Since x_k should be confined by both $X_{ek}^{i_a,i_s}$ and $X_M^{i_a}$, a refined state set is further obtained by using both $X_{ek}^{i_a,i_s}$ and $X_M^{i_a}$, that is

$$x_k \in X_{ek}^{i_a,i_s} \cap X_M^{i_a}.$$

In the proposed scheme, considering that the concrete state-estimation values are needed for updating the MPC controller at each time instant, a pragmatic state estimation is proposed, i.e.,

$$\hat{x}_k = \text{center}(X_{ek}^{i_a,i_s} \cap X_M^{i_a}), \tag{35}$$

where $\hat{x}_k$ denotes the state estimation used to update the corresponding MPC controller and center$(\cdot)$ denotes the center of a set. In this paper, the involved sets are convex, which means that the set intersection in (35) is also convex. Thus, if $X_{ek}^{i_a,i_s} \cap X_M^{i_a}$ is not a centered set, $\hat{x}_k$ in (35) denotes the center of the smallest box of the intersection $X_{ek}^{i_a,i_s} \cap X_M^{i_a}$.

Remark 4.2 *In the proposed scheme, (35) is used as the state estimation by considering that the center of an estimated set is the most representative point of the set. Actually, any points inside the set $X_{ek}^{i_a,i_s} \cap X_M^{i_a}$ cannot be as state estimations of MPC controller in the proposed scheme.*

Notice that only when the system is at the steady state of the corresponding system status, it can be guaranteed that the real states are always inside the intersection $X_{ek}^{i_a,i_s} \cap X_M^{i_a}$. Thus, the proposed state estimation (35) is only used during the steady-state operation of the proposed scheme.

4.3 Stability and Feasibility

Before FD, the scheme uses the RPI sets to describe system behaviors in different actuator statuses. In order to construct the RPI sets, a precondition that the plant dynamics should be stable is required (see Assumption 2.1). Thus, as long as the input vector is bounded, the system can always keep BIBO (bounded-input, bounded-output) stability. Note that the details about how to design an MPC controller able to stabilize the system are out of the scope of this paper. The readers are referred to [4, 8].

As aforementioned, the feasibility of MPC open-loop optimization problem is fundamental not only for the MPC controller but also system constraint satisfaction under the relevant assumptions. In this paper, the feasibility is considered in two different phases, respectively:

- the feasibility of MPC open-loop optimization problem during the steady-state operation,
- the feasibility of MPC open-loop optimization problem during the transient-state operation.

According to [4], if the terminal state-constraint set of MPC controller is an RCI set, as long as the state estimation is always inside the RCI set, the feasibility can be guaranteed by using the state estimation. As seen in (9), the MRCI set $X_M^{i_a}$ is used as the terminal state-constraint set, which is mainly used to guarantee the feasibility of MPC open-loop optimization problem. Thus, during the steady-state operation, based on (35), the feasibility of the corresponding MPC open-loop optimization problem can always be guaranteed. However, (35) can only guarantee the feasibility during the steady-state operation.

If the system is during the transition induced by faults, (33) or (34) may not hold, which will result in that (35) may not accurately or cannot be obtained. This means that (35) cannot be used as the state estimation during the whole operation including the steady-state and transient-state phases. Instead, a new state estimation strategy should be further proposed specially for the transient-state operation.

Definition 4.3 *The transient-state phase is defined as the time period from the steady-state phase of a health/fault status to that of another status (fault/health).*

According to Definition 4.3, without loss of generality, the transient-state operation is mainly divided into two phases. The first phase covers from the fault occurrence to FD while the second phase goes from FD to FI. Note that the transition after FI is omitted here, but it will be discussed in the following subsection. Since before FD, it is not known whether or not a fault has appeared and no measures can be taken to deal with the fault during this period, the following assumption has to be made.

Assumption 4.4 *During the first phase of the transition, the system constraints are satisfied and the MPC feasibility is preserved.*

Thus, in this paper, only the second phase of the transition is considered. It is assumed that the system is in the i_a-th actuator and i_s-th sensor status and the system becomes faulty at time instant k_d. According to the proposed FI strategy, the input-constraint set of the corresponding MPC controller should be adjusted from U to U_k^{f,i_a,i_s} at the FD time to implement FI for $k \geq k_d + 1$. The procedure of adjusting the input-constraint set of the corresponding MPC controller set is that, at time instant k_d, the input-constraint set is adjusted from U to $U_{k_d+1}^{f,i_a,i_s}$ and it is tested whether the fault can be isolated at time instant $k_d + 1$ or not. If the fault can be isolated at time instant $k_d + 1$, it means that the FI task has been completed at this time instant and the whole system will be reconfigured at time instant $k_d + 1$. Otherwise, the adjustment continues from $U_{k_d+1}^{f,i_a,i_s}$ to $U_{k_d+2}^{f,i_a,i_s}$ at time instant $k_d + 1$ to test if the fault can be isolated at time instant $k_d + 2$. For $k \geq k_d$, this procedure will be repeated till the fault is isolated and the system is reconfigured. Note that, according to the proposed FI strategy, the FI time will not be longer than the time span of the designed input set sequence.

By adjusting the input-constraint set of the corresponding MPC controller from U to the input set sequence U_k^{f,i_a,i_s} ($k \geq k_d + 1$), it can be guaranteed that a status can be isolated in a finite period. However, except for FI, in this scheme, the feasibility and constraint satisfaction should also be guaranteed during the FI phase. In this paper, the idea to guarantee the feasibility is to simultaneously adjust the terminal state-constraint set of MPC controller together with the adjustment of input-constraint sets. Thus, for the i_a-th actuator status, corresponding to the input set sequence U_k^{f,i_a,i_s} ($k \geq k_d + 1$) and the state-constraint set X, an MRCI set sequence $X_{M,k}^{i_a}$ ($k \geq k_d + 1$) can be constructed. Considering that the MPC controller is always based on the i_a-th model of the plant before FI/reconfiguration/accommodation, in order to guarantee the feasibility, the pair ($U_{k_d+1}^{f,i_a,i_s}$, $X_{M,k}^{i_a}$) is used as the input-constraint and terminal state-constraint sets of the corresponding MPC controller, respectively, for $k \geq k_d + 1$.

Moreover, during the FI phase, a point inside each terminal state-constraint set $X_{M,k}^{i_a}$ for $k \geq k_d + 1$ is chosen as the state estimation. As a result, corresponding to the designed input set sequence, a state-estimation sequence with the same time length can be obtained, which is used to update the MPC controller and guarantee the feasibility during the FI phase. Additionally, except for feasibility guarantee, the state estimation sequence should also be able to guarantee constraint satisfaction during the transition. For the sake of clarity in presentation, the following notation is used to express state estimations:

$$\hat{x}_k = p(X_{M,k}^{i_a}), \quad k \geq k_d + 1, \tag{36}$$

where $p(\cdot)$ denotes a selected point inside $X_{M,k}^{i_a}$ such that at time instant k, the feasibility can be guaranteed and the state constraints can be satisfied.

Note that, the constraint satisfaction during the transition should be considered in case of sensor-status and actuator-status changing. Actually, as long as the MPC controller keeps feasible during the transition, the generated control actions should always satisfy $u_k \in U$. This means that, if a sensor-status changing occurs, because the state dynamics do not change, the system states should still stay inside the corresponding RPI set and the state constraints will not be violated. In this case, the chosen state estimation sequence only needs to guarantee the feasibility. But, if an actuator-status changing occurs, the state dynamics are changed. However, under Assumption 4.1, the RPI sets corresponding to all actuator statuses should be inside the state-constraint set. Thus, the state estimation sequence can be designed to guarantee the state-constraint satisfaction during the transition.

Remark 4.3 *In this paper, the state estimations $\hat{x}_k$ used to update the corresponding MPC controller during the transition are chosen by off-line simulations.*

Thus, by using the proposed state estimations during both transient-state and steady-state operation, respectively, the scheme can be feasible, the system constraints can be satisfied and the faults can be isolated.

4.4 Fault-tolerant Control

It is assumed that a fault is isolated at time instant k_i, the proposed scheme will take measures to accommodate/reconfigure the system to tolerate the effect of the fault. In this paper, FTC is implemented by choosing the state-input setpoint pair or/and the MPC controller, and the dynamical model corresponding to the current actuator/sensor status are used to adjust/switch the MPC controller to match the current actuator/sensor status and to update the internal model of the set-valued observer, respectively.

In order to explain the FTC procedure, it is assumed that the system first operates in the i_a-th actuator and i_s-th sensor status and then the j_a-th actuator status has occurred and been isolated. At the FI time, according to the proposed FTC strategy, the i_a-th MPC controller should be switched to the j_a-th MPC controller and the input-state setpoint pair $(x^*_{j_a,i_s}, u^*_{j_a,i_s})$ is used for this new MPC controller. Furthermore, the j_a-th model of the plant should be used to update the internal model of the set-valued observer and the state, input and terminal constraint sets of the j_a-th MPC controller should be X, U and $X_M^{j_a}$ again, respectively. However, due to the switching of MPC controller, the states at the FI time may not be inside the terminal state-constraint set $X_M^{j_a}$. In this case, it cannot be guaranteed that the current MPC controller is always feasible and the system constraints are always satisfied after accommodation/reconfiguration. After accommodation/reconfiguration, an approach similar to(36) is used (i.e., choose a point inside $X_M^{j_a}$ to guarantee the feasibility of MPC open-loop optimization problem and system constraint satisfaction at each time instant during the period before the system enters into steady-state operation in the new status)

$$\hat{x}_k = p(X_M^{j_a}), \quad k \geq k_i. \tag{37}$$

Remark 4.4 *Although each MPC in the bank controllers is designed to be stable, the stability on switching among them is not guaranteed. The design of a bank of switched min-max MPC controllers preserving stability is a complex topic not yet solved in the literature. There exists some very recent results addressing this issue in the case of tube-based MPC [26], that can be useful for addressing the stability during switching.*

Remark 4.5 *The expression presented in (37) is only used as the state estimation before entering the steady state of a new system status after accommodation/reconfiguration, where the time length of its use is given by T.*

Thus, after the time period T elapses, it implies that the states have entered into the RPI set corresponding to a new system status. In this case, the state estimation (35) is used again and then the system can operate in this new system status. For illustrative purposes, in the j_a-th actuator status, (35) should be rewritten as

$$\hat{x}_k = \text{center}(X_{ek}^{j_a,i_s} \cap X_M^{j_a}). \tag{38}$$

Note that, in order to summarize the proposed FTMPC scheme, it is assumed that the system is in the i_a-th actuator and i_s sensor status and Algorithm 1 is made for clarity of understanding.

5 Case Study

In this paper, the two-tank system presented in [19] is revisited as a case study, which is shown in Figure 3 (taken from [12]). The control objective is to regulate both tank levels towards a given setpoint. Figure 3 shows the two manipulated variables u_1 and u_2 corresponding to a pump and a valve, respectively. The equilibrium levels of the two tanks are denoted as $\bar{x}_1$ and $\bar{x}_2$ and x_1 and x_2 are used to denote the incremental water levels in the two tanks relative to their equilibrium values. With the water levels of the two tanks as states and outputs and under the condition $\bar{x}_1 < l_v < \bar{x}_2$, where l_v is the height of the valve, the linearized model of the two-tank system around $\bar{x}_1$ and $\bar{x}_2$ can be written as

$$x_{k+1} = \begin{bmatrix} -0.25 & 0 \\ 0.25 & -0.25 \end{bmatrix} x_k + \begin{bmatrix} 1 & -0.5 \\ 0 & 0.5 \end{bmatrix} F_{i_a} u_k + \omega_k, \tag{39a}$$

$$y_k = G_{i_s} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + \eta_k, \tag{39b}$$

where F_{i_a} and G_{i_s} are used to model the actuator and sensor statuses (healthy or faulty), respectively. Additionally, it is considered that $|\omega| \leq \begin{bmatrix} 0.0001 & 0.0001 \end{bmatrix}^T$ and $|\eta| \leq \begin{bmatrix} 0.0001 & 0.0001 \end{bmatrix}^T$.

In this example, the performance loss of the pump and valve and the performance degradation of the two water-level sensors are considered and the healthy and faulty statuses are as follows:

$$F_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; F_1 = \begin{bmatrix} 0.7 & 0 \\ 0 & 1 \end{bmatrix}, \; F_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.7 \end{bmatrix}, \; G_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; G_1 = \begin{bmatrix} 0.8 & 0 \\ 0 & 1 \end{bmatrix}, \; G_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.7 \end{bmatrix},$$

Algorithm 1: FTMPC algorithm

```
Initialization (the $i_a$-th actuator and $i_s$-th sensor status, controller, set-valued observer, etc.);
At time instant $k$: FD $\leftarrow$ FALSE, FI $\leftarrow$ FALSE, $y_k \in \bar{Y}_k^{i_a,i_s}$, $\hat{x}_k \leftarrow \mathrm{center}(X_{ek}^{i_a,i_s} \cap X_M^{i_a})$;
(Fault detection)
while FD $\neq$ TRUE do
  $k \leftarrow k+1$;
  Obtain $y_k$ and $\bar{Y}_k^{i_a,i_s}$;
  if $y_k \notin \bar{Y}_k^{i_a,i_s}$ then
    FD $\leftarrow$ TRUE;
  end if
end while
(Fault isolation)
At time instant $k = k_d$:
Adjust $U$ and $X_M^{i_a}$ to $U_{k_d+1}^{f,i_a,i_s}$ and $X_{M,k_d+1}^{i_a}$;
$\hat{x}_{k_d+1} \leftarrow p(X_{M,k_d+1}^{i_a})$;
while FI $\neq$TRUE do
  $k \leftarrow k+1$;
  Obtain $y_k$;
  Test (31) for actuator/sensor FI;
  if a fault is isolated by (31) then
    FI $\leftarrow$ TRUE;
  else
    Use $U_k^{f,i_a,i_s}$ and $X_{M,k}^{i_a}$ as MPC constraints for $k_d+2 \leq k \leq k_i$;
    $\hat{x}_k \leftarrow p(X_{M,k}^{i_a})$ for $k_d+2 \leq k \leq k_i$;
  end if
end while
(Fault-tolerant control)
At time instant $k_i$:
1. Reconfigure/accommodate the closed-loop system as explained in Subsection 4.4;
2. Adjust MPC constraints to $U$ and $X_M^{j_a}/X_M^{i_a}$ for the $j_a$-th actuator/$j_s$-th sensor status;
3. Select a new state-input setpoint pair as explained in Subsection 4.4;
4. Update the internal model of the set-valued observer as explained in Subsection 4.4;
for $k_i < k \leq k_i + T$ do
  Use $\hat{x}_k \leftarrow p(X_{M,k}^{i_a})$;
end for
Use (37) as the state estimation;
The system will operate in steady state of the new status;
return
```

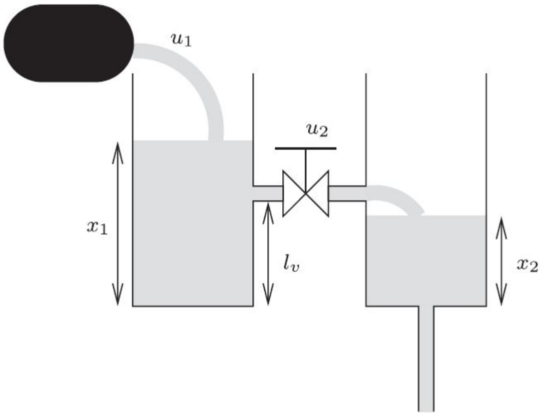

Figure 3: Two-tank system

where F_0 and G_0 mean that all the actuators (pump, valve) and sensors are healthy, F_1 and G_1 mean the performance degradation of the pump and the water-level sensor for the first tank, respectively, and F_2 and G_2 mean the performance degradation of the valve and the water-level sensor for the second tank, respectively.

Moreover, it is considered that the system inputs and states are constrained in sets

$$U = \left\{ u : \begin{bmatrix} 0.15 \\ 0 \end{bmatrix} \leq u \leq \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix} \right\},$$
$$X = \left\{ x : \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq x \leq \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} \right\}.$$

A set-valued observer with the models of the corresponding considered system statuses as its internal models is designed to monitor the plant, whose parameter is designed as

$$\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where Λ can be arbitrarily chosen as long as the set-valued observer can work well (see [9]).

For obtaining the RPI sets that can contain the estimated state sets, $\alpha = 1$ in (14) is chosen by means of a set of simulations. Furthermore, three MPC controllers are designed to control the plant under the corresponding statuses. The prediction horizon is given as $N = 2$ and the other parameters of the MPC controllers are designed as

$$Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, P_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_1 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, P_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Without loss of generality, this example only considers the process from healthy to faulty. Thus, totally four scenarios are considered:

- Scenario 1: from time instant $k = 1$ to $k = 50$, the system is healthy, while from time instant $k = 50$ to $k = 100$, a fault in the first actuator occurs,

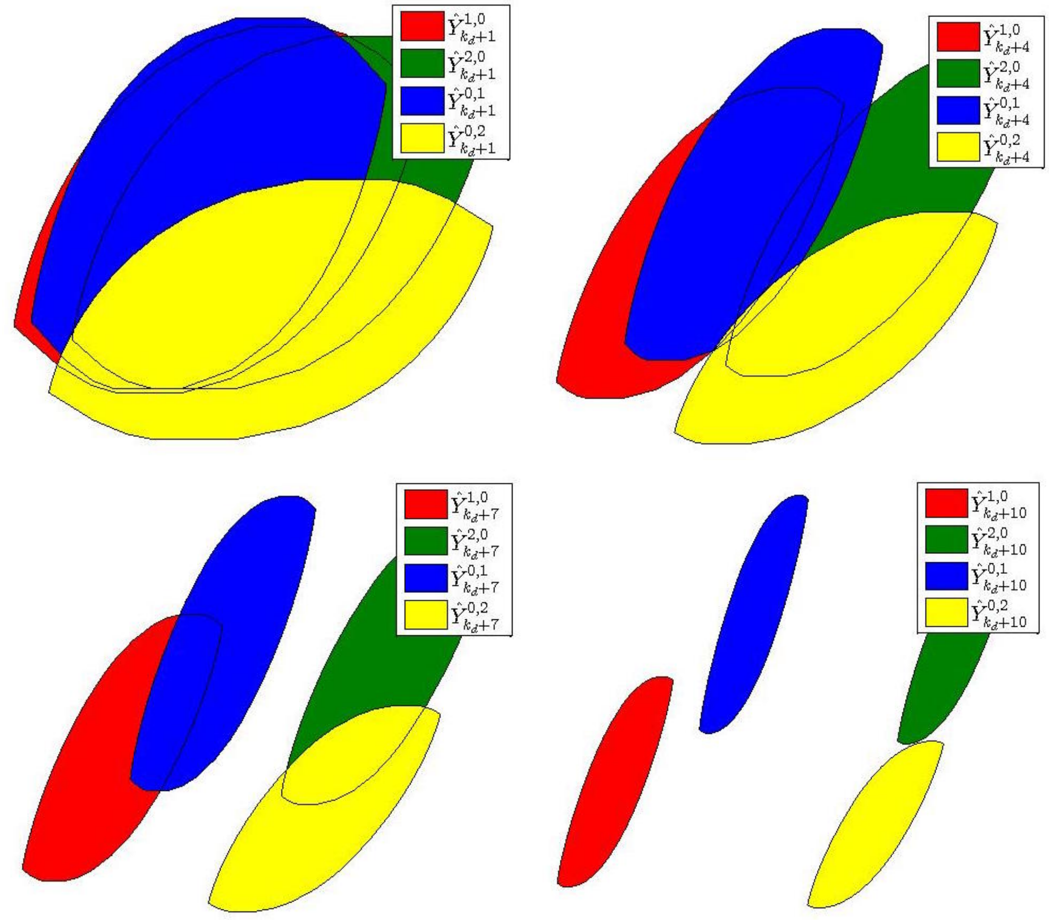

Figure 4: Estimated output sets

- Scenario 2: from time instant $k = 1$ to $k = 50$, the system is healthy, while from time instant $k = 50$ to $k = 100$, a fault in the second actuator occurs,
- Scenario 3: from time instant $k = 1$ to $k = 50$, the system is healthy, while from time instant $k = 50$ to $k = 100$, a fault in the first sensor occurs,
- Scenario 4: from time instant $k = 1$ to $k = 50$, the system is healthy, while from time instant $k = 50$ to $k = 100$, a fault in the second sensor occurs.

Furthermore, the set-points for the outputs are defined as $y^*_{0,0} = y^*_{0,1} = y^*_{0,2} = y^*_{1,0} = y^*_{2,0} = \begin{bmatrix} 0.4 & 0.6 \end{bmatrix}^T$. Thus, under the considered statuses, the corresponding state and input setpoints can be computed as

$$x^*_{0,0} = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix},\ x^*_{0,1} = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix},\ x^*_{0,2} = \begin{bmatrix} 0.4 \\ 0.8571 \end{bmatrix},\ x^*_{1,0} = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix},\ x^*_{2,0} = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix},$$
$$u^*_{0,0} = \begin{bmatrix} 0.15 \\ 0.1 \end{bmatrix},\ u^*_{0,1} = \begin{bmatrix} 0.15 \\ 0.05 \end{bmatrix},\ u^*_{0,2} = \begin{bmatrix} 0.2143 \\ 0.2286 \end{bmatrix},\ u^*_{1,0} = \begin{bmatrix} 0.2146 \\ 0.1 \end{bmatrix},\ u^*_{2,0} = \begin{bmatrix} 0.15 \\ 0.1429 \end{bmatrix}.$$

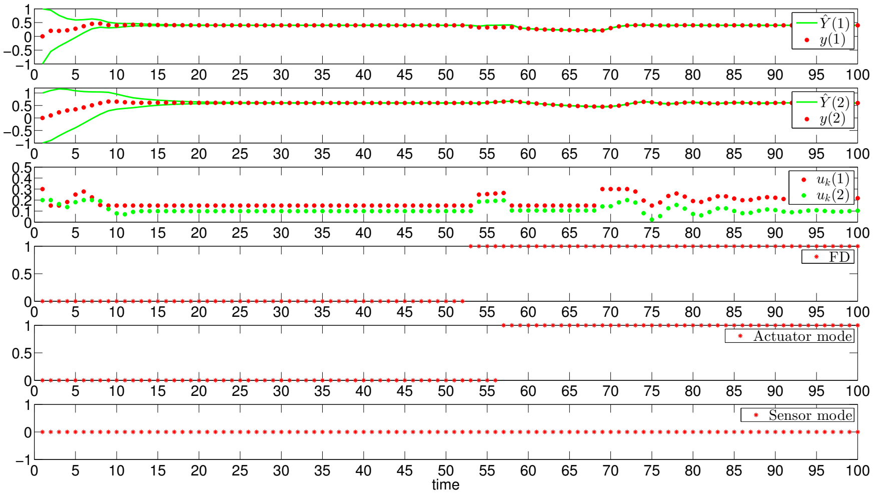

Figure 5: Scenario 1

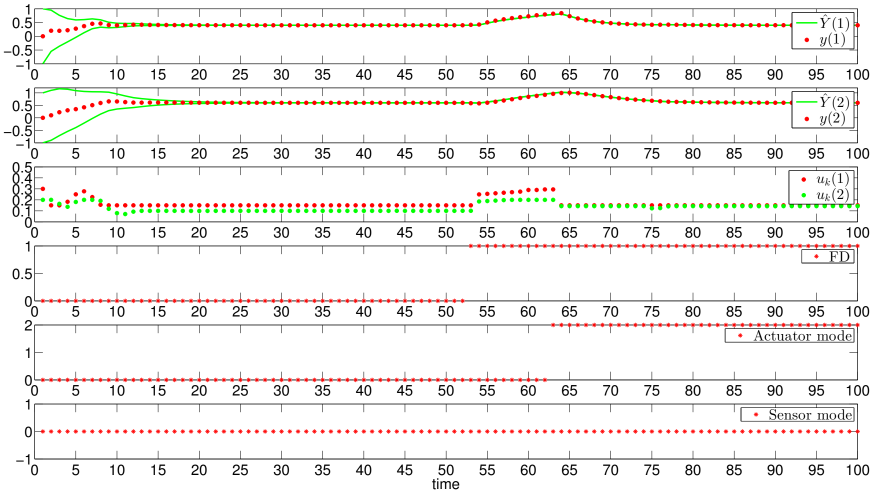

Figure 6: Scenario 2

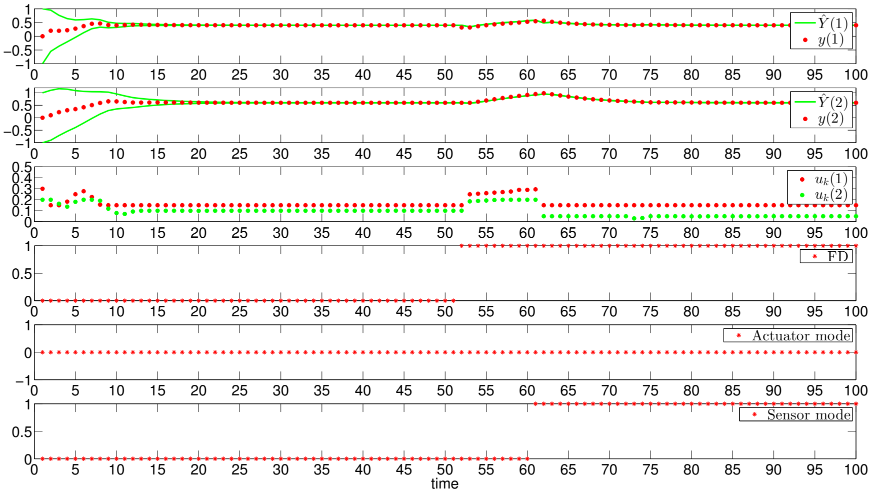

Figure 7: Scenario 3

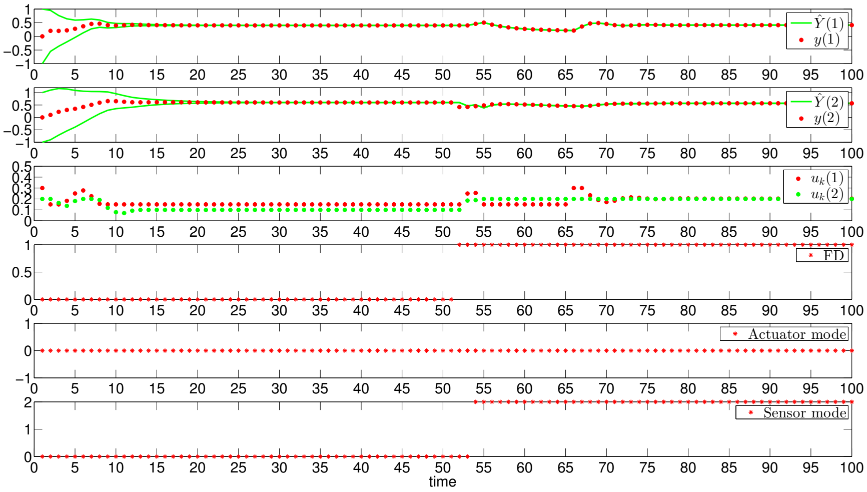

Figure 8: Scenario 4

Remark 5.1 *From the setpoint pairs of states and inputs above, it can be seen that all the state and input setpoints can satisfy their constraints except $u^*_{0,2}$. This means that a degree of performance has to be sacrificed when the fault G_2 occurs. However, for the rest of faulty statuses, the expected performance can be achieved.*

Moreover, in order to differentiate the four considered faulty statuses, an input set sequence should be carefully designed. In this example, according to the proposed method, it is verified that at most ten steps of input sets are needed to guarantee to separate all estimated output sets of the corresponding faulty statuses, which are shown in Figure 4 and the input set sequence is presented as

$$U^{f,0,0}_{k_d+1} = \left\{u : \begin{bmatrix}0.250\\0.150\end{bmatrix} \leq u \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},\ U^{f,0,0}_{k_d+2} = \left\{x : \begin{bmatrix}0.255\\0.155\end{bmatrix} \leq x \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},$$
$$U^{f,0,0}_{k_d+3} = \left\{u : \begin{bmatrix}0.260\\0.160\end{bmatrix} \leq u \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},\ U^{f,0,0}_{k_d+4} = \left\{x : \begin{bmatrix}0.265\\0.165\end{bmatrix} \leq x \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},$$
$$U^{f,0,0}_{k_d+5} = \left\{u : \begin{bmatrix}0.270\\0.170\end{bmatrix} \leq u \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},\ U^{f,0,0}_{k_d+6} = \left\{x : \begin{bmatrix}0.275\\0.175\end{bmatrix} \leq x \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},$$
$$U^{f,0,0}_{k_d+7} = \left\{u : \begin{bmatrix}0.290\\0.190\end{bmatrix} \leq u \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},\ U^{f,0,0}_{k_d+8} = \left\{x : \begin{bmatrix}0.290\\0.190\end{bmatrix} \leq x \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},$$
$$U^{f,0,0}_{k_d+9} = \left\{u : \begin{bmatrix}0.295\\0.195\end{bmatrix} \leq u \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\},U^{f,0,0}_{k_d+10} = \left\{u : \begin{bmatrix}0.295\\0.195\end{bmatrix} \leq u \leq \begin{bmatrix}0.3\\0.2\end{bmatrix}\right\}.$$

With the input set sequence above, all the corresponding estimated output sets from $k = k_d + 1$ to $k = k_d + 10$ can be plotted. However, due to the space limit, only the figures at time instants $k = k_d + 1$, $k = k_d + 4$, $k = k_d + 7$ and $k = k_d + 10$ are shown in Figure 4. It can be seen that the four faulty statuses can be distinguished in at most ten steps after FD with the designed input sequence. However, this does not mean that ten steps are always needed to complete the FI task. In particular, for this case study, ten steps only represent the maximal number of steps guaranteeing isolation of the four statuses, which is verified by the proposed method by off-line computation. However, in real time, it is possible to isolate faults during a period shorter than 10 steps.

For Scenario 1, the FDI and FTC results are shown in Figure 5, where a fault in the first actuator is detected at time instant $k_d = 53$ and then the fault is isolated at time instant $k = 57$. Figure 5 shows that the first actuator fault can be well tolerated with the framework of the proposed FTC scheme.

For Scenario 2, a fault is detected at time instant $k_d = 53$ and isolated at time instant $k_i = 63$, see Figure 6. It shows that in Scenario 2, the second actuator becomes faulty. From the first two plots of Figure 6, it can be seen that the second actuator fault can also be well tolerated and the expected outputs can be well kept after the fault.

For Scenario 3, a fault is detected at time instant $k_d = 52$ and isolated at time instant $k_i = 61$, see Figure 7. It shows that in Scenario 3, the first sensor becomes faulty. The first two plots of Figure 6 can show that the first sensor fault can also be well tolerated and the expected outputs can be well reached after fault tolerance as well.

For Scenario 4, a fault is detected at time instant $k_d = 52$ and isolated at time instant $k_i = 54$, which are presented in Figure 7. It is shown that, in Scenario 4, the second sensor becomes faulty. Different from Scenarios 1, 2 and 3, the first two plots of Figure 7 shows that a degree of performance has to be sacrificed since the expected input setpoint $u^*_{0,2}$ does not satisfy the input constraint. However, from the figure, it can be seen that, in spite of degradation, the output performance is still acceptable.

According to the results from the given scenarios, it can be concluded that the proposed scheme can effectively detect, isolate and tolerate faults in actuators and sensors, which can sufficiently show the effectiveness of this proposed FTMPC approach.

6 Conclusions

This paper has proposed an integrated actuator and sensor active fault-tolerant model predictive control (FTMPC) scheme. In this scheme, a set-valued observer-based method is chosen as the robust FDI strategy in the proposed scheme. One important advantage of this FTMPC scheme is that it has integrated the advantages of actuator and sensor fault diagnosis and fault-tolerant control, robustness, efficient/fact fault isolation (FI) and system constraint-handling ability into one scheme. Moreover, a novel manner to implement active FI is also proposed, whose objective is to reduce FI conservatism as much as possible by making full use of the transient-state sets after faults. So far, input sets for active FI were designed based on the ultimate sets (i.e., RPI sets), which could obtain FI guarantees but it did not fully use the transient-state information to reduce FI conservatism. As a result, obtained results were more conservative when compared with other active FI methods such as computing separating inputs on-line. However, on-line design methods of separating inputs are not able to obtain FI guarantees in advance, showing also higher computational complexity, which were disadvantages over the off-line methods computing inputs (or input sets). The feature of the proposed active FI method is able to obtain a balance of the on-line and off-line methods, which consists in a trade-off of computational complexity and FI conservatism, since the proposed active FI method constructs off-line input set sequences with considering the transition and uses them on-line for FI. As a future research, a methodology for systematically design the input for active FI will be developed considering the preliminary results presented in [14]. Moreover, the design of a bank of min-max MPC controllers to preserve stability will be investigated following the ideas suggested in [26].

A Appendix

Definition A.1 *A set $\mathcal{X} \subseteq X$ is an RCI set of the dynamics $x_{k+1} = Ax_k + Bu_k + \omega_k$ if for any $x_k \in \mathcal{X}$, there always exists $u_k \in U$ for all $\omega_k \in W$ such that $x_{k+1} \in \mathcal{X}$ holds for all $k \geq 0$ [4].*

Definition A.2 *A set $\mathcal{X}_{\mathcal{M}} \subseteq X$ is said to be the MRCI set of the dynamics $x_{k+1} = Ax_k + Bu_k + \omega_k$, if it is RCI and contains all RCI sets inside X [4].*

Definition A.3 *A set $\mathcal{X}$ is an RPI set of the dynamics $x_{k+1} = Ax_k + E\omega_k$ if for $x_k \in \mathcal{X}$ and $\omega_k \in W$, $x_{k+1} \in A\mathcal{X} + EW \subseteq \lambda\mathcal{X}$ $(0 < \lambda \leq 1)$ always holds, where $0 < \lambda < 1$ implies that $\mathcal{X}$ is also attractive to the state trajectory [7, 11].*

Definition A.4 *The mRPI set of the dynamics $x_{k+1} = Ax_k + E\omega_k$ is an RPI set contained in any closed RPI set and the mRPI set is unique and compact [7, 11].*

Definition A.5 *An r-order zonotope Z is defined as $Z = g \oplus H\mathbb{B}^r$, where g and H are the center and segment (or generator) matrix of this zonotope, $\mathbb{B}^r$ is a box composed of r unitary intervals and the symbol $\oplus$ denotes the Minkowski sum [1].*

Property A.1 *Given $X_1 = g_1 \oplus H_1\mathbb{B}^{r_1} \subset \mathbb{R}^n$ and $X_2 = g_2 \oplus H_2\mathbb{B}^{r_2} \subset \mathbb{R}^n$, their Minkowski sum is $X_1 \oplus X_2 = \{g_1 + g_2\} \oplus [H_1 \quad H_2]\mathbb{B}^{r_1+r_2}$.*

Property A.2 *Given $X = g \oplus H\mathbb{B}^r \subset \mathbb{R}^n$ and a suitable matrix K, $KX = Kg \oplus KH\mathbb{B}^r$.*

Property A.3 *Given a zonotope $X = g \oplus H\mathbb{B}^r \subset \mathbb{R}^n$ and an integer s (with $n < s < r$), denote by $\hat{H}$ the matrix resulting from the reordering of the columns of the matrix H in decreasing Euclidean norm. $X \subseteq g \oplus [\hat{H}_T \quad Q]\mathbb{B}^s$ where $\hat{H}_T$ is obtained from the first $s - n$ columns of matrix $\hat{H}$ and $Q \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose elements are $Q_{ii} = \sum_{j=s-n+1}^{r} \mid \hat{H}_{ij} \mid$, $i = 1, \ldots, n$[1]).*

Property A.4 *Given a matrix $\Lambda \in \mathbb{R}^{n\times m}$, a zonotope $Z = g \oplus H\mathbb{B}^r$, and an H-polytope $P = \{x \in \mathbb{R}^n : |Cx - d| \leq [\phi_1, \phi_2, ..., \phi_m]^T\}$, with $C \in \mathbb{R}^{m\times n}$, $d \in \mathbb{R}^m$, $\phi_i \in \mathbb{R}_+$ ($i = 1, 2, ..., m$), define a vector $\hat{g}(\Lambda) = g + \Lambda(d - Cg)$ and a matrix $\hat{H}(\Lambda) = [(I - \Lambda C)H \quad \Lambda\Phi]$, with a diagonal matrix $\Phi = diag(\phi_1, \phi_2, ..., \phi_m)$. Then a family of zonotopes (parameterized by the matrix Λ) that contains the intersection of the zonotope Z and the polytope P is obtained as $Z \cap P \subseteq \hat{Z}(\Lambda) = \hat{g} \oplus \hat{H}\mathbb{B}^{r+m}$ [9].*

Acknowledgements

This work has been partially funded by the Spanish Government (MINECO) through the project CICYT ECOCIS (ref. DPI2013-48243-C2-1-R), by MINECO and FEDER through the project CICYT HARCRICS (ref. DPI2014- 58104-R). The authors also express sincere gratefulness to Prof. Sorin Olaru from the Automatic Control Department, Supeléc, France for the interesting discussions.

References

[1] T. Alamo, J.M. Bravo, and E.F. Camacho. Guaranteed state estimation by zonotopes. *Automatica*, 41(6):1035–1043, 2005.

[2] M. Blanke, M. Staroswiecki, and N. E. Wu. Concepts and methods in fault-tolerant control. In *Proceedings of the 2001 American Control Conference*, Virginia, USA, June 2001.

[3] J. Blesa, V. Puig, J. Romera, and J. Saludes. Fault diagnosis of wind turbines using a set-membership approach. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 28 August - 2 September 2011.

[4] F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*. Model Predictive Control Lab, UC-Berkeley, USA, 2014.

[5] G. Franzè, F. Tedesco, and D. Famularo. Actuator fault tolerant control: a set-theoretic approach. In *In proceedings of the 51st IEEE Conference on Decesion and Control*, December 2012.

[6] D.A. Joosten, T.J.J. van den Boom, and T.J.J. Lombaerts. Fault-tolerant control using dynamic inversion and model-predictive control applied to an aerospace benchmark. In *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea, July 2008.

[7] E. Kofman, H. Haimovich, and M.M. Seron. A systematic method to obtain ultimate bounds for perturbed systems. *International Journal of Control*, 80(2):167–178, 2007.

[8] J. Löfberg. *Min-max Approaches to Robust Model Predictive Control*. PhD thesis, Department of Electrical Engineering, Linköping University, Sweden, 2003.

[9] V.T.H. Le, C.N. Stoica, T. Alamo, E.F. Camacho, and D. Dumur. Zonotope-based set-membership estimation for multi-output uncertain systems. In *Proceedings of 2013 IEEE international Symposium on Intelligent Control (ISIC), Part of 2013 IEEE Multi-Conference on Systems and Control*, Hyderabad, India, August 2013.

[10] G. Battistelli M. Baglietto and L. Scardovi. Active mode observation of switching systems based on set-valued estimation of the continuous state. *International Journal of Robust and Nonlinear Control*, 19:1521C1540, 2009.

[11] S. Olaru, J.A. De Doná, M.M. Seron, and F. Stoican. Positive invariant sets for fault tolerant multisensor control schemes. *International Journal of Control*, 83(12):2622–2640, 2010.

[12] E.N. Osella, H. Haimovich, and M.M Seron. Integration of invariant-set-based FDI with varying sampling rate virtual actuator and controller. *International Journal of Adaptive Control and Signal Processing.*

[13] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

[14] D.M. Raimondo, G. Roberto Marseglia, R.D. Braatz, and J.K. Scott. Fault-tolerant model predictive control with active fault isolation. In *Proceedings of 2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, Nice, France, October 9-11 2013.

[15] P. Rosa, C. Silvestre, J.S. Shamma, and M. Athans. Fault detection and isolation of LTV systems using set-valued observers. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Hilton Atlanta Hotel, Atlanta, GA, USA, December 15-17 2010.

[16] Joseph K. Scott, Rolf Findeisen, Richard D. Braatz, and Davide M. Raimondo. Input design for guaranteed fault diagnosis using zonotopes. *Automatica*, 50(6):1580 – 1589, 2014.

[17] M.M. Seron and J.A. De Doná. Actuator fault tolerant multi-controller scheme using set separation based diagnosis. *International Journal of Control*, 83(11):2328–2339, 2010.

[18] M.M. Seron, J.A. De Doná, and J.H. Richter. Integrated sensor and actuator fault-tolerant control. *International Journal of Control*, 86(4):689–708, 2013.

[19] T. Steffen. *Control Reconfiguration of Dynamical Systems.* Springer, Germany, 2005.

[20] S. Sun, L. Dong, L. Li, and S. Gu. Fault-tolerant control for constrained linear systems based on MPC and FDI. *International Journal of Information and Systems Sciences*, 4(4):512 –23, 2008.

[21] F. Xu, S. Olaru, V. Puig, C. Ocampo-Martinez, and S. Niculescu. Sensor-fault tolerance using robust MPC with set-based state estimation and active fault isolation. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, Los Angeles, CA, USA, December 15-17 2014.

[22] F. Xu, V. Puig, C. Ocampo-Martinez, S. Olaru, and F. Stoican. Set-theoretic methods in robust detection and isolation of sensor faults. *International Journal of Systems Science*, pages 1–18, 2014.

[23] F. Xu, V. Puig, C. Ocampo-Martinez, F. Stoican, and S. Olaru. Actuator-fault detection and isolation based on set-theoretic approaches. *Journal of Process Control*, 24(6):947 – 956, 2014.

[24] X. Yang and J.M. Maciejowski. Fault-tolerant model predictive control of a wind turbine benchmark. In *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Mexico City, Mexico, August 2012.

[25] A. Yetendje, M. M. Seron, and J. A. De Doná. Robust MPC design for fault tolerance of constrained multisensor linear systems. In *Proceedings of the 2010 International Conference on Control and Fault-Tolerant Systems*, Nice, France, October 6 - 8 2010.

[26] Lixian Zhang, Songlin Zhuang, and Richard D. Braatz. Switched model predictive control of switched linear systems: Feasibility, stability and robustness. *Automatica*, 67:8 – 21, 2016.

[27] Y.M. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229 – 252, 2008.