Proceedings of the 36th Chinese Control Conference
July 26-28, 2017, Dalian, China

Fault-tolerant control of a 2-DOF robot manipulator using multi-sensor switching strategy

Yilin Mi, Feng Xu, Junbo Tan, Xueqian Wang* and Bin Liang

Shenzhen Engineering Laboratory of Geometry Measurement Technology, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, P.R. China
E-mail: wang.xq@sz.tsinghua.edu.cn
miyilin@126.com

Abstract: In this paper, a fault-tolerant multi-sensor switching strategy is used for sensor-fault tolerance of robot manipulator with redundant sensors in each joint to provide the angular position of the joint. This strategy employs a linear-parameter-varying (LPV) model to describe the nonlinear dynamics of the robot manipulator. Then, based on the obtained LPV model, a zonotopic set-based robust fault detection (FD) mechanism is further designed to monitor the system behaviors. At each time instant, the FD mechanism checks whether there are some sensors that have become faulty to make FD decisions. A switching controller is designed to choose the best estimation among the healthy sensors to generate closed-loop control laws. Once a sensor has become faulty and is detected by the FD mechanism, the fault-tolerant strategy discards the faulty sensor and uses the healthy one to implement closed-loop control. At the end, some simulation results are presented to illustrate the effectiveness of the proposed fault-tolerant control (FTC) strategy.

Key Words: robot manipulator; fault-tolerant control; multi-sensor switching scheme; bounded sets; zonotopes

1 Introduction

The robot manipulator has been widely used in many operating scenarios such as space mission, rescue operation and medical services. The reliability and safety of the manipulator system has gained significant attention in recent years. System faults can lead to unsafe operations which may result in mission abortion or even unpredictable damage to the equipment, the surroundings or human. Thus, it is paramount to detect and identify potential abnormalities and faults as early as possible and implement fault tolerance for minimizing performance degradation and avoiding dangerous situations [1].

In general, there are a collection of sensors equipped in single joint of a manipulator, e.g., the Hall sensor of brushless motor, the encoder (usually at the end of the motor) and angular sensor at the output port of the joint. The redundancy of the sensors makes fault-tolerant control (FTC) possible when a fault occurs in sensors. Besides the redundancy of the sensors, FTC strategy for sensors is also needed.

Various methods have been investigated in Fault Detection, Isolation and Accommodation (FDIA) for robot manipulator [2-9]. Some works ([2-4]) studied the Fault Detection and Isolation (FDI) of both actuator and sensor faults while others ([5-8]) focused only on sensor faults. However, most of these works only dealt with the FDI problem without discussing the control problem under the effect of faults.

In [10], a multi-sensor switching control scheme was proposed and later applied to automotive vehicle longitudinal control, motor control [11], magnetic levitation system [12] and other systems [13]. This scheme was first designed for linear system and implemented on some nonlinear system using linearizing method such as linearizing at equilibrium point [12]. Recently in [14], María M. Seron et.al presented a method that could extend this scheme to LPV system. The concept of invariant set has been used to prove the close-loop stability of the switching control method. More importantly, the use of invariant sets reduces the computational load at runtime, which makes this method practical to apply to real systems. However, it is difficult for the manipulator to compute the invariant sets satisfying the set-separation condition, because the reference related sets are usually not centered away from zero and so does the "fault noise" sets, in which case a preliminary fault detection mechanisms in the sensors should be used[12].

Based on these facts, we consider the situation that each joint of the manipulator has a brushless motor with hall sensor and an encoder. Under this situation, we propose to use a multi-sensor switching control scheme for FTC of robot manipulator. For different combinations of the sensors of the joints, a state estimator based on an LPV model of the manipulator is designed along with a set-based observer, where the model uncertainties, modeling errors, process disturbances and measurement noises are considered bounded based on zonotopes. Using zonotopes to represent bounding sets can simplify the propagation of set-based dynamics [21] and make online computing of the bounding set possible. Then a residual set can be computed, which should include the zero vector. At each time instant, the FD decisions are made using the set-based criterion. Thus, we provide an online computing method using zonotopes to implement fault detection and combine with the switching control to realize FTC. The main contribution is that we provide a practical FTC framework for the robot manipulator to well employ its existing sensor redundancy. To the best of our knowledge, no other works have implement this method on robot manipulator. Simulations on the model of CRS

*This work is supported by the Natural Science Foundation of Guangdong Province (2014A030310318, 2015A030313881) and the Basic Research Program of Shenzhen (JCYJ20140509172959962, JCYJ-20160301153317415, JCYJ20160301100921349).

A465 robot manipulator show the effectiveness of the method.

This paper is organized as follows. Section II presents the system description and the LPV model of the manipulator. Section III introduces the multi-sensor switching scheme for FTC and the set-valued observer for FD. In Section IV, the simulation is conducted to show the effectiveness of the scheme. The last section gives the conclusion on the proposed sensor FTC strategy.

2 Description and LPV Modeling of A 2-DOF Robot Manipulator

2.1 Description

The model of CRS A465 robot manipulator shown in Fig. 1 has been well studied in [16]. Considering the available parameters of the model, we use this model for our research. Although the manipulator has six DOFs, for simplicity we only consider the second and third joints of this robot, referred to as joint 1 and joint 2 respectively, and other links are fixed during simulation without loss of generality.

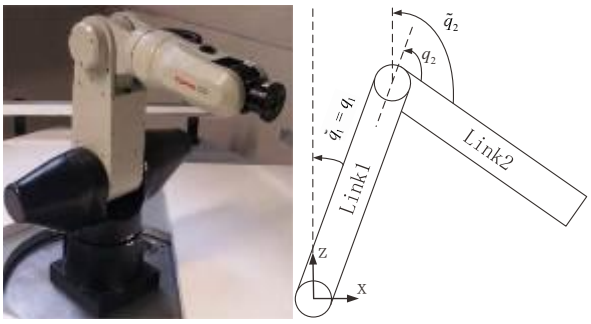

Fig. 1: The 2-DOF robot manipulator and its side view sketch

The rigid-body dynamic model of the 2-DOF manipulator is described by using the Euler-Lagrange formulation as

$$\boldsymbol{\tau}(t) = \mathbf{D}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{c}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}(t)), \quad (1)$$

where $\boldsymbol{\tau} \in \mathbb{R}^2$ is the vector of joint torques (control inputs), and $\mathbf{q} = [q_1,\ q_2] \in \mathbb{R}^2$ is the configuration coordinates, $\mathbf{D}$ is the inertia matrix and $\mathbf{c}$, $\mathbf{g}$ and $\boldsymbol{\tau}_f$ denotes the Coriolis-centrifugal forces, gravity and joint friction torques respectively.

Based on the results in [16], an affine parameter dependent LPV model can be developed by introducing new joint position variables $\tilde{q}_i$ with a fixed coordinate reference as follows:

$$\tilde{q}_1 = q_1, \tilde{q}_2 = q_1 + q_2. \quad (2)$$

Using (2), the model of the manipulator (1) can be rewritten as

$$\boldsymbol{\tau}(t) = \tilde{\mathbf{D}}(\tilde{\mathbf{q}}(t))\ddot{\tilde{\mathbf{q}}}(t) + \tilde{\mathbf{c}}(\tilde{\mathbf{q}}(t), \dot{\tilde{\mathbf{q}}}(t)) + \tilde{\mathbf{g}}(\tilde{\mathbf{q}}(t)) + \tilde{\boldsymbol{\tau}}_f(\dot{\tilde{\mathbf{q}}}(t)), \quad (3)$$

$$\tilde{\mathbf{D}} = \begin{bmatrix} b_1 + b_3\cos(\tilde{q}_2 - \tilde{q}_1) & b_2 + b_3\cos(\tilde{q}_2 - \tilde{q}_1) \\ b_3\cos(\tilde{q}_2 - \tilde{q}_1) - b_8 & b_7 \end{bmatrix},$$

$$\tilde{\mathbf{c}} = \begin{bmatrix} b_3\dot{\tilde{q}}_1\sin(\tilde{q}_2 - \tilde{q}_1) - b_3\dot{\tilde{q}}_2\sin(\tilde{q}_2 - \tilde{q}_1) \\ b_3\dot{\tilde{q}}_1\sin(\tilde{q}_2 - \tilde{q}_1) \end{bmatrix}, \quad (4)$$

$$\tilde{\mathbf{g}} = \begin{bmatrix} -b_4\sin(\tilde{q}_2) - b_5\sin(\tilde{q}_1) \\ -b_4\sin(\tilde{q}_2) \end{bmatrix}, \quad \tilde{\boldsymbol{\tau}}_f = \begin{bmatrix} b_6\dot{\tilde{q}}_1 \\ b_9(\dot{\tilde{q}}_2 - \dot{\tilde{q}}_1) \end{bmatrix},$$

where the coefficients $b_l (l = 1,...,9)$ are linear combinations of dynamic and kinematic parameters given specifically in the Appendix A in [16].

2.2 Quasi-LPV model of robot manipulator

Consider a continuous-time state-space LPV model

$$\begin{aligned} \dot{x} &= A_{ss}(\theta)x + B_{ss}(\theta)u, \\ y &= Cx + Du, \end{aligned} \quad (5)$$

where θ is the external scheduling parameter vector, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the input vector, $y \in \mathbb{R}^{n_y}$ is the output vector, $A_{ss}(\theta)$ is the system matrix and $B_{ss}(\theta)$ is the input matrix, both depending on the time-varying parameter vector $\theta = [\theta^{(1)} \quad \theta^{(2)} \ldots \theta^{(L)}] \in \mathbb{R}^L$, C is the measurement matrix and D is the feedforward matrix.

To obtain the LPV description, a mapping $f : \mathbb{R}^n \to \mathbb{R}^L$ such that $\theta = f(x)$ should be constructed. We select $\mathbf{x} = [\tilde{q}_1 \quad \tilde{q}_2 \quad \dot{\tilde{q}}_1 \quad \dot{\tilde{q}}_2]^T$ as the state vector and $u = [\tau_1 \quad \tau_2]^T$ as the input vector. Then $A_{ss}(\theta)$ and $B_{ss}(\theta)$ can be obtained from (3):

$$A_{ss}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ h_1(\boldsymbol{\theta}) & h_2(\boldsymbol{\theta}) & h_3(\boldsymbol{\theta}) & h_4(\boldsymbol{\theta}) \\ h_5(\boldsymbol{\theta}) & h_6(\boldsymbol{\theta}) & h_7(\boldsymbol{\theta}) & h_8(\boldsymbol{\theta}) \end{bmatrix}, \quad (6)$$

$$B_{ss}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ h_9(\boldsymbol{\theta}) & h_{10}(\boldsymbol{\theta}) \\ h_{11}(\boldsymbol{\theta}) & h_{12}(\boldsymbol{\theta}) \end{bmatrix}, \quad (7)$$

where $h_i(\boldsymbol{\theta})$ is affine with respect to $\boldsymbol{\theta}$ while $\boldsymbol{\theta}$ is affine with respect to $\mathbf{x}$, both given in the Appendices B and C in [16].

As the observer and the switching controller are designed in discrete-time form, the quasi-LPV model should be discretized first. Thus, we define $x_k \triangleq x(kT)$ as the state value sampled at time kT, where T is the sampling period. Then we can obtain the following discretized form of quasi-LPV model (5) as

$$x_{k+1} = A_d(\theta_k)x_k + B_d(\theta_k)u_k, \quad (8)$$

where $A_d(\theta_k)$ and $B_d(\theta_k)$ are obtained from (6) and (7) using ZOH polynomial discretization approach [17] and $\theta_k = [\theta_k^{(1)} \quad \theta_k^{(2)} \ldots \theta_k^{(L)}] \in \mathbb{R}^L$. For simplicity, we denote by x^+ the successor of the current state x in the following context and use the superscript to represent the scheduling vector, that is A_d^{θ} for $A_d(\theta)$. Hence, the model (8) can be rewritten as

$$x^+ = A_d^{\theta}x + B_d^{\theta}u. \quad (9)$$

In order to improve the accuracy of the LPV model, the effect of process disturbances, modeling and discretizing errors, and parametric uncertainties are further considered and described as an unknown but bounded vector ω, leading to a modified linear discrete-time LPV model:

$$x^+ = A_d^{\theta}x + B_d^{\theta}u + E\omega, \quad (10)$$

where $A_d{}^{\theta}$ and $B_d{}^{\theta}$ are as in (9) and E is a parametric matrix.

3 Multi-sensor Switching Scheme

For robot manipulators, brushless motors with hall sensors are often used as the actuator and encoders are used as the measurements of the joint angle. Although the accuracy of measurements of the joint angle using the hall sensors is low, these hall sensors provide hardware redundancy for the robot manipulator to perform FTC. Accordingly, we propose a multi-sensor control scheme for the robot manipulator depicted in Fig. 2. The other elements in the figure will be explained in detail in this section.

A typical control problem of robot manipulator is to design a closed-loop control law such that the manipulator tracks a predefined or online computed trajectory which is denoted by x_{ref} in Fig. 2. Note that x_{ref} satisfies a given dynamics:

$$x_{ref}^{+} = A_d{}^{\theta} x_{ref} + B_d{}^{\theta} u_{ref}, \tag{11}$$

where u_{ref} is reference control input.

3.1 Sensor Measurements and Estimators

The angular position of the *j*-th joint can be measured by two sensors: a hall sensor, $S_{j,H}$, $j \in \{1,2\}$ embedded in the brushless motor, which measures the angular position of the motor shaft in a low accuracy and an encoder, $S_{j,E}$, which provides the angular position of the motor shaft in a high accuracy. Notice that the measurements provided by the hall sensor of brushless motor are often used by motor driver for control purpose, so if faults occur in the hall sensor $S_{j,H}$, the motor driver cannot control the motor. However, if fault occur in sensor $S_{j,E}$, the angular position of the *j*-th joint can still be measured as long as sensor $S_{j,H}$ remains healthy and the system is observable. So for this two-DOF manipulator, we can use the measurements provided by $S_{1,H}$, $S_{1,E}$, $S_{2,H}$ and $S_{2,E}$ combined into four sensor-pairs: $\begin{bmatrix} S_{1,H} & S_{2,H} \end{bmatrix}$, $\begin{bmatrix} S_{1,E} & S_{2,H} \end{bmatrix}$, $\begin{bmatrix} S_{1,H} & S_{2,E} \end{bmatrix}$ and $\begin{bmatrix} S_{1,E} & S_{2,E} \end{bmatrix}$.

In this paper, we assume that a fault occurs in the encoder in the 1-st joint of the manipulator at a given time. Under this specific setting, only $y_1 = \begin{bmatrix} S_{1,H} & S_{2,E} \end{bmatrix}^T$ and $y_2 = \begin{bmatrix} S_{1,E} & S_{2,E} \end{bmatrix}^T$ are considered as the system output.

Usually, there are two types of encoders used by robot manipulator: absolute encoder and incremental encoder. The incremental encoder is more widely used because it is cheaper and easier to use. However, incremental encoders indicate the position by the number of pulses, requiring the software counter to record the position by accumulating the number of pulses. So if faults occur in incremental encoders, the counter will remain at the position value recorded at the time instant of fault occurrence. In this paper we use incremental encoder and only total outage fault is considered. Under healthy condition, the output equations of the manipulator are:

$$y_i = C_i x + \eta_i, \quad i \in \{1, 2\}, \tag{12}$$

where $C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix}$, $\eta_i = \begin{bmatrix} \eta_{i,\alpha} & \eta_{i,\beta} \end{bmatrix}^T$ and $|\eta_i| \le \bar{\eta}_i$ is a bounded measurement noise with $\bar{\eta}_i$ being a vector with non-negative entries. According to the fault scenario considered before, if faults occur in the encoder in the 1-st joint of the manipulator, the output equation of y_2 will turn into:

$$y_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t_f) + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} x + \eta_1, \tag{13}$$

where $x(t_f)$ is the state at time instant t_f.

For each sensor-pair output y_i, $i \in \{1, 2\}$, we design an estimator E_i to estimate the state of the manipulator. The estimators are designed as

$$\begin{aligned} \hat{x}_i^{+} &= A_d{}^{\theta} \hat{x}_i + B_d{}^{\theta} u + L_i{}^{\theta} (y_i - \hat{y}_i), \\ \hat{y}_i &= C_i \hat{x}_i, \end{aligned} \tag{14}$$

where $\hat{x}_i$ is the *i*-th estimate of the state *x* of the manipulator. The estimator gain $L_i{}^{\theta}$ is designed such that $\hat{x}_i$ is asymptotically converge to *x*. This can be achieved by first using PCA based method to reduce the vertices of the LPV model[16] and then using gain-scheduled techniques[18] to find proper $L_i{}^{\theta}$. Since the actual state of the system cannot be measured, we define the error of state estimation as follow:

$$\hat{\xi}_i \triangleq \hat{x}_i - x_{ref}. \tag{15}$$

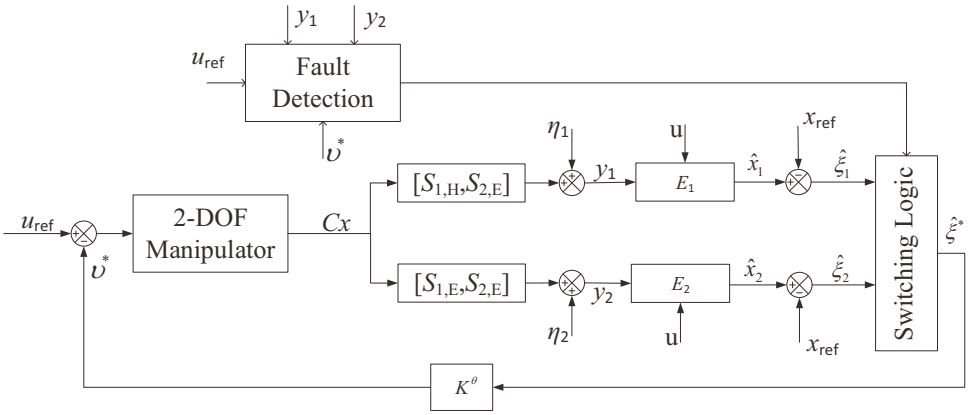

Fig. 2: Multi-sensor control scheme for a 2-DOF robot manipulator

3.2 Set-valued Observer for Fault Detection

The set-valued observer we apply in the following aims at computing the bounding set of the estimated error. Because of the dynamics of the system and bounded disturbances, the estimated error will stay in the bound set on healthy condition. If the estimated error is inside the bounding set, we then assume that there is no fault occurrence in the system. Otherwise, if the estimated error is out of the bounding set, we can assure that faults occurs and fault detection is realized.

There are several ways to represent sets and bounds. In this paper, zonotope is used to describe sets and bounds because some of its features can simplify the propagation of set-based dynamics and improve the computational speed compared with using polytope or other geometry forms. Some definitions and properties of zonotopes related to this thesis is listed as follows [21].

Definition 1. An *m*-order zonotope Z is defined as $Z = g \oplus H\mathbb{B}^m$, where g and H are called center and segment matrix, respectively.

Property 1. Given two zonotopes $Z_1 = g_1 \oplus H_1\mathbb{B}^{m1}$ and $Z_2 = g_2 \oplus H_2\mathbb{B}^{m2}$, $Z_1 \oplus Z_2 = (g_1 + g_2) \oplus [H_1\ H_2]\mathbb{B}^{m1+m2}$.

Property 2. Given a zonotope $Z = g \oplus H\mathbb{B}^m \subset \mathbb{R}^n$ and a compatible matrix *K*, $KZ = Kg \oplus KH\mathbb{B}^m$.

Definition 2. The interval hull $\square Z$ of a zonotope $Z = g \oplus H\mathbb{B}^m \subset \mathbb{R}^n$ is the smallest box containing *Z*, i.e.,

$$\square Z = \{x : |x_i - g_i| \le \|H_i\|_1\},$$

where H_i is the *i*-th row of *H*, x_i and g_i are the *i*-th components of *x* and *g*, respectively.

The complexity of zonotopes is described by their order. The higher the order is, the more complex a zonotope is. Using complex zonotopes can lead to computational problems. Thus, it is needed to approximate high order zonotopes with lower-order zonotopes. Property 3 presents a method to control the order of zonotopes during online propagation, which is used to implement set computation.

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

In the following part, we use capital letters to represent the zonotopes, i.e. *X* is the zonotope for *x*, Ω for ω and *V* for η , etc. Recall the discrete-time model of the manipulator described in (10) and the output function (12), the dynamics of the manipulator is as follows:

$$\begin{aligned} x^+ &= A_d{}^\theta x + B_d{}^\theta u + E\omega, \\ y_i &= C_i x + \eta_i, \end{aligned} \tag{16}$$

where ω and η_i are bounded signals. We define their bounding sets as $\omega \in \Omega$ and $\eta_i \in V_i$. We also assume an initial state estimation set satisfying $\hat{x}_i(0) \in \hat{X}_i(0)$. Similar as (14), we can construct an observer that takes model uncertainty and noise into account as

$$\begin{aligned} \hat{x}_i^+ &= A_d{}^\theta \hat{x}_i + B_d{}^\theta u + L_i{}^\theta (y_i - \hat{y}_i) + E\omega, \\ \hat{y}_i &= C_i \hat{x}_i + \eta_i. \end{aligned} \tag{17}$$

Based on this observer and transforming it into set-based form, we can derive the set-valued observer in the following iterative form [15]:

$$\begin{aligned} \hat{X}_i^+ &= (A_d^\theta - L_i^\theta C_i)\hat{X}_i \oplus \{Bu\} \oplus \{L_i^\theta y_i\} \oplus (-L_i^\theta V_i) \oplus E\Omega, \\ \hat{Y}_i &= C\hat{X}_i \oplus V_i. \end{aligned} \tag{18}$$

Under the initial condition $\hat{x}_i(0) \in \hat{X}_i(0)$, $y_i \in \hat{Y}_i$ can be assured at each time instant as long as the system operates in the healthy situation. Further, we define residual sets R_i as

$$R_i = \{y_i\} \oplus (-\hat{Y}_i). \tag{19}$$

Then $0 \in R_i$ can be assured as long as the system operates in the healthy situation. Thus, during system operation, the FD decisions are made by testing whether $0 \in \square R_i$ holds online at each time instant. If $0 \in \square R_i$ holds for all *i*, the system is considered to be still healthy. Otherwise, it is guaranteed that faults occur in the sensors corresponding to the *i*-th output.

3.3 Switching Controller

The controller we apply to the manipulator will switch among each sensors-estimator pairs and compute the feedback control action based on the error of state estimation obtained from the selected estimator.

If the sensors are all healthy, in order to decide which sensors-estimator pairs should be chosen, the following optimization will be solved at each sample time:

$$\hat{\xi}^* = \arg\min \left\{ \hat{\xi}' P \hat{\xi} : \hat{\xi} \in \left\{ \hat{\xi}_1, \hat{\xi}_2 \right\} \right\}, \tag{20}$$

where *P* is a positive semi-definite matrix, and the control action is computed as

$$u = u_{ref} + \upsilon^*, \tag{21}$$

$$\upsilon^* = -K^\theta \hat{\xi}^*, \tag{22}$$

where K^θ is a properly selected feedback gain to guarantee the system can achieve an expected performance [11]. Similar as the way we find $L_i{}^\theta$, we can first use PCA based method to reduce the vertices of the LPV model[16] and then use gain-scheduled techniques[18] to find proper K^θ .

If the set-valued observer detect faults occurred in the sensors corresponding to the *i*-th output, the controller will only use the healthy sensor-estimator pair to implement control action. Thus, during the system operation stage, the system will select the optimal sensors-observer pair to generate the expected performance and tolerate the effect of faults in sensors. In this way, sensor FTC of the robot manipulator can be performed by making well use of the sensor hardware redundancy in the actual system.

4 Simulation

The FTC strategy is tested using the model of the CRS A465 robot manipulator. The inertial and friction parameters in (4) are shown in table 1 in [16]. According to [22], the ranges of q_1 and q_2 are -1.57 $\leqslant q_1 \leqslant$ 1.57 (*rad*) and -1.92 $\leqslant q_2 \leqslant$ 1.92 (*rad*), respectively. The max speed of joint 1 and joint 2 is 3.14 *rad/s*. These performance specification limits

are considered when generating the joint space trajectories. The max continuous torque that joint 1 can generate is $\bar{\tau}_1 = 66.08\ N{\cdot}m$ and for joint 2 this value is $\bar{\tau}_2 = 39.50\ N{\cdot}m$. The gear ratio from the motor shaft to the output port is 100:1. The resolution of the hall sensor is 6 pulses per revolution of the motor and the resolution of the incremental encoder is 2000 pulses per revolution of the motor. Transforming these resolutions by simple calculation, we can obtain the minimum resolution of output port of the robot joint, that is, the minimum resolution of output port that can be measured by the hall sensor is $2\pi/100/6 = 0.0105\ rad$ and the minimum resolution of output port that can be measured by the incremental encoder is $2\pi/100/2000 = 3.142\times10^{-5}\ rad$. These two minimum resolutions are considered when setting the bound of the measurement noise η_i. The parametric matrix E is set to be identity matrix, i.e. $E = diag[1 \quad 1 \quad 1 \quad 1]$.

The initial conditions in this simulation are given as $x(0) = \hat{x}_i(0) = [0 \quad 0 \quad 0 \quad 0]^T$,

$$V_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0.011 & 0 \\ 0 & 0.000032 \end{bmatrix} \mathbb{B}^2,$$

$$V_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0.000032 & 0 \\ 0 & 0.000032 \end{bmatrix} \mathbb{B}^2,$$

$$\Omega = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0.00004 & 0 & 0 & 0 \\ 0 & 0.00004 & 0 & 0 \\ 0 & 0 & 0.00007 & 0 \\ 0 & 0 & 0 & 0.00007 \end{bmatrix} \mathbb{B}^4,$$

$$\hat{X}_i(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \mathbb{B}^4.$$

The sampling time is $0.001s$. Then compute the scheduled gain L_i^{θ} and K^{θ} by using the approximate LPV model. The weight matrix P is set as

$$P = \begin{bmatrix} 2 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In the first simulation, the total simulation time is set as $20s$. The joint space trajectories of joint 1 and joint 2 are generated by sine function as shown in Fig. 3 in blue solid line. We set a total outage fault to happen at $t_f = 9.999s$. It can be observed in Fig. 4 that $0 \in \square R_1$ holds during the simulation and in Fig. 5, $0 \in \square R_2$ holds before the time instant $t = 10.000s$. However, the interval hull of the bounding set of R_2 does not contain the zero point first at time instant $t = 10.000s$, that is, $0 \notin \square R_2$. This means that the fault is detected at this time. Fig. 6 shows that before $t = 10.000s$, the controller chooses the error of state estimation between $\hat{\xi}_1$ and $\hat{\xi}_2$ according to (20). After $t = 10.000s$, the controller only selects $\hat{\xi}_1$ to perform close-loop control. The trajectory tracking results in joint space are plotted in red dashed line in Fig. 3. Fig. 7 shows the tracking error in joint space. It can be noticed that the tracking error of joint 1 became larger when the controller switch to $\hat{\xi}_1$ after fault is detected. This is because the resolution of the hall sensor is relatively low. The results show that in spite of the fault occurrence in encoder, the robot manipulator can still be controlled using the measurement by the hall sensor.

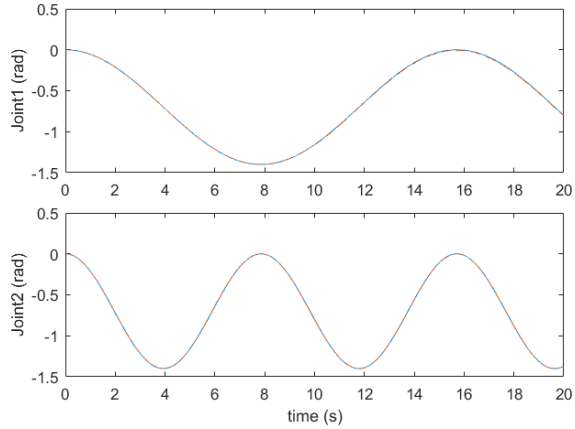

Fig. 3: Reference trajectory in joint space (solid blue line) and tracking results (dashed red line).

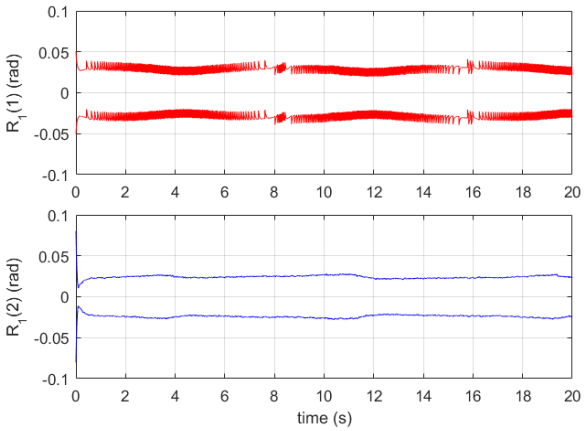

Fig. 4: Residual bounding set of R1.

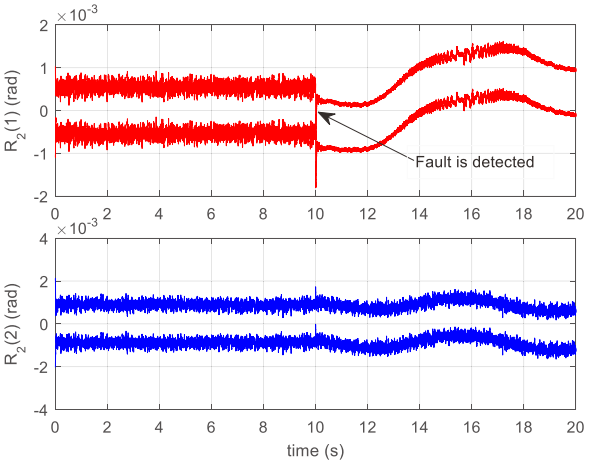

Fig. 5: Residual bounding set of R2.

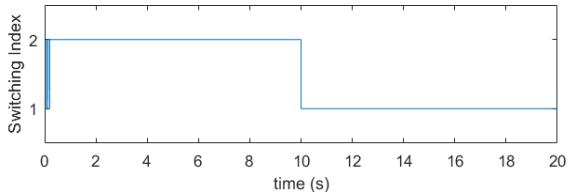

Fig. 6: Switching results of the controller.

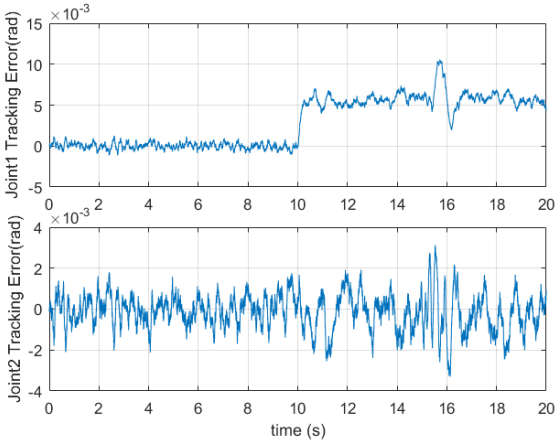

Fig. 7: Tracking error in joint space

In the second simulation, the total simulation time is set as 20s. The joint space trajectories of joint 1 and joint 2 are generated by sine function as shown in Fig. 8 in blue solid line. We set a total outage fault to happen at t_f = 7.855s. It can be observed in Fig. 9 that $0 \in \square R_1$ holds during the simulation and in Fig. 10, $0 \in \square R_2$ holds before the time instant $t = 8.297$s. However, the interval hull of the bounding set of R_2 does not contain the zero point first at time instant t = 8.297s, that is, $0 \notin \square R_2$. This means that the fault is detected at this time. Fig. 11 shows that before $t = 8.297$s, the controller chooses the error of state estimation between $\hat{\xi}_1$ and $\hat{\xi}_2$ according to (20). After t = 8.297s, the controller only selects $\hat{\xi}_1$ to perform close-loop control. The trajectory tracking results in joint space are plotted in red dashed line in Fig. 8. Fig. 12 shows the tracking error in joint space. It can be noticed that the tracking error has an abrupt change from the time instant t_f = 7.855s to t = 8.297. The fault is not detected very fast because the fault happens at a time when the magnitude of the fault is small. However, after the detection of fault, the controller can still control the manipulator back to normal.

In this section, plenty of simulations with different sine wave trajectories computed by different magnitudes and frequencies were conducted. Consequently, the following general conclusions on the current application of the multi-sensor switching FTC strategy to the robot manipulator are obtained.

(1) When the fault happens at the time that the velocity of the corresponding joint is slow and the controller happens to choose the sensor-estimator pair containing no faults, the manipulator tracks the trajectory as normal and the fault will be detected after a short time.

(2) When the fault happens at the time that the velocity of the corresponding joint is fast and the controller happens to choose the sensor-estimator pair containing no faults, the manipulator tracks the trajectory as normal and the fault will be detected very quickly.

(3) When the fault happens at the time that the velocity of the corresponding joint is slow and the controller happens to choose the sensor-estimator pair containing faults, the fault will be detected after a short time and there will be an abrupt change in the tracking error hence an abrupt change in the control torque. But after the fault is detected, the controller can control the manipulator back to normal. This is the situation in the second simulation presented in this paper. To deal with the abrupt change in the tracking error, we can set a limit torque so that if the control torque is beyond this limit torque, the controller will stop the manipulator for safety reasons. As the velocity of the corresponding joint is slow in this situation, stop the manipulator at this time will do no harm.

(4) When the fault happens at the time that the velocity of the corresponding joint is fast and the controller happens to choose the sensor-estimator pair containing faults, the fault is detected very fast and the controller will switch to the healthy sensor-estimator pair and the manipulator tracks the trajectory very well. This is the situation in the first simulation presented in this paper.

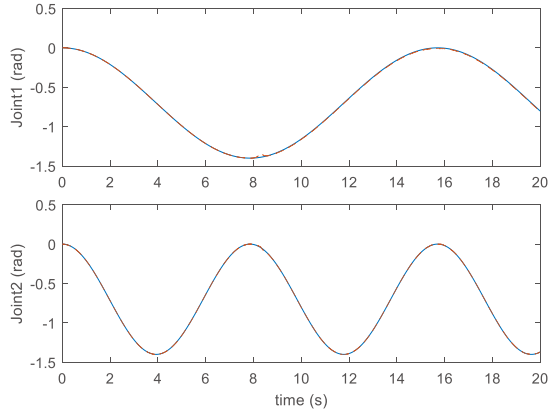

Fig. 8: Reference trajectory in joint space (solid blue line) and tracking results (dashed red line).

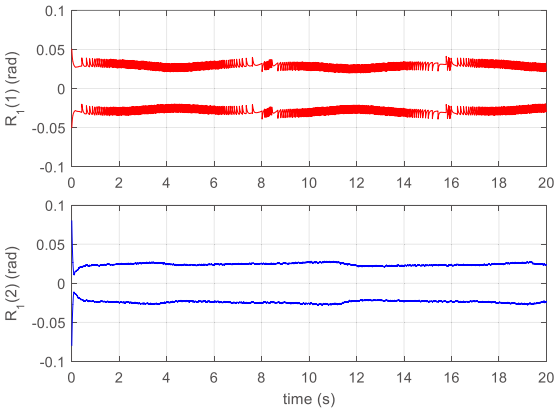

Fig. 9: Residual bounding set of R1.

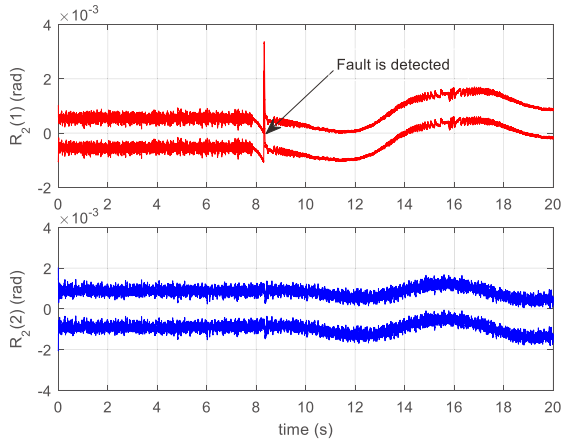

Fig. 10: Residual bounding set of R2.

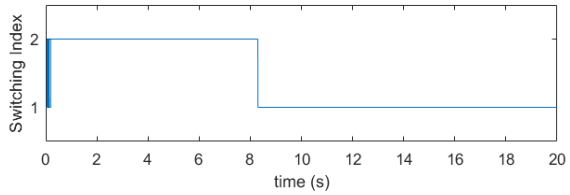

Fig. 11: Switching results of the controller.

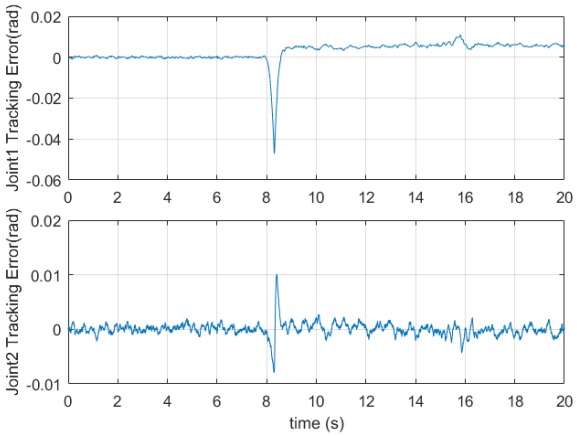

Fig. 12: Tracking error in joint space

5 Conclusion

This paper focuses on a sensor FTC strategy of a 2-DOF manipulator. Using the set-based FD approach and the optimization-based control approach, the fault-tolerant controller can choose the sensors-estimator combination that guarantees the closed-loop stability of the manipulator in the healthy and sensor-fault situations. This paper proposes a framework for the robot manipulator that have the similar hardware configuration against sensor faults. Although only the fault occurrence in the encoder in joint 1 is considered, this method can be used for every joint of the manipulator.

It should be noticed that this multi-sensor schemes consider the model uncertainties, modeling errors, process disturbances and measurement noises as bounding sets. The accuracy of the value of the bounding sets determine the sensitivity of fault detection. Another problem is that because this is a robust FD approach, small faults may not be detected on time after their occurrence, which may lead to performance degradation of FD and FTC and consequently results in abrupt changes of angular speeds of joints. Thus, in our future research, our focus will be to propose better strategies to improve the FD performance of the multi-sensor switching scheme and finally apply the proposed sensor FTC strategy into real robot manipulator systems.

References

[1] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Industrial Electronics*, vol. 62, pp. 3757-3767, 2015.

[2] D. Brambilla, L. M. Capisani, A. Ferrara, and P. Pisu, "Fault detection for robot manipulators via second-order sliding modes," *IEEE Trans. Industrial Electronics*, vol. 55, pp. 3954-3963, 2008.

[3] M. Defoort, K. C. Veluvolu, J. J. Rath, and M. Djemai, "Adaptive sensor and actuator fault estimation for a class of uncertain Lipschitz nonlinear systems," *Int. Journal of Adaptive Control and Signal Processing*, vol. 30, pp. 271-283, 2016.

[4] H.-J. Ma and G.-H. Yang, "Simultaneous fault diagnosis for robot manipulators with actuator and sensor faults," *Information Sciences*, vol. 366, pp. 12-30, 2016.

[5] V. Reppa, M. M. Polycarpou, and C. G. Panayiotou, "Decentralized isolation of multiple sensor faults in large-scale interconnected nonlinear systems," *IEEE Trans. Automatic Control*, vol. 60, pp. 1582-1596, 2015.

[6] X. Zhang, "Sensor bias fault detection and isolation in a class of nonlinear uncertain systems using adaptive estimation," *IEEE Trans. Automatic Control*, vol. 56, pp. 1220-1226, 2011.

[7] G. Paviglianiti, F. Pierri, F. Caccavale, and M. Mattei, "Robust fault detection and isolation for proprioceptive sensors of robot manipulators," *Mechatronics*, vol. 20, pp. 162-170, 2010.

[8] A. M. Pertew, H. J. Marquez, and Q. Zhao, "LMI-based sensor fault diagnosis for nonlinear Lipschitz systems," *Automatica*, vol. 43, pp. 1464-1469, 2007.

[9] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. Automatic Control*, vol. 47, pp. 576-593, 2002.

[10] M. M. Seron, X. W. Zhuo, J. A. De Doná, and J. J. Martínez, "Multisensor switching control strategy with fault tolerance guarantees," *Automatica*, vol. 44, pp. 88-97, 2008.

[11] M. E. Romero, M. M. Seron, and J. A. D. Dona, "Sensor fault-tolerant vector control of induction motors," *IET Control Theory & Applications*, vol. 4, pp. 1707-1724, 2010.

[12] A. Yetendje, M. M. Seron, J. A. D. Doná, and J. J. Martínez, "Sensor fault - tolerant control of a magnetic levitation system," *Int. J. Robust and Nonlinear Control*, vol. 20, pp. 2108-2121, 2010.

[13] Florin Stoican, "Fault tolerant control based on set-theoretic methods, " Ph.D. dissertation, Automatic Control Department, École Supérieure d'Électricité, Paris, France, 2011.

[14] M. M. Seron and J. A. De Doná, "Robust fault estimation and compensation for LPV systems under actuator and sensor faults," *Automatica*, vol. 52, pp. 294-301, 2015.

[15] F. Xu, V. Puig, C. Ocampo-Martinez, et al. "Actuator-fault detection and isolation based on interval observers and invariant sets, " *Decision and Control (CDC), 2013 IEEE 52nd Annu. Conf.*, pp. 4385-4390

[16] S. M. Hashemi, H. S. Abbas, and H. Werner, " Low-complexity linear parameter-varying modeling and control of a robotic manipulator," *Control Engineering Practice*, vol. 20, pp. 248-257, 2012.

[17] R. Tóth, F. Felici, P. Heuberger, and P. Van den Hof, "Crucial aspects of zero-order hold LPV state-space system discretization," *IFAC Proc. Volumes*, vol. 41, pp. 4952-4957, 2008.

[18] D. Rotondo, "Advances in gain-scheduling and fault tolerant control techniques, " Ph.D. dissertation, Technical Univ. of Catalonia, Barcelona, Spain, 2015.

[19] O. Sename, P. Gaspar, and J. Bokor, *Robust control and linear parameter varying approaches: application to vehicle dynamics*. Springer, 2013, vol. 437.

[20] W. M. H. Heemels, J. Daafouz, and G. Millerioux, "Observer-based control of discrete-time LPV systems with uncertain parameters," *IEEE Trans. Automatic Control*, vol. 55, pp. 2130-2135, 2010.

[21] F. Xu, "Diagnosis and fault-tolerant control using set-based methods, " Ph.D. dissertation, Technical Univ. of Catalonia, Barcelona, Spain, 2014.

[22] "*A465 Robot Arm User Guide for Use with a C500C Controller*, " UMI-33-465, CRS Robotics Cooperation, 2000.