

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Self-Healing Control Framework Against Actuator Fault of Single-Rotor Unmanned Helicopters

Xin Qi, Zhong Liu, Yuqing He, Liying Yang,
Yuqing He and Jianda Han

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62480>

Abstract

Unmanned helicopters (UHs) develop quickly because of their ability to hover and low speed flight. Facing different work conditions, UHs require the ability to safely operate under both external environment constraints, such as obstacles, and their own dynamic limits, especially after faults occurrence. To guarantee the postfault UH system safety and maximum ability, a self-healing control (SHC) framework is presented in this chapter which is composed of fault detection and diagnosis (FDD), fault-tolerant control (FTC), trajectory (re-)planning, and evaluation strategy. More specifically, actuator faults and saturation constraints are considered at the same time. Because of the existence of actuator constraints, usable actuator efficiency would be reduced after actuator fault occurrence. Thus, the performance of the postfault UH system should be evaluated to judge whether the original trajectory and reference is reachable, and the SHC would plan a new trajectory to guarantee the safety of the postfault system under environment constraints. At last, the effectiveness of proposed SHC framework is illustrated by numerical simulations.

Keywords: fault detection and diagnosis, fault-tolerant control, invariant set, self-healing control framework, trajectory (re-)planning

1. Introduction

Safe operation and reliability are important performances of unmanned aerial vehicles (UAVs). Unmanned helicopters (UHs) as a kind of UAVs develop quickly because of their ability to hover and low speed flight. Nevertheless, traditional control and planning strategies

cannot guarantee their safety and reliability in the face of malfunctions such as sensor faults, actuator faults, and component faults. Furthermore, taking physical limits such as actuator constraints and state constraints into account, the situation would be more serious. In order to ensure the reliability and safety of UHs, fault detection and diagnosis (FDD) and fault-tolerant control (FTC) approaches become focus of research and many related research results have been presented [1]. However, physical limits are not well considered in conventional FTC approaches which may affect their efficiency. On the other hand, besides UH's dynamic limits, external environment constraints, such as obstacles, also affect UH's safety. However, traditional planning methods cannot consider environment constraints and vehicles' limits at the same time which makes these methods helpless in the face of faults. In this chapter, a self-healing control (SHC) framework is proposed against actuator faults and constraints of single-rotor UHs under external environment constraints. The SHC framework is shown in **Figure 1** which involves FDD module, reconfigurable controller, trajectory (re-)planning module, and evaluator module.

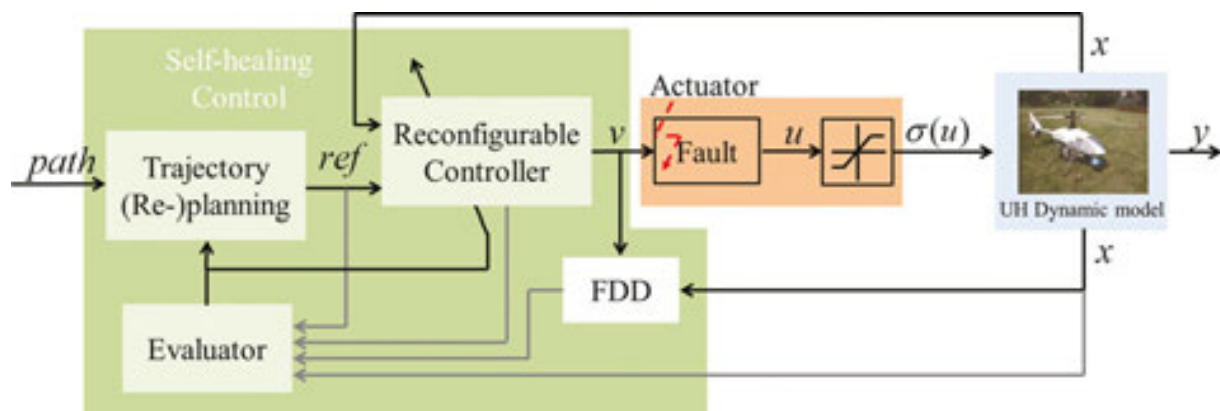


Figure 1. SHC framework.

The tasks of the above modules are as follows:

1. FDD module: Estimate the actuator healthy coefficient (AHC) [2] in real time. AHCs can indicate healthy conditions of actuators and actuator fault information is the basis of controller reconfiguration.
2. Reconfigurable controller: This module realizes the same functions as conventional FTC approaches and considers actuator constraints at the same time. After fault occurrence, the fault-free controller will be configured as a postfault controller according to AHCs provided by FDD module to guarantee the stability of the postfault UH system.
3. Trajectory (re-)planning module: Compute realizable trajectory based on desired path and UH dynamic model under environment constraints. This module calculates a feasible trajectory under both fault-free and postfault conditions. Furthermore, controller references are computed directly in this module which can be used by reconfigurable controller.

4. Evaluator module: Evaluate system performance of UHs according to AHCs, UH dynamic model, and controller and trajectory information. If the original controller and trajectory are not feasible, this module will ask related modules to reconfigure controller and replan trajectory.

The remaining part of this chapter is organized as follows: single-rotor UH and AHC models are simply introduced in Section 2; Section 3 investigates FDD approach against actuator faults based on extended Kalman filter (EKF) and linear neural network (LNN). In Section 4, the postfault controller is designed to guarantee the stability of UH system under both actuator faults and constraints; Section 5 presents invariant-set based planning (ISBP) approach that can compute controller reference according to desired path and UH dynamic model. Evaluation strategy is introduced in Section 6 and numerical simulations are shown in Section 7. Finally, Section 8 summarizes the chapter with conclusions.

2. Model of UH and AHCs

2.1. UH model

The 6-DOF dynamics of UH is given by the following Newton–Euler equations:

$$\begin{aligned}\dot{V}_b &= -\omega_b \times V_b + \frac{F_b}{m} + \frac{F_g}{m} \\ \dot{\omega}_b &= -[M_b - \omega_b \times (I_b \omega_b)] / I_b\end{aligned}$$

where $V_b = [u \ v \ w]$ is the velocity vector, $\omega_b = [p \ q \ r]$ is the angular velocity vector, F_b is the aerodynamic force vector, F_g is the gravity force vector, I_b is the moment of inertia matrix, M_b is the aerodynamic moment vector, and m is the UH mass.

Aerodynamic forces and moments can be calculated by

$$\begin{aligned}F_b &= f_f(\delta_{lon}, \delta_{lat}, \delta_{col}, \delta_{ped}, \omega_r) \\ M_b &= f_m(\delta_{lon}, \delta_{lat}, \delta_{col}, \delta_{ped}, \omega_r)\end{aligned}$$

where δ_{lon} , δ_{lat} , δ_{col} , and δ_{ped} are the longitude, latitude, main rotor collective pitch, and tail rotor collective pitch control inputs, respectively. ω_r is control input of main rotor rotating speed. Typically, f_f and f_m are nonlinear functions and their details can be found in [3, 4].

Note that δ_{lon} , δ_{lat} , δ_{col} are the nominal control inputs and they do not represent the actual actuator outputs. The relationship between the nominal and actual inputs is determined by the structure of swashplate [5] and their values can be converted to each other.

In order to guarantee the accuracy of AHC estimation, nonlinear model is used to design FDD approach. However, the above nonlinear model is too complex to design EKF filter so that a simplified nonlinear discrete-time model is used [6]:

$$\begin{cases} x(k) = f(x(k-1), u(k-1)) + \omega(k-1) \\ y(k) = h(x(k)) + v(k) \end{cases} \quad (1)$$

where $x \in R^n$ is the system state vector, $y \in R^p$ is the system output vector, $u \in R^m$ is the system control input vector, ω and v are the process noise and measurement noise that are assumed to be Gaussian white noise with zero mean and uncorrelated from each other. Considering swashplate configuration [5], the system control input vector is $u = [\theta_l, \theta_r, \theta_b, \theta_t, \omega_r]$ where $\theta_l, \theta_r, \theta_b$ are left, right, back swashplate actuator outputs, and $\theta_t = \delta_{ped}$ is the tail actuator output.

On the other hand, the linear dynamic model is used to design controller and trajectory planning which is shown as follows:

$$\begin{cases} x_h(k+1) = A_h x_h(k) + B_h u_h(k) \\ y_h(k) = C_h x_h(k) \end{cases} \quad (2)$$

where $x_h \in R^n = x - x_{trim}$, $y_h \in R^p = y - y_{trim}$, $u_h \in R^m = u - u_{trim}$ and x_{trim} , y_{trim} , u_{trim} are the trim values. A_h , B_h , and C_h are all constant matrices.

2.2. AHC model

Suppose v is the control signal given by controller called control variable and u is the actual actuator output called manipulated variable, their relationship is represented by

$$u(k) = \Lambda v(k) + (I - \Lambda) \bar{u}$$

where Λ is a diagonal matrix with $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$, $\lambda_i \in [0, 1]$, and $\bar{u} \in R^m$ is a constant vector. In addition, λ_i and \bar{u}_i are the proportional effectiveness and fault bias of the i th actuator's AHC. $\lambda_i = 1$ and $\bar{u}_i = 0$ represent that the i th actuator is fault-free; otherwise, the i th actuator has fault. Thus, the fault condition of actuators can be represented by AHCs.

3. EKF- and LNN-based FDD approach

Because of the serious nonlinear coupling between all control channels, we assume that an actuator has fault whereas others are still well, and the actuator for rotate speed control, ω_r , is

always well. Here, the multiplicative fault of an actuator as a parameter that is indicated by the AHC for this kind of fault is more likely to happen. The following will introduce the realization of fault diagnosis in the order of fault identification module, fault isolation module, and fault detection module. In this section, $x(k)$ is written as x_k for simplicity and $\hat{x}_{k|k-1}$ represents the pre-estimation value of x_k based on the information of $k-1$ step.

3.1. Fault identification module

The task of fault identification is to determine the size of the fault and the fault time after fault isolation. The size of a fault is indicated by the proportional effectiveness of AHC, so we should estimate the actual manipulated variable vector u according to state estimations from EKF and actual control signal v to calculate and obtain the proportional effectiveness.

As shown in Eq. (1), in a typical nonlinear discrete-time system, the input vector is obtained from actuators, and the current state vector x_k is calculated based on the state vector of the last step. The estimations of $\hat{x}_{k|k}$ and $\hat{x}_{k-1|k-1}$ can be obtained by EKF [7]. Therefore, the state equation can be transformed into the equation as follows:

$$\hat{x}_{k|k} = M(\hat{x}_{k-1|k-1})u + N(\hat{x}_{k-1|k-1}) \quad (3)$$

For the calculation, processes of forces and torques of the simplified nonlinear model have been linearized, Eq. (3) is a linear equation set, and the values of M and N are connected with state estimations of the last step only.

Typically, Eq. (3) is an overdetermined linear equation set and could not be solved directly [8]. After knowing which actuator is faulty, many output estimations of this actuator can be obtained by the faultless manipulated variables of other actuators according to Eq. (3). Multiply the sample column vectors formed by these estimations by a weight matrix and correct this result with an offset. Then, the manipulated variable estimation of this faulty actuator can be obtained.

To obtain the above weight matrices and offsets, consider a linear neuron as a simple LNN, and its input–output relation is expressed as follows:

$$a = f(Wp + b) = Wp + b$$

where a is the output vector, p is the input sample vector, W is the weight matrix, and b is the offset. The weight matrices and offsets can be obtained by training. Assume that an actuator has a fault, and let the output estimations of this actuator be a set of input of the linear neuron. After simulation, multiple sets of input samples can be obtained, and let the actual manipulated variable be the target value to train the weight matrix and offset of this actuator. Determine the weight matrix and offset of every faulty actuator, and make use of them in the online experience to calculate the estimation of every manipulated variable. The above process is

closely related to the estimation process of EKF; more certainly, EKF and LNN complete the joint estimation of system states and AHCs together.

3.2. Fault isolation module

The task of fault isolation is to determine the location and type of the fault after fault detection. The structure of this module is shown in **Figure 2** to record the completion time and pass the serial number of the faulty actuator to fault identification module.

Four fault identification modules are in parallel in this module, and we assume that there are different actuator faults in different identification modules. Therefore, every fault identification module can obtain the proportional effectiveness of AHC called isolation AHC. Make use of the isolation AHC of the i th actuator and let these values of other actuators be both 1 to calculate the manipulated variables called isolation-manipulated variables according to control variable vector v . Based on these variables, we can obtain the system state vector and define it as x_{iso} . In this way, we define the isolation residual of the i th actuator as follows:

$$res_i = (x_{iso} - x_{est})^T (x_{iso} - x_{est})$$

where x_{est} is state estimation. If the isolation residual of an actuator is less than the isolation residuals of others, there will be fault in this actuator. For example, if res_i is less than other isolation residuals, the i th actuator can be considered to be faulty.

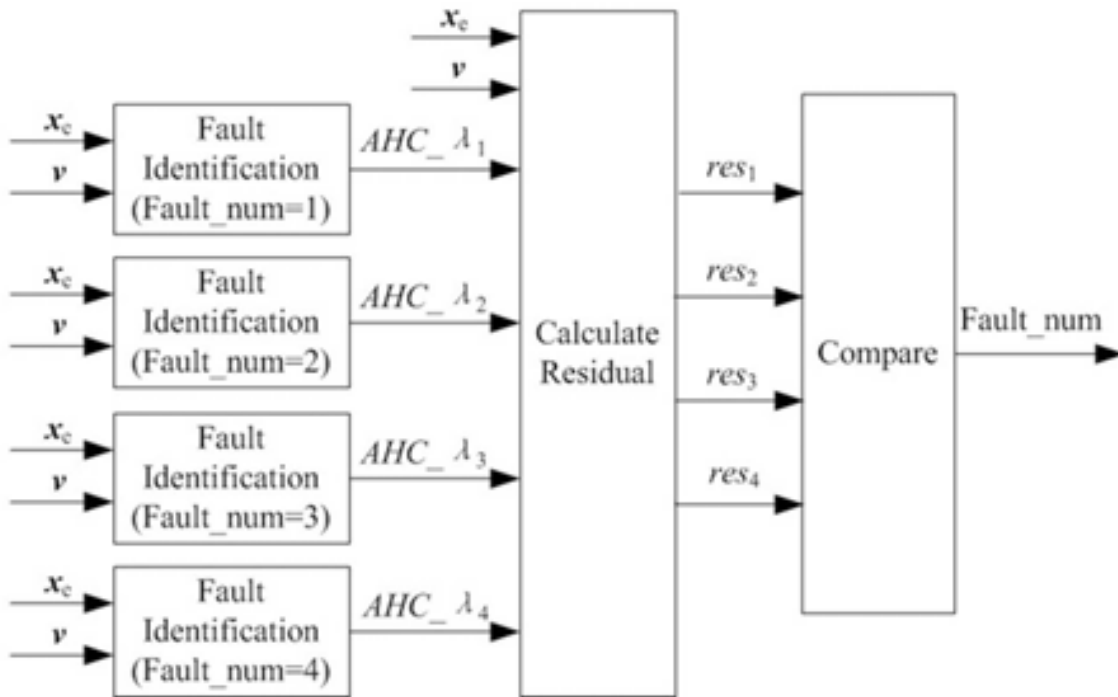


Figure 2. The structure of fault isolation.

3.3. Fault detection module

The basis of the above two modules is that the fault is detected quickly and accurately. Known from the EKF process, the filter obtains the state vector pre-estimation $\hat{x}_{k|k-1}$ by the estimations of the last step and updates $\hat{x}_{k|k-1}$ by the measurement information to obtain the current estimation of the state vector $\hat{x}_{k|k}$. If there is a fault in the actuator, there will be difference between the pre-estimations of states and the actual states with fault. However, the estimations of states would track the actual states after the update. Define a filter residual as follows:

$$r_k = \hat{x}_{k|k} - \hat{x}_{k|k-1}$$

Perform the weighted sum of squared residuals (WSSR) operation [9]:

$$\text{WSSR}_k = r_k^T \Sigma r_k$$

where the weight matrix $\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$ and σ_i is the standard deviation of the filter residual of the i th state variable to make the residual that can indicate faults work.

According to the threshold value γ set in advance, the criteria for fault detection are as follows: if $\text{WSSR}_k > \gamma$, fault; if $\text{WSSR}_k \leq \gamma$, no fault.

4. Reconfigurable controller design

In this section, AHC-based antiwindup controller design method is introduced which can be achieved by solving a set of linear matrix inequalities (LMIs). At the same time, the related invariant-set-based safety region is also calculated. The definition of invariant set and the controller design under actuator constraints are given as follows.

4.1. Invariant set

Considering a linear system

$$x(k+1) = Ax(k)$$

the (positively) invariant set can be given by the following definition.

Definition 1 [10]: The set $S \subset R^n$ is said invariant for the above system if for all $x(0) \in S$ the solution $x(k) \in S$. If $x(0) \in S$ implies $x(k) \in S$ for $k > 0$, then we say that S is positively invariant.

Based on Definition 1, the definition of robustly controlled invariant set for the following system is provided:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + D\omega(k) \\ y(k) = Cx(k) \end{cases}$$

Definition 2 [10]: The set $S \subset R^n$ is said robustly controlled invariant for the above system if there exists a continuous feedback control law $u(k) = Kx(k)$, which ensures the existence and uniqueness of the solution on R^+ such that S is positively invariant for the closed loop system.

An invariant set S can be constructed by the following theorem.

Theorem 1 [11]: The set $S_\rho = \{x(k) \in R^n \mid V(x(k)) \leq \rho\}$ with $\Delta V(x(k)) \leq 0$ for all $x(k) \in S_\rho$ is positively invariant where V is a function.

For convenience, ρ is chosen equal to 1 and simplify S_ρ as $S = \{V(x(k)) \leq 1\}$. Hence, the construction of S finds a function V , which implies $\Delta V(x(k)) \leq 0$ when $V(x(k)) \leq 1$. Obviously, if we assume $V(x(k)) \geq 0$, V would be a candidate of the Lyapunov function. More specially, if the Lyapunov candidate function is in quadratic form such that $V(x(k)) = x^T(k)Px(k)$, the invariant set will be an ellipsoid and this nature will be used later.

4.2. Controller design under actuator constraints

Consider linear discrete UH model, Eq. (2), with actuator constraints:

$$\begin{cases} x_h(k+1) = A_h x_h(k) + B_h \sigma(u_h(k)) \\ y_h(k) = C_h x_h(k) \end{cases}$$

Where $\sigma(u_h(k))$ represents constrained control inputs such that $-1 \leq u(k) \leq 1$ and the saturation feature can be defined by $\sigma(u_i(k)) = \text{sgn}(u_i(k)) \min\{1, |u_i(k)|\}$, where $\text{sgn}(\cdot)$ is a sign function and $i = 1, \dots, m$. Then, considering AHC model, the above system can be rewritten as

$$\begin{cases} x_h(k+1) = A_h x_h(k) + B_{h0} \sigma(v(k)) + B_{hf} \bar{u} \\ y_h(k) = C_h x_h(k) \end{cases}$$

Where $B_{h0} = B_h \Lambda$ and $B_{hf} = B_h (I - \Lambda)$. In the following discussion, assume the number of $\lambda_i = 0$ is m_f , which implies m_f actuators cannot respond to the control signal and $m_0 = m - m_f$ is the number of fault-free/partial-fault actuators that can respond to the control signal. In other words, m_0 represents the number of nonzero columns of matrix B_{h0} . In this way, the postfault system with different actuator faults can be represented.

In order to achieve set-point tracking offset free and compensate actuator saturation, an integrator with a saturation compensator is introduced as follows:

$$e(k+1) = e(k) + T_s (\text{ref} - C_h x_h(k)) - T_s E_c \varphi(v(k))$$

where $e(k)$ is the integrator state vector, E_c is the pending compensator matrix, $\varphi(v(k)) = v(k) - \sigma(v(k))$ is the difference of controller outputs and actuator outputs under saturation [12], $\text{ref} \in \Omega_{\text{ref}} = \{\text{ref} \mid \text{ref}^T Q_r \text{ref} \leq 1\}$ is a set-point reference, Q_r is a constant matrix, and T_s is sampling period. Thus, a new open-loop system with extend state vector $x(k) = [x_h^T(k) \ e^T(k)]^T$ can be expressed by

$$\begin{cases} x(k+1) = A_e x(k) + Bv(k) + D\omega(k) - (B + RE_c)\varphi(v(k)) \\ y(k) = Cx(k) \end{cases} \quad (4)$$

where $\omega(k) = [\bar{u}^T \ \text{ref}^T]^T$,

$$A_e = \begin{bmatrix} A_h & 0 \\ -T_s C_h & I \end{bmatrix}, B = \begin{bmatrix} B_{h0} \\ 0 \end{bmatrix}, D = [D_1 \ D_2] = \begin{bmatrix} B_{hf} & 0 \\ 0 & T_s I \end{bmatrix}, \\ R = \begin{bmatrix} 0 \\ T_s I \end{bmatrix} \text{ and } C = [C_h \ 0]$$

Note that, if $m_f = 0$, $\omega(k) = \text{ref}$, then $D = [0 \ (T_s I)^T]^T$ will hold.

Based on linear quadratic regulator (LQR) theory, state-feedback controller $v(k) = Kx(k)$ can be calculated. Thus, the closed-loop postfault system can be expressed by

$$\begin{cases} x(k+1) = Ax(k) + D\omega(k) - (B + RE_c)\varphi(v(k)) \\ y(k) = Cx(k) \end{cases}$$

where $A = (A_e + BK)$. In order to guarantee the closed-loop system stability and to track offset-free under external input $\omega(k)$ and actuator constraints, the following theorem is proposed. As a basis, first a lemma is given which defines a set of system states related to actuator saturation.

Lemma 1 ([12]): Define the following polyhedral set with matrix $G \in \mathbb{R}^{m_0 \times (n+p)}$:

$$\mathcal{E} = \{x(k) \mid |(K_i - G_i)x(k)| \leq 1, i = 1, \dots, m\}$$

where K_i and G_i represent the i th row of matrices K and G . If $x(k) \in \mathcal{E}$, then the relation

$$\varphi(Kx(k))^T T [\varphi(Kx(k)) - Gx(k)] \leq 0$$

is verified for any positive-definite matrix $T \in R^{m_0 \times m_0}$.

Based on this lemma, saturation compensator can be designed in the following theorem.

Theorem 2: Given LQR parameter matrices Q_u, Q_x , if there exist a symmetric positive-definite matrix $W \in R^{(n+p) \times (n+p)}$, matrices $X \in R^{m_0 \times (n+p)}$, $Y \in R^{m_0 \times (n+p)}$, $Z \in R^{p \times m_0}$, diagonal positive-definite matrices $S \in R^{m_0 \times m_0}$, $W_r \in R^{p \times p}$, and a positive scale η satisfying

$$\inf_{W, X, Y, Z, S, W_r} \eta$$

$$\begin{bmatrix} 1 & X_i - Y_i \\ * & W \end{bmatrix} \geq 0, I = 1, \dots, m_0$$

$$\begin{bmatrix} -W & Y^T & 0 & 0 & WA^T & -C^T & X^T & W \\ * & -2S & 0 & 0 & -SB^T - Z^T R^T & 0 & 0 & 0 \\ * & * & -\eta & 0 & D_1^T & 0 & 0 & 0 \\ * & * & * & -W_r & D_2^T & I & 0 & 0 \\ * & * & * & * & -W & 0 & 0 & 0 \\ * & * & * & * & * & -I & 0 & 0 \\ * & * & * & * & * & * & -Q_u^{-1} & 0 \\ * & * & * & * & * & * & * & -Q_x^{-1} \end{bmatrix} < 0$$

where X_i, Y_i represent the i th row of matrices X, Y and $K = XP, G = YP$. Then, the saturation compensator will be $E_c = ZS^{-1}$ and the closed-loop postfault system will be stable if $x(k)$ inside the stability domain $S = \{x(k) \mid x^T(k)Px(k) \leq 1\}$ with $P = W^{-1}$. Note that S is defined as the safety region in this chapter. Ω_{ref} is also achieved with $Q_r = W_r^{-1}$. The proof can be found in [13].

According to Theorem 2, the safety region S is an invariant set which means that if initial states and steady states of a system are inside the set, the state trajectory from initial states to steady states will also be inside the set. In this way, a postfault system with a fault-tolerant controller can be seen as a new closed-loop system with the initial state $x(k_f)$, where k_f is instant when the fault is detected. Clearly, a postfault system will be safe if initial state $x(k_f)$ is inside the safety region S .

On the other hand, the related admissible set of reference Ω_{ref} is also achieved by Theorem 2 and it has a closed relationship with the controller. In the next section, the controller design and the reference design will be composed together.

5. ISBP approach

This section proposes an ISBP approach that can plan a feasible trajectory based on the desired path nodes under both external environment constraints and their own dynamic limits [14]. In order to consider environment constraints and dynamic constraints at the same time, invariant set is used as a bridge to connect the two kinds of constraints because the invariant set is calculated based on the Lyapunov function that is linked to dynamic model, and on the other hand, the invariant set has certain geometrical shapes, such as ellipsoid, so that it can easily be represented in work space with obstacles.

Note that, in this section, we have assumed that the heading of UH is kept to be 0, pitch angle and roll angle of UH are kept small so that the position of UH in the world coordinate system can be considered as the integration of velocities in the body coordinate system.

5.1. Reachable set

Considering System (4) with controller $v(k)=Kx(k)$, the definition of reachable set is given.

Definition 3 [15]: The reachable set \mathcal{S}_r is defined as

$$\mathcal{S}_r = \{y(k) = Cx(k) \mid x(k) \in \mathcal{S}_{rx}\}$$

$$\mathcal{S}_{rx} = \left\{ x(k) \left| \begin{array}{l} x(k), \omega(k), \text{ref satisfy (4),} \\ v(k) = Kx(k) \in \Omega_v, \\ \text{ref} \in \Omega_{\text{ref}}, x(0) = 0, k \geq 0 \end{array} \right. \right\}$$

where $\Omega_v = \{v(k) \mid -1 \leq v_i \leq 1, i=1, \dots, m_0\}$.

5.2. ISBP method

For the convenience of discussion, a 2D work space is used in the following whereas the proposed ISBP approach can be expanded in 3D condition directly. Consider a preknown environment as shown in **Figure 3**, where the initial location is L_0 and the goal location is L_5 . Based on the preknown environment, a lot of path planning approaches can be used to find a feasible path against external environment constraints such as obstacles. Suppose this path is represented by a series of nodes such as $L_0 \sim L_5$ as shown in **Figure 3**.

The target of the proposed ISBP approach is to calculate feasible controller reference inputs to move the UH from the initial location to the goal location based on the feasible path and satisfy both external environment constraints and UH's limits.

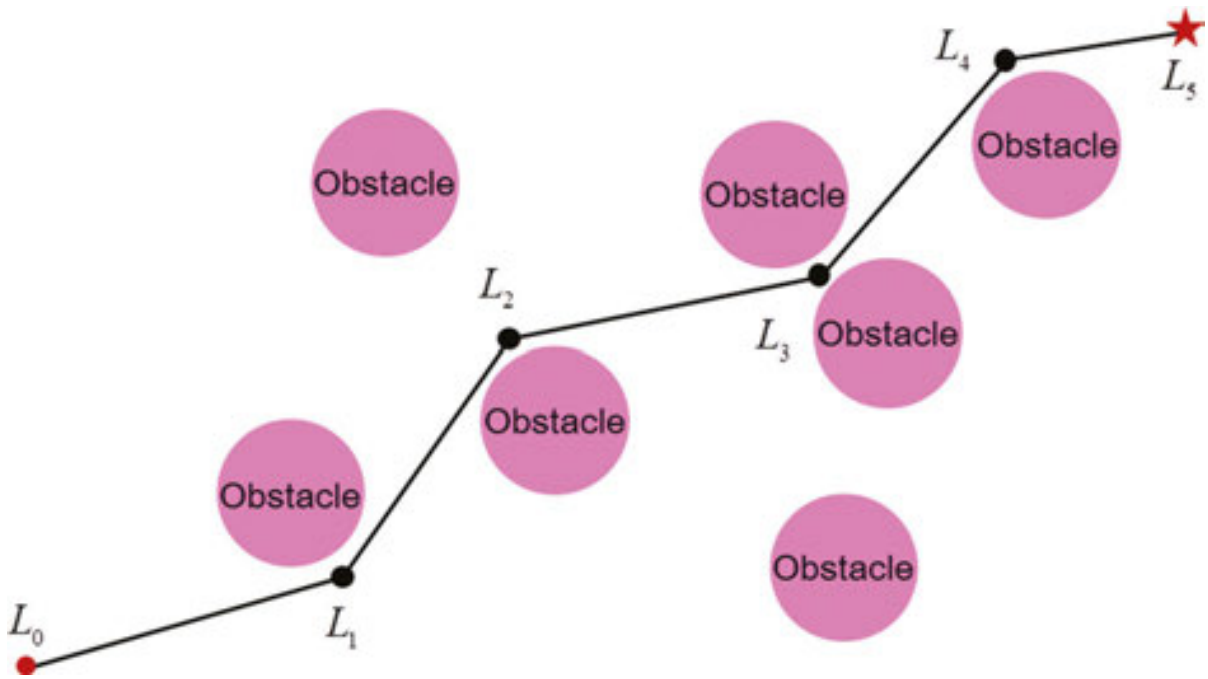


Figure 3. Two-dimensional environment and feasible path.

Consider a simple example in the preknown environment as shown in **Figure 4(a)** where the initial location is L_0 and the goal location is L_1 . Suppose a feasible path is represented by two nodes L_0 and L_1 and their connection is shown in the same figure.

Clearly, the work space is state space; thus, the location in work space is equality to the state such as the initial state, which is equal to L_0 . Thus, an invariant set S can be constructed according to plant dynamic model (4) under controller $v(k)=Kx(k)$, which also guarantees obstacle avoidance. Suppose the invariant set is S_0 and the related reachable set is S_{r0} as shown in **Figure 4(b)**. Clearly, because of the existence of constraints such as obstacles and UH's limits, the invariant set S_0 may not contain the goal location L_1 as shown in **Figure 4(b)**. Closely, the goal location L_1 is also outside the reachable set S_{r0} , which means L_1 is unreachable under this condition. Thus, the system states should be continued moving.

Suppose the intersection of reachable set S_{r0} and feasible path $\overline{L_0 L_1}$ is c_1 as shown in **Figure 4(b)**. In order to move state from c_1 to L_1 , a new invariant set, whose center is c_1 , is required. Suppose the state of the new system is $\bar{x}(k)=x(k)-x_{c1}$, where x_{c1} is a constant vector whose position is equal to c_1 and other elements are 0. Based on the new system, we can construct the second invariant set S_1 whose center is c_1 as shown in **Figure 4(c)**. The related reachable set S_{r1} is also shown. Obviously, the goal location L_1 is inside the reachable set S_{r1} now, which means L_1 is reachable under this condition. In this way, we can choose the center of S_1 , c_1 , and the goal location L_1 as a sequence of the controller reference input. Finally, Figure 4(d) shows the practical trajectory that is obtained by the UH actual operation.

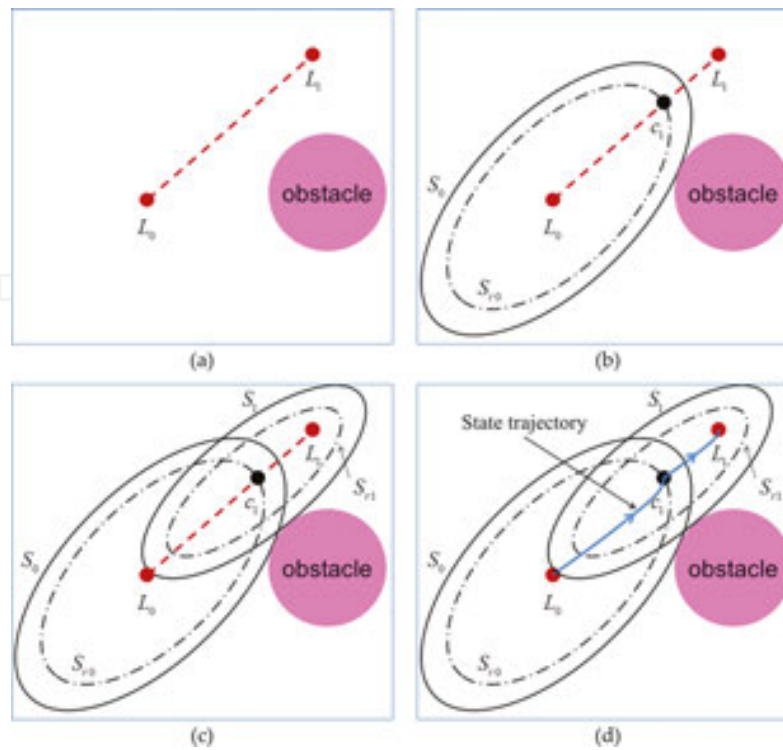


Figure 4. ISBP in 2D condition.

5.3. Calculation of the invariant set and the reachable set

According to Theorem 2, the invariant set S and the reference set Ω_{ref} are obtained. Assume the reachable set S_r defined by Definition 3 is included in the invariant set S such that $S_r \subset S$. Because the reference is bounded such as $ref \in \Omega_{ref}$ and the integrator can guarantee all references to be reachable such as $y(\infty) = ref$, the reachable set S_r can be denoted by $S_r = \{ref \mid ref^T Q_r ref \leq 1\}$. Thus, the invariant set and reachable set neglecting external environment constraints are obtained.

However, in order to guarantee controlled plant obstacle avoidance, the state trajectory should be kept outside all of spheres that represent obstacles. For achieving it, the invariant set S should not intersect with all of obstacle spheres. Suppose $S_0 \subset R^{n+p}$ is a sphere that is defined by $S_0 = \{x(k) \mid x^T(k) Q_0 x(k) \leq 1\}$, where $Q_0 = \text{diag}[q_1, \dots, q_{n+p}]$, $q_i = 1/d_{min}^2$, d_{min} is the distance between the center of the nearest obstacle and the current system equilibrium point.

According to the above analysis, the following theorem is proposed to calculate the maximum invariant set S and the reachable set S_r of system (4) under both external environment constraints and UH's limits.

Theorem 3: Given $\eta > 0$, LQR parameter matrices Q_u , Q_x , symmetric positive-definite matrices W_0 and R , if there exist a symmetric positive-definite matrix $W \in R^{(n+p) \times (n+p)}$, matrices

$X \in R^{m_0 \times (n+p)}$, $Y \in R^{m_0 \times (n+p)}$, $Z \in R^{p \times m_0}$, diagonal positive-definite matrices $S \in R^{m_0 \times m_0}$, $W_r \in R^{p \times p}$, and a positive scale λ satisfying

$$\inf_{W, X, Y, Z, S, W_r} \lambda$$

$$\begin{bmatrix} 1 & X_i - Y_i \\ * & W \end{bmatrix} \geq 0, I = 1, \dots, m_0$$

$$\begin{bmatrix} -W & Y^T & 0 & 0 & WA^T & -C^T & X^T & W \\ * & -2S & 0 & 0 & -SB^T - Z^T R^T & 0 & 0 & 0 \\ * & * & -\eta & 0 & D_1^T & 0 & 0 & 0 \\ * & * & * & -W_r & D_2^T & I & 0 & 0 \\ * & * & * & * & -W & 0 & 0 & 0 \\ * & * & * & * & * & -I & 0 & 0 \\ * & * & * & * & * & * & -Q_u^{-1} & 0 \\ * & * & * & * & * & * & * & -Q_x^{-1} \end{bmatrix} < 0$$

$$\begin{bmatrix} I_p & 0_{p \times n} \end{bmatrix}^T W_r \begin{bmatrix} I_p & 0_{p \times n} \end{bmatrix} \leq W$$

$$\begin{bmatrix} \lambda R & I \\ * & W_r \end{bmatrix} \leq 0$$

$$W \leq W_0$$

$$\lambda > 0$$

where $W_0 = Q_0^{-1}$, then obstacle avoidance and $-1 \leq v(k) \leq 1$ would be guaranteed, furthermore, the ellipsoid $S = \{x(k) \mid x^T(k) P x(k) \leq 1\}$ is an invariant set with $P = W^{-1}$ and $S_r = \{\text{ref} \mid \text{ref}^T Q_r \text{ref} \leq 1\}$ is a reachable set with $Q_r = W_r^{-1}$, which satisfies $S_r \subset S$.

Clearly, Theorem 2 is included in Theorem 3 because the invariant set is a bridge to connect environment and UH dynamic constraints. Thus, by the ISBP approach, the controller design and controller reference calculation are achieved simultaneously when the preknown work space, the desired path nodes, and the UH dynamic model are given.

The relationship between the related sets and the obstacles are shown in **Figure 5** assuming that all sets are projected to a 2D plane.

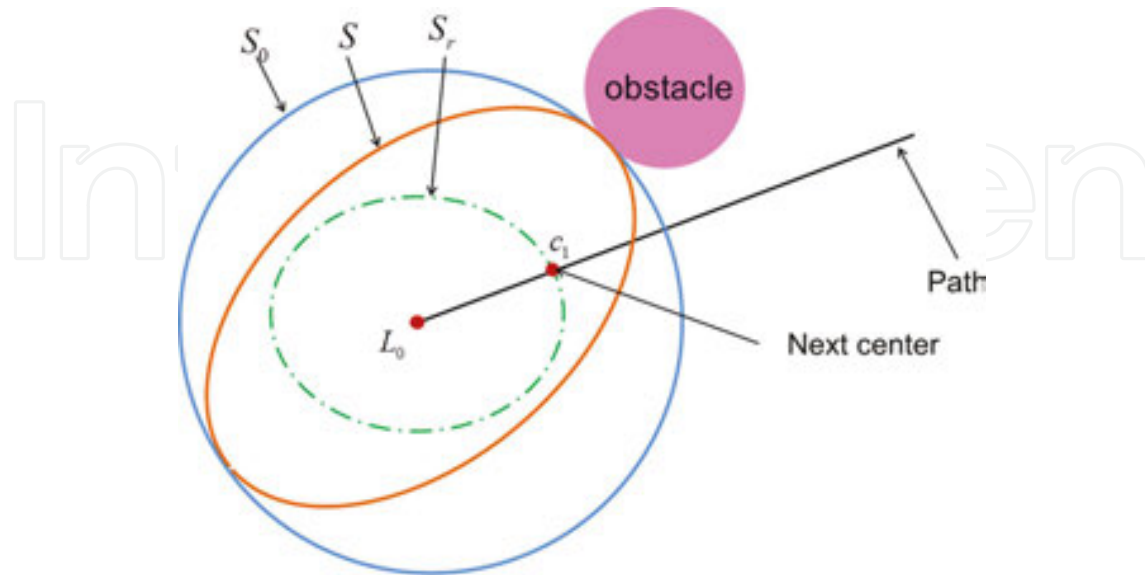


Figure 5. The relationship of related sets.

In other words, based on the calculated invariant set S and the reachable set S_r , a new center c_1 can be computed which is the intersection point of S_r and the feasible path. The next step is to calculate the second invariant set and the reachable set whose center is c_1 until the last reachable set covers the final path point.

6. Evaluation strategy

The open-loop model (2) of UH is unstable and the actuator outputs are limited. Thus, the only regional stability of UH can be guaranteed and the stability region is determined by the system structure and actuator constraints [16]. In other words, if the states of UH are inside the stability region, the UH would be of safety; otherwise, the UH may be in danger. Obviously, after actuator fault occurrence, actuator efficiency will be reduced. Hence, the safety region of the postfault system will be different with the fault-free case as shown in **Figure 6(a)**. Suppose that the safety region of the fault-free system with a fault-free controller is Ω_{ff} and the postfault system with a FTC controller is Ω_{pf} as shown in **Figure 6(b)**; furthermore, the initial state of the fault-free system is $x_0 \in \Omega_{ff}$ and an actuator fault is detected at k_{f1} . Clearly, the state $x(k_{f1})$ is outside the safety region of the postfault system Ω_{pf} so that the postfault system may be unstable (as shown by state trajectory $x(k_{f1})x'(k_{f1})$) at last. That is to say, the actuator fault cannot be compensated. In contrast, suppose that the other fault is detected at k_{f2} and $x(k_{f2}) \in \Omega_{pf}$ is valid, which represents that the fault can be compensated. However, the states

in steady case are also determined by references such as x_{f1} which may be outside the safety region Ω_{pf} . Obviously, this reference is unreachable and tracking such reference may lead UH unsafe (as shown by the state trajectory $x(k_{f2})x'_{f1}$). Hence, reference reachability should be analyzed before system motion and a new reachable reference is necessary. Compared with x_{f1} , x_{f2} may be more reasonable which is inside Ω_{pf} .

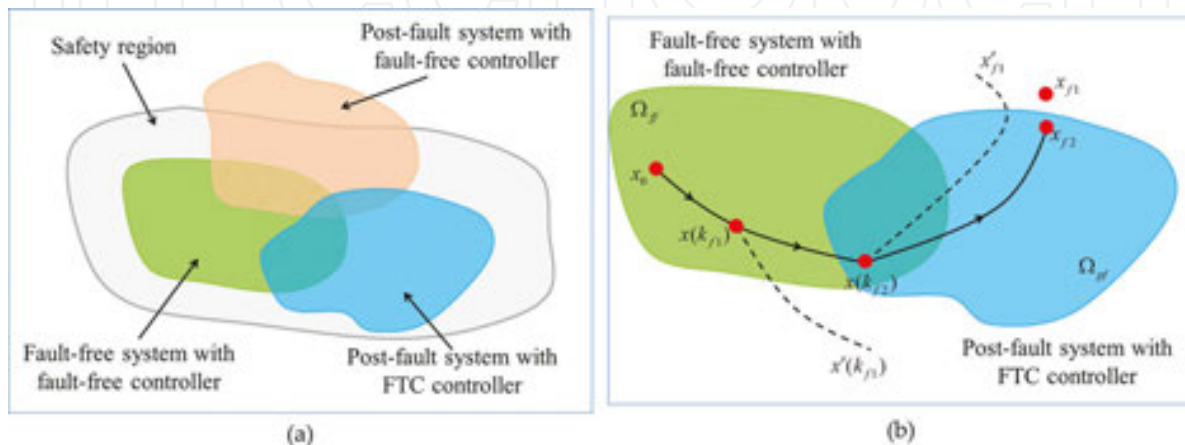


Figure 6. A sample of safety region of fault-free and postfault system in 2D state space.

According to Theorem 2, the safety region S is an invariant set. Clearly, a postfault system will be safe if initial states are inside the safety region S_f of the postfault system. In this way, the initial states of the postfault system can be evaluated. Second, the steady states will be analyzed. In steady-state case, actuators are not expected to be saturated so that the remaining efficiency of actuators can be used for disturbance defence. Hence, the original reference should be inside the reachable set of the postfault system such as $ref \in S_{rf}$, where S_{rf} is the reachable set of the postfault system; otherwise, the original reference ref is not reachable and tracking the original one may lead UH unsafe. The reason is that the actuator efficiency is reduced in the postfault system and tracking unreachable reference will lead fault-free actuator saturated which implies that UH cannot respond to control signal correctly. Under this condition, a new optimal reference is required which can be calculated by the trajectory replanning approach. In other words, if the original reference is not reachable after detecting the actuator fault, the ISBP approach should be called to calculate new trajectory and controller reference based on the postfault dynamic model of UH.

7. Numerical simulation

The parameters of nonlinear UH dynamic model used in numerical simulations is obtained from reference [6]. This model is used as the simulation model in this section. The step size of this simulation is 0.02s, which is the same as the control period of the real UH flight platform.

7.1. FDD simulation

Based on the simplified nonlinear model, the EKF is designed and LNN is used to train weight matrices and offsets of faulty actuators. The parameters of the EKF filter and the training results of the left swashplate actuator by LNN are as follows: $Q = \text{diag}(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $R = 2.5 \times 10^{-5} \cdot \text{diag}(40, 40, 40, 1, 1, 1, 1, 1, 1, 1, 1, 1)$, $W = [-0.0095, -0.00015, 0.000073, 0.076, 0.01023, -0.01008]$, $b = 0.03099$

where Q and R are covariance matrices of the process noise and the measurement noise, respectively. Then, we introduce a multiplicative fault on the left swashplate actuator as follows:

$$\begin{cases} \lambda_1 = 1 & 0 \leq t < 20 \\ \lambda_1 = 0.6 & t \geq 20 \end{cases}$$

The fault detection module based on the filter residuals can get the WSSR curve shown in **Figure 7**. When no fault occurs, the WSSR curve cannot exceed the threshold.

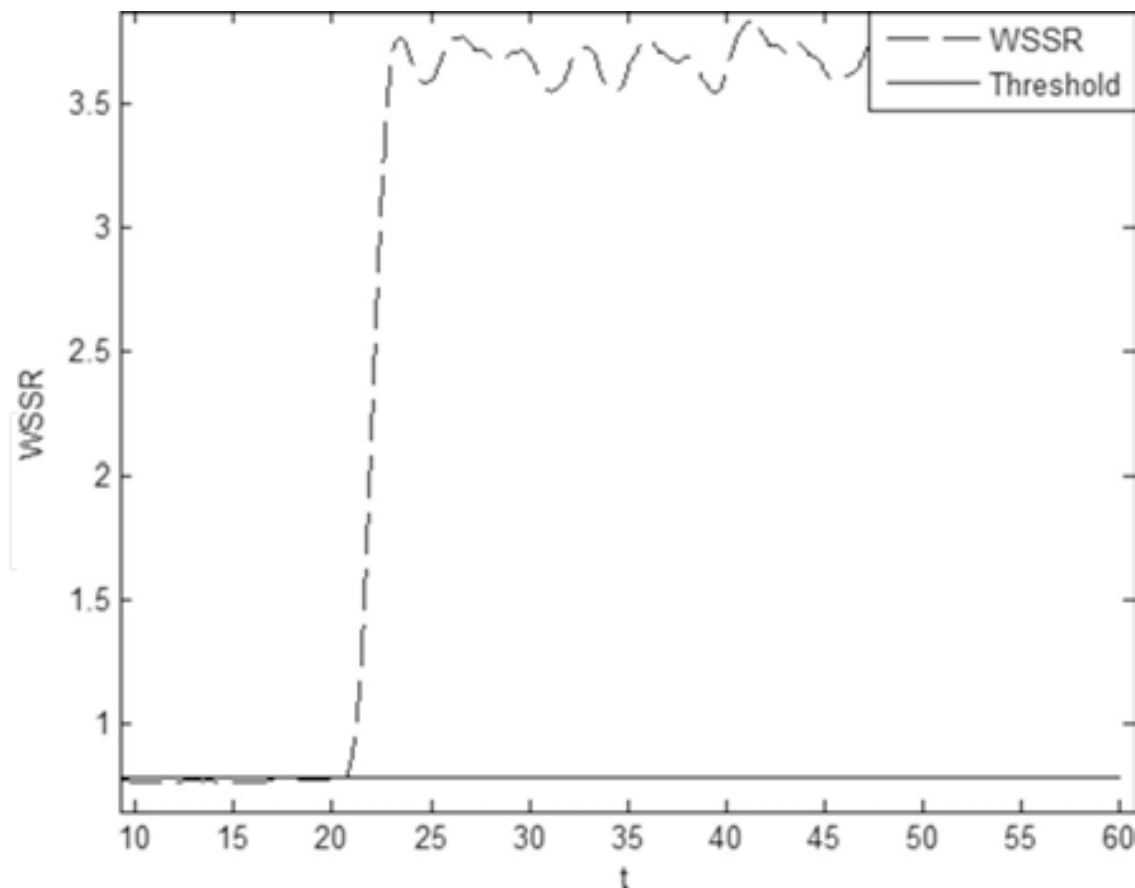


Figure 7. The curve of filter WSSR under the left servo fault.

After fault detection, isolation residuals are calculated to isolate this fault. The curves of four isolation residuals are shown in **Figure 8**. According to this figure, the isolation residual of the left swashplate actuator is less than other residuals. That is to say, the left actuator of swashplate is faulty. The fault isolation module cannot provide the right isolation results as soon as the completion of fault detection because of the vibration caused by the measurement noise. There may be misdiagnosis of the fault location within a very short time.

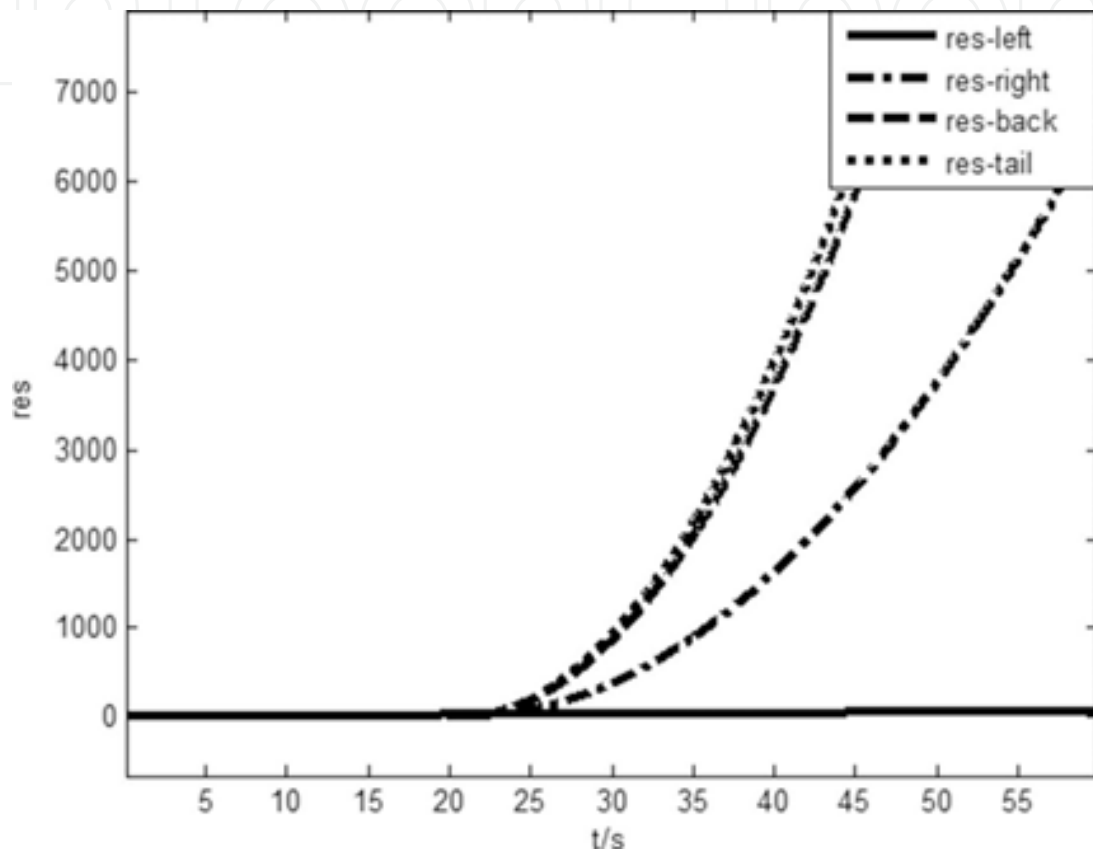


Figure 8. Curves of isolation residuals of every actuator under the left servo fault.

The curve of the proportional effectiveness of the left actuator under the fault is shown in **Figure 9**. After fault isolation, the isolation module passes the serial number of the faulty actuator to fault identification module to identify the AHC of this faulty actuator, and the simulation results indicate the size of this multiplicative fault. Because the weight matrices and offsets are trained by the LNN under the faults whose AHCs' parameters are all 0.5 for the online simulations, there may be error between the identification results and the actual size of this fault. By the calculation of average, the fault identification result is equal to the actual fault approximately.

The simulation results show that fault detection and isolation modules could detect and isolate a fault quickly, and the identification result is accurate enough; moreover, this method is with good real-time performance.

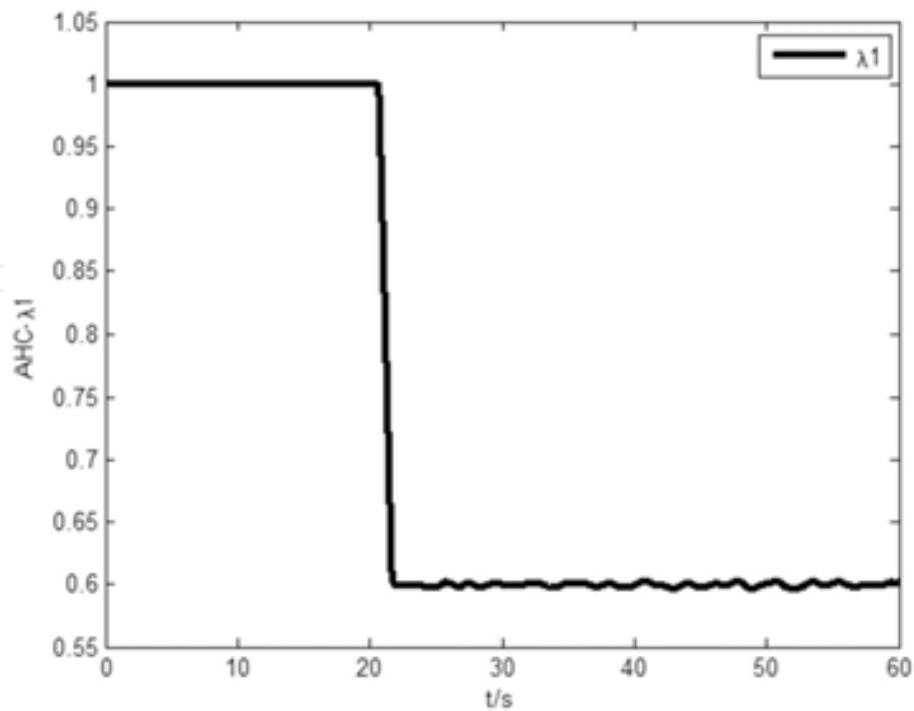


Figure 9. The curve of AHC proportional effectiveness under the left servo fault.

7.2. Controller and trajectory (re-)planning simulation

A linear UH model (2) with added three-axis positions is used in this part and related parameters can be found in [13]. The added positions are considered integration of velocities in the body coordinate. Thus, the system states, control inputs, and outputs are

$$x_h = [p_x \ p_y \ p_z \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r \ a_{1s} \ b_{1s}]$$

$$u = [\theta_l \ \theta_r \ \theta_b \ \theta_t \ \omega_r], \quad y_h = [p_x \ p_y \ p_z \ \psi]$$

Assume that the left swashplate actuator has fault at 6.4s with $\lambda_1=0.6$, and the LQR parameter matrices are

$$Q_x = \text{diag}(0 \ 0 \ 0 \ 0.01 \ 0.01 \ 0.01 \ 3 \ 1 \ 0.02 \ 0 \ 0 \ 0 \ 0.1 \ 0.1 \ 0.001 \ 0.001 \ 0.003 \ 0.01)$$

$$Q_u = \text{diag}(1110.50.1)$$

The preknown work space is shown in **Figure 10** where the start point is (0,0,0) and the target point is (20,20,20). The external environment constraint considered here is a sphere obstacle whose position is (10,10,10) and radius is 1. The feasible path is given by path nodes, such as (0,0,0), (8,8,13), and (20,20,20), which can guide UH from the start point to the target point with obstacle avoidance.

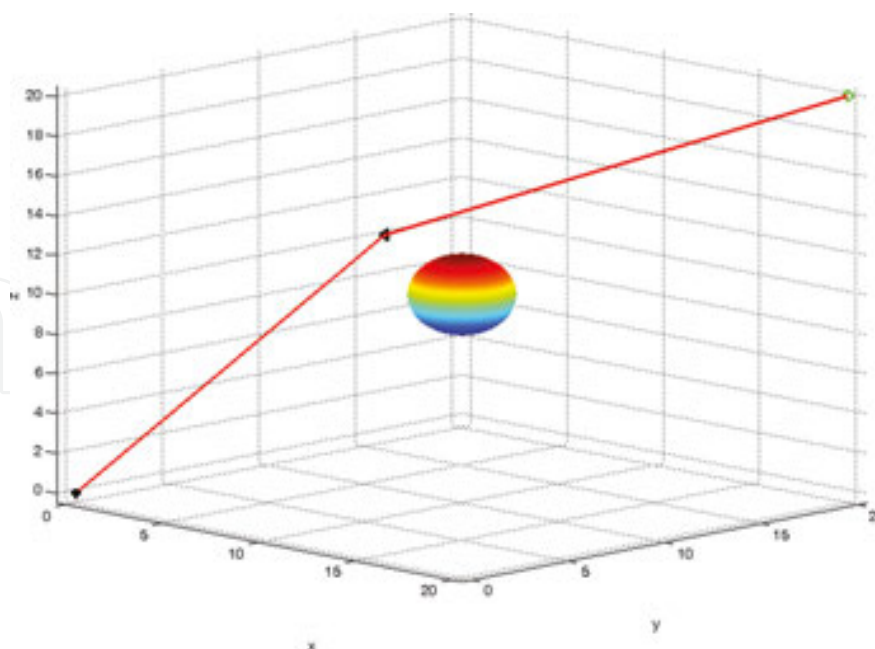


Figure 10. Preknown work space and feasible path.

According to Theorem 3, a series of controllers K , invariant sets S , and reachable sets S_r satisfying obstacle avoidance and UH dynamic limits can be computed as shown in **Figure 11** where the green ellipsoids are the reachable sets in 3D and the centers of ellipsoids, blue stars, are controller references. According to these references, the UH can run from the start point to the target point and the actual trajectory is shown by the blue curve. Clearly, the trajectory is kept inside the green ellipsoids so that it is also inside the invariant sets.

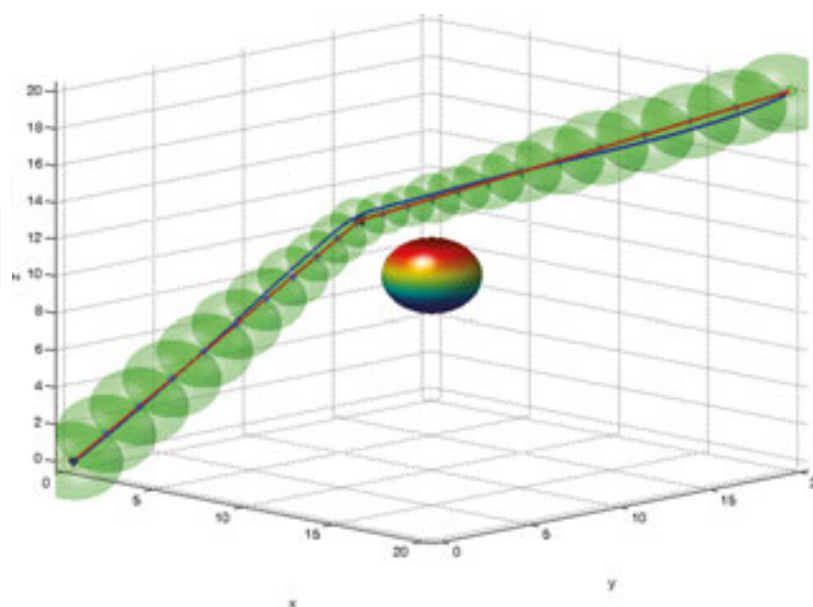


Figure 11. Reachable sets and UH actual trajectory in fault-free condition.

Compared with fault-free case as shown in **Figure 11**, **Figure 12** shows the results of the postfault case without the SHC framework. The actuator fault is detected at 6.4s and the related state is marked by red points in the figure. According to the postfault dynamic model, a new reachable set, red ellipsoid, is calculated. It is easy to see that the original reference, black point, is outside the reachable set of the postfault system which implies that the original reference is unreachable. Under this condition, if it does nothing, the UH system may be in danger as shown by the blue curve. The related manipulated variables are shown in **Figure 13** where the blue curves are the actuator outputs and the red dashed lines are the actuator constraints. Clearly, the actuator outputs are saturated at last which leads the UH system out of order.

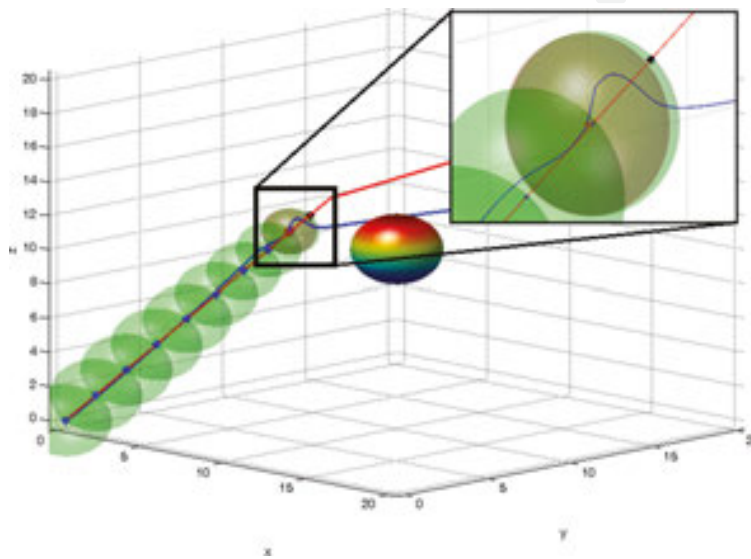


Figure 12. Reachable sets and UH actual trajectory in postfault condition without SHC.

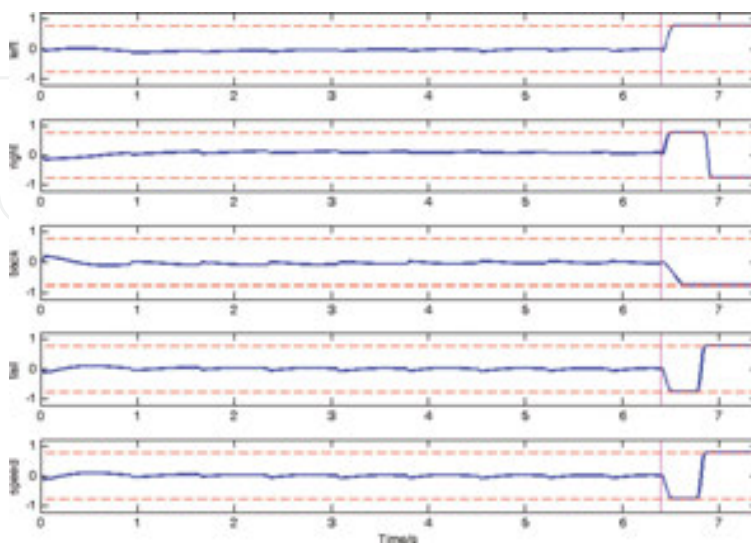


Figure 13. Actuator outputs in postfault condition without SHC.

Figure 14 shows the results of the postfault system with a SHC framework. After fault detection, Theorem 3 is recalled to calculate the new controllers, invariant sets, and reachable sets to evaluate the performance and guarantee the postfault UH system to be stable. At last, the UH can reach the target point with obstacle avoidance as shown by the actual trajectory.

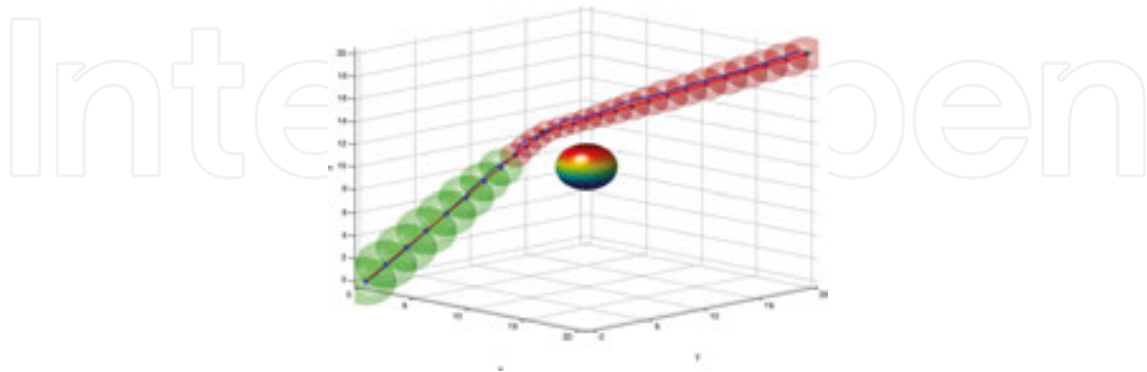


Figure 14. Reachable sets and UH actual trajectory in postfault condition with SHC.

8. Conclusions

In this chapter, a self-healing control framework is proposed for UH systems. The SHC framework aims at providing a solution to guarantee UHs safety and maximum ability to achieve the desired missions under both fault-free and postfault conditions. The EKF- and LNN-based FDD approach is used to detect and diagnosis actuator faults modeled by AHCs. Then, the AHC-based reconfigurable controller design method is proposed to calculate the fault-tolerant controller and the related safety region against both actuator faults and constraints by solving a set of LMIs. Third, the ISBP approach is presented for planning a feasible trajectory and computing the related controller reference under both external environment constraints and UH dynamic limits at the same time. After fault occurrence, based on the calculated safety region and controller reference, the performance of the postfault UH system can be evaluated, which can provide information whether the fault can be compensated and the original reference can be reached. If the original reference is not reachable, the ISBP approach will be recalled to calculate the new trajectory and reference again according to the postfault dynamic model. Finally, numerical simulations illustrate the effectiveness of the proposed SHC framework.

Acknowledgements

This work was supported by Key Program of National Natural Science Foundation of China under Grant 61433016 and National Natural Science Foundation of China under Grant 61503369.

Author details

Xin Qi^{1,2}, Zhong Liu^{1,2}, Yuqing He^{1*}, Liying Yang¹, Yuqing He¹ and Jianda Han¹

*Address all correspondence to: heyuqing@sia.cn

1 Shenyang Institute of Automation Chinese Academy of Sciences, Shenyang, P.R.China

2 University of Chinese Academy of Sciences, Beijing, P.R.China

References

- [1] Qi X., Qi J., Theilliol D., Zhang Y., Han J., Song D., Hua C. A review on fault diagnosis and fault tolerant control methods for single-rotor aerial vehicles. *Journal of Intelligent & Robotic Systems*. 2014;73(1):535–555. DOI: 10.1007/s10846-013-9954-z
- [2] Qi J., Song D., Wu C., Han J., Wang T. KF-based adaptive UKF algorithm and its application for rotorcraft UAV actuator failure estimation. *International Journal of Advanced Robotic Systems*. 2012;9:1–9. DOI: 10.5772/51893
- [3] Cai G., Chen B.M., Lee T.H. *Unmanned Rotorcraft System*. London: Springer; 2011. 270 pp. DOI: 10.1007/978-0-85729-635-1
- [4] Shim H. Hierarchical flight control system synthesis for rotorcraft-based unmanned aerial vehicles [thesis]. Berkeley: University of California; 2000.
- [5] Enns R., Si J. Helicopter flight-control reconfiguration for main rotor actuator failures. *Journal of Guidance, Control, and Dynamics*. 2003;26(4):572–584.
- [6] He Y., Han J. Acceleration-feedback-enhanced robust control of an unmanned helicopter. *Journal of Guidance, Control, and Dynamics*. 2010;33(4):1236–1250.
- [7] Ruschmann M., Wu N., Shin J.Y. Actuator fault diagnosis using two-stage extended Kalman filters. In: *AIAA Guidance, Navigation, and Control Conference*, 2–5 August, Toronto, Canada; 2010. pp. 1–21. DOI: 10.2514/6.2010-7703
- [8] Ji H. Algebra foundation of the theory of control [dissertation]. Hefei: University of Science and Technology of China Press; 2008. 5 pp.
- [9] Qiao C., Sun S. Fault detection for systems with missing measurements. In: *The 25th Chinese Control and Decision Conference*, 25–27 May, Guizhou, China; 2013. pp. 1050–1053.
- [10] Blanchini F. Set invariance in control. *Automatica*. 1999;35(11):1747–1767. DOI: 10.1016/S0005-1098(99)00113-2

- [11] Khalil H.K., Grizzle J.W. Nonlinear Systems. 3rd ed. New Jersey: Prentice- Hall; 2001. 750 pp.
- [12] da Silva Jr. J.M.G., Tarbouriech S. Antiwindup design with guaranteed regions of stability: an LMI-based approach. IEEE Transactions on Automatic Control. 2005;50(1): 106–111. DOI: 10.1109/TAC.2004.841128
- [13] Qi X., Qi J., Theilliol D., Song D., Zhang Y., Han J. Self-healing control design under actuator fault occurrence on single-rotor unmanned helicopters. Journal of Intelligent & Robotic Systems. 2016;online first,1–15. DOI: 10.1007/s10846-016-0341-4
- [14] Qi X., Theilliol D., Song D., Han J. Invariant-set-based planning approach for obstacle avoidance under vehicle dynamic constraints. In: 2015 International Conference on Robotics and Biomimetics, IEEE, Zhuhai, China, 6–9 December,; 2015.
- [15] Boyd S., Ghaoui E.L., El Ghaoui F., Venkataramanan B. Linear Matrix Inequalities in Systems and Control Theory. Philadelphia: SIAM; 1994. 205 pp.
- [16] Hu T., Lin Z., Qiu L. An explicit description of null controllable regions of linear systems with saturating actuators. Systems & Control Letters. 2002;47(1):65–78. DOI: 10.1016/S0167-6911(02)00176-7