

# Robust LPV Control for Attitude Stabilization of a Quadrotor Helicopter under Input Saturations

Seif-El-Islam Hasseni<sup>1,\*</sup>, Latifa Abdou<sup>2</sup>

<sup>1</sup>LMSE Laboratory, <sup>2</sup>LI3CUB Laboratory,  
Electrical Engineering Department, University of Biskra, Algeria

Received 18 March 2019; received in revised form 29 September 2019; accepted 15 October 2019

DOI: <https://doi.org/10.46604/aiti.2020.3953>

## Abstract

This article investigates the robust stabilization of the rotational subsystem of a quadrotor against external inputs (disturbances, noises, and parametric uncertainties) by the LFT-based LPV technique. By establishing the LPV attitude model, the LPV robust controller is designed for the system. The weighting functions are computed by Cuckoo Search, a meta-heuristic optimization algorithm. Besides, the input saturations are also taken into account through the Anti-Windup compensation technique. Simulation results show the robustness of the closed-loop system against disturbances, measurement noises, and the parametric uncertainties.

**Keywords:** quadrotor, LPV systems, optimal weighting functions, cuckoo search, input saturations

## 1. Introduction

Several researchers have focused on quadrotor aircraft because of its advantages such as vertical landing and stationary flight [1]. However, the quadrotor is a nonlinear complex vehicle and very sensible, which motivates the researchers to develop robust controllers. In this article, the development of a robust controller for the quadrotor attitude subsystem is investigated.

Recently, researchers have been interested in Linear-Parameter-Varying (LPV) theory because of its advantage that allowing them to represent the nonlinear model as the quasi-LPV system. There are several approaches to develop the LPV system from its nonlinear original model. The common ones are the polytopic approach [2] and the Linear-Fractional-Transformation (LFT) [3]. In this article, the LFT is used because it can exact representative the nonlinear model. On the other hand, one of the efficient robust controllers is the  $H_\infty$ , which is created for the Linear-Time-Invariant systems (LTI). In the LPV case, there are no frequency norms. There is a just relative gain in the  $L_2$  sense (in the time-domain). In the LTI case, the  $L_2$  gain and the  $H_\infty$  norm are equal. Hence, the LPV method via the  $L_2$  gain concept can be seen as the extension of  $H_\infty$  synthesis [4].

One of the existed problems on  $H_\infty$  controller design is weighting functions' selection. There are many structures such as Loop-Shaping which is proposed by McFarlane and Glover [5]. Loop-Shaping is the simplest and the most commonly used method [6]. The other common structure is the mixed sensitivity structure [7]. It is very hard to give a general formula for weighting functions that work in every case [8]. This problem depends on experimental skills [9], but Zhou and Doyle proposed guidelines to adjust the weighting functions' parameters for a SISO system [10]. In [11], the authors propose to consider each system as a combination of first and second-order sub-systems, which make it easier to find a good weighting function to be used in  $H_\infty$  control methodology. Yet there is no general methodology to adjust the parameters especially when performance specification is difficult to define, plus it gets more complex for MIMO systems.

\* Corresponding author. E-mail address: [seif.hasseni@univ-biskra.dz](mailto:seif.hasseni@univ-biskra.dz)

On the other side, the Cuckoo Search algorithm is a swarm-based optimization algorithm. It is inspired by a kind of bird (cuckoo) behavior in dumping its eggs into random nests. A cuckoo never builds its own nest but lets other species doing it. The algorithm's essential operations are the global random walk and the local walk. The global walk is similar to Practical Swarm Optimization (PSO) and Simulated Annealing (SA) with Lévy flight transition probability. The local walk is a differential operation with such probability, so it is similar to the Differential Evolution algorithm (DE). Therefore, each algorithm (SA, PSO, and DE) could be considered as a special case of the CS algorithm. Conversely, CS is a good and efficient combination of SA, PSO, and DE in one algorithm [12].

Many researchers have used a nature-inspired optimization algorithm to determine weighting functions' parameters for the LPV system [13-14]. This approach is much more required in the MIMO systems. In this article, the mixed sensitivity structure is applied, and then the Cuckoo Search (CS) algorithm is exploited to find optimal weighting functions' parameters.

The second point investigated in this article is the input saturation. It is a real constraint of the actuators. The actuators can't provide additional effort although the outputs don't reach the reference signal. Such a technique called Anti-Windup has been used in [15]. An Anti-Windup compensator has been designed with the same nature and structure of LFT-LPV plant to reject the input saturation effect [16].

The article's outline is as follows; in Section 2, the representation of the quadrotor attitude model as the LPV system and the synthesis controller are recalled [17]. Section 3 is about a description of the CS algorithm characteristics. Besides that, the optimization of the weighting functions problem for the nominal system (without considering the input saturation) is discussed in Section 3.2. In Section 4, the Anti-Windup compensator design is investigated. The simulation results will be shown in Section 5, and finally, the conclusion in Section 6.

## 2. LFT-LPV Controller Design

In this section, the control of the quadrotor's attitude by the LPV technique is recalled [17].

### 2.1. LFT-LPV representation of quadrotor attitude

As noticed previously, this work focusses on the rotational subsystem of the quadrotor, the framework of three degrees of freedom system is presented on Fig. 1. The angles are the three degrees of freedom; roll ( $\varphi$ ), the rotation around the  $x$  axis; pitch ( $\theta$ ), the rotation around the  $y$  axis; yaw ( $\psi$ ), the rotation around the  $z$  axis.

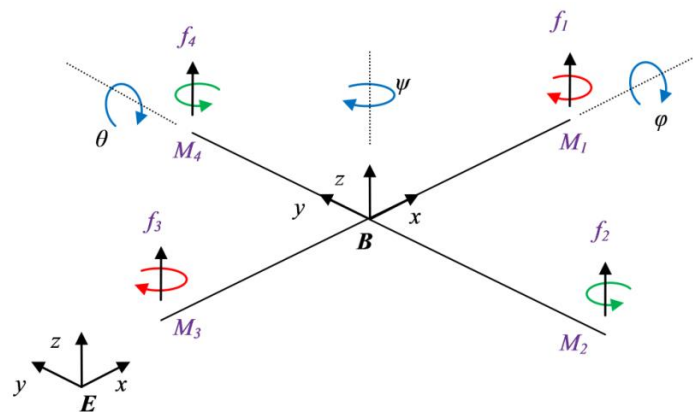


Fig. 1 The geometric structure of the quadrotor

These motions are achieved by four rotors,  $M_i (i=1 \dots 4)$ . The nonlinear dynamic model of the rotational subsystem is shown on Eqs. (1)-(2). It has six states; roll, pitch, yaw;  $[\varphi \ \theta \ \psi]$  and their derivatives;  $[\dot{\varphi} \ \dot{\theta} \ \dot{\psi}]$ . It has three inputs  $[U_\varphi \ U_\theta \ U_\psi]$ .

$$\begin{cases} \ddot{\phi} = \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} - \frac{J_r \Omega_r}{I_x} \dot{\theta} + \frac{l}{I_x} (f_4 - f_2) \\ \ddot{\theta} = \frac{I_z - I_x}{I_y} \dot{\phi} \dot{\psi} + \frac{J_r \Omega_r}{I_y} \dot{\phi} + \frac{l}{I_y} (f_3 - f_1) \\ \ddot{\psi} = \frac{I_x - I_y}{I_z} \dot{\phi} \dot{\theta} + \frac{1}{I_z} (T_1 - T_2 + T_3 - T_4) \end{cases} \quad (1)$$

where

$$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad (2)$$

The considered inputs are expressed as:

$$\begin{cases} U_\phi = f_4 - f_2 = b(\omega_4^2 - \omega_2^2) \\ U_\theta = f_3 - f_1 = b(\omega_3^2 - \omega_1^2) \\ U_\psi = T_1 - T_2 + T_3 - T_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (3)$$

Table 1 OS4 parameters

Parameter	Description	Value	Unit
$l$	Arm length	0.23	m
$b$	Thrust coefficient	$3.13 \cdot 10^{-5}$	$N.s^2$
$d$	Drag coefficient	$7.5 \cdot 10^{-7}$	$N.m.s^2$
$I_x, I_y$	Inertia on x and y axis	$7.5 \cdot 10^{-3}$	$Kg.m^2$
$I_z$	Inertia on z axis	$1.3 \cdot 10^{-2}$	$Kg.m^2$
$J_r$	Rotor inertia	$6 \cdot 10^{-5}$	$Kg.m^2$
$\omega_i$	Rotor speed	[0, 500]	$rad.s^{-1}$

Table 1 shows the model's parameters from the selected project, OS4 [18]. By considering the assumption that the selected system is symmetric ( $I_x = I_y$ ), the yaw model can be linearized as:

$$\ddot{\psi} = \frac{1}{I_z} U_\psi \quad (4)$$

There are two subsystems i.e., the roll-pitch ( $\phi, \theta$ ) subsystem and the linear one, yaw ( $\psi$ ) subsystem. Fig. 2 show the sheme of the quadrotor attitude. After some mathematical operations (for more details see [17]), the LPV model of the nonlinear subsystem (*roll-pitch*) with LFT representation is presented as:

$$\begin{bmatrix} \dot{x} \\ z_p \\ y \end{bmatrix} = \begin{bmatrix} A & B_p & B_1 \\ C_p & D_{pp} & D_{p1} \\ C_1 & D_{1p} & D_{11} \end{bmatrix} \begin{bmatrix} x \\ w_p \\ u \end{bmatrix} \quad (5)$$

and

$$w_p = \Theta z_p \quad (6)$$

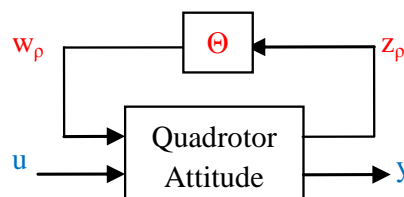


Fig. 2 LFT-LPV scheme of the quadrotor attitude

where the elements of the matrices in Eq. (5) can be obtained by:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

$$B_p = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{I_y - I_z}{I_x} & -\frac{J_r}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{I_z - I_x}{I_y} & \frac{J_r}{I_y} \end{bmatrix} \tag{8}$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ \frac{l}{I_x} & 0 \\ 0 & 0 \\ 0 & \frac{l}{I_y} \end{bmatrix} \tag{9}$$

$$C_p = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{11}$$

$$D_{pp} = 0_{4 \times 4} \tag{12}$$

$$D_{p1} = 0_{4 \times 2} \tag{13}$$

$$D_{1p} = 0_{2 \times 4} \tag{14}$$

$$D_{11} = 0_{2 \times 2} \tag{15}$$

where the state vector  $x = [\varphi \ \dot{\varphi} \ \theta \ \dot{\theta}]^T$ , the input vector  $u = [U_\varphi \ U_\theta]^T$ , the outputs  $y = [\varphi \ \theta]^T$ ,  $z_p$  and  $w_p$  are the inputs and outputs of the parameters block,  $\Theta = \text{diag}(p_1, p_2, p_1, p_2)$ . Table 2 presents the parameter range of roll-pitch subsystem. The yaw subsystem is a simple double integrator system with 2 states  $[\psi \ \dot{\psi}]^T$ , and just one input ( $U_\psi$ ).

Table 2 Varying-parameters range

Parameter	Description	Range
$p_1$	$\dot{\psi}$	$[-2, 2] \text{rad.s}^{-1}$
$p_2$	$\Omega_r$	$[-500, 500] \text{rad.s}^{-1}$

## 2.2. The Controller synthesis

In the  $H_\infty$  controller design, either the LPV subsystem (roll-pitch) or LTI subsystem (yaw) needs to interconnect the system with external inputs (disturbances and noises) and the actuator dynamics.

The rotor's dynamic in the selected project, [18] can be expressed as:

$$Act(s) = \frac{\omega_i}{\omega_{di}} = \frac{0.936}{0.178s + 1} \quad (16)$$

where  $\omega_i$  is the actual rotor speed,  $\omega_{di}$  is the desired rotor speed. From Eq. (3), it is possible to extract the inputs' dynamics.

The actuators' dynamic for each input ( $U_\varphi$ ,  $U_\theta$ ,  $U_\psi$ ) of pitch, roll, and yaw controls can be expressed as:

$$U_a = Act^2(s)U_d \quad (17)$$

where  $U_d$  is the desired control signal generated by the controller, and  $U_a$  is the actual control signal.

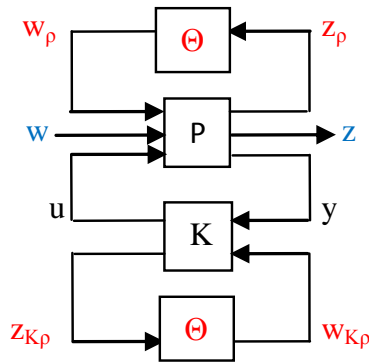


Fig. 3 LFT-LPV representation of closed loop plant-controller

As noticed before, both the plant and controller of the *roll-pitch* subsystem are LPV with LFT representation (Fig. 3). On the contrary, the yaw subsystem is a simple LTI. The augmented plant ( $P$ ) includes actuators dynamics, external inputs, and weighting functions. The plant's state space is presented in Eqs. (18)-(22), and the controller's state space is expressed as:

$$\dot{x} = Ax + B_p w_p + B_1 w + B_2 u \quad (18)$$

$$z_p = C_p x + D_{pp} w_p + D_{p1} w + D_{p2} u \quad (19)$$

$$z = C_1 x + D_{1p} w_p + D_{11} w + D_{12} u \quad (20)$$

$$y = C_2 x + D_{2p} w_p + D_{21} w + D_{22} u \quad (21)$$

$$w_p = \Theta z_p \quad (22)$$

where  $x$  is the states vector;  $u$  is the input vector;  $w$  is the exogenous inputs; disturbances and noises;  $y$  is the measured outputs;  $z$  is the controlled outputs;  $z_p$  and  $w_p$  are inputs and outputs of the block,  $\Theta = \text{diag}(p_1 I_{r1}, p_2 I_{r2}, \dots, p_k I_{rk})$ .

$$\dot{x}_K = A_K x_K + B_{K1} y + B_{Kp} w_{Kp} \quad (23)$$

$$u = C_{K1} x_K + D_{K11} y + D_{K1p} w_{Kp} \quad (24)$$

$$z_{Kp} = C_{Kp} x_K + D_{Kp1} y + D_{Kpp} w_{Kp} \quad (25)$$

$$w_{Kp} = \Theta z_{Kp} \quad (26)$$

where  $x_K$  is the controller states vector,  $y$  is the outputs from the plant,  $u$  is the controller outputs,  $z_{Kp}$ , and  $w_{Kp}$  are inputs and outputs of the block parameter.

After getting the LPV model of attitude in section 2.1, establishing an interconnection between the plant and the different inputs is needed. The inputs are reference signals, disturbances, and noises. The actuator dynamic is also added by the block  $Act^2$  as shown in Eqs. (16)-(17). Furthermore, the weighting functions which represent the desired performance are ordered in mixed sensitivity structure by the errors ( $e_{\varphi, \theta, \psi}$ ) and the input signals ( $U_{\varphi, \theta, \psi}$ ). In this article, three types of external inputs are considered, i.e., reference signals ( $r_{\varphi, \theta, \psi}$ ), disturbances ( $d_{\varphi, \theta, \psi}$ ), and noises ( $n_{\varphi, \theta, \psi}$ ) as presented in Fig. 4. The controller of each subsystem, the LTI controller for the yaw subsystem and the LPV controller for the *roll-pitch* subsystem, will be designed with the same nature as the plant.

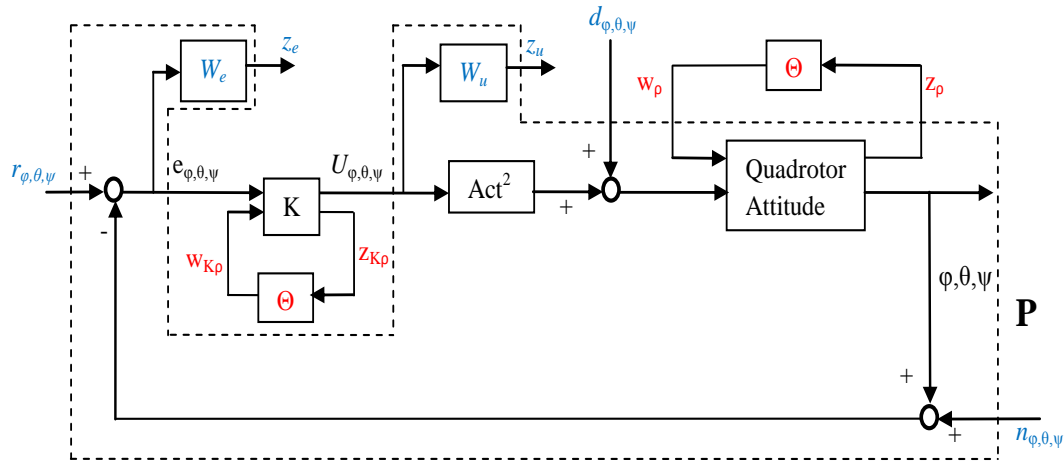


Fig. 4 Closed-loop plant-controller of the quadrotor

### 3. Optimal Weighting Functions

#### 3.1. The cuckoo search algorithm

The Cuckoo Search (CS) algorithm is a swarm-based intelligent algorithm. It was developed by Yang and Deb in 2009 [19] and inspired by the cuckoo's behavior in laying its eggs. A cuckoo doesn't build its own nest but selects another species' nest to let them brood young cuckoos. In this article, a simple way of application of CS is used; a nest corresponds to a cuckoo and also to an egg. The standard CS procedure strike a balance between a local and a global random walk. The global walk is presented by Lévy flight [19] where a cuckoo searches for a new nest. Furthermore, there is a probability  $p_a$  of cases where the host birds discover that the eggs are not theirs, so they abandon the eggs. The cuckoo seeks for a new solution (nest) far from the actual solution; that's the local random walk.

<b>Algorithm 1: Improved Cuckoo Search</b>	
1	Initialize solutions randomly;
2	<b>While</b> max_Generation not meet <b>do</b>
3	Search by fraction ( $p_c$ );
4	Get a cuckoo randomly by Lévy Flight;
5	Evaluate its fitness $F_i$ ;
6	Get a nest randomly ( $j$ )
7	<b>if</b> $F_i$ is better than $F_j$
8	Replace the worse nest( $j$ ) by the trial one;
9	<b>end if</b>
10	Fraction ( $p_a$ ) of worse nests are abandoned and build new ones;
11	<b>end while</b>

The *Improved Cuckoo Search (Algorithm 1)* was developed by Ouaarab et al. [20-21]. Its extension to the standard CS is to add a fraction  $p_c$  of smart cuckoos which seek new solutions in other areas by Lévy flight operation, and can be expressed as:

$$x_i^{t+1} = x_i^t + \alpha L(\lambda) \quad (27)$$

where  $L$  is the Lévy flight function [22],  $\alpha$  is the scaling factor. After getting a new solution (*trial one*), it is compared with a random one ( $j$ ). If the *trial solution* is better than  $j$  (*the worse*), it should be replaced by the new solution. In addition to the fraction  $p_c$ , there are other eggs with probability  $p_a$  which are abandoned by the host birds. Therefore, the cuckoos search for new nests in the local area is:

$$x_i^{t+1} = \begin{cases} x_i^t + s(x_p^t - x_q^t) & \text{if } r > p_a \\ x_i^t & \text{otherwise} \end{cases} \quad (28)$$

where  $x_p$  and  $x_q$  are random solutions different from  $x_i$ , and  $s$  is a scaling factor. The parameters' values are chosen on the base of the recommendation of optimal ranges from experts in the field [20], and are shown in Table 3.

Table 3 Parameters setting of Improved CS

Parameter	Value
<i>max_Generation</i>	20
Population size	50
$p_c$	0.5
$\alpha$	0.1
$\lambda$	1.5
$p_a$	0.25
$s$	0.2

### 3.2. The optimal weighting functions

The good choice of weighting functions significantly affects the system's nominal performance. In [17], the selection of the weighting functions is implemented by trial-and-error. In this article, the implementation is done by an optimal algorithm (Cuckoo Search). The weighting functions are expressed as:

$$W_e = \begin{pmatrix} W_{e\varphi} & 0 & 0 \\ 0 & W_{e\theta} & 0 \\ 0 & 0 & W_{e\psi} \end{pmatrix} \quad (29)$$

$$W_u = \begin{pmatrix} W_{u\varphi} & 0 & 0 \\ 0 & W_{u\theta} & 0 \\ 0 & 0 & W_{u\psi} \end{pmatrix} \quad (30)$$

Because the pitch and the roll have the same dynamics, their functions are equal. The weighting function of error is considered as a first-order filter. The weighting function of the control is considered as a scalar gain, i.e.,

$$W_{e\varphi} = W_{e\theta} = k_e \frac{\tau_1 s + 1}{\tau_2 s + 1} \quad (31)$$

$$W_{u\varphi} = W_{u\theta} = k_u \quad (32)$$

Concerning the *yaw* subsystem, the weighting function of the error is considered as a first-order filter. The weighting function of the control is considered as a simple gain and expressed as:

$$W_{e\psi} = k_e \frac{\tau_1 s + 1}{\tau_2 s + 1} \tag{33}$$

$$W_{u\psi} = k_u \tag{34}$$

The parameters' selection of the *roll-pitch* subsystem and *yaw* subsystem is achieved separately. In the optimization process, the solutions of roll-pitch problem and yaw problem can be obtained by:

$$S_\varphi = [k_e \quad \tau_1 \quad \tau_2 \quad k_u] \tag{35}$$

$$S_\psi = [k_e \quad \tau_1 \quad \tau_2 \quad k_u] \tag{36}$$

It should be noticed that the solutions are presented in real-coded. The optimization process of each problem is solved by the presented CS optimization algorithm in Section 2. The fitness functions are the Integral Square of Error (ISE) is used for the fitness function, that is,

$$f_{roll-pitch} = \int_0^t e_\varphi^2(\tau) d\tau + \int_0^t e_\theta^2(\tau) d\tau \tag{37}$$

$$f_{yaw} = \int_0^t e_\psi^2(\tau) d\tau \tag{38}$$

The fitness (ISE) evolutions of each criterion, Eqs. (37)-(38), are presented in Fig. 5 and Fig. 6. The efficiency of the CS algorithm is shown by the presented convergences where the fitness of the *roll-pitch* subsystem (Fig. 5) reaches its optimal (minimum) at 10<sup>th</sup> generation, and the fitness of the *yaw* subsystem (Fig. 6) reaches its optimal (minimum) at the 8<sup>th</sup> generation. The optimal weighting functions parameters are presented in Table 4.

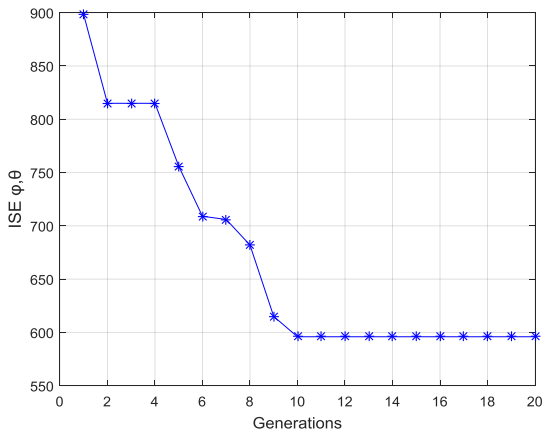


Fig. 5 Fitness evolution of the *roll-pitch* subsystem ( $ISE_\varphi + ISE_\theta$ )

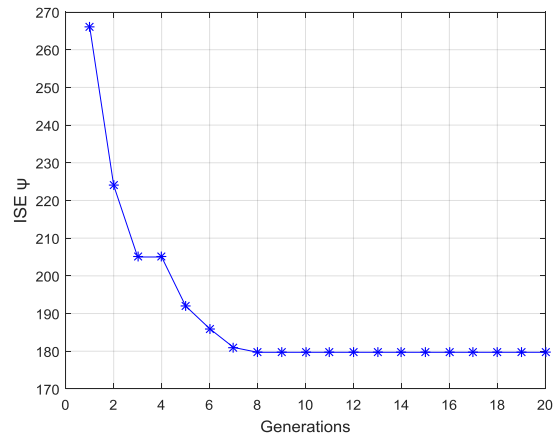


Fig. 6 Fitness evolution of the *yaw* subsystem ( $ISE_\psi$ )

Table 4 Optimal parameters of  $W_e$  and  $W_u$

<i>roll-pitch</i>			
$W_{e\varphi}, W_{e\theta}$	$k_e$	$\tau_1$	$\tau_2$
	12.66	3.112	65.359
$W_{u\varphi}, W_{u\theta}$	$k_u=0.902$		
<i>yaw</i>			
$W_{e\psi}$	$k_e$	$\tau_1$	$\tau_2$
	2.5	0.3388	8.8968
$W_{u\psi}$	$k_u=0.28$		

Fig. 7 presents the inverse sensitivity functions of both subsystems. The presented weights are the error weighting functions  $W_{e\varphi}$  and  $W_{e\psi}$ , when the weights in the control criteria ( $W_{u\varphi}$  and  $W_{u\psi}$ ) are just scalar gains. The inverse of



sensitivity function of the *roll-pitch* subsystem ( $1/W_{e\phi}$ ) is a high pass filter, the steady-state error to step input ( $\varepsilon = 0.078$ ), the overshoot magnitude (peak sensitivity,  $M_s = 1.65$ ), and limit closed-loop bandwidth ( $\omega_b = 0.1937$  rad/s). The inverse of the sensitivity function of the yaw subsystem ( $1/W_{e\psi}$ ) is a high pass filter also, the steady-state error to step input ( $\varepsilon = 0.4$ ), the overshoot magnitude (peak sensitivity,  $M_s = 1.55$ ), and limit closed-loop bandwidth ( $\omega_b = 0.281$  rad/s). These results are in line with the recommended guidelines of [10].

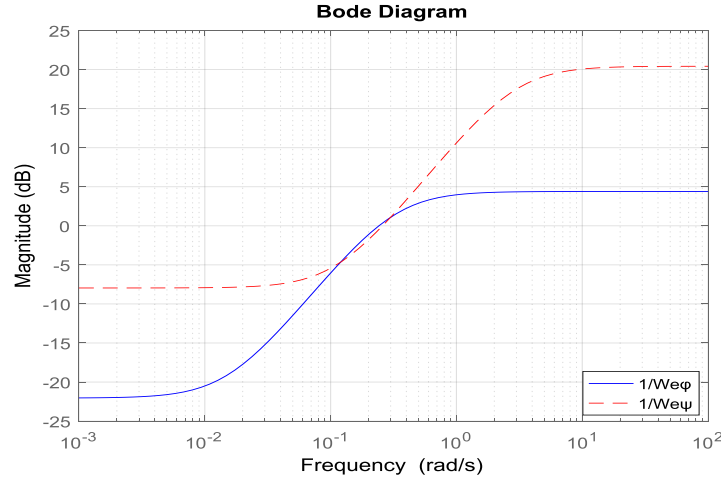


Fig. 7 Desired closed-loop performance

Let  $K_{LPV}$  denote the controller of the *roll-pitch* subsystem, and  $K_{LTI}$  denote the controller of the yaw subsystem, they can be expressed as:

$$K_{LPV} = \begin{bmatrix} A_{K1} & B_{K1} & B_{Kp} \\ C_{K1} & D_{K11} & D_{K1p} \\ C_{Kp} & D_{Kp1} & D_{Kpp} \end{bmatrix} \quad (39)$$

$$K_{LTI} = \begin{bmatrix} A_{K2} & B_{K2} \\ C_{K2} & D_{K2} \end{bmatrix} \quad (40)$$

The augmented controller  $K$  (Fig. 4) is obtained by combining the two controllers, and that is:

$$A_K = \begin{bmatrix} A_{K1} & 0 \\ 0 & A_{K2} \end{bmatrix} \quad (41)$$

$$B_K = \begin{bmatrix} B_{K1} & 0 & B_{Kp} \\ 0 & B_{K2} & 0 \end{bmatrix} \quad (42)$$

$$C_K = \begin{bmatrix} C_{K1} & 0 \\ 0 & C_{K2} \\ C_{Kp} & 0 \end{bmatrix} \quad (43)$$

$$D_K = \begin{bmatrix} D_{K11} & 0 & D_{K1p} \\ 0 & D_{K2} & 0 \\ D_{Kp1} & 0 & D_{Kpp} \end{bmatrix} \quad (44)$$

In this stage, the problem of the optimal weighting functions' parameters is solved, but the problem of the input saturation has not been considered yet. The advantage of this method is that the controller design and the anti-windup compensator synthesis can be carried out independently. The next section is to discuss the anti-windup compensator's design step.

### 4. Anti-Windup Compensator

The system is a mini-drone which has actuators, rotors, with low energy consumption. If the input saturation is imposed, the controller design considering the input saturation is needed to stabilize the system. The saturation function is presented in Eq. (45):

$$sat(u) = \begin{cases} -u_m & \text{if } u < -u_m \\ u & \text{if } -u_m < u < u_m \\ u_m & \text{if } u > u_m \end{cases} \quad (45)$$

In general nonlinear systems, hyperbolic tangent ( $sat(u) \approx \tanh(u)$ ) is used the approximate function of the saturation e.g., the missile control [23] and ship control [24]. Many researchers have focused their attention on the Anti-Windup problem, hence, a modern anti-windup compensator design is developed for the LTI systems called coprime factor [25].

#### 4.1. Theoretical background

The main advantage of this approach is that the controller and the Anti-Windup are designed independently. The Anti-Windup compensator has the same nature and the same representation of the plant and the controller (LFT-LPV).

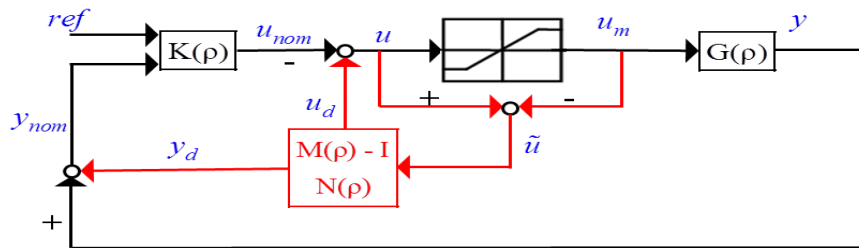


Fig. 8 General scheme of the coprime factor anti-windup

The Anti-Windup compensator is shown in the block diagram (Fig. 8). The windup's effect is rejected by adding  $y_d$ , and the plant can be created as multiplier factors. To design the factors, firstly a matrix  $F(\rho)$  is created according to some LMI conditions. By putting  $M(\rho) = F(\rho) + I$ , it could be possible to generate the other factor  $N(\rho)$ ;  $N(\rho) = G(\rho)M(\rho)$ . So, the plant  $G(\rho)$  could be presented as multiplier factors;  $G(\rho) = N(\rho)M^{-1}(\rho)$ . Therefore, the design of the Anti-Windup compensator is based on determination  $N(\rho)$  and  $M(\rho)$  of  $G(\rho)$ . The key is to find  $F(\rho)$  that minimize the  $L_2$  gain. For more details about computing  $F(\rho)$ , see [16].

Previously, the LPV controller for the nominal system was designed without considering the saturation. As mentioned, the compensator's design is independent of the controller's design. What's next is the proof:

By assuming  $N(\rho) = G(\rho)M(\rho)$  in Fig. 8 it yields,

$$\begin{aligned} u_m &= u - \tilde{u} \\ &= u_{nom} - M(\rho)\tilde{u} \end{aligned} \quad (46)$$

$$\begin{aligned} y_{nom} &= y_d + y \\ &= N(\rho)\tilde{u} + G(\rho)u_m \\ &= N(\rho)\tilde{u} + G(\rho)(u_m - M(\rho)\tilde{u}) \\ &= G(\rho)u_{nom} \end{aligned} \quad (47)$$

where  $u_{nom}$  is the nominal input generated by the controller,  $u_m$  is the actuator input.  $y_{nom}$  is the nominal output.  $y$  is the actual output,  $u_d$  and  $y_d$  are the outputs of the Anti-Windup compensator.

The Anti-Windup state-space is presented as:

$$AW(\rho) = \begin{cases} \dot{\zeta} = (A + FB_u)\zeta + B_\rho w_\rho + B_u \tilde{u} \\ z_\rho = (C_\rho + FD_{\rho u})\zeta + D_{\rho\rho} w_\rho + D_{\rho u} \tilde{u} \\ y_M = M(\rho)\tilde{u} = F\zeta + \tilde{u} \\ y_d = (C_y + FD_{yu})\zeta + D_y w_\rho + D_{yu} \tilde{u} \\ w_\rho = \Theta z_\rho \end{cases} \quad (48)$$

4.2. The compensator design

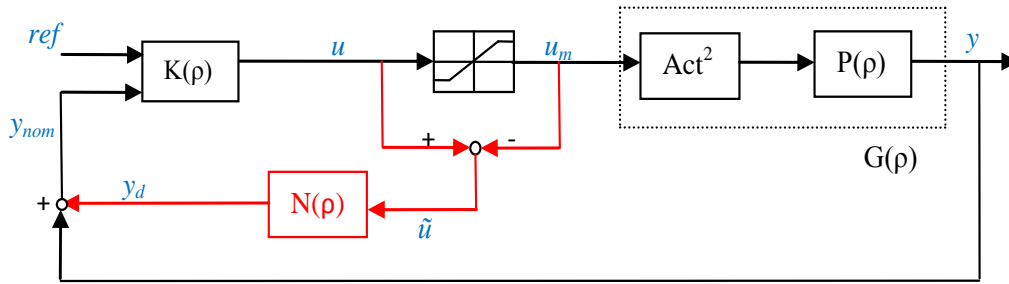


Fig. 9 Implemented Anti-Windup compensator

In the case of the stabilization of the quadrotor attitude, the rejecting of the input saturation's effect is done by the form of Internal Model Control (IMC) [26]. That means  $F(\rho) = 0$ ; so,  $M(\rho) = I$  and  $N(\rho) = G(\rho)$ . The only output of the Anti-Windup compensator is  $y_d$  (Fig. 9). And where the augmented plant,  $G(\rho)$ , contains the plant  $P(\rho)$  which is the LFT-LPV representation of the quadrotor attitude and the actuators function ( $Act^2$ ). From Table 2, the rotor's saturation is  $0 < \omega_i < 500 \text{ rad}\cdot\text{s}^{-1}$ . By combing it with Eq. (3), the saturation values are presented as:

$$sat(.) = \text{diag}(7.825, 7.825, 0.375) \quad (49)$$

5. Simulation Results

To show the efficiency of the designed controller, the controller is applied to the nonlinear model itself (Eq. (1)). The selected initial angles are  $[\varphi_0 \ \theta_0 \ \psi_0] = [\frac{\pi}{3} \ -\frac{\pi}{3} \ \frac{\pi}{3}]^T$  rad, where the objective is to stabilize the quadrotor at its origin (0, 0, 0). In the next simulation, the nominal performance is taken without considering the disturbances. The application of the controller is done for three modes: the nominal one where no saturation is introduced; the second mode where the input saturation is introduced, and the third mode where the input saturation is introduced by adding the Anti-Windup compensator.

Fig. 10 presents the angles' dynamics of the three modes. Fig. 11 and Fig. 12 present the inputs and its zoom with the three modes. The benefit of Anti-Windup compensation is clearly shown. The solid line presents the nominal responses (without saturation). It's possible to observe a swift response and short settling time with impossible input (reaches 100 N). However, when the saturation is introduced without Anti-Windup (dotted line), it's possible to observe a longer settling time and bigger overshoot than the nominal one. By adding the Anti-Windup compensator (dashed line), the responses are better than the obtained by the saturation without Anti-Windup compensator, and the responses are almost identical to those in the nominal responses. In fact, this is the objective of the Anit-Windup compensator.

The efficiency of the robustness  $H_\infty$  controller is shown against the external inputs and the parametric uncertainties with taking the Anti-Windup compensator. Permanent measurement noises of  $\pm 2^\circ (\pm 0.034 \text{ rad})$  are applied on all of the the angles.

The input disturbances are introduced as impulses with  $d_\varphi = 0.1 N$  in  $[6, 6.07] s$ ,  $d_\theta = 0.1 N$  in  $[10, 10.07] s$ , and  $d_\psi = 0.02 N.m$  in  $[14, 14.1] s$ . The simulation results are presented in Fig. 13. It is observed that the stability is kept despite the presence of disturbances and noises.

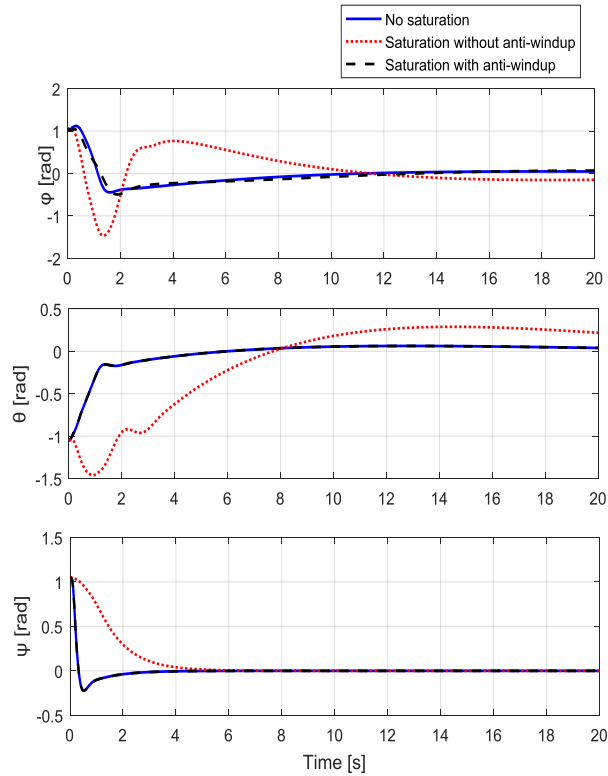


Fig. 10 The angles responses

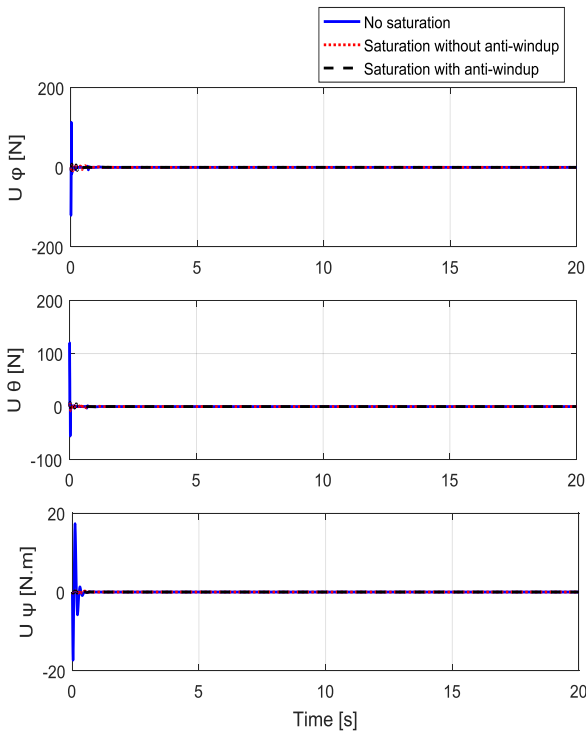


Fig. 11 The plant control inputs

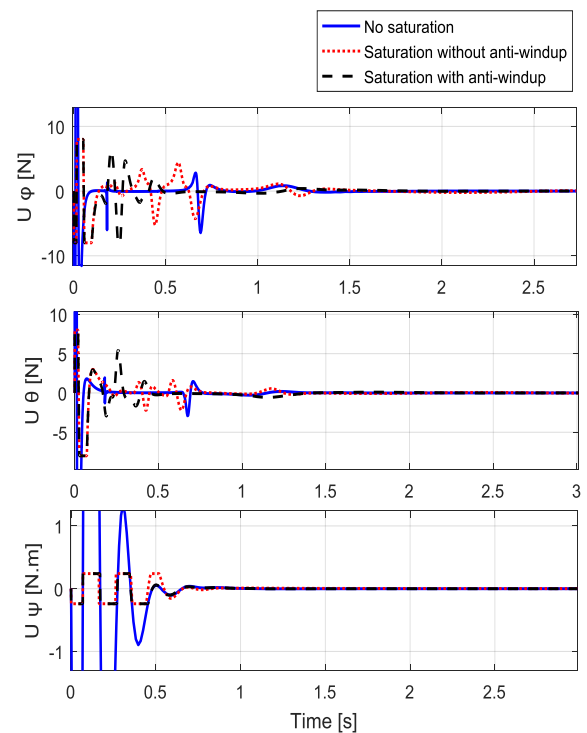


Fig. 12 The plant control inputs (zoom)

By adding 50% of the nominal values of the inertia ( $I_x, I_y, I_z, J_r$ ) after 5 seconds as uncertainties, its simulation is shown in Fig. 14. Compared with the nominal system, it is clearly noticed that there are no big changes in the responses. In addition to disturbances and noises, the system is robust and also against the parametric uncertainties.

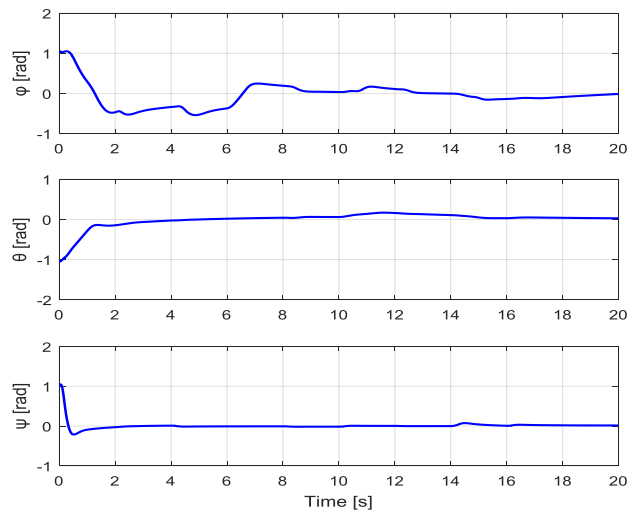


Fig. 13 The angles responses with presence of input disturbances and measurement noises

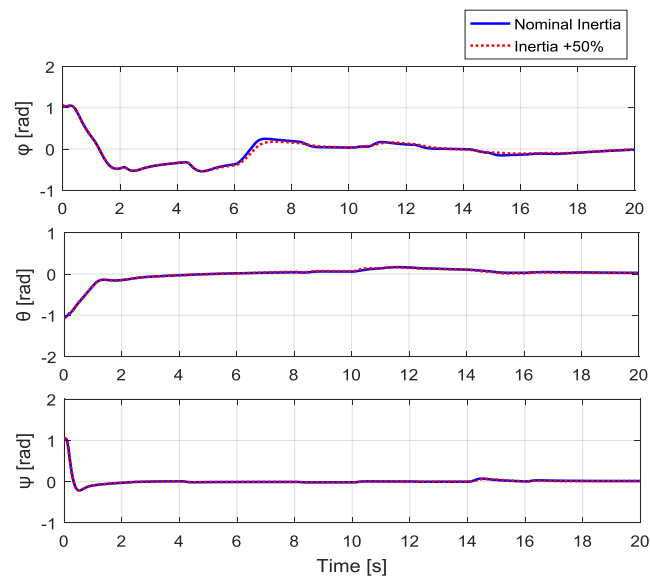


Fig. 14 The angles responses with presence of the uncertainties

## 6. Conclusions

In this article, the robust control of a quadrotor attitude has been carried out by using an LFT-LPV  $H_\infty$  controller. Two issues are discussed. First, the weighting functions selection where an optimization meta-heuristic algorithm is exploited, CS, to provide the parameters of these functions to get optimal performances. Second, the problem of the input saturation which is imposed by the actuators. This problem has been solved by using an Anti-Windup compensator. The advantage of this approach is that the LPV controller and the Anti-Windup compensator could be designed separately without any conflict in the system stability's analysis. The simulation results showed that the system is robust and against the external inputs as well as the uncertainties.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

- [1] L. Carrillo, A. López, R. Lozano, and C. Pégard, Quad Rotorcraft Control. London: Springer-Verlag, 2013.
- [2] N. Vafamand and A. Khayatian, "Model predictive-based reset gain-scheduling dynamic control law for polytopic LPV systems," ISA Transactions, vol. 81, pp. 132-140, 2018.

- [3] H. Behzad, A. Casavola, F. Tedesco, and M. A. Sadrnia, "Use of LPV-LFT unknown input observers for the design of fault tolerant sensor reconciliation schemes," Proc. Iranian Conference on Electrical Engineering, Mashhad, Iran, May 2018, pp. 899-905.
- [4] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled  $H_\infty$  controllers," IEEE Transactions on Automatic Control, vol. 40, no. 5, pp. 853-864, 1995.
- [5] D. McFarlane and K. Glover, Robust Controller Design Using Normalized Coprime Factor Plant Descriptions, vol. 138 of Lecture Notes in Control and Information Sciences, Berlin: Springer-Verlag, 1990.
- [6] A. D. Rosaline and S. Ushakumari, "Robust loop-shaping controller for load frequency control of an uncertain deregulated power system," Journal of the Institution of Engineers (India): Series B, vol. 100, no. 4, pp. 357-369, August 2019.
- [7] L. Yu, C. Che, J. Guo, and H. Wang, "The design of  $H_\infty$  mixed sensitivity controller for fin/tank roll stabilizer," Proc. 35<sup>th</sup> Chinese Control Conference, Chengdu, China, July 2016, pp. 2929-2933.
- [8] S. Skogestad and I. Postlethwaite, Multivariable feedback control, analysis and design, 2<sup>nd</sup> ed. Chichester: John Wiley & Sons, 2005.
- [9] R. W. Beaven, M. T. Wright, and D. R. Seaward, "Weighting function selection in the  $H_\infty$  design process," Control Engineering Practice, vol. 4, no. 5, pp. 625-633, May 1996.
- [10] K. Zhou and J. C. Doyle, Essentials of Robust Control, New Jersey: Prentice Hall, 1998.
- [11] J. Hu, C. Bohn, and H. R. Hu, "Systematic  $H_\infty$  weighting function selection and its application to the real-time control of a vertical take-off aircraft," Control Engineering Practice, vol. 8, no. 3, pp. 241-252, March 2000.
- [12] X. S. Yang, Nature-Inspired Optimizations Algorithms, X.S. Yang (ed.). London: Elsevier, pp. 129-139, 2014.
- [13] A. L. Do, O. Sename, L. Dugard, and B. Soualmi, "Multi-objective optimization by genetic algorithms in  $H_\infty$ /LPV control of semi-active suspension," IFAC Proceedings Volumes, vol. 44, no. 1, pp. 7162-7167, January 2011.
- [14] V. T. Vu, O. Sename, L. Dugard, and P. Gaspar, "Multi objective  $H_\infty$  active anti-roll bar control for heavy vehicles," IFAC PapersOnLine, vol. 50, no. 1, pp. 13802-3807, July 2017.
- [15] J. Fu, Y. Lv, and T. Huang, "Distributed anti-windup approach for consensus tracking of second-order multi-agent systems with input saturation," Systems & Control Letters, vol. 130, pp. 1-6, August 2019.
- [16] E. Prempain, M. C. Turner, and I. Postlethwaite, "Coprime factor based anti-windup synthesis for parameter-dependent systems," Systems & Control Letters, vol. 58, no. 12, pp. 810-817, December 2009.
- [17] S. Hasseni and L. Abdou, "Robust attitude stabilization of a quadrotor by using LFT Linear-Parameter-Varying  $H_\infty$ ," Proc. 26<sup>th</sup> Mediterranean Conference on Control and Automation, Zadar, Croatia, August 2018, pp. 241-246.
- [18] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. thesis, EPFL, Switzerland, 2007.
- [19] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," Proc. World Congress on Nature and Biologically Inspired Computing, Coimbatore, India, December 2009, pp. 210-214.
- [20] A. Ouaarab, B. Ahiod, and X. S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," Neural Computing and Applications, vol. 24, no. 7-8, pp. 1659-1669, June 2014.
- [21] A. Ouaarab, B. Ahiod, and X. S. Yang, "Improved and discrete cuckoo search for solving the travelling salesman problem," Cuckoo Search and Firefly Algorithm, Theory and Applications, X. S. Yang (ed.). Cham, Switzerland: Springer, 2014, pp. 63-84.
- [22] X. S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," Computers & Operations Research, vol. 40, no. 6, pp. 1616-1624, June 2013.
- [23] B. Liu, M. Hou, and Y. Li, "Field-of-view and impact angle constrained guidance law for missiles with time-varying velocities," IEEE Access, vol. 7, pp. 61717-61727, May 2019.
- [24] M. Li, and C. Liu, "New second order sliding mode control design for course-keeping control of ship with input saturation," Proc. International Conference on Advanced Mechatronic Systems, Xiamen, China, March 2018, pp. 462-466.
- [25] P. F. Weston and I. Postlethwaite, "Linear conditioning for systems containing suturing actuators," Automatica, vol. 36, no. 9, pp. 1347-1354, September 2000.
- [26] M. C. Turner, G. Herrmann, and I. Postlethwaite, "Incorporating robustness requirements into antiwindup design," IEEE Transactions on Automatic Control, vol. 52, no. 10, pp. 1842-1855, 2007.

