Parameter tuning of active disturbance rejection control in quad tilt rotor based on particle swarm optimization

Zhigang Wang¹,* , Zhichao Lyu¹ and Jianbo Li²

¹Yangzhou Collaborative Innovation Research Institute CO., LTD. Yangzhou, China
²National key laboratory of Aeromechanics, Nanjing University of Aeronautics and Astronautics, Nanjing, China

*Email: wangzhigang@nuaa.edu.cn

Abstract. The quad tilt rotor (QTR) has complex dynamics characteristics, especially in transition mode. It is difficult to model the QTR dynamics and the environmental factors have a great influence on it. To solve the problem of control in transition mode of QTR, this paper carries out the design of the controller based on active disturbance rejection control (ADRC). ADRC has many parameters to be tuned, and the coupling effect is more serious, so it is very difficult to tune parameters. Particle swarm optimization (PSO) algorithm has faster computing speed and better global search ability in dynamic and multi-objective optimization. So, using PSO algorithm to realize self-tuning of ADRC parameters in ADRC parameters tuning. By comparing the control performance of ADRC before and after optimization, the rationality and effectiveness of ADRC parameters tuning algorithm in QTR are verified in both time domain and frequency domain.

1. Introduction
The quad tilt rotor (QTR) is a novel vehicle[1-7] which combines the characteristics of helicopter and fixed wing aircraft. The QTR has three flight modes: helicopter mode, fixed wing mode and transition mode. Due to the special configuration of quad tilt rotor, the aerodynamic characteristics and stability of quad tilt rotor will change significantly with the change of tilt angle and forward flight speed in the transition mode. The whole process of change is not only time-varying but also high-order nonlinear strong coupling. This brings great difficulty to the design of flight control system.

The classical PID controller is still widely used due to its simplicity and effectiveness. The standard procedure of flight control design is linearizing the flight dynamic around its trimming points based on the small perturbation assumption. Papachristos[8] designed a PID controller, however, due to the nonlinear coupling and the indeterminate complex structure of quad tilt rotor, the control effect of the PID controller is limited. Öner[9] designed a LQR controller. The flight control method based on the optimal control needs to obtain an accurate linear model by linearizing the nonlinear model. However, for the nonlinear model, the control range of the controller is limited to the vicinity of the trimming point. Han[10] proposed an innovative philosophy called ADRC based on the feedback linearization approach, which has remarkable adaptive ability and robustness. However, there are many parameters to be tuned in classical ADRC[11-17], and many parameter tuning methods mainly rely on experience. References developed a linear ADRC, which is easy to design and reduced the parameters, however, the control effect of linear ADRC is limited.
In the developments of optimization theory, particle swarm optimization (PSO) algorithm[18-23], as a new parallel optimization algorithm, has been widely used in the fields of science and engineering. The parameters tuning problem of ADRC can be reduced to a nonlinear dynamic optimal problem with state and control constrains. The problem can be solved by PSO method. PSO algorithm is basically applied to control parameters optimization.

Based on the above research background, in order to solve the problems of QTR time-varying, nonlinear control and ADRC parameter tuning, we use PSO algorithm to self-tune the parameter of ADRC. Firstly, we establish the dynamic model, and then we analyze the ADRC and PSO algorithms. Finally, we simulate and verify the self-tuning algorithm by comparing the ADRC control algorithm before and after optimization based on PSO algorithm in time and frequency domain.

2. Aerodynamic model

The quad tilt rotor with partial tilt wing in this paper is shown in figure 1, including four groups of propellers, front and rear wings, fuselage, elevator, motors, tilting mechanism, undercarriage and flight control system. Both ends of the front and rear wings are designed with a tilt nacelle. The tilt wing is connected to the nacelle and turns with the tilting of the propeller in the nacelle.

The parameters of the tilt quad rotor with partial tilt wing is shown in table 1.

### Table 1. Parameters of the quad tilt rotor with partial tilt wing.

| Parameters            | Value | Unit |
|-----------------------|-------|------|
| Takeoff weight        | 20    | kg   |
| Front wing area       | 0.278 | m²   |
| Front wing length     | 1.4   | m    |
| Back wing length      | 1.8   | m    |
| Back wing area        | 0.5   | m²   |
| Vertical tail area    | 0.07  | m²   |
| Propeller diameter    | 0.51  | 0.95 |

2.1. Propeller aerodynamic model

The propeller is modeled according to the Goldstein vortex theory, and figure 2 shows the velocity and force acting on the blade element. \( \mathbf{R}(\mathbf{m}) \) is radius, \( r \) is the distance from hub center to any point of propeller profile, \( x \) is dimensionless value of \( r \), \( \sigma \) is propeller solidity, \( \omega (\text{rad/s}) \) is rotational speed of propeller, \( V (\text{m/s}) \) is inflow velocity, \( \lambda \) is inflow ratio, \( \phi \) is blade element angle, \( \phi_i \) is blade element inflow angle at propeller tip, \( V_e \) is resultant velocity, \( \omega_2, \omega_3 \) are axial and circumferential induced velocity.

\[
w_i = \frac{B^T}{4\pi rK}
\]
\[
\frac{w_c}{\omega R} = \frac{1}{2} \left( \lambda + \sqrt{\lambda^2 + 4 \frac{w_c}{\omega R} (x - \frac{w_c}{\omega R})} \right)
\]

where, \( \Gamma \) is blade element circulation, \( \kappa \) is coefficient of Goldstein.

\[
F = \frac{2}{\pi} \cos^{-1} \left( \frac{B(1-x)}{2 \sin \phi} \right)
\]

Thrust coefficient \( C_T \), power coefficient \( C_p \) can calculate from equation(4) and(5), \( C_L \) and \( C_D \) is lift coefficient and drag coefficient respectively.

\[
C_T = \frac{\pi^2}{8} \int_{\phi_1}^{\phi_2} \left( C_L \cos(\phi + \alpha) - C_D \sin(\phi + \alpha) \right) d\phi
\]

\[
C_p = \frac{\pi^2}{8} \int_{\phi_1}^{\phi_2} \left( C_L \sin(\phi + \alpha) + C_D \cos(\phi + \alpha) \right) d\phi
\]

The thrust \( T \) and moment \( M \) can calculated from equation(6). Where, \( \Omega / \text{rpm} \) propeller rotational speed, \( \rho \) is atmospheric density and \( D \) is the diameter of propeller.

\[
\begin{align*}
\eta &= \frac{\Omega}{60} \\
T &= C_m \rho n^2 D^4 \\
P &= C_m \rho n^2 D^4 \\
M &= P / \Omega
\end{align*}
\]

According to equations(7) and(8), the force and moment of the hub system are converted to the propeller system:

\[
\begin{bmatrix}
F_{x_{\text{rear}}} \\
F_{y_{\text{rear}}} \\
F_{z_{\text{rear}}}
\end{bmatrix}
= 
\begin{bmatrix}
T \sin \beta_m \\
0 \\
-T \cos \beta_m
\end{bmatrix}
\]

\[
\begin{bmatrix}
M_x \\
M_y \\
M_z
\end{bmatrix}
= 
\begin{bmatrix}
-M \sin \beta_m \\
0 \\
M \cos \beta_m
\end{bmatrix}
\]

The moment caused by the propeller in the fuselage axis system is shown in equation(9).

\[
\begin{bmatrix}
M_{x_{\text{rear}}} \\
M_{y_{\text{rear}}} \\
M_{z_{\text{rear}}}
\end{bmatrix}
= 
\begin{bmatrix}
0 & -z_r & y_r \\
z_r & 0 & -x_r \\
y_r & x_r & 0
\end{bmatrix}
\begin{bmatrix}
F_{x_{\text{rear}}} \\
F_{y_{\text{rear}}} \\
F_{z_{\text{rear}}}
\end{bmatrix}
+ 
\begin{bmatrix}
M_x \\
M_y \\
M_z
\end{bmatrix}
\]

2.2. Wing aerodynamic model

The velocity of the pneumatic center in the free flow area is the sum of the free flow velocity and the velocity caused by the angular velocity, as shown in equation(11).

\[
\begin{bmatrix}
u_{\text{wing,FL}} \\
v_{\text{wing,FL}} \\
w_{\text{wing,FL}}
\end{bmatrix}
= 
\begin{bmatrix}
u \\
v \\
w
\end{bmatrix}
+ 
\begin{bmatrix}
0 & 0 & -z_{\text{wing,FL}} \\
0 & -z_{\text{wing,FL}} & 0 \\
-x_{\text{wing,FL}} & 0 & -x_{\text{wing,FL}}
\end{bmatrix}
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix}
\]

The angle of attack in the free area is shown in equation(12). The angle of attack in the sliding flow area is shown in equation(13).
Thus, the force on the left side of the front wing of the wind axis is shown in equation.

\[
\begin{align*}
\alpha_{\text{wing,FL}} &= \frac{\pi}{2} u_{\text{wing,FL}} = 0 \\
\alpha_i &= \frac{u_{\text{wing,FL}}}{u_{\text{wing,FL}}} \\
\alpha_{\text{wing,FL}} &= \varphi_f + \alpha_i \\
\alpha_{\text{wing,FL}} &= \varphi_f + \alpha_i + \left( \frac{\pi}{2} - \beta_u \right) \\
L_{\text{wing,FL}} &= q \left( \frac{S_{\text{wing,FL}}}{2} C_{L_{\text{wing,FL}}} \right) \\
D_{\text{wing,FL}} &= q \left( \frac{S_{\text{wing,FL}}}{2} C_{D_{\text{wing,FL}}} \right)
\end{align*}
\]

Similarly, the forces and moment on the right side of the front wing are obtained. Compared with the front wing, the rear wing adds lift caused by rudder deflection. The slant angles of the left and right rudder surfaces of the rear wing are defined as \( \delta_{\text{wing,RL}} \) and \( \delta_{\text{wing,RR}} \):

\[
\begin{align*}
\delta_{\text{wing,RL}} &= -\delta_{\text{pack}} + \delta_{\text{val}} \\
\delta_{\text{wing,RR}} &= -\delta_{\text{pack}} - \delta_{\text{val}}
\end{align*}
\]

2.3. Vertical tail aerodynamic model

The velocity of the pneumatic center on the vertical tail is the sum of the free velocity and the velocity caused by the angular velocity, as shown in equation(18). Where, \( x_{VT}, y_{VT}, z_{VT} \) are the coordinate of the relative center of gravity of the fuselage under the fuselage axis.

\[
\begin{align*}
u_{VT} &= u + 0 \quad 0 \quad y_{VT} \\
v_{VT} &= v + 0 \quad 0 \quad y_{VT} \\
w_{VT} &= w + \left( 0 \quad -z_{VT} \quad 0 \right) \\
\end{align*}
\]

The force on the vertical tail of the wind axis is shown in equation(19).

\[
\begin{align*}
L_{VT} &= q S_{VT} C_{L_{VT}} \left( \alpha_{VT} - \alpha_{0,VT} \right) \\
D_{VT} &= q S_{VT} C_{D_{VT}}
\end{align*}
\]

The force and moment on the fuselage in the fuselage axis are shown in equation(20).

\[
\begin{align*}
\begin{bmatrix}
F_{x_{VT}} \\
F_{y_{VT}} \\
F_{z_{VT}} \\
M_{x_{VT}} \\
M_{y_{VT}} \\
M_{z_{VT}}
\end{bmatrix}
= q S_{VT} \begin{bmatrix}
\cos \alpha_{VT} & 0 & -\sin \alpha_{VT} \\
0 & 1 & 0 \\
\sin \alpha_{VT} & 0 & \cos \alpha_{VT} \\
0 & z_{VT} & y_{VT} \\
-\delta_{VT} & 0 & -x_{VT} \\
-\delta_{VT} & 0 & x_{VT}
\end{bmatrix}
\end{align*}
\]

2.4. Fuselage aerodynamic model
The fuselage aerodynamic center velocity is the sum of the free velocity and the velocity caused by the angular velocity, as shown in equation (21). Where, \( x_{\text{fus}} \), \( y_{\text{fus}} \), \( z_{\text{fus}} \) is the coordinate of the fuselage relative to the center of gravity.

\[
\begin{bmatrix}
  u_{\text{fus}} \\
  v_{\text{fus}} \\
  w_{\text{fus}}
\end{bmatrix} =
\begin{bmatrix}
  u \\
  v \\
  w
\end{bmatrix} +
\begin{bmatrix}
  0 & -z_{\text{fus}} & y_{\text{fus}} \\
  z_{\text{fus}} & 0 & -x_{\text{fus}} \\
  -y_{\text{fus}} & x_{\text{fus}} & 0
\end{bmatrix}
\begin{bmatrix}
  \rho \\
  \dot{q} \\
  0
\end{bmatrix}
\]  

(21)

The force and moment on the fuselage under the fuselage axis are shown in equation (22).

\[
\begin{bmatrix}
  F_{\text{x fus}} \\
  F_{\text{y fus}} \\
  F_{\text{z fus}} \\
  M_{\text{x fus}} \\
  M_{\text{y fus}} \\
  M_{\text{z fus}}
\end{bmatrix} =
\begin{bmatrix}
  \cos \alpha_{\text{fus}} & 0 & -\sin \alpha_{\text{fus}} \\
  0 & 1 & 0 \\
  \sin \alpha_{\text{fus}} & 0 & \cos \alpha_{\text{fus}} \\
  0 & -z_{\text{fus}} & y_{\text{fus}} \\
  z_{\text{fus}} & 0 & -x_{\text{fus}} \\
  -y_{\text{fus}} & x_{\text{fus}} & 0
\end{bmatrix}
\begin{bmatrix}
  -D_{\text{fus}} \\
  0 \\
  -L_{\text{fus}} \\
  F_{\text{x fus}} \\
  F_{\text{y fus}} \\
  F_{\text{z fus}}
\end{bmatrix}
\]  

(22)

3. Design of flight control law

Flight control law is the core of the flight control system. The complete control system structure is shown in figure 3, which includes position, speed and attitude controller. Firstly, the velocity correction under the body axis is given by the position controller; Secondly, the attitude correction and collective are given by the speed controller; Finally, the attitude manipulation correction is calculated by the attitude controller.

![Figure 3. The control system of tilt quad rotor.](image)

3.1. Active disturbance rejection controller

ADRC technology does not depend on the model, which can handle various internal uncertainties, and has strong robustness. ADRC consists of three parts; TD, ESO and NLSEF. The structure of active disturbance rejection controller is shown in figure 4. Where \( z_1, z_2, z_3 \) is the output of ESO and \( v_1, v_2 \) is the output of TD.

![Figure 4. ADRC structure.](image)

3.1.1 Tracking differentiator (TD)

TD is tracking differentiator that arranges a transition process for the attitude angle command and obtains its differential signal, that is, the attitude angular rate command:

\[
\begin{align*}
  r_1(k + 1) &= r_1(k) + hr_2(k) \\
  r_2(k + 1) &= r_2(k) + hfr1(r_1(k) - v(k), r_2(k), h)
\end{align*}
\]  

(23)
where, $r_i(k)$ is tracking signal of $v(k)$, $r_i(k)$ is the differential of $v(k)$, $\delta$ is the parameter that determines the speed of tracking, $h$ is step size. The nonlinear function $f_i(x_i, x_{\cdot}, \delta, h)$ expression is as follows:

$$f_i(x_i, x_{\cdot}, \delta, h) = \begin{cases} \delta \text{sign}(a), & |a| > d \\ \frac{\delta a}{d}, & |a| \leq d \end{cases}$$  \hspace{1cm} (24)

$$a = \begin{cases} x_i + \frac{(a_0 - d)}{2} \text{sign}(y), & |y| > d_0 \\ x_i + \frac{y}{h}, & |y| \leq d_0 \end{cases}$$  \hspace{1cm} (25)

$$d = \delta h \\ d_x = h d \\ y = x_i + h x_{\cdot} \\ a_0 = \sqrt{d^2 + 8 \delta |y|}$$  \hspace{1cm} (26)

3.1.2 Extend State Observer (ESO)

The ESO can simultaneously estimate the system states and total external disturbance. The structure of the ESO can be described as:

$$\begin{bmatrix} e \\ \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} = \begin{bmatrix} e \\ \zeta_1 - \beta_0 e \\ \zeta_2 - \beta_0 \text{sat}(e, \alpha, \delta) + b u \\ \zeta_3 - \beta_0 \text{sat}(e, \alpha, \delta) \end{bmatrix}$$  \hspace{1cm} (27)

where, $\beta_{0i} > 0 (i = 1, 2, 3), \alpha_1 = 0.5, \alpha_2 = 0.25$. The function of saturation function is to suppress signal jitter.

$$\text{sat}(e, \alpha, \delta) = \begin{cases} e, & |e| \leq \delta \\ e^+ \text{sgn}(e), & |e| > \delta \end{cases}$$  \hspace{1cm} (28)

3.1.3 Nonlinear State Error Feedback (NLSEF)

NLSEF is a nonlinear state error feedback that nonlinearly combines the deviation between the output of the tracking differentiator and the state of the system. The traditional NLSEF control law of ADRC is as follows:

$$\begin{bmatrix} e_i \\ e_i \\ e_i \end{bmatrix} = \begin{bmatrix} v_i - z_i \\ v_i - z_i \\ v_i - z_i \end{bmatrix}$$  \hspace{1cm} (29)

where, $0 < \alpha_1 < 1, k_p = \beta_1, k_d = \beta_2, e_i$ is error between the command signal and the controlled object position output, $e_i$ is the error between the command differential signal and the controlled object speed output.

3.2. Parameters tuning of ADRC based on PSO

The parameters $\beta_{0i}, \beta_{2i}, \beta_{0i}$ determine the estimated energy of ESO and need to be debugged. In the nonlinear feedback control law, the nonlinear allocation coefficients $\beta, \beta_i$ play a key role in the performance of the controller. In this way, for a class of objects, except $\{\beta_{0i}, \beta_{2i}, \beta_{0i}, \beta_{2i}\}$ need to be debugged frequently, other controller parameters can be set to fixed parameters.

Assuming that the search space has $D$ dimensions, and the total number of particles is $N$, where the particle $i$ is represented as a $D$ dimensional vector:

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), i = 1, 2, \cdots, N$$  \hspace{1cm} (30)

The "flight" velocity of the particle $i$ is also a $D$ dimensional vector, denoted as:

$$V_i = (v_{i1}, v_{i2}, \cdots, v_{iD}), i = 1, 2, \cdots, N$$  \hspace{1cm} (31)
The optimal position searched so far for the particle $i$ becomes the individual extreme value, denoted as:

$$p_{best} = (p_{i1}, p_{i2}, \ldots, p_{id}), i = 1, 2, \ldots, N$$

(32)

The optimal position searched by the whole particle swarm up to now becomes the global extreme value, denoted as:

$$g_{best} = (g_{1}, g_{2}, \ldots, g_{d}), i = 1, 2, \ldots, N$$

(33)

The particle updates its speed and position according to equation (34) and (35):

$$v_{i}(t + 1) = wv_{i}(t) + c_{1}r_{1}(t)\left[p_{i}(t) - x_{i}(t)\right] +$$

$$c_{2}r_{2}(t)\left[p_{best}(t) - x_{i}(t)\right]$$

(34)

$$x_{i}(t + 1) = x_{i}(t) + v_{i}(t + 1)$$

(35)

where, $c_{1}, c_{2}$ are learning factors; $r_{1}, r_{2}$ are uniform random numbers within the range of \([0, 1] ; i = 1, 2, \ldots, D ; v_{i}$ is particle speed, $v_{i} \in [v_{min}, v_{max}]$; $x_{i} \in [x_{min}, x_{max}]$. $w$ is called the inertia factor.

$$w = w_{max} - \frac{(w_{max} - w_{min})t}{T_{max}}$$

(36)

where, $T_{max}$ is the maximum number of evolutions; $w_{min}$ is minimum inertia factor; $w_{max}$ is maximum inertia factor; represents the number of current evolution.

The fitness function is:

$$J = \int_{0}^{t} e(t)dt$$

(37)

where, $e(t)$ is system error.

After the fitness function is determined, the parameters can be optimized. Under the constraint condition, the parameter corresponding to the minimum fitness function value is the optimal controller parameter. The specific steps of the algorithm are as follows:

Step1: initializes related parameters. Particle population size $N$, search space dimension $D$ , learning factor $c_{1}, c_{2}$, maximum and minimum inertia factor $w_{max}, w_{min}$ etc. Randomly initialize the particle position $x_{i}$, speed $v_{i}$, individual optimum $p_{i}$, global optimum $g_{i}$, and ADRC related parameters;

Step2: the position vectors of each particle are taken successively as the controller parameters of ADRC to simulate the system, and then the fitness function values of each particle are calculated by using equation (37);

Step3: the fitness value of each particle is compared with $p_{best}$, and update $p_{best}$ and $g_{best}$;

Step4: the velocity of each particle is updated according to equation (34). If $v_{i}(t + 1) \geq v_{max}$, then $v_{i}(t + 1) = v_{max}$; if $v_{i}(t + 1) \leq v_{min}$, then $v_{i}(t + 1) = v_{min}$;

Step5: the position of each particle is updated according to equation (35). If $x_{i}(t + 1) \geq x_{max}$, then $x_{i}(t + 1) = x_{max}$; if $x_{i}(t + 1) \leq x_{min}$, then $x_{i}(t + 1) = x_{min}$;

Step6: judge whether the iteration times reach the maximum set value. And if not, return to Step2; otherwise, the optimization ends and the final $g_{best}$ is output, namely the optimization parameters of the controller.

**4. Results and discussion**

The controller transfer function can be modeled as:

$$\frac{\Theta(s)}{\delta(s)} = \frac{k_{d}T_{d}s + 1}{s^{2} + Ts + 1}$$

(38)

The command signal of the three attitude is step signal with amplitude of 1, and the external disturbance signal is $1.2 \sin(\pi t)$. We study the parameters tuning results of three attitude angles in different tilt angle (0°, 30°, 60°, 90°). The structure block diagram of the parameters tuning algorithm is shown in figure 5. Parameters before and after optimization are shown in table 2 and table 3. And the
simulation results before and after the parameters tuning of the three attitude angles in different tilt angle are shown in figure 6 and figure 7.

![Structure block diagram of the parameters tuning algorithm](image)

**Figure 5.** The structure block diagram of the parameters tuning algorithm.

**Table 2.** Parameters before optimization.

| Parameter | TD | ESO | NLSEF | PSO |
|-----------|----|-----|-------|-----|
| $h$       | 0.01 |     |       |     |
| $b$       | 50  |     |       |     |
| $\theta$  | 50  | 675 | 375   | 375 |
| $\phi$    | 600 | 100 | 100   | 100 |
| $\psi$    | 10  |     |       |     |

Table 3. Parameters after optimization.

| $\beta_a$ | Attitude | $\beta_{t1}$ | $\beta_{t2}$ | $\beta_{t3}$ | $\beta_{t4}$ |
|-----------|----------|--------------|--------------|--------------|--------------|
| $0^\circ$ | $\theta$ | 190.888      | 39.035       | 1022.524     | 1303.858     | 1811.473     |
|           | $\phi$   | 279.227      | 608.562      | 2001.379     | 7903.306     | 1731.210     |
|           | $\psi$   | 128.873      | 378.369      | 2404.698     | 1034.848     | 1800.009     |
| $30^\circ$| $\theta$ | 168.491      | 172.015      | 1749.921     | 2871.245     | 5557.164     |
|           | $\phi$   | 287.988      | 731.850      | 2354.654     | 5901.947     | 1160.516     |
|           | $\psi$   | 211.471      | 38.083       | 6106.224     | 1816.823     | 1093.225     |
| $60^\circ$| $\theta$ | 116.738      | 189.090      | 8424.483     | 3951.584     | 6144.513     |
|           | $\phi$   | 285.989      | 852.765      | 7166.081     | 4008.517     | 799.840      |
|           | $\psi$   | 196.358      | 264.616      | 1488.540     | 4870.641     | 7940.965     |
| $90^\circ$| $\theta$ | 200.049      | 97.893       | 1446.577     | 4262.878     | 1871.982     |
|           | $\phi$   | 81.470       | 39.610       | 4684.341     | 4239.806     | 2782.550     |
|           | $\psi$   | 156.810      | 18.986       | 2108.375     | 3929.466     | 1155.937     |

The ADRC in figure 6 and 7 represent the unit step response of the system under ADRC control based on manual method. The parameters of controller are shown in table 2. We can see that the output of the system has oscillation characteristics. Although the parameters are adjusted many times, it is difficult to tune appropriate parameters to eliminate the oscillation of the system due to the interaction between the parameters.

However, under the PSO-ADRC optimization control, the system has good dynamic performance from the analysis of time domain and has good reject distraction performance from the analysis of frequency domain. The rise time of the system is faster, and the system basically has no overshoot. The steady-state error of the system is basically zero, and the system oscillation is effectively eliminated. In PSO algorithm, the convergence rate is fast, and the global optimal value is basically reached after 20 iterations. The specific dynamic performance parameters of PSO-ADRC control algorithm are shown in table 4.
Figure 6. Comparison of frequency characteristics.

Figure 7. Response of the system.

Table 4. Dynamic performance parameters of PSO-ADRC control algorithm.

| $\beta_{a}$ | Attitude | $t_s$ (s) | $M_s$ (%) | $e$ | $e_{ret}$ | $T_{ret}$ |
|------------|----------|-----------|-----------|-----|-----------|-----------|
| $\theta$  | 0°       | 0.31      | 1.4       | 0.0001 | 0.3354    | 61        |
| $\psi$    | 0.33     | 0.33      | 0.7       | 0.0001 | 0.3375    | 54        |
| $\theta$  | 0.34     | 0.34      | 0.6       | 0.0001 | 0.3360    | 88        |
| $\psi$    | 0.32     | 0.34      | 0.4       | 0.0001 | 0.3364    | 53        |
| $\theta$  | 0.33     | 0.33      | 0.6       | 0.0001 | 0.3388    | 80        |
| $\psi$    | 0.32     | 0.33      | 0.9       | 0.0001 | 0.3383    | 40        |
| $\theta$  | 0.34     | 0.32      | 0.4       | 0.0001 | 0.3346    | 53        |
| $\psi$    | 0.33     | 0.33      | 0.3       | 0.0001 | 0.3305    | 64        |
| $\theta$  | 0.33     | 0.33      | 0.5       | 0.0001 | 0.3361    | 86        |
| $\psi$    | 0.32     | 0.32      | 0.6       | 0.0001 | 0.3372    | 31        |
| $\theta$  | 0.32     | 0.32      | 0.2       | 0.0001 | 0.3365    | 70        |

5. Conclusion

Compared with other ADRC parameters tuning algorithms, this algorithm is simple and effective. Simulation studies the feasibility of the algorithm in time domain and frequency domain, which can greatly improve the control performance of the system.

Acknowledgments

A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] Beaumier P, Decours J, Lefebvre T. Aerodynamic and Aero-Acoustic Design of Modern Tilt-Rotors : the Onera Experience[J]. 26th International Congress of the Aeronautical Sciences, 2008:1-11.
[2] Peng C, Wang X M, Chen X. Design of Tiltrotor Flight Control System in Conversion Mode Using Improved Neutral Network PID [J]. Advanced Materials Research, 2014, 850-851: 640-3.

[3] Ridgely D B, Mcfarland M B. Tailoring Theory to Practice in Tactical Missile Control [J]. IEEE Control Systems, 2002, 19(6): 49-55.

[4] Song S, Wang W, Lu K, et al. Nonlinear attitude control using extended state observer for tilt-rotor aircraft 2015.

[5] Sato M, Muraoaka K. Flight Controller Design and Demonstration of Quad-Tilt-Wing Unmanned Aerial Vehicle [J]. Journal of Guidance Control & Dynamics, 2014, 38(6): 1-12.

[6] Haixu L, Xiangju Q. Multi-body Motion Modeling and Simulation for Tilt Rotor Aircraft [J]. Chinese Journal of Aeronautics, 2010, 23(4): 415-22.

[7] Song Y, Wang H. Design of Flight Control System for a Small Unmanned Tilt Rotor Aircraft [J]. Chinese Journal of Aeronautics, 2009, 22(3): 250-6.

[8] Papachristos C, Alexis K, Tzes A B T C. Hybrid model predictive flight mode conversion control of unmanned Quad-TiltRottors 2013.

[9] Öner K T, Oner K T, Çetinsoy E, et al. Dynamic Model and Control of a New Quadrotor Unmanned Aerial Vehicle with Tilt-Wing Mechanism [J]. Waset, 2008.

[10] Han J. From PID to Active Disturbance Rejection Control [J]. IEEE Transactions on Industrial Electronics, 2009, 56(3): 900-6.

[11] Wu Q, Sun M, Chen Z, et al. Tuning of Active Disturbance Rejection Attitude Controller for Statically Unstable Launch Vehicle [J]. Journal of Spacecraft and Rockets, 2017, 54(6): 1383-9.

[12] Li Y, Chen Z Q, Sun M W, et al. Attitude control for quadrotor helicopter based on discrete-time active disturbance rejection control [J]. Kongzhi Lilun Yu Yingyong/Control Theory and Applications, 2015, 32(11).

[13] Lou Z, Zhang K, Wang Y, et al. Active Disturbance Rejection Station-Keeping Control for Solar-Sail Libration-Point Orbits [J]. Journal of Guidance, Control, and Dynamics, 2016, 39(8): 1917-21.

[14] Zhang L, Wu R, Wei C, et al. Quaternion-Based Reusable Launch Vehicle Composite Attitude Control via Active Disturbance Rejection Control and Sliding Mode Approach 2017: 1-14.

[15] Xiong H, Yi J, Fan G, et al. Anti-crosswind Autolanding of UAVs based on Active Disturbance Rejection Control 2013: 1-14.

[16] Matras A, Balas M. Modification of Adaptive Disturbance Rejection Control for an Active Magnetic Bearing San Francisco, California: 2012: 1-10.

[17] Torres S, Mehiel E. Nonlinear Direct Adaptive Control and Disturbance Rejection for Spacecraft 2013: 1-13.

[18] Da Y, Xiurun G. An improved PSO-based ANN with simulated annealing technique [J]. Neurocomputing, 2005, 63(none): 527-33.

[19] Chuang L Y, Chang H W, Tu C J, et al. Improved binary PSO for feature selection using gene expression data [J]. Computational Biology & Chemistry, 2008, 32(1): 29-38.

[20] Li B B, Wang L, Liu B. An Effective PSO-Based Hybrid Algorithm for Multiobjective Permutation Flow Shop Scheduling [J]. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 2008, 38(4): 818-31.

[21] Boy M G, Wang C, Wilkinson B E, et al. Double-blind, placebo-controlled, dose-escalation study to evaluate the pharmacologic effect of CP-690,550 in patients with psoriasis [J]. Journal of Investigative Dermatology, 2009, 129(9): 2299-302.

[22] Wang G, Guo J, Chen Y, et al. A PSO and BFO-Based Learning Strategy Applied to Faster RCNN for Object Detection in Autonomous Driving [J]. IEEE Access, 2019, 7(99): 18840-59.

[23] Rout N K, Das D P, Panda G. PSO based adaptive narrowband ANC algorithm without the use of synchronization signal and secondary path estimate [J]. Mechanical Systems & Signal Processing, 2019, 114: 378-98.