Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

# The combination of Q-learning based tuning method and active disturbance rejection control for SISO systems with several practical factors

**Sen Chen** *, **Wenyan Bai** **, **Zhixiang Chen** ***, **Zhiliang Zhao** *

* *School of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710119, Shaanxi, P. R. China.*
** *Beijing Aerospace Automatic, Control Institute, Beijing 100854, P. R. China.*
*** *College of Missile Engineering, Rocket Force University of Engineering, Xi'an 710025, Shaanxi, R P China.*

Abstract: The paper studies the control design and parameter tuning for SISO systems with several practical factors, including nonlinear uncertainty, time delay, input saturation and measurement noise. An active disturbance rejection control design is proposed to actively compensate for the above nonlinear factors. Moreover, an automatic tuning method based on Q-learning is proposed, which is featured with model-free and data-driven properties. By the tentative actions in the proposed Q-learning algorithm, the optimized control parameters can be obtained.

*Keywords:* reinforcement learning, Q-learning, active disturbance rejection control, nonlinear uncertain system, delay, saturation, measurement noise.

## 1. INTRODUCTION

The control science and technology aim at solving the control problem for practical plant, i.e., the design, tuning and realization of the controller. Since plenty of nonlinear factors, including nonlinear uncertainty, input saturation, time delay and measurement noise, exist in the physical plant, the control problem becomes extremely challenging.

In the past few decades, many researches focusing on the control problem under a specific practical situation have been substantially developed. The control design and the closed-loop performance for systems with saturation are rigorously studied in Hu and Lin (2001). Niculescu (2001) discusses the effect of time delay in control systems and proposes several robust approaches for systems with time delay. For systems with nonlinear uncertainties, many active disturbance rejection based designs are innovatively proposed (see Chen et al. (2016); Han (2009)). Although there are numerous successfully practical applications by the above methods (see Su et al. (2004); Sun et al. (2020a); Xue et al. (2017); Sun et al. (2020b)), the control task requires the advanced control design for systems with more complicated situations.

However, for the more complicated model, the less contributions can be made by mathematics. Some studies for systems with several practical factors, i.e., time delay, saturation, noise and uncertainties, have been presented in recent years, such as Chen et al. (2018); Xue et al. (2015); Zhao and Guo (2017). For the sake of rigorous mathematical derivation, the assumptions are conservative in these studies, which greatly limit the available region of control parameters and are hard to be verified in practice. Moreover, these studies only show the qualitative analysis of the closed-loop system, which cannot directly help practitioners tune the controller parameters. It is of urgent need to study an effective control design and an automatic tuning method for systems with several practical factors.

Machine learning has been rapidly developed and drawn more attentions due to the successful applications in image processing, medical diagnosis and competitive game (see LeCun et al. (2015); Mnih et al. (2013); Bar et al. (2015)). Some researchers have attempted to integrate the learning method in control design (see Lillicrap et al. (2015); Karimi et al. (2009)). However, the typical design for SISO systems with several practical factors has not been studied. The main difficulty is establishing the suitable relationship between the conceptions in control task and learning algorithm.

In this paper, an active disturbance rejection control (ADRC) based design is proposed for SISO (single-input single-output) systems with measurement noise, nonlinear uncertainty, delay and input saturation. Moreover, a Q-learning algorithm is innovatively proposed to automatically tune the ADRC's parameters. By the proposed Q-learning algorithm, the optimized controller parameters can be obtained. The simulations for the longitudinal attitude control of a hypersonic aircraft model illustrate

This work was supported by the National Natural Science Foundation of China under Grant No. 61973202 and the Fundamental Research Funds for the Central Universities under Grant No. GK202003008.

the effectiveness of the proposed method. The main contributions of the paper are shown as follows:

(i) An ADRC design is proposed for SISO systems with several practical factors, including uncertainties, input saturation, measurement noise and time delay.

(ii) The tuning method based on Q-learning is presented, which is a model-free and data-driven technology. Moreover, the optimized controller parameters can be obtained by the proposed learning algorithm.

The rest of the paper has the following organization. In Section 2, the control problem for systems with several practical factors is presented. An ADRC design is presented in Section 3. The tuning method based on Q-learning is proposed in Section 4. Section 5 shows the simulation results. The conclusion and future works are presented in Section 6.

## 2. PROBLEM FORMULATION

Consider the following SISO systems with several practical factors:

$$\begin{cases} \dot{X}(t) = f(X(t), sat(u(t-\tau)), t), \\ y(t) = h(X(t), t), \\ y_m(t) = y(t) + n_m, \end{cases} \quad t \geq t_0, \tag{1}$$

where $X \in R^n$ is the state vector, $u \in R$ is the control input, $y \in R$ is the system output to be controlled, $y_m \in R$ is the measured output, $n_m$ is the zero-mean measurement noise, $f$ is the unknown nonlinear dynamics, $h$ is the unknown nonlinear measurement model, $t_0$ is the initial time, $\tau$ is the unknown input delay and the function $sat(\cdot)$ is the input saturation given by the following equation.

$$sat(u) = \begin{cases} s_{\max}, \text{ if } u \geq s_{\max}, \\ u, \text{ if } s_{\min} < u < s_{\max}, \\ s_{\min}, \text{ if } u \leq s_{\min}, \end{cases} \tag{2}$$

where $s_{\max}$ and $s_{\min}$ are constants satisfying $s_{\max} > s_{\min}$.

The control objective is to design the control input $u$ such that the output $y$ can track the time-varying reference signal $y_r(t)$.

We assume that the relative degree of the system (1), which is determined by the control mechanism of the physical plant, is known and invariant:

*Assumption 1.* For the system (1), the relative degree from $u$ to $y$ is $n_r$.

Since there are several practical factors in the system (1), including unknown nonlinear dynamics, nonlinear measurement, measurement noise, time delay and input saturation, it is challenging to design the controller to achieve the tracking objective. Moreover, due to the trade-off of these nonlinear factors, it is difficult to determine the feasible region of the controller's parameters. Optimizing the control parameters for satisfied tracking performance becomes a critical issue.

In the paper, an ADRC design and a tuning method based on Q-learning are presented for the system (1).

## 3. ACTIVE DISTURBANCE REJECTION CONTROL

In this section, an ADRC design is presented.

From Assumption 1, the relative degree of the system (1) is $n_r$, which implies that the minimum number of the integrators from the control input to the controlled output is $n_r$. Hence the following $n_r$-th integrator chain from the control input to the controlled output can be presented, which is equivalent to the system (1).

$$\begin{cases} y^{(n_r)} = f_1(y, y^{(1)}, \cdots, y^{(n_r-1)}, \xi, sat(u(t-\tau)), t), \\ \dot{\xi} = f_2(y, y^{(1)}, \cdots, y^{(n_r-1)}, \xi, t), \end{cases} \tag{3}$$

where $\xi \in R^{n-n_r}$ is the state vector of zero dynamics. In addition, the functions $f_1$ and $f_2$ are determined by $(f, h)$ and the transformation $X = \phi_x(y, y^{(1)}, \cdots, y^{(n_r-1)}, \xi)$, where $\phi_x$ is the local diffeomorphism between $X$ and $[y\ y^{(1)}\ \cdots\ y^{(n_r-1)}\ \xi]^T$ (see Chen et al. (2020) for details).

Assume that the nominal control model of $f_1$ is $\bar{b} \cdot \overline{sat}(u(t-\bar{\tau}))$, where

$$\overline{sat}(u) = \begin{cases} \bar{s}_{\max}, \text{ if } u \geq \bar{s}_{\max}, \\ u, \text{ if } \bar{s}_{\min} < u < \bar{s}_{\max}, \\ \bar{s}_{\min}, \text{ if } u \leq \bar{s}_{\min}, \end{cases} \tag{4}$$

and $(\bar{b}, \bar{\tau}, \bar{s}_{\max}, \bar{s}_{\min})$ are constants.

*Remark 1.* In practice, $\bar{b}$ is commonly obtained by linearizing the physical plant (see Ren et al. (2015)). $\bar{\tau}$ represents for the nominal value of $\tau$. The function $\overline{sat}(\cdot)$ is the nominal design of input saturation. In the paper, these nominal values will be tuned by Q-learning based method.

Then the integrator chain (3) can be reformulated as

$$\begin{cases} y^{(n_r)}(t) = \bar{b} \cdot \overline{sat}(u(t-\bar{\tau})) + f_\Delta(y, y^{(1)}, \cdots, y^{(n_r-1)}, \xi, t), \\ \dot{\xi} = f_2(y, y^{(1)}, \cdots, y^{(n_r-1)}, \xi, t), \end{cases} \tag{5}$$

where $f_\Delta$ represents for the total disturbance, containing unmodeled dynamics, external disturbances and parametric perturbations.

Based on the system (5) and the measured output $y_m$, the following linear extended state observer (ESO) is presented to timely estimate the total disturbance and the derivatives of output.

$$\begin{cases} \dot{\hat{y}}_i(t) = \hat{y}_{i+1}(t) - \beta_i(\hat{y}_1(t) - y_m(t)),\ 1 \leq i \leq n_r - 1, \\ \dot{\hat{y}}_{n_r}(t) = \bar{b} \cdot \overline{sat}(u(t-\bar{\tau})) + \hat{f}_\Delta(t) - \beta_{n_r}(\hat{y}_1(t) - y_m(t)), \\ \dot{\hat{f}}_\Delta(t) = -\beta_{n_r+1}(\hat{y}_1(t) - y_m(t)), \end{cases} \tag{6}$$

where $[\hat{y}_1(t)\ \cdots\ \hat{y}_{n_r}(t)]$ is the estimation for $[y\ \cdots\ y^{(n_r-1)}]$, $\hat{f}_\Delta$ is the estimation for the total disturbance $f_\Delta$ and $[\beta_1\ \cdots\ \beta_{n_r+1}]$ is the ESO's parameter vector satisfying that the polynomial $s^{n_r+1} + \beta_1 s^{n_r} + \cdots + \beta_{n_r+1}$ is Hurwitz. Owing to the method in Yoo et al. (2007), the ESO's parameters can be designed as

$$\beta_i = \phi_{i,\beta}\omega_o^i,\ \phi_{i,\beta} = \frac{(n_r+1-i)!i!}{(n_r+1)!},\ \omega_o \geq 0, \tag{7}$$

where $\omega_o$ is a positive constant to be designed.

Via the estimation from the ESO (6), the control input is designed as follows:

$$u(t) = \frac{y_r^{(n_r)} - \hat{f}_\Delta - \Sigma_{i=0}^{n_r-1} k_i(\hat{y}_{i+1} - y_r^{(i)})}{\bar{b}}, \tag{8}$$

where $k_i$ is the feedback gain to be chosen. Similar with (7), $k_i$ can be simply designed as

$$k_i = \phi_{i,k}\omega_c^i,\ \phi_{i,k} = \frac{(n_r-i)!i!}{n_r!},\ \omega_c \geq 0, \tag{9}$$

where $\omega_c$ is a positive constant to be designed.

For the ADRC design (6)–(9), the parameters

$$\bar{b}, \bar{\tau}, \bar{s}_{\max}, \bar{s}_{\min}, \omega_o, \omega_c \tag{10}$$

need to be tuned for the specific physical plant. The tuning of control parameters is a significant problem for practitioners. In the next section, a Q-learning based tuning method featured with model-free and data-driven properties is proposed.

## 4. Q-LEARNING BASED TUNING METHOD

Reinforcement learning (RL) is a classical method in the field of machine learning. In RL problem, a agent can take several actions to interact with the environment, and a reward is received after the agent takes an action. The RL problem is to find a policy to map the state of the environment to an action which maximizes the long-term rewards. The components of RL problem are shown as follows (Sutton and Barto (2018)):

- State $s$: series of information describing the state of the agent and environment.
- Action $a$: decision made by the agent that will affect the environment.
- Reward $r$: scalar value that determines how close to the objective.
- Policy $\pi$: map from states to actions.
- Action value function $Q^\pi(s,a)$: the expected return, or expected discount reward, when starting as state $s$, taking action $a$, and using policy $\pi$.

The RL problem can be reformulated as finding the optimal policy $\pi^*$ to maximize the action value function for all states and actions, i.e., $Q^{\pi^*}(s,a) = \max_\pi Q^\pi(s,a)$.

When knowing transition probabilities and rewards, the RL problem can be solved by the Bellman equation. However, these model information is not available in most cases, especially the tuning problem. Thus, the paper focus on Q-learning, which is a model-free and data-driven learning algorithm. Q-learning is one of the widest used temporal difference algorithm, which updates the estimation for the action value function by each tentative action. The updating law of the action value function at the step $k$ is presented as follows:

$$\begin{aligned} Q(s_k,a_k) \leftarrow & (1-\alpha_Q)Q(s_k,a_k) \\ & + \alpha_Q(r(s_k,a_k) + \gamma_Q \max_{a'} Q(s_{k+1},a')), \end{aligned} \tag{11}$$

where $\alpha_Q \in (0,1)$ is the learning rate and $\gamma_Q \in (0,1)$ is the discounted factor. With (11), the Q-learning algorithm is shown in Algorithm 1.

*Remark 2.* With the smaller $\alpha_Q$, the updating law is dependent more on the previous learning result $Q(s_k,a_k)$ rather than the immediate reward $r(s_k,a_k)$. The larger $\gamma_Q$ leads to the more trust on $\max_{a'} Q(s_{k+1},a')$, which can be regarded as the remembered reward.

Next, we design the states, actions and reward for the tuning problem of ADRC's parameters $(\bar{b}, \bar{\tau}, \bar{s}_{\max}, \bar{s}_{\min}, \omega_o, \omega_c)$.

Let $\varepsilon_r$ be a small positive. For the reference signal $y_r$, the $\varepsilon_r$-neighbourhood divides the state plant into the following three regions:

**Algorithm 1** Q-learning algorithm.

**Input:** learning rate $\alpha_Q$, discounted factor $\gamma_Q$, initial state $s_1$.
**Output:** action value function $Q$.
1: Initialize $k=1$, $Q(s,a)=0$, $\forall(s,a)$.
2: **while** the objective is not achieved **do**
3: Generate the action $a_k$ by a policy $\pi$ (e.g. $\varepsilon$-greedy policy).
4: Observe the new state $s_{k+1}$ and the reward $r_k$.
5: Update $Q(s_k,a_k)$ with (11).
6: $k \leftarrow k+1$.
7: **return** $Q$.

$$\begin{aligned} &\text{Region A: } R_A \triangleq \{y|\ y > y_r + \varepsilon_r\}, \\ &\text{Region B: } R_B \triangleq \{y|\ |y - y_r| \le \varepsilon_r\}, \\ &\text{Region C: } R_C \triangleq \{y|\ y < y_r - \varepsilon_r\}. \end{aligned} \tag{12}$$

We select a time series $\{t_i\}_{i=1}^{n_s}$ where $n_s$ is a positive integer and $t_{i+1} > t_i > t_0$ for $1 \le i \le n_s$. The state in Q-learning is designed as $s \triangleq [s(1)\ \cdots\ s(n_s)] \in R^{n_s}$, where $s(i)$ satisfies

$$s(i) = \begin{cases} 1, & \text{if } y(t_i) \in R_A, \\ 0, & \text{if } y(t_i) \in R_B, \\ -1, & \text{if } y(t_i) \in R_C. \end{cases} \tag{13}$$

Fig. 1 shows the sketch for designing the state.

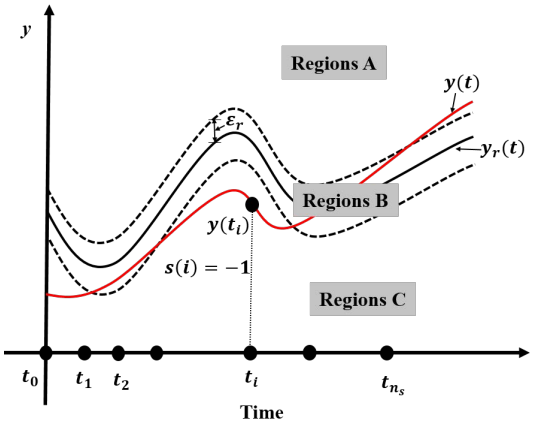


Figure 1. Sketch for designing state in Q-learning.

*Remark 3.* As the number $n_s$ increases, the state $s$ can sufficiently describe the error between $y(t)$ and $y_r(t)$, which contains the information of classical performance index such as overshoot, rise time and settling time. In practice, the time node $t_i$ can be selected as the critical time at which the system faces vast uncertainties.

The actions in Q-learning are designed as adjusting the ADRC's parameters. In each action, the ADRC's parameters are updated by

$$pr \leftarrow pr + \Delta pr, \quad pr = \bar{b}, \bar{\tau}, \bar{s}_{\max}, \bar{s}_{\min}, \omega_o, \omega_c, \tag{14}$$

where $pr$ is the symbol for the ADRC's parameters and $\Delta pr$ represents for the corresponding variation. The action set is shown in Tab. 1, where $\delta\bar{b}$, $\delta\bar{\tau}$, $\delta\bar{s}_{\max}$, $\delta\bar{s}_{\min}$, $\delta\omega_o$ and $\delta\omega_c$ are positive constants.

The reward function describes the tracking performance of the current ADRC's parameters. In the paper, the reward is selected as the minus of the integral of the square tracking error (MISE):

Table 1. Action set.

| Action | $\Delta\bar{b}$ | $\Delta\bar{\tau}$ | $\Delta\bar{s}_{\max}$ | $\Delta\bar{s}_{\min}$ | $\Delta\omega_o$ | $\Delta\omega_c$ |
|---|---|---|---|---|---|---|
| $a_1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $a_2$ | $+\delta\bar{b}$ | 0 | 0 | 0 | 0 | 0 |
| $a_3$ | $-\delta\bar{b}$ | 0 | 0 | 0 | 0 | 0 |
| $a_4$ | 0 | $+\delta\bar{\tau}$ | 0 | 0 | 0 | 0 |
| $a_5$ | 0 | $-\delta\bar{\tau}$ | 0 | 0 | 0 | 0 |
| $a_6$ | 0 | 0 | $+\delta\bar{s}_{\max}$ | 0 | 0 | 0 |
| $a_7$ | 0 | 0 | $-\delta\bar{s}_{\max}$ | 0 | 0 | 0 |
| $a_8$ | 0 | 0 | 0 | $+\delta\bar{s}_{\min}$ | 0 | 0 |
| $a_9$ | 0 | 0 | 0 | $-\delta\bar{s}_{\min}$ | 0 | 0 |
| $a_{10}$ | 0 | 0 | 0 | 0 | $+\delta\omega_o$ | 0 |
| $a_{11}$ | 0 | 0 | 0 | 0 | $-\delta\omega_o$ | 0 |
| $a_{12}$ | 0 | 0 | 0 | 0 | 0 | $+\delta\omega_c$ |
| $a_{13}$ | 0 | 0 | 0 | 0 | 0 | $-\delta\omega_c$ |

**Algorithm 2** Q-learning based tuning method.

**Input:** the initial parameters $(\bar{b}, \bar{\tau}, \bar{s}_{\max}, \bar{s}_{\min}, \omega_o, \omega_c)$.
**Output:** the action value function $Q$ and the optimized parameters $(\bar{b}^*, \bar{\tau}^*, \bar{s}^*_{\max}, \bar{s}^*_{\min}, \omega^*_o, \omega^*_c)$.

1: Initialize $k = 1$, $Q(s, a) = 0$, $\forall(s, a)$.
2: **while** the satisfied tracking performance is not achieved **do**
3: Generate the action $a_k$ from Tab. 1 by a policy $\pi$ (e.g. $\varepsilon$-greedy policy).
4: Do an experiment or simulation.
5: Observe the new state $s_{k+1}$ (13) and the reward $r_k$ (15).
6: Update $Q(s_k, a_k)$ with (11).
7: $k \leftarrow k + 1$.
8: **return** $Q$ and $(\bar{b}, \bar{\tau}, \bar{s}_{\max}, \bar{s}_{\min}, \omega_o, \omega_c)$.

$$r = -\int_{t=t_0}^{t_T} (y(\theta) - y_r(\theta))^2 d\theta, \tag{15}$$

where $t_T > t_0$ is the stop time.

With the designed states, actions and reward function, the Q-learning based tuning method is shown in Algorithm 2. By Algorithm 2, the optimized parameters can be obtained. Moreover, the tuning logic, i.e., action value function Q, is learned, which provides the strategy of tuning parameters for the better closed-loop performance based on the information for the current simulation or experimental result.

*Remark 4.* The most noticeable difference with the existing iterative learning control and adaptive dynamic programming is that the proposed Q-learning tuning method is model-free and data-driven, which is capable to optimize the controller parameters despite complicated physical limitations, i.e., uncertain nonlinear dynamics, time delay, input saturation and measurement noise.

## 5. SIMULATION

In this section, the simulation for the longitudinal attitude control of a hypersonic aircraft model is presented.

From Wang and Stengel (2000), the following longitudinal dynamics of the generic hypersonic aircraft, including an inverse-square-law gravitational model and the centripetal acceleration for the non-rotating earth, is presented as

Table 2. Parameter description.

| Parameters | Physical definitions |
|---|---|
| $a$ | speed of sound, $ft/s$ |
| $\bar{c}$ | reference length, $ft$ |
| $D$ | drag, $lbf$ |
| $H$ | altitude, $ft$ |
| $J_z$ | moment of inertia, $slug \cdot ft^2$ |
| $L$ | lift, $lbf$ |
| $M_z$ | pitching moment, $lbf \cdot ft$ |
| $m$ | mass, $slugs$ |
| $q$ | pitch rate, $rad/s$ |
| $r$ | radius of the Earth, $ft$ |
| $S$ | reference area, $ft^2$ |
| $T$ | thrust, $lbf$ |
| $V$ | velocity, $ft/s$ |
| $\alpha$ | angle of attack, $rad$ |
| $\alpha_t$ | angle of attack at trim condition, $rad$ |
| $\theta$ | flight-path angle, $rad$ |
| $\delta_e$ | elevator deflection, $rad$ |
| $\delta_T$ | throttle setting, $18.3 \times 100\%$ |
| $\mu$ | gravitational constant, $1.39 \times 10^{16}\ ft^3/s^2$ |
| $\rho$ | density of air, $slug/ft^3$ |
| $C_D$ | drag coefficient |
| $C_L$ | lift coefficient |
| $C_T$ | thrust coefficient |
| $M$ | Mach number |
| $c_e$ | coefficient of elevator deflection, 0.0292 |
| $v_i$ | coefficients of uncertainties |
| $C_{M,\omega_z}$ | pitching moment coefficient by pitch rate |
| $C_{M,\alpha}$ | pitching moment coefficient by angle of attack |
| $C_{M,\delta_e}$ | pitching moment coefficient by elevator deflection |

$$\begin{cases} \dot{V} = \dfrac{T\cos\alpha - D}{m} - \mu\dfrac{\sin\theta}{r^2}, \\ \dot{\theta} = \dfrac{L + T\sin\alpha}{mV} - \dfrac{(\mu - V^2 r)\cos\theta}{Vr^2}, \\ \dot{H} = V\sin\theta, \\ \dot{\alpha} = \omega_z - \dot{\theta}, \\ \dot{\omega_z} = \dfrac{M_z}{J_z}, \end{cases} \tag{16}$$

where

$$\begin{aligned} &L = QSC_L, \quad D = QSC_D, \quad T = QSC_T, \\ &M_z = QS\bar{c}[C_{M,\omega_z} + C_{M,\delta_e} + C_{M,\alpha}], \\ &Q = \frac{1}{2}\rho V^2, \quad r = H + 20903500. \end{aligned} \tag{17}$$

Since the flying environment frequently varies, the aerodynamic coefficients $C_a(a = L, D, T, M)$ have uncertainties compared with the nominal value obtained from the wind tunnel test or numerical simulations. The 28 coefficients of uncertainties $v_i(i = 1, 2, \cdots, 28)$ are multiplied by the system coefficients to fit the biases around the nominal cruising condition. The aerodynamic coefficients $C_a(a = L, D, T, M)$ and inertial data are presented as follows:

$$m = m_0 v_1, \quad J_z = J_{z,0} v_2, \quad S = S_0 v_3, \quad \bar{c} = \bar{c}_0 v_4, \quad M = V/a,$$
$$\rho = 57.12 v_8 e^{-H}, a = v_5(8.99 \times 10^{-9} v_6 H^2 - 9.16 \times 10^{-4} v_7 H + 996),$$
$$C_L = v_9\alpha(0.493 + 1.91 v_{10}/M),$$
$$C_D = 0.0082 v_{11}(171 v_{12}\alpha^2 + 1.15 v_{13}\alpha + 1) \times (0.0012 v_{14} M^2 - 0.054 v_{15} M + 1),$$
$$C_T = \begin{cases} 0.0105 v_{16}[1 - 164 v_{17}(\alpha - \alpha_t)^2] \\ \quad \times(1 + 17 v_{18}/M)(1 + 0.15 v_{19})\delta_T, \text{ if } \delta_T < 1, \\ 0.0105 v_{16}[1 - 164 v_{17}(\alpha - \alpha_t)^2] \\ \quad \times(1 + 17 v_{18}/M)(1 + 0.15 v_{19}\delta_T), \text{ if } \delta_T \geq 1, \end{cases}$$
$$C_{M,\alpha} = v_{20} 10^{-4}(0.06 - e^{-v_{21} M/3})(-v_{22} 6565\alpha^2 + v_{23} 6875\alpha + 1),$$

$$C_{M,\omega_z} = (\bar{c}/2V)\omega_z v_{24}(-v_{25}0.025M + 1.37) \times (-v_{26}6.83\alpha^2 + v_{27}0.303\alpha - 0.23),$$
$$C_{M,\delta_e} = v_{28}c_e(sat(\delta_e(t-\tau)) - \alpha).$$

The physical definitions of the parameters in the system (16) are shown in Tab. 2.

The control objective is to design the angle of elevator deflection $\delta_e$ such that the angle of attack can track the reference signal $\alpha_r$. In this simulation, we consider the down-phase control problem, where $\alpha_r$ satisfies

$$\alpha_r = \begin{cases} 0.0524\ (rad), & \text{if } 0 \le t \le 30, \\ 0.0022t^2 - 0.2774t + 9.3236\ (rad), & \text{if } 30 < t \le 80. \end{cases}$$

Due to the physical limitation of elevator deflection, the saturation function is clearly known:

$$sat(\delta_e) = \begin{cases} \frac{25\pi}{180}\ (rad), & \text{if } \delta_e \ge \frac{25\pi}{180}, \\ \delta_e\ (rad), & \text{if } -\frac{25\pi}{180} < \delta_e < \frac{25\pi}{180}, \\ -\frac{25\pi}{180}\ (rad), & \text{if } \delta_e \le -\frac{25\pi}{180}. \end{cases} \quad (18)$$

The exact input delay $\tau$ is unknown. Combined with the analysis in Bai et al. (2017), the parameters in the nominal control model $\bar{b} \cdot \overline{sat}(u(t-\bar{\tau}))$ satisfy

$$\bar{b} = (\frac{28.56c_eS_0\bar{c}_0}{J_{z,0}})e^{-H}V^2, \bar{\tau} = 0, \bar{s}_{\max} = \frac{25\pi}{180}, \bar{s}_{\min} = -\frac{25\pi}{180}, \quad (19)$$

where $(J_{z,0}, S_0, \bar{c}_0) = (7 \times 10^6, 3603, 80)$ are the nominal value of $(J_z, S, \bar{c})$. Moreover, from the analysis in Bai et al. (2017), the relative degree from the elevator deflection to the angle of attack is 2. The ADRC controller (6)–(9) is designed with $n_r = 2$.

Based on the known saturation function (18), we uses the proposed Q-learning to get the optimized control parameters $(\bar{b}, \bar{\tau}, \omega_c, \omega_o)$. Since the altitude $H$ and the velocity $V$ are measurable and time-varying, the update of $\bar{b}$ is designed as $\pm\delta\bar{b}\cdot e^{-H}V^2$. The initial control parameters $(\omega_c, \omega_o)$ are selected with the same values in Bai et al. (2017):

$$\omega_c = 5, \quad \omega_o = 30. \quad (20)$$

Moreover, due to Bai et al. (2017), the following three typical variations of parameters are considered:

$$\begin{aligned} &\text{Case 1: } v_i = 0,\ 1 \le i \le 28, \\ &\text{Case 2: } v_2 = 0.2,\ v_3 = v_4 = v_8 = v_{28} = -0.2, \\ &\qquad v_j = 0, j \ne 2, 3, 4, 8, 28, \\ &\text{Case 3: } v_2 = -0.2,\ v_3 = v_4 = v_8 = v_{28} = 0.2, \\ &\qquad v_j = 0, j \ne 2, 3, 4, 8, 28, \end{aligned} \quad (21)$$

where Case 1 represents for the nominal situation, Case 2 represents for the situation that the system (16) has the weak control ability and Case 3 represents for the situation that the system (16) has the strong control ability. In the simulations, the sum of the MISEs for Cases 1-3 are used as the reward function $r$. The states in Q-learning depend on the trajectory of Case 1 with the time series $\{t_1 = 2(s), t_2 = 32(s), t_3 = 61(s), t_4 = 80(s)\}$ and the error bound $\varepsilon_r = 0.02$. The $\varepsilon$-greedy policy with the probability $\varepsilon = 0.1$ is utilized. The parameters in Q-learning are designed as follows:

$$\alpha_Q = \gamma_Q = 0.9,$$
$$\delta\bar{b} = \frac{28.56c_eS_0\bar{c}_0}{10J_z},\ \delta\bar{\tau} = 0.01,\ \delta\omega_c = 0.2,\ \delta\omega_o = 1.$$

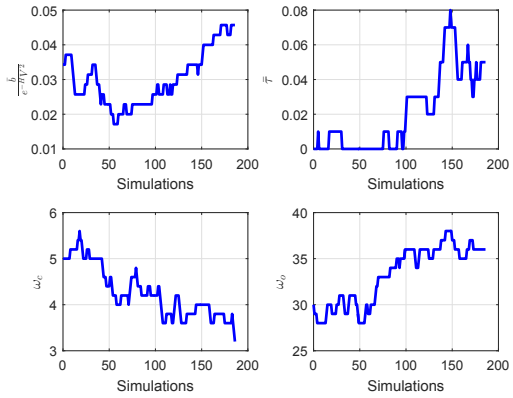


Figure 2. The updating process of control parameters.

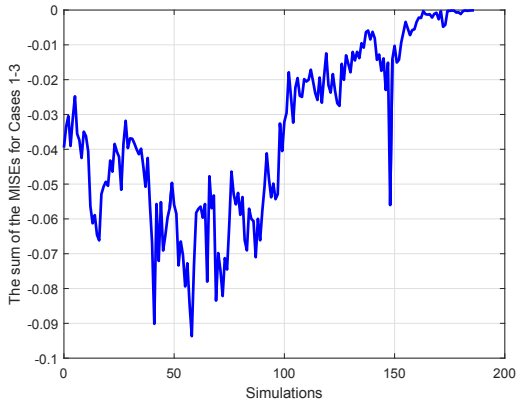


Figure 3. The sum of the MISEs for Cases 1-3.

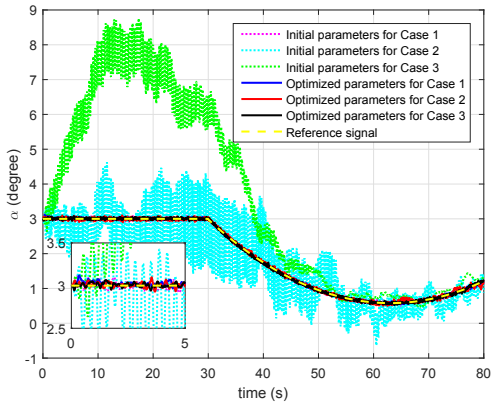


Figure 4. The tracking results for the initial parameters and optimized parameters for Cases 1-3.

The initial conditions of the system (16) are chosen as $(M_0, H_0, V_0, \theta_0, \alpha_t) = (15, 110000, 15060, -0.1, 0.0315)$. The time delay is set as $\tau = 0.05(s)$ and the measurement noise of the angle of attack obeys the normal distribution $N(0, (\frac{0.04*\pi}{180})^2)$.

The learning process of control parameters is shown in Fig. 2 and Fig. 3. It is significant to point out that the proposed tuning method tries to increase the nominal value of the delay $\bar{\tau}$ around the 150-th trial, while the reward shown in Fig. 3 decreases. Then it is automatically learned that the larger $\bar{\tau}$ means the worse performance. Final, the optimized control parameters are shown as follows:

$$\bar{b} = 0.0457e^{-H}V^2, \quad \bar{\tau} = 0.05, \quad \omega_c = 3.2, \quad \omega_c = 36. \quad (22)$$

The simulations for the initial parameters (19)–(20) and the optimized parameters (22) are shown in Fig. 4. From Fig. 4, the optimized parameters (22) greatly improve the tracking accuracy for Cases 2-3, which illustrates the effectiveness of the proposed automatic tuning method.

## 6. CONCLUSION AND FUTURE WORKS

The paper studies the control problem for SISO systems with several practical factors, including nonlinear uncertainty, input saturation, time delay and measurement noise. To overcome the challenges caused by these nonlinear factors, an ADRC design is proposed. Moreover, the tuning of control parameters is agonisingly difficult for practitioners due to these nonlinear factors. An automatic tuning method based on Q-learning is presented. By suitably establishing the relationship between the conceptions in control task and learning algorithm, the control parameters can be automatically optimized by several trials.

The following future works will be considered:

(1) The theoretical analysis of the paper lacks due to the complicated practical factors. We will theoretically study the effectiveness of learning algorithm in the future.

(2) The designed Q-learning is innovative but subjective. Since there are plenty of feasible designs, how to construct the learning algorithm with the suitable relationship between conceptions in control task and learning algorithm is a critical job.

## REFERENCES

Bai, W., Chen, S., Lu, K., Huang, W., and Liu, P. (2017). Stable and robust control design for a benchmark hypersonic aircraft model. In *2017 36th Chinese Control Conference (CCC)*, 3443–3448. doi: 10.23919/ChiCC.2017.8027891.

Bar, Y., Diamant, I., Wolf, L., and Greenspan, H. (2015). Deep learning with non-medical training used for chest pathology identification. In *Medical Imaging 2015: Computer-Aided Diagnosis*, volume 9414, 94140V. International Society for Optics and Photonics.

Chen, S., Bai, W., Hu, Y., Huang, Y., and Gao, Z. (2020). On the conceptualization of total disturbance and its profound implications. *Science China Information Sciences*, 63(2), 1–129201.

Chen, S., Xue, W., Zhong, S., and Huang, Y. (2018). On comparison of modified adrcs for nonlinear uncertain systems with time delay. *Science China Information Sciences*, 61(7), 70223.

Chen, W., Yang, J., Guo, L., and Li, S. (2016). Disturbance-observer-based control and related methods: An overview. *IEEE Transactions on Industrial Electronics*, 63(2), 1083–1095.

Han, J. (2009). From PID to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, 56(3), 900–906.

Hu, T. and Lin, Z. (2001). *Control systems with actuator saturation: analysis and design*. Springer Science & Business Media.

Karimi, A., Eftekharnejad, S., and Feliachi, A. (2009). Reinforcement learning based backstepping control of power system oscillations. *Electric Power Systems Research*, 79(11), 1511–1520.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Niculescu, S.I. (2001). *Delay effects on stability: a robust control approach*, volume 269. Springer Science & Business Media.

Ren, B., Zhong, Q., and Chen, J. (2015). Robust control for a class of nonaffine nonlinear systems based on the uncertainty and disturbance estimator. *IEEE Transactions on Industrial Electronics*, 62(9), 5881–5888.

Su, J., Ma, H., Qiu, W., and Xi, Y. (2004). Task-independent robotic uncalibrated hand-eye coordination based on the extended state observer. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4), 1917–1922.

Sun, L., Jin, Y., and You, F. (2020a). Active disturbance rejection temperature control of open-cathode proton exchange membrane fuel cell. *Applied Energy*, 261, 114381.

Sun, L., Li, G., Hua, Q., and Jin, Y. (2020b). A hybrid paradigm combining model-based and data-driven methods for fuel cell stack cooling control. *Renewable Energy*, 147, 1642–1652.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.

Wang, Q. and Stengel, R.F. (2000). Robust nonlinear control of a hypersonic aircraft. *Journal of guidance, control, and dynamics*, 23(4), 577–585.

Xue, W., Bai, W., Yang, S., Song, K., Huang, Y., and Xie, H. (2015). Adrc with adaptive extended state observer and its application to air–fuel ratio control in gasoline engines. *IEEE Transactions on Industrial Electronics*, 62(9), 5847–5857.

Xue, W., Madonski, R., Lakomy, K., Gao, Z., and Huang, Y. (2017). Add-on module of active disturbance rejection for set-point tracking of motion control systems. *IEEE Transactions on Industry Applications*, 53(4), 4028–4040.

Yoo, D., Yau, S.S.T., and Gao, Z. (2007). Optimal fast tracking observer bandwidth of the linear extended state observer. *International Journal of Control*, 80(1), 102–111.

Zhao, Z.L. and Guo, B.Z. (2017). A novel extended state observer for output tracking of mimo systems with mismatched uncertainty. *IEEE Transactions on Automatic Control*, 63(1), 211–218.