

考虑执行器退化的容退控制器设计

靳荟敏¹, 曾建潮², 牛慧芳¹, 原艳丹¹, 王文杰¹

(1. 中北大学 电气与控制工程学院, 山西 太原 030051;

2. 中北大学 计算机科学与技术学院, 山西 太原 030051)

摘要: 执行器在故障前会逐渐退化, 其退化程度直接影响控制系统的性能, 但现有研究主要针对执行器故障的容错控制, 因此需要研究考虑执行器退化过程的容退控制策略。本文以线性系统为对象, 首先, 建立了考虑系统状态与执行器退化状态相互影响的系统动态模型和执行器退化模型。然后, 基于该模型设计了考虑执行器退化的容退控制器。此外, 利用状态观测器和卡尔曼滤波算法联合估计系统状态和退化状态。实验结果表明, 容退控制器能确保执行器退化过程中控制系统稳定且性能最优。

关键词: 控制系统; 执行器退化; Wiener过程; 最优控制; 状态观测器; 卡尔曼滤波

中图分类号: TP273 **文献标识码:** A **doi:** 10.62756/jnuc.issn.1673-3193.2025.01.0013

引用格式: 靳荟敏, 曾建潮, 牛慧芳, 等. 考虑执行器退化的容退控制器设计[J]. 中北大学学报(自然科学版), 2025, 46(5): 622-631.

JIN Huimin, ZENG Jianchao, NIU Huifang, et al. Design of tolerance-degradation controller considering actuator degradation [J]. Journal of North University of China (Natural Science Edition), 2025, 46(5): 622-631.

Design of Tolerance-Degradation Controller Considering Actuator Degradation

JIN Huimin¹, ZENG Jianchao², NIU Huifang¹, YUAN Yandan¹, WANG Wenjie¹

(1. School of Electrical and Control Engineering, North University of China, Taiyuan 030051 China;

2. School of Computer Science and Technology, North University of China, Taiyuan 030051 China)

Abstract: The actuator will gradually deteriorate before a failure occurs, and the degree of degradation of the actuator directly affects the performance of the control system. The existing research mainly focuses on the fault-tolerant control of actuator faults. Therefore, it is necessary to study the tolerance control strategy considering the degradation process of the actuator. This paper took the linear system as the object. Firstly, the system dynamic model and the actuator degradation model considering the mutual influence between the system state and the actuator degradation state were established. Then, based on this model, a tolerance-degradation controller was designed. Furthermore, the state observer and the Kalman filtering algorithm were jointly utilized to estimate the system states and degradation states. The

收稿日期: 2025-01-13

基金项目: 国家自然科学基金(61703297, 72071183); 山西省回国留学人员科研资助项目(2021-135, 2022-161); 山西省留学回国人员科技活动择优资助项目(20220029); 山西省自然科学基金资助项目(202103021223206, 20210302124165, 202203021211194); 山西省科技成果转化引导专项资助项目(202204021301051); 山西省专利转化专项计划项目(202306002)

作者简介: 靳荟敏(2001-), 女, 硕士生, 主要从事复杂系统寿命预测及健康管理的研究。

通信作者: 牛慧芳(1986-), 女, 讲师, 博士, 主要从事复杂系统寿命预测及健康管理的研究。E-mail: hfn@nuc.edu.cn。

experimental results show that the tolerance-degradation controller can ensure the stability and optimal performance of the control system during the actuator degradation process.

Key words: control system; actuator degradation; Wiener process; optimal control; state observer; Kalman filtering

0 引言

随着工业自动化技术的发展,控制系统的结构变得越来越复杂,确保控制系统运行的高效性和安全性变得愈发关键^[1]。执行器作为控制系统的关键部件,其性能对于控制系统来说至关重要^[2-4]。然而,在实际应用中,执行器受到内外环境等多种因素的影响,其性能会逐渐下降,并最终失效,从而影响控制系统的稳定性以及动态性能^[5-7]。因此,需要研究有效的控制策略来容忍执行器退化对控制系统的影响。然而,现有的研究大多是针对执行器故障对控制系统的影响来开展的,即容错控制。

容错控制按设计方法分为主动容错和被动容错。主动容错控制是在故障发生后才使用容错控制器。1991年,Stengel^[8]首次提出了智能故障容错控制的概念,这被视为主动容错控制研究的起点。随后,研究者们逐步将主动容错控制应用于航天器、无人机、机器人等复杂系统中^[9-11]。近年来,神经网络和模糊逻辑等先进算法被引入主动容错控制中,进一步提升了系统的鲁棒性和适应性^[12]。被动容错控制是在控制器设计之初就考虑了故障。Lunze等^[13]对被动容错控制进行了系统性总结,并提出了基于 H_∞ 方法^[14]的鲁棒控制器设计理论。随后,滑模控制理论成为被动容错控制的重要工具之一^[15]。近年来,被动容错控制在能源管理、传感器网络等领域得到了广泛应用^[16]。

对于被动容错控制来说,其只能适应几种明确的特定类型的故障,且往往以牺牲性能代价来换取容错能力。对于主动容错控制来说,其高度依赖于故障检测与隔离机制,而这些机制往往存在误报或漏报的风险,且其通常依赖于预定义的容错控制器集合。因此,虽然被动容错和主动容错在处理执行器故障情况时都表现出良好的效果,但是它们都不能灵活地应对执行器的退化。事实上,执行器在完全失效之前会逐渐退化,进而影响控制系统的稳定性和动态性能^[17-18],并且这种退化是随机的。因此,仅针对执行器故障来设计容错控制器显然是不够的,有必要研究在执行器退化过程中保持控制系统稳定

性及动态性能的控制策略。

本文针对线性系统中执行器的退化问题,设计了执行器退化过程中能够保证控制系统稳定性以及最优性能的容退控制器。首先在分析执行器退化对控制系统稳定性的影响关系的基础上,建立了系统状态和退化状态相互影响的系统动态模型和退化模型。然后,使用状态观测器和卡尔曼滤波算法联合估计系统状态和退化状态,进而设计了容退控制器,保证控制系统的稳定性以及最优性能。最后,通过数值仿真证明了本文所提方法的有效性。

本研究的创新之处如下:

- 1) 建立了系统状态和退化状态相互影响的系统动态模型和执行器退化模型。
- 2) 利用状态观测器和卡尔曼滤波算法联合估计系统状态和退化状态。
- 3) 设计了考虑执行器退化的容退控制器,不仅保证了执行器退化过程中控制系统的稳定性,还使得控制系统的性能最优。

1 问题描述

1.1 不考虑执行器退化的最优控制器设计

对于线性系统,其状态空间方程为

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t), \end{cases} \quad (1)$$

式中: $\boldsymbol{x}(t) \in R^{n \times 1}$ 为系统状态; $\boldsymbol{u}(t) \in R^{m \times 1}$ 为控制输入; $\boldsymbol{y}(t) \in R^{p \times 1}$ 为系统输出; $\boldsymbol{A} \in R^{n \times n}$, $\boldsymbol{B} \in R^{n \times m}$, $\boldsymbol{C} \in R^{p \times n}$ 是实数矩阵,分别为系统的状态、输入、输出矩阵。

设计状态反馈控制器 $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t)$,其中, \boldsymbol{K} 为反馈增益矩阵。控制目标是最小化如式(2)所示的二次型性能指标^[19],旨在最小化控制成本,同时确保闭环系统具有良好的动态响应特性。

$$J = \int_0^{\infty} (\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t))dt, \quad (2)$$

式中: \boldsymbol{Q} 和 \boldsymbol{R} 分别为正定的状态权重矩阵和控制权重矩阵。

为实现设计目标,采用线性二次最优调节器的

方法来确定状态反馈增益。具体步骤总结如下:

1) 定义权重矩阵: 根据系统的物理特性和控制要求, 合理选择状态权重矩阵 Q 和控制权重矩阵 R 。 Q 用于调整状态变量的重要性, 其中的元素越大, 表明越重视状态的偏离, 通过调整矩阵 Q 可以影响闭环系统的性能。 R 用于平衡控制能量与系统性能之间的关系, 其中的元素越大, 意味着更倾向于降低控制成本。

2) 求解代数 Riccati 方程: 该方程是 LQR 问题的核心。代数 Riccati 方程为

$$A^T P + PA - PBR^{-1}B^T P + Q = 0, \quad (3)$$

式中: A 为系统状态矩阵; B 为输入矩阵; P 为对称正定的解矩阵。式(3)确保了闭环系统的稳定性, 并提供了最优控制所需的反馈增益矩阵的计算基础。

3) 计算最优增益向量: 利用求解式(3)得到的矩阵 P , 进一步计算出最优反馈增益矩阵 K 。

$$K = R^{-1}B^T P. \quad (4)$$

4) 验证 K 是否能够保证系统稳定: 通过检查闭环系统的特征值是否都位于复平面的左半部分来完成。如果所有特征值均具有负实部, 则说明闭环系统是稳定的, 从而能够保证所设计控制器的有效性。

为展示上述方法的应用, 以文献[20]中的例子来说明, 具体参数如下:

$$A = \begin{bmatrix} 0 & 1 & 1 & 2 \\ -1 & -1 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

对应的状态方程为

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ -1 & -1 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}. \quad (5)$$

按照前述步骤, 得出最优状态反馈控制器:

$$u(t) = -Kx(t) = -\begin{bmatrix} 1.39 & 2.07 & 1.98 & 1.85 \\ 0.80 & 0.92 & 0.74 & 1.12 \end{bmatrix} x(t). \quad (6)$$

此控制律下, 闭环系统特征值的实部分别为

-1.172 0, -1.172 0, -1.958 0 和 -1.958 0, 均为负值, 这意味着闭环系统是稳定的。

1.2 执行器退化对控制系统的影响

文献[20]考虑了执行器退化, 其容错控制器的设计主要针对一个执行器完全失效, 即中断故障, 另外一个执行器完好的情况。然而, 实际中执行器的退化是一个渐进的过程。因此, 有必要全面考量执行器退化过程对控制系统的影响。

两个执行器退化的情况下, 系统的状态方程为

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ -1 & -1 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} M(t)u(t), \quad (7)$$

式中: $M(t) = \begin{bmatrix} \rho_1(t) & 0 \\ 0 & \rho_2(t) \end{bmatrix}$, 是执行器有效性矩阵, 为对角矩阵。执行器的有效性由与其退化状态有关的有效性因子 $\rho_i(t)$ 来量化, 其中 $i=1, 2$ 。

$\rho_i(t)=1$, 表示执行器未发生退化; $\rho_i(t)=0$ 表示执行器退化至完全失效; $\rho_i(t)$ 介于 0 到 1 之间时, 执行器处于退化状态。

控制系统运行的前提是保证其稳定性, 因此首要任务是分析退化对控制系统稳定性的影响。无执行器退化时, 闭环特征方程为 $|\lambda I - (A - BK)|$, λ 是闭环特征值。存在执行器退化时, 闭环特征方程为 $|\lambda I - (A - BM(t)K)|$ 。那么, 执行器的退化会引起闭环特征值的变化, 甚至让闭环特征值实部变为非负, 导致控制系统不稳定。

两个执行器在不同程度退化下对控制系统稳定性的影响如图 1 所示。由图 1 可见, 当 (ρ_1, ρ_2) 的值落在绿色区域时, 控制系统是稳定的; 落在黄色区域时, 控制系统是不稳定的。例如, 当 $\rho_1=0.5, \rho_2=0.2$ 时控制系统是稳定的; 当 $\rho_1=0.2, \rho_2=0.2$ 时控制系统是不稳定的。由此可见, 执行器在故障之前的退化已经造成控制系统的不稳定。显然只考虑执行器故障来进行容错控制器的设计是不够的。还需要进行能够容忍执行器退化的控制器的设计。

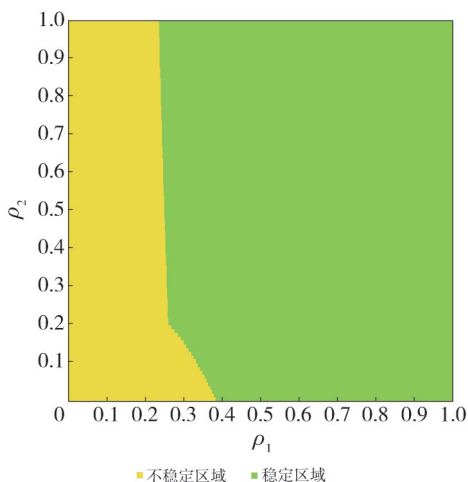


图1 执行器退化对控制系统稳定性的影响

Fig. 1 Influence of actuator degradation on the stability of the control system

2 模型的建立

2.1 考虑执行器退化的系统建模

执行器的退化过程会导致其性能逐渐下降, 这种变化在系统状态空间方程中具体表现为控制输入矩阵 B 的变化。随着执行器性能的衰退, 矩阵 B 中的元素会出现渐进性的数值变化, 这直接反映了执行器对控制输入响应特性的改变, 进而影响系统的性能。因此, 考虑执行器退化的线性系统状态空间方程为

$$\begin{cases} \dot{x}(t) = Ax(t) + BM(t)u(t), \\ y(t) = Cx(t), \end{cases} \quad (8)$$

式中: 对角矩阵 $M(t) \in R^{m \times m}$ 是执行器有效性矩阵, 其中的元素 $\rho_i(t) \in [0, 1] (i = 1, 2, \dots, m)$ 是第 i 个执行器的有效性因子, 与其退化状态相关。

2.2 执行器退化建模

执行器性能衰退是一个受多种随机因素和时间累积效应影响的复杂渐进过程。这些不确定性和随机因素随着时间逐渐积累, 并导致性能下降。非线性 Wiener 过程作为一种随机模型, 不仅能捕捉这种不确定性的退化行为, 还能通过引入时间变量反映退化的累积特性。同时, 它允许对非线性退化速率进行灵活建模, 适应不同阶段和条件下退化速度的变化, 从而更准确地模拟真实的退化模式。该模型提供了强大的数学工具, 能够描述和分析执行器退化的动态特性, 是模拟执行器退化过程的理想选择。因此, 执行器退化状态

$z_i(t)$ 建模为非线性维纳过程^[21]

$$z_i(t) = z_i(0) + \int_0^t \mu(x(\tau), \tau) d\tau + \sigma_{B,i} B(t), \quad (9)$$

式中: $z_i(0)$ 表示初始退化状态, 不失一般性, 假设其为 0; 退化速率与系统状态相关, $\mu(x(t), t) = a_i e^{N_i x(t)}$, 其中 N_i 为相关性向量, 体现其与每个系统状态的具体关系; $\sigma_{B,i}$ 为扩散系数; $B(t) \sim N(0, t)$ 为标准布朗运动, 表征了退化过程的随机动力学。

退化状态向量为

$$z(t) = z(0) + aH + \sigma_B B(t), \quad (10)$$

其中, $z(t) = [z_1(t), z_2(t), \dots, z_m(t)]^T$,

$$a = \text{diag}\{a_1, a_2, \dots, a_m\},$$

$$H = \left[\int_0^t e^{N_1 x(\tau)}, \int_0^t e^{N_2 x(\tau)}, \dots, \int_0^t e^{N_m x(\tau)} \right]^T,$$

$$\sigma_B = [\sigma_{B,1}, \sigma_{B,2}, \dots, \sigma_{B,m}]^T.$$

鉴于能够直接估计的是执行器的退化状态而非其有效性因子 $\rho_i(t)$, 因此需要建立退化状态和有效性因子之间的关系。两者关系应该满足: 当执行器退化状态为 0 时, $\rho_i(t) = 1$, 此时系统的行为与无退化情况一致; 随着退化状态的增大, $\rho_i(t)$ 逐渐减小; 直到退化状态为 1 时, $\rho_i(t) = 0$ 时, 表示执行器完全失效。此外, 考虑到实际退化状态可能超出预设范围, 需对其进行归一化处理。归一化后的退化状态 $z'_i(t)$ 定义为

$$z'_i(t) = \frac{z_i(t) - \min(z_i(t))}{\max(z_i(t)) - \min(z_i(t))}, \quad (11)$$

式中: 不失一般性, 假设退化状态最小值 $\min(z_i(t))$ 为 0; 假设最大的退化状态 $\max(z_i(t))$ 已知, 且等于 h_i 。

由此, 实际退化状态和有效性因子之间的关系可以表述为

$$\rho_i(t) = 1 - \bar{z}'_i(t) = 1 - \frac{1}{h_i} \bar{z}_i(t), \quad (12)$$

式中: \bar{z}_i 表示实际退化状态的平均值。

故

$$M(t) = I_{m \times m} - \text{diag} \left\{ \frac{\bar{z}_1(t)}{h_1}, \frac{\bar{z}_2(t)}{h_2}, \dots, \frac{\bar{z}_m(t)}{h_m} \right\}. \quad (13)$$

3 容退控制器设计

针对系统(8), 本文设计了状态负反馈容退控制器 $u(t) = -K(t)x(t)$, 使性能指标(14)最小, 且反馈增益能够根据执行器的退化状态进行实时

调整。这样不仅保证了控制系统在执行器退化过程中的稳定性,还使得控制系统性能最优。

$$J = \int_0^{\infty} (x^T(t)Qx(t) + u^T(t)Ru(t))dt. \quad (14)$$

根据最优控制理论,定义哈密顿函数(15)。

$$H(x(t), u(t), \lambda(t)) = x^T(t)Qx(t) + u^T(t)Ru(t) + \lambda^T(t)(Ax(t) + BM(t)u(t)), \quad (15)$$

式中: $\lambda(t)$ 为协态变量。

根据庞特里亚金极小值原理,最优控制律 $u(t)$ 应该满足

$$\frac{\partial H}{\partial u} = 2Ru(t) + M(t)B^T\lambda(t) = 0. \quad (16)$$

解得

$$u(t) = -R^{-1}M(t)B^T\lambda(t). \quad (17)$$

注:在这个过程中,“2”这个因子实际被合并到了矩阵 R 中,因为 R 的具体数值通常是根据系统的设计要求选择的,只是一个比例的选择,不影响控制器的本质特征。

协态变量 $\lambda(t)$ 满足协态方程

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x} = -(Qx(t) + A^T\lambda(t)). \quad (18)$$

注: $-\frac{\partial H}{\partial x} = -(2Qx(t) + A^T\lambda(t))$, “2”这个因子实际被合并到了矩阵 Q 中。

将式(17)代入系统状态方程(8)中可以得到闭环系统的状态方程

$$\dot{x}(t) = Ax(t) - BM(t)R^{-1}M(t)B^T\lambda(t). \quad (19)$$

假设存在一个矩阵 $P(t)$,使得协态变量可以表示为状态 $x(t)$ 的线性函数,即 $\lambda(t) = P(t)x(t)$ 。

那么,式(19)变为

$$\dot{x}(t) = Ax(t) - BM(t)R^{-1}M(t)B^T P(t)x(t). \quad (20)$$

式(18)变为

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x} = -Qx(t) - A^T P(t)x(t). \quad (21)$$

并且

$$\dot{\lambda}(t) = \dot{P}(t)x(t) + P(t)[Ax(t) - BM(t)R^{-1}M(t)B^T P(t)x(t)]. \quad (22)$$

结合式(21)和(22)可得

$$\begin{aligned} & \dot{P}(t)x(t) + P(t)[Ax(t) - \\ & BM(t)R^{-1}M(t)B^T P(t)x(t)] = \\ & -Qx(t) - A^T P(t)x(t). \end{aligned} \quad (23)$$

由于 $x(t)$ 是任意的非零向量,式(23)对所有的 $x(t)$ 成立,因此系数矩阵相等,即

$$\dot{P}(t) = P(t)BM(t)R^{-1}M(t)B^T P(t) - P(t)A -$$

$$Q - A^T P(t). \quad (24)$$

稳态情况下, $\dot{P}(t) = 0$ 。从而得出代数 Riccati 方程

$$A^T P(t) + P(t)A - P(t)BM(t)R^{-1}M(t)B^T P(t) + Q = 0. \quad (25)$$

最优控制器为

$$u(t) = -K(t)x(t) = -R^{-1}M(t)B^T P(t)x(t). \quad (26)$$

4 状态观测与估计

假设退化状态是可测量的,那么,退化状态方程以及退化测量方程可表示为

$$\begin{cases} z(t) = aH + \sigma_B B(t), \\ g(t) = Fz(t) + \nu(t), \end{cases} \quad (27)$$

式中:矩阵 F 是状态 $z(t)$ 的测量矩阵; $\nu(t)$ 是测量噪声。

系统模型(8)是确定性系统,而退化模型(27)是随机系统。因此,对于系统状态,需要使用状态观测器进行观测;对于退化状态,需要使用滤波器进行估计。

为了便于系统状态和退化状态的估计,采用欧拉离散化方法将系统模型(8)和退化模型(27)离散化。

$$\begin{cases} x_{k+1} = (I + A\Delta t)x_k + BM_k\Delta t u_k, \\ y_k = Cx_k, \\ z_{k+1} = z_k + aH_k + \epsilon_k, \\ g_k = Fz_k + \nu_k, \end{cases} \quad (28)$$

式中: I 为单位矩阵; Δt 为时间间隔; $M_k = I_{m \times m} - \text{diag}\left\{\frac{z_{1,k}}{h_1}, \frac{z_{2,k}}{h_2}, \dots, \frac{z_{m,k}}{h_m}\right\}$; $H_k = [e^{N_1 x_k \Delta t}, e^{N_2 x_k \Delta t}, \dots, e^{N_n x_k \Delta t}]^T$ 。

ϵ_k 是式(10)中标准布朗运动产生的过程噪声矩阵,其中的元素服从 $N(0, \sigma_{B,i}^2 \Delta t)$, $i = 1, \dots, m$ 。

由离散模型(28)可见,在每个步长下,对系统状态 x 观测时, z 可看作常数;而对退化状态 z 进行滤波估计时, x 可看作常数。因此,使用状态观测器进行系统状态的观测,使用卡尔曼滤波算法进行退化状态的估计,并且将两者结合起来,具体流程如下:

1) 初始化及参数设置:设置 x 和 z 的初始状态以及相关参数。

2) 状态观测与估计:

系统状态 x 的观测:已知 t_k 时刻观测值 \tilde{x}_k 以及估计值 \hat{z}_k ,设计状态观测器进行 x 的观测,即

$$\tilde{x}_{k+1} = (I + A\Delta t)\tilde{x}_k + BM_k\Delta t u_k + L(y_k - C\tilde{x}_k), \quad (29)$$

式中： $M_k = I - \text{diag}\left\{\frac{\hat{z}_1}{h_1}; \frac{\hat{z}_2}{h_2}; \dots; \frac{\hat{z}_m}{h_m}\right\}$ ； $u_k = -M_k B^T P \tilde{x}_k$ ， $P = \text{care}(A, BM_k, Q, R)$ ； L 为观测器的增益，使用极点配置法确定。

退化状态 z 的估计：

预测部分为

$$\hat{z}_{k+1|k} = \hat{z}_{k|k} + a e^{N \hat{z}_k} \Delta t, \quad (30)$$

$$P_{z, k+1|k} = P_{z, k|k} + Q_{z_0} \quad (31)$$

式中： $\hat{z}_{k+1|k}$ 为 $k+1$ 时刻状态的预测值； $P_{z, k+1|k}$ 为预测的状态协方差矩阵； Q_z 为 z 的过程噪声协方差矩阵。

更新部分为

$$K_{k+1} = P_{z, k+1|k} F^T (F P_{z, k+1|k} F^T + R_z)^{-1}, \quad (32)$$

$$\hat{z}_{k+1|k+1} = \hat{z}_{k+1|k} + K_{k+1} (g_k - F \hat{z}_{k+1|k}), \quad (33)$$

$$P_{k+1|k+1} = (I - K_{k+1} F) P_{k+1|k}, \quad (34)$$

式中： K_k 为卡尔曼增益； F 为 z 的测量矩阵； R_z 为测量噪声协方差矩阵； g_k 为实际测量值； I 为单位矩阵。

3) 返回步骤(2)进行新的循环迭代。

将上述状态估计过程总结为算法 1，如表 1 所示。

表 1 状态估计算法

Tab. 1 State estimation algorithm

步骤	描述
1	设置状态的初始值及相关参数
2	根据上个时间步长得到的 \tilde{x}_k 及 \hat{z}_k ，根据观测器(29)观测系统状态，根据式(30)~式(34)估计得到退化状态
3	返回步骤2进行新的循环迭代

综上可得总控制框图如图 2 所示。

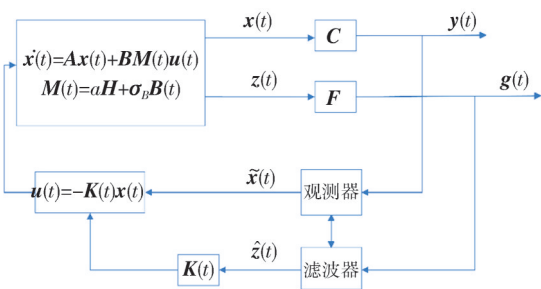


图 2 控制框图

Fig. 2 Control block diagram

5 数值仿真

5.1 模型及参数

给定线性系统以及退化状态的参数：

$$A = [0, 1, 1, 2; -1, -1, 1, 0; 2, 2, 0, 1; 0, 1, 0, 0];$$

$$B = [0, 0; 2, 0; 0, 0; 0, 1];$$

$$C = [1, 0, 0, 0; 0, 1, 0, 0; 0, 0, 1, 0; 0, 0, 0, 1];$$

$$F = [1, 0; 0, 1]; x(0) = [1; 1; 1; 1];$$

$$h_1 = 10, h_2 = 7, a_1 = 0.3, a_2 = 0.1;$$

$$N_1 = [0.1, 0.1, 0.1, 0.1], N_2 = [0.2, 0.2, 0.2, 0.2];$$

$$\sigma_{B,1} = \sigma_{B,2} = 0.1; \Delta t = 0.1;$$

$$\nu_1 \sim N(0, 0.0064), \nu_2 \sim N(0, 0.0064).$$

考虑执行器退化时相应的状态空间方程如式(35)所示。

$$\begin{cases} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 2 \\ -1 & -1 & 1 & 0 \\ 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} M(t) u(t), \\ y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(t), \end{cases} \quad (35)$$

$$\text{其中, } M(t) = I_{2 \times 2} - \text{diag}\left\{\frac{z_1(t)}{10}, \frac{z_2(t)}{7}\right\}.$$

退化模型为

$$\begin{cases} z(t) = \begin{bmatrix} 0.3 \int_0^t e^{[0.1 \ 0.1 \ 0.1 \ 0.1]x(\tau)} d\tau \\ 0.1 \int_0^t e^{[0.2 \ 0.2 \ 0.2 \ 0.2]x(\tau)} d\tau \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} B(t), \\ g(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} z(t) + \nu(t). \end{cases} \quad (36)$$

5.2 控制器设计及状态估计仿真结果

针对系统(35)，设计容退控制器，使得性能指标(37)最小。

$$J = \int_0^{\infty} (x^T(t) Q x(t) + u^T(t) R u(t)) dt, \quad (37)$$

$$\text{其中, } Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

文献[20]针对两个执行器中一个故障另一个完好的情况设计了容错控制器。考虑故障之前执行器的退化时，原来控制器的控制效果如图 3 所示，执行器的退化状态如图 4 所示。

结合图 3 和图 4 可以发现，在两个执行器退化的前期，控制系统能够保持稳定，这是由于控制器自身具有一定的调节作用。然而，当执行器退化严重时，即在时间步长为 400 步之后，系统状态会明显发散。可见，原控制器不能在执行器退化的整个过程中发挥很好的调节作用。因此，有必要设计容退控制器来保证执行器退化过程中控制系统稳定且性能最优。

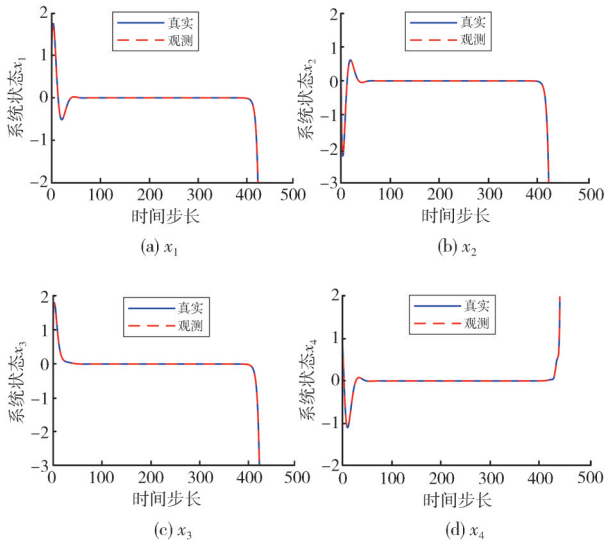


图3 文献[20]中不考虑执行器退化时控制器的系统状态

Fig. 3 The system states of controller without considering actuator degradation in reference [20]

在实际应用中,由于受环境、运行方式等因素的影响,即使是两个完全相同的执行器,其退化程度也不相同。因此,另外,选择3组不同参数

进行仿真,结果如表2所示。一个执行器退化严重,一个执行器退化缓慢的情况下,在退化前期,控制器能够保持控制系统的稳定,而当执行器退化严重时,控制系统不能保持稳定,两个执行器退化都迅速的情况下,无论执行器退化是否严重,都不能保持控制系统的稳定。总之,无论两个执行器处于何种退化情况下,原来不考虑执行器退化的控制器都存在局限性。

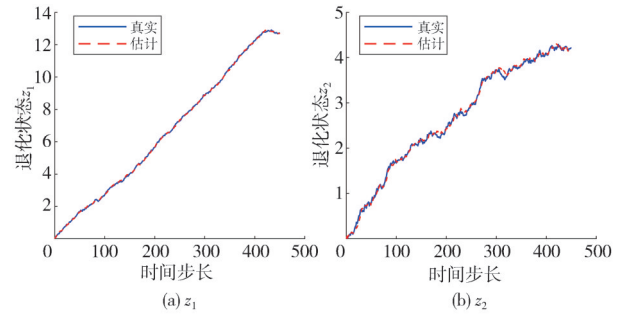


图4 文献[20]中不考虑执行器退化时控制器的退化状态

Fig. 4 The degradation states of controller without considering actuator degradation in reference [20]

表2 仿真结果对比

Tab. 2 Comparison of simulation results

执行器退化参数	执行器退化情况	不考虑执行器退化的控制器的控制效果	本文设计的控制器的控制效果
$a_1=0.3, a_2=0.1,$ $N_1=[0.1, 0.1, 0.1, 0.1],$ $N_2=[0.2, 0.2, 0.2, 0.2]$	执行器1: 缓慢 执行器2: 缓慢	退化前期控制系统稳定,退化严重(400步之后)控制系统不稳定	整个退化过程控制系统稳定,且性能最优
$a_1=1, a_2=0.1,$ $N_1=[0.3, 0.3, 0.3, 0.3],$ $N_2=[0.2, 0.2, 0.2, 0.2]$	执行器1: 迅速 执行器2: 缓慢	退化前期控制系统稳定,退化严重(110步之后)控制系统不稳定	整个退化过程控制系统稳定,且性能最优
$a_1=0.3, a_2=1,$ $N_1=[0.1, 0.1, 0.1, 0.1],$ $N_2=[0.3, 0.3, 0.3, 0.3]$	执行器1: 缓慢 执行器2: 迅速	退化前期控制系统稳定,退化严重(150步之后)控制系统不稳定	整个退化过程控制系统稳定,且性能最优
$a_1=1, a_2=1,$ $N_1=[0.3, 0.3, 0.3, 0.3],$ $N_2=[0.3, 0.3, 0.3, 0.3]$	执行器1: 迅速 执行器2: 迅速	控制系统一直不稳定	整个退化过程控制系统稳定,且性能最优

对本文设计的控制器的控制效果进行仿真,选取与图3和图4一样的退化参数,即

$$a_1=0.3, a_2=0.1,$$

$$N_1=[0.1, 0.1, 0.1, 0.1],$$

$$N_2=[0.2, 0.2, 0.2, 0.2].$$

仿真结果如图5和图6所示。结合图5和图6可知,本文设计的容退控制器不仅在执行器退化前期能够使得控制系统稳定,而且在执行器退化严重,即400步之后,同样能够保持控制系统的稳定,且性能最优。由此可见,本文设计的容退控制器在执行器退化的整个过程中都能够实现很好的控制效果。

为进一步验证本文方法的有效性,依然选择表2中后3组执行器退化参数进行仿真。

1) $a_1=1, a_2=0.1, N_1=[0.3, 0.3, 0.3, 0.3], N_2=[0.2, 0.2, 0.2, 0.2]$ 。

本文设计的容退控制器仿真结果如图7和图8所示。结合图7和图8可知,执行器1退化迅速,执行器2退化缓慢时,在本文所设计的容退控制器下,系统状态很快平稳,且系统性能最优。同时,即使是在退化严重时,仍能够保持控制系统的稳定。

2) $a_1=0.3, a_2=1, N_1=[0.1, 0.1, 0.1, 0.1], N_2=[0.3, 0.3, 0.3, 0.3]$ 。

本文设计的容退控制器的仿真结果如图9和图10所示。结合图9和图10可知,执行器1退化缓慢,执行器2退化迅速时,无论执行器退化是否严重,本文设计的容退控制器都能够保持控制系统的稳定,且性能最优。

3) $a_1=1, a_2=1, N_1=[0.3,0.3,0.3,0.3], N_2=[0.3,0.3,0.3,0.3]$ 。

本文设计的容退控制器的仿真结果如图 11 和图 12 所示。结合图 11 和图 12 可知,两个执行器退化都比较迅速时,本文所设计的容退控制器依然能够保持控制系统的稳定以及最优性能。同时,即使是在执行器退化严重时控制系统仍然稳定。

由以上不同参数下本文所设计的控制器仿真图 5~图 12 可知,无论两个执行器处于何种退化情况下,其都能够保持控制系统的稳定性以及良好的性能,而且在退化严重时也没有破坏控制系统的稳定。总之,以上仿真体现了本文方法的有效性。

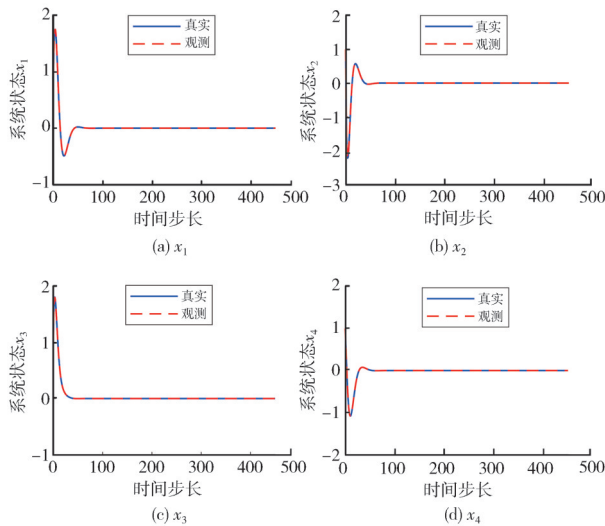


图 5 $a_1=0.3, a_2=0.1, N_1=[0.1,0.1,0.1,0.1], N_2=[0.2, 0.2,0.2,0.2]$ 时容退控制器下的系统状态

Fig. 5 The system states under the tolerance-degradation controller at $a_1=0.3, a_2=0.1, N_1=[0.1,0.1,0.1,0.1], N_2=[0.2,0.2, 0.2,0.2]$

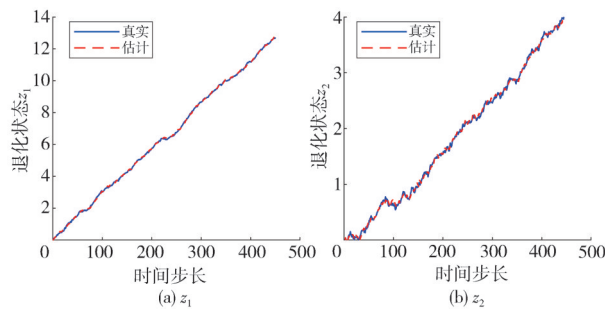


图 6 $a_1=0.3, a_2=0.1, N_1=[0.1,0.1,0.1,0.1], N_2=[0.2, 0.2,0.2,0.2]$ 时容退控制器下的退化状态

Fig. 6 The degradation states under the tolerance-degradation controller at $a_1=0.3, a_2=0.1, N_1=[0.1,0.1,0.1,0.1], N_2=[0.2,0.2,0.2,0.2]$

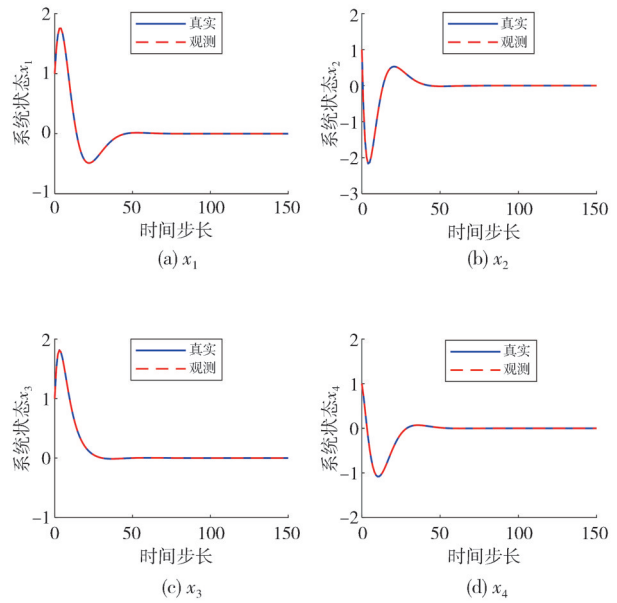


图 7 $a_1=1, a_2=0.1, N_1=[0.3,0.3,0.3,0.3], N_2=[0.2, 0.2,0.2,0.2]$ 时容退控制器下的系统状态

Fig. 7 The system states under the tolerance-degradation controller at $a_1=1, a_2=0.1, N_1=[0.3,0.3,0.3,0.3], N_2=[0.2,0.2, 0.2,0.2]$

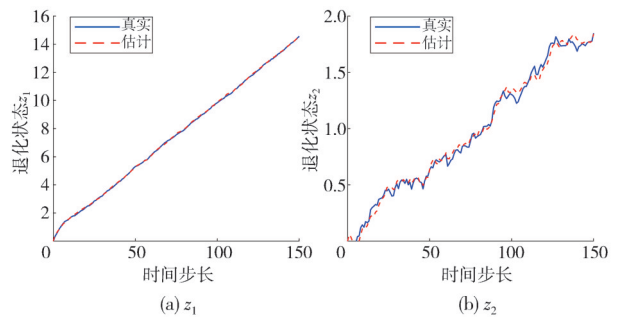
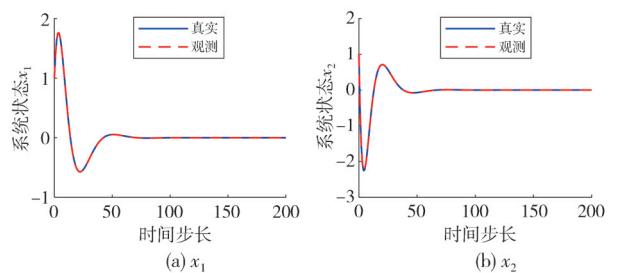


图 8 $a_1=1, a_2=0.1, N_1=[0.3,0.3,0.3,0.3], N_2=[0.2, 0.2,0.2,0.2]$ 时容退控制器下的退化状态

Fig. 8 The degradation states under the tolerance-degradation controller at $a_1=1, a_2=0.1, N_1=[0.3,0.3,0.3,0.3], N_2=[0.2, 0.2,0.2,0.2]$



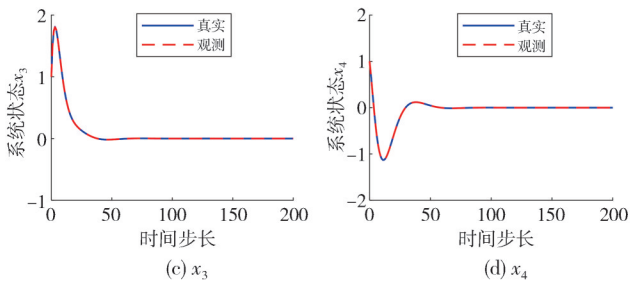


图9 $a_1=0.3, a_2=1, N_1=[0.1, 0.1, 0.1, 0.1], N_2=[0.3, 0.3, 0.3, 0.3]$ 时容退控制器下的系统状态

Fig. 9 The system states under the tolerance-degradation controller at $a_1=0.3, a_2=1, N_1=[0.1, 0.1, 0.1, 0.1], N_2=[0.3, 0.3, 0.3, 0.3]$

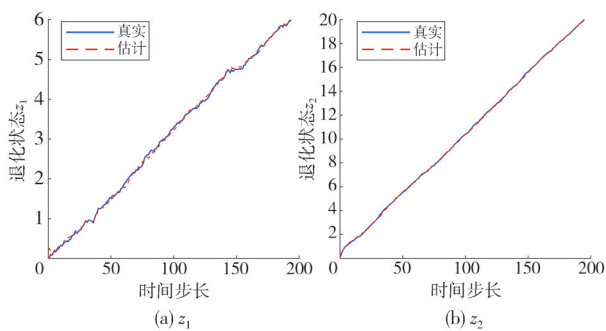


图10 $a_1=0.3, a_2=1, N_1=[0.1, 0.1, 0.1, 0.1], N_2=[0.3, 0.3, 0.3, 0.3]$ 时容退控制器下的退化状态

Fig. 10 The degradation states under the tolerance-degradation controller at $a_1=0.3, a_2=1, N_1=[0.1, 0.1, 0.1, 0.1], N_2=[0.3, 0.3, 0.3, 0.3]$

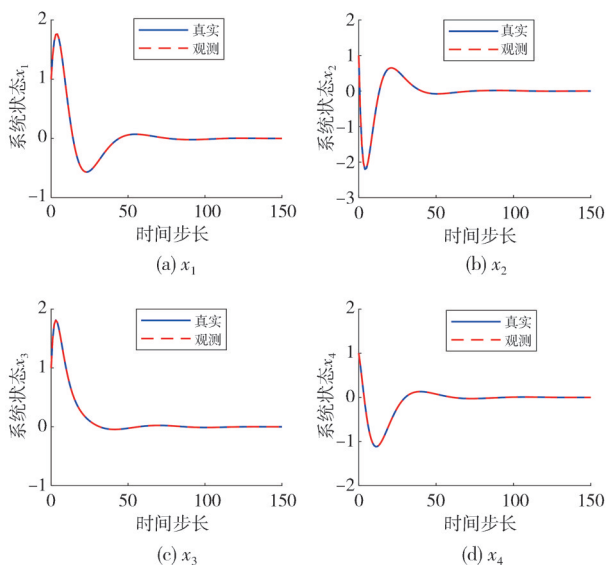


图11 $a_1=1, a_2=1, N_1=[0.3, 0.3, 0.3, 0.3], N_2=[0.3, 0.3, 0.3, 0.3]$ 时容退控制器下的系统状态

Fig. 11 The system states under the tolerance-degradation controller at $a_1=1, a_2=1, N_1=[0.3, 0.3, 0.3, 0.3], N_2=[0.3, 0.3, 0.3, 0.3]$

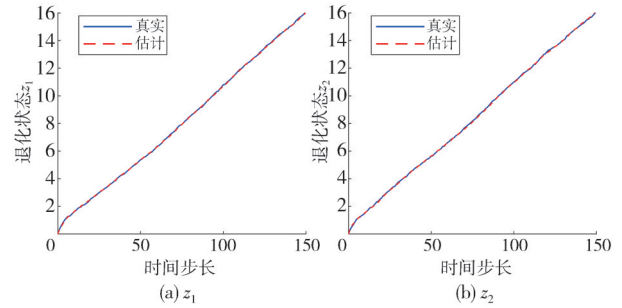


图12 $a_1=1, a_2=1, N_1=[0.3, 0.3, 0.3, 0.3], N_2=[0.3, 0.3, 0.3, 0.3]$ 时容退控制器下的退化状态

Fig. 12 The degradation states under the tolerance-degradation controller at $a_1=1, a_2=1, N_1=[0.3, 0.3, 0.3, 0.3], N_2=[0.3, 0.3, 0.3, 0.3]$

6 结论

本文分析了执行器退化对控制系统性能的影响,并且设计了容退控制器,以确保控制系统在整个退化过程中的稳定性和最佳性能,具体结论如下:1)建立了系统状态和退化状态相互影响的系统动态模型和执行器退化模型,该模型能够捕捉执行器退化的随机性和累积效应,提供了一个更贴近实际情况的框架来描述系统的动态特性;2)提出了基于二次型性能指标的容退控制器设计方法,基于此方法的控制器不仅能够执行器退化过程中保持控制系统的稳定,还使其性能达到最优;3)采用状态观测器和卡尔曼滤波算法相结合的方法进行状态估计,这种方法有效解决了系统状态和退化状态相互影响下的状态估计问题。然而,本研究也存在一定的局限性:1)假设了退化状态是可测量的,而实际当中退化状态通常是隐含不可测的;2)执行器的退化是缓慢的变化过程,而系统状态的变化相较于执行器的退化非常迅速,因此,需要考虑两者不同的时间尺度。后续将研究退化状态隐含不可测时,如何估计得到退化状态,并且考虑不同的时间尺度来进行控制器设计。这些研究将有助于进一步完善执行器退化问题的解决方案,提高控制系统的整体性能和可靠性。

参考文献:

- [1] GAO Z W. Special issue on "modelling, monitoring, control and optimization for complex industrial processes"[J]. Processes, 2023, 11(1): 207.
- [2] BARAMBONES O, CORTAJARENA J A, ALKORTA P. New control schemes for actuators[J].

- Actuators, 2024, 13(3): 99-104.
- [3] ROUSSEL J, BUDINGER M, RUET L. Preliminary sizing of the electrical motor and housing of electromechanical actuators applied on the primary flight control system of unmanned helicopters [J]. Aerospace, 2022, 9(9): 473-473.
- [4] TOPCZEWSKI S, ŻUGAJ M, and BIBIK P. Impact of actuators backlash on the helicopter control during landing on the moving vessel deck [J]. Aircraft Engineering and Aerospace Technology, 2021, 93 (9) : 1387-1394.
- [5] VIEIRA J P, GALVÃO R K H, YONEYAMA T. Predictive control for systems with loss of actuator effectiveness resulting from degradation effects [J]. Journal of Control, Automation and Electrical Systems, 2015, 26(6): 589-598.
- [6] YANG Z M, LI X P, XU J C, et al. Study of dynamic performance and control strategy of variable stiffness actuator system based on two-inertial-system [J]. Mathematics, 2023, 11(5): 1166-1166.
- [7] FENG C C, UNIVERSITY I S. Control systems in the presence of actuators or sensors degradation [J]. International Journal of Trend in Scientific Research and Development, 2017, 1(5): 97-104.
- [8] STENGEL R F. Intelligent failure-tolerant control [J]. IEEE Control Systems Magazine, 1991, 11(4): 14-23.
- [9] SADEGHZADEH I, MEHTA A, CHAMSEDDINE A, et al. Active fault tolerant control of a quadrotor UAV based on gainscheduled PID control [C]//2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). IEEE, 2012: 1-4.
- [10] SHEN Q, YUE C, and GOH C H, et al. Active fault-tolerant control system design for spacecraft attitude maneuvers with actuator saturation and faults [J]. IEEE Transactions on Industrial Electronics, 2019, 66 (5): 3763-3772.
- [11] NGUYEN N P, XUAN MUNG N, HA L N N T, et al. Fault-tolerant control for hexacopter UAV using adaptive algorithm with severe faults [J]. Aerospace, 2022, 9(6): 304.
- [12] ALSUWIAN T, USMAN M H, AMIN A A. An autonomous vehicle stability control using active fault-tolerant control based on a fuzzy neural network [J]. Electronics, 2022, 11(19): 3165.
- [13] LUNZE J, RICHTER J H. Reconfigurable fault-tolerant control: A tutorial introduction [J]. European Journal of Control, 2008, 14(5): 359-386.
- [14] ZHANG L P, GONG D L. Passive fault-tolerant control for vehicle active suspension system based on H₂/H_∞ approach [J]. Journal of Vibroengineering, 2018, 20(4): 1828-1849.
- [15] BENOSMAN M, LUM K Y. Passive actuators' fault tolerant control for affine nonlinear systems [J]. IFAC Proceedings Volumes, 2008, 41 (2) : 14229-14234.
- [16] ROTONDO D, NEJJARI F, PUIG V, et al. Fault tolerant control of the wind turbine benchmark using virtual sensors/actuators [J]. IFAC Proceedings Volumes, 2012, 45(20): 114-119.
- [17] SHI Y T, XIANG X, WANG L, et al. Stochastic model predictive fault tolerant control based on conditional value at risk for wind energy conversion system [J]. Energies, 2018, 11(1): 193.
- [18] QI Y, JING H Z, WU X W. Variable structure PID controller for satellite attitude control considering actuator failure [J]. Applied Sciences, 2022, 12(10): 5273.
- [19] ZHANG W. Based on the state feedback optimal controller design [J]. Applied Mechanics and Materials, 2012, 241/242/243/244: 1265-1268.
- [20] VEILLETTE R J. Reliable linear-quadratic state-feedback control [J]. Automatica, 1995, 31(1): 137-143.
- [21] SI X, REN Z, HU X, et al. A novel degradation modeling and prognostic framework for closed-loop systems with degrading actuator [J]. IEEE Transactions on Industrial Electronics, 2020, 67(11): 9635-9647.