algorithms

MDPI

Article

Data-Driven Fault-Tolerant Model Predictive Control for Soft Electro-Hydraulic Actuators Using Koopman Operator and Chebyshev Neural-Network Compensation

Gaofeng Che

School of Artificial Intelligence and Computer Science, Liaocheng University, Liaocheng 252000, China; chegaofeng@lcu.edu.cn

Abstract

Soft electro-hydraulic actuators (SEHAs) combine compliant deformation, fast response and high force density, therefore making them attractive for soft robotics and human–machine interaction. Their electro-hydro-solid multi-physics coupling, however, produces strong hysteresis, severe nonlinearity and unmodeled dynamics, so that accurate modeling and reliable fault-tolerant control remain difficult once the actuator degrades and the measurements drift. Purely first-principle models of SEHAs rely on restrictive geometric and material assumptions, whereas black-box data-driven models are poorly interpretable and hard to embed in constrained predictive control. Finite-dimensional Koopman predictors retain interpretability but leave an approximation residual that is amplified by partial actuator failure and measurement drift; existing Koopman-based controllers neither compensate this residual explicitly nor provide fault-tolerance guarantees, and most of them are validated only in simulation. This paper develops a residual-corrected, provably bounded data-driven framework. (i) A Chebyshev-network compensator guided by a terminal sliding-mode surface (CNCTSMS) estimates the Koopman approximation residual and the lumped disturbance online, without any first-principle equation. (ii) A Koopman actor–critic model predictive control (KACMPC) is constructed, in which the critic evaluates the long-horizon value gradient and the actor emits a bounded supplementary input, so that partial actuator failure and measurement drift are tolerated without a fault-detection and isolation (FDI) module. (iii) A composite Lyapunov argument, which explicitly accounts for input saturation, actuator effectiveness loss and bounded measurement disturbances, establishes uniform ultimate boundedness(UUB) and ϵ-approximate optimality of the resulting policy. On a joint-type SEHA, the compensator lowers the identification RMSE from 2.00×10^{-1} to 3.61×10^{-3} in simulation and from 2.80×10^{-3} to 5.70×10^{-5} at 3 kV in experiment. In simulation, KACMPC reduces the tracking RMSE by 25.1% with respect to Koopman MPC under nominal conditions and by 84.0% under persistent measurement drift, as well as keeping the lowest error at every noise level tested; over 15 repeated runs, the spread of the RMSE is below 4%. In hardware experiments, only the actuator model and the one-step prediction are validated, at 2 kV and 3 kV; closed-loop fault-tolerant tracking on the physical prototype is not yet demonstrated and is stated explicitly as future work. With $N_p = 20$, the average and worst-case optimizer execution times are 3.76 ms and 5.10 ms, which fit a 10 ms (100 Hz) control period.

Keywords: soft electro-hydraulic actuators (SEHAs); koopman operator; Chebyshev network; model predictive control; actor–critic network; fault-tolerant control

Check for updates

Academic Editor: Jin Guo

Received: 30 October 2026
Revised: 16 November 2026
Accepted: 18 November 2026
Published: 21 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

This section establishes the background of soft electro-hydraulic actuation and reviews the modeling and control literature that motivates the present study. It first explains why SEHAs are attractive and why their multi-physics coupling makes accurate modeling and robust control difficult, then summarizes three technical routes—model-driven active robust control, pure data-driven modeling, and Koopman-operator-based data-driven control—together with the model-free, error-driven controller families to which the proposed scheme is later compared. On this basis, four unresolved research gaps are identified, and the four contributions of the paper are stated.

1.1. Motivation

Soft electro-hydraulic actuators (SEHAs) [1–3] feature superior comprehensive performances including compliant deformation capability, fast response, high force density, favorable energy efficiency, silent operation, and low thermal emission. Such merits enable their widespread application prospects in soft robotics [4], human–machine interaction systems [5], and biomedical engineering fields [6]. Nevertheless, the inherent nonlinear characteristics and intricate multi-physics coupling effects make it challenging to precisely characterize the dynamic behaviors of SEHAs [7]. Establishing accurate dynamic models and developing high-robustness control strategies remain core challenges hindering the practical engineering deployment of SEHAs [8].

SEHAs integrate electrostatic actuation with dielectric liquid power transmission and hydrodynamic actuation mechanisms [9]. As one of the most typical SEHA configurations, hydraulically amplified self-healing electrostatic (HASEL) actuators suffer from complex multi-field coupled dynamic behaviors, whose output performances are highly sensitive to inhomogeneous electric fields and geometric nonlinearities [10]. In existing studies, such actuators are generally simplified to regular geometric models for electromechanical characteristic analysis. Although these model simplifications rely on strong physical assumptions and reduce geometric modeling complexity, they have achieved acceptable predictive accuracy under ideal operating conditions. However, practical working scenarios involve diverse external disturbances, including unknown time-varying external loads, progressive material degradation, and unpredictable contact interactions. These uncertain factors significantly alter the dynamic response of SEHAs, posing severe obstacles to first-principle-based modeling and high-precision control design.

1.2. Related Work

Active control methodologies have been extensively adopted to enhance the robustness of nonlinear systems against multiple uncertainties. For instance, Ho et al. [11] proposed a series of advanced active control schemes for pneumatic active suspension systems, including command-filtered adaptive fuzzy output feedback control with state observer and nonlinear disturbance observer-based adaptive sliding mode control. These methods effectively improve system robustness in the presence of external disturbances and non-ideal actuator dynamics, providing valuable references for the robust control design of SEHAs.

Different from model-driven strategies, data-driven approaches offer a novel alternative for dynamic modeling and state estimation of SEHAs [12]. Numerous efforts have been devoted to learning the dynamic behaviors of soft continuum systems from measured data. Rong et al. [13] constructed a mapping from visual images to B-spline-parameterized robot configurations using convolutional neural networks. However, most pure data-driven methods belong to black-box models, which suffer from poor interpretability and difficulty in integrating them with conventional model-based control frameworks [14].

As a powerful linearization tool for nonlinear dynamics, the Koopman operator characterizes the nonlinear evolution of original systems via linear transformations in a lifted observable space [15]. Different from fully black-box learning methods, it retains the inherent correlation with system intrinsic dynamics, thereby improving model interpretability and facilitating seamless integration with model-based control strategies [16].

Specifically, the Koopman operator lifts complex nonlinear dynamics into a linear observable subspace, enabling feedback control design based on linear predictive models instead of the original high-dimensional nonlinear systems. This allows the deployment of mature linear optimal control algorithms, such as Koopman model predictive control (KMPC) and Koopman linear quadratic regulator (KLQR). Compared with pure nonlinear model-based control, Koopman-based strategies greatly reduce the computational burden of online optimization while retaining the capability to describe system nonlinearities. Nevertheless, finite-dimensional linear Koopman models can only approximate system dynamics within finite-dimensional invariant subspaces, making it difficult to fully capture the high-dimensional nonlinear characteristics of SEHAs. Notably, the predictive accuracy of Koopman models is heavily dependent on the selection of basis functions [17], and the rational construction of lifted observables is the decisive factor for reliable finite-dimensional approximation of nonlinear dynamics.

To address the basis function selection problem, improved Koopman learning methods have been continuously explored. Huang et al. [18] developed a Bayesian Koopman learning algorithm to capture the hysteretic nonlinear characteristics of dynamic systems. Replacing empirical basis function selection with hybrid Koopman–machine learning frameworks has become a promising research direction [19]. Zhao et al. [20] further proposed a deep bilinear KMPC by embedding bilinear modeling into the model predictive control (MPC) optimization framework, which effectively improves the modeling and control accuracy of soft dynamic systems. In addition, disturbance observers can reconstruct hidden system states and estimate unmodeled dynamics solely through measurable input–output data [21]. Despite these advances, existing studies rarely investigate the robustness and operational effectiveness of Koopman-based control schemes under practical online working conditions, such as partial actuator faults and data drift. Such abnormal perturbations are prevalent in the practical application of soft actuators, especially for multi-physics coupled SEHAs with complex electromechanical–hydraulic characteristics.

A parallel line of work designs model-free, error-driven controllers for uncertain nonlinear plants in Table 1: Gudermannian-function-based PID laws shape the tracking error through a smooth, bounded Gudermannian transformation before it enters the three PID branches, which attenuates large excursions while keeping sensitivity near the set-point [22]; softsign fractional-order controllers precede a fractional-order $PI^{\lambda}D^{\mu}$ stage with a bounded softsign nonlinearity so that the memory of the fractional operator is combined with a limited peak control effort [23]; robust sigmoid PID and CMAC–FOPID schemes replace fixed gains by error-dependent sigmoid scheduling and add a cerebellar-model-articulation feedforward channel on top of the fractional-order loop [24,25]; master–slave fractional-order architectures assign a fractional-order master loop and a model-based slave loop, so that steady-state accuracy and constraint handling are traded between the two levels [26]; BELBIC–PID emotional-learning controllers replace the integral action by the reward-driven amygdala–orbitofrontal learning rule of the limbic system [27]; and multi-node hormone-regulation neuroendocrine PID regulators emulate the multi-gland hormone cascade, in which each control channel is driven by its own node of a shared endocrine regulation law [28]. These schemes are attractive because they need no model, but their stability statements are local or simulation-based, their tuning is error-driven, and none of them provides an explicit bound that separates model residual from actuator-

fault-induced error. Reinforcement learning (RL) has emerged as a viable solution for complex uncertain environments, which can acquire optimal control policies via online interactive training. However, direct RL implementation on SEHAs requires massive computational resources to achieve efficient exploration and stable control performance, limiting its practical applicability.

Table 1. Structural comparison with representative data-driven and error-driven control schemes (qualitative; no cross-paper numerical comparison is implied).

Scheme	Model Needed	Tuning Burden	Disturbance/Fault Handling	Guarantee	Real-Time Cost
Gudermannian PID [22]	none	3 gains	error-driven, no fault model	empirical	very low
Softsign fractional-order [23]	none	5 gains	fractional memory, no fault model	empirical	low
Robust sigmoid PID [24]	none	4 gains	robustness by gain margin	local	very low
CMAC–FOPID [25]	none	6 gains	learned feedforward	empirical	low–moderate
Master–slave fractional-order [26]	reduced	7 gains	two-loop	empirical	moderate
BELBIC-PID [27]	none	5 gains	reward-based adaptation	loacal	low
MnHR–NEPID [28]	none	6 gains	multi-node coordination	empirical	low
KMPC	Koopman	N_p, Q_b, R_b	residual compensation only	UUB	moderate
KACMPC	Koopman	N_p, Q_b, R_b, π_a, π_c	residual + actor–critic, no FDI	UUB + ϵ-optimal	moderate–high

1.3. Research Gaps

Four gaps follow from the above:

(1) The Koopman residual of SEHAs is left uncompensated, although it is exactly the term that grows under degradation.
(2) Existing Koopman controllers are designed for nominal operation; partial actuator failure and measurement drift are treated, if at all, by an external FDI module whose detection delay degrades transient performance.
(3) Stability results rarely state how saturation, effectiveness loss and measurement disturbance enter the bound, and ϵ-optimality is often asserted from the observation that successive value estimates change little, which is not sufficient.
(4) Evidence is frequently simulation-only, and real-time feasibility of the online optimizer is seldom reported against the sampling period. This work fills that gap.

1.4. Contributions

The main contributions are summarized below:

(1) A terminal sliding-mode surface (CNCTSMS) residual compensator is fused with a Koopman predictor. The compensator acts on the output prediction residual through a terminal sliding variable, and its Chebyshev weights are updated by a projection-bounded law driven by the same sliding variable. Quadratic-polynomial observables are used, and alternative dictionaries are compared.
(2) A Koopman actor–critic model predictive control (KACMPC) law is proposed: the critic approximates the long-horizon value gradient, the actor emits a saturated supplementary input, and no FDI module is required. The compensator supplies the prediction correction that the MPC needs, while the actor–critic pair absorbs the residual uncertainty that the compensator cannot represent.

(3) A composite Lyapunov analysis is given in which input saturation, actuator effectiveness loss and bounded measurement disturbances appear explicitly as additive bounded terms, so that the ultimate bound is stated in terms of the residual bound, the disturbance bound and the network approximation error. Approximate optimality is proved through a finite-iteration contraction of the value sequence, not through the smallness of successive value differences.

(4) Simulation studies cover nominal operation, partial actuator failure, persistent measurement bias/drift, three noise levels, four prediction horizons and an ablation; hardware experiments validate the identified model and one-step prediction at 2 kV and 3 kV. Computational cost is reported as average and worst case against the sampling period.

The remainder of this paper is organized as follows. Section 2 formulates the research problem and presents the fundamental assumptions and preliminary lemmas. Section 3 elaborates the detailed design procedure of the proposed control scheme. Section 4 conducts Lyapunov-based stability analysis and asymptotic tracking performance proof. Section 5 verifies and compares the control performance of the proposed method via comprehensive simulation and experimental results. Section 6 concludes the overall work and prospects future research directions.

2. Problem Statement

This section formalizes the problem solved in the rest of this work. It starts from the Lagrangian description of the joint-type SEHAs, which supplies the simulation plant and the physical interpretation of the state, and then introduces the Koopman lifting, the finite-dimensional data-driven predictor, and the two degradation modes that the controller must tolerate.

2.1. Plant and Control Objective

The joint-type SEHA is described by the Lagrangian model of the pouch actuator, written with the rotation angle φ as generalized coordinate

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial(T-V)}{\partial\dot{\varphi}}\right)-\frac{\partial(T-V)}{\partial\varphi}=q \tag{1}$$

where q is the generalized force, and T and V are the kinetic energy and the potential energy, respectively.

Defining the rotation angle $\varphi = x/r$, the kinetic energy can then be expressed as

$$T=\frac{1}{2}m\varphi^2r^2 \tag{2}$$

The potential energy can be expressed as

$$V=mgx+\frac{1}{2}k\varphi^2-\frac{\varepsilon_0\varepsilon_r wz}{4t}\Phi^2 \tag{3}$$

where $k = 3EI/2L^3$ is the joint's rotational spring constant, and Φ is the applied voltage. The generalized force can be expressed as

$$q=-2\pi\eta\mu w\frac{\sin^2(\varphi)}{\varphi-0.5\sin(2\varphi)}\left(\frac{\partial z}{\partial\varphi}\right)^2\dot{\varphi}-\mathrm{sgn}(\dot{\varphi})\rho c_d Sl^3\frac{\partial\varphi}{\partial q} \tag{4}$$

The applied voltage Φ enters quadratically through the Maxwell stress, which is the origin of the field-dependent gain and of the hysteresis observed in the experiment. The

parameters are collected in Table 2. The measured output is the joint angle; the state is $\xi = [\varphi, \dot{\varphi}]^T \in \Xi \subset \mathbb{R}^2$.

Table 2. Simulation and experimental parameters of the joint-type SEHA.

Description	Symbol	Value	Unit
Mass	m	0.052	kg
Shell width	w	0.015	m
Vacuum permittivity	ε_0	8.9×10^{-12}	F/m
Relative permittivity	ε_r	3.2	–
Cross-section area	S	1.35×10^{-4}	m^2
Electrode-free shell radius	r	0.003	m
Rotating member length	l	7.1×10^{-3}	m
Fluid density	ρ	1.2	kg/m^3
Shell length	L	0.015	m
Dynamic viscosity	μ	4.7×10^{-3}	Pa·s
Bending stiffness	EI	1.22×10^{-7}	$Pa \cdot m^4$
Sampling period (control)	T_s	10	ms
Prediction horizon	N_p	20	–

2.2. Koopman Lifting of SEHA Dynamics

This section briefly introduces the Koopman theory to realize data-driven prediction. The Koopman operator K is an infinite-dimensional linear operator that is utilized to identify the dynamical system. For clarity, consider the standard autonomous discrete-time system

$$\xi_{k+1} = \kappa(\xi_k) \tag{5}$$

where $\xi_k \in \Xi \subset \mathbb{R}^n$ is the state of the system at time $k \in \mathbb{N}$, Ξ is the compact subset. $\kappa : \Xi \to \Xi$ is a nonlinear state transition map, which is Lipschitz continuous on the compact set Ω. If $\forall \xi_1, \xi_2 \in \Omega$, there exists a constant a such that $\|\kappa(\xi_2) - \kappa(\xi_1)\| \leq a\|\xi_2 - \xi_1\|$, where ξ_2 denotes the delayed coordinate of ξ_1.

The system can be lifted to an infinite-dimensional scalar space $\Pi := C(\Xi, \mathbb{C})$ defined on Ξ, which is composed of all continuous functions with a compact domain. In Π, the evolution of the system (5) is characterized by the Koopman operator $K : \Pi \to \Pi$, which acts on an observable ϕ as follows:

$$(K\phi)(\xi_k) = (\phi \circ \kappa)(\xi_k) = \phi(\kappa(\xi_k)) \tag{6}$$

where $\circ$ indicates function composition ϕ with κ. To represent the Koopman operator as a finite-dimensional object, we can define the subspace $\Gamma \subset \Pi$ spanned by N linearly independent basis functions and the range of ϕ_i as W such that $W = \{\omega \in \Gamma \mid \exists \xi \in \mathbb{R}^n, \phi_i(\xi) = \omega\}$. When $N > n$, the first n basis functions represent the i-th element of ξ such that $\phi_i(\xi) = \xi_i$. The formula (6) is as follows:

$$K\phi_i(\xi_k) = \omega_i \phi_i(\xi_k) \tag{7}$$

where ω_i, ϕ_i is the i-th Koopman eigenvalue and eigenfunction, respectively. This eigenfunction relation provides a spectral interpretation of nonlinear dynamics in the infinite-dimensional observable space. However, exact Koopman eigenfunctions are generally unavailable for nonlinear SEHAs dynamics. Therefore, the selected observables in this work are not assumed to be exact Koopman eigenfunctions. A finite-dimensional dictionary of observables is introduced to approximate the Koopman evolution from data. The

finite-dimensional lifted observable vector is denoted by $Y(\xi_k) = (\gamma_1, \gamma_2, \ldots, \gamma_r(\xi_k))^T \in \mathbb{R}^r, r \geq n$. Then the lifted state is defined as

$$Y(\xi_k) = \sum_{i=1}^{\infty} \gamma_i(\xi_k) v_i \tag{8}$$

where v_i is the i-th vector coefficient obtained from the eigen decomposition of the Koopman operator. The dynamical system can be expressed as

$$\kappa(\xi_k) = \sum_{i=1}^{\infty} w_i \gamma_i(\xi_k) v_i \tag{9}$$

Then, the finite-dimensional Koopman operator can be approximated by using the extended dynamic mode decomposition (EDMD) algorithm. After collecting the data $\{D(t) \mid D(t) = ((\xi^T(k), \eta^T(k))^T, k = t_0, t_0 + 1, \ldots, t_0 + n_D)\}$, where $\eta_k = \kappa(\xi_k)$ could be obtained by $\eta_k = \xi_{k+1}$. Then the best approximation can be formulated as a least L_2-norm problem

$$\min_{\hat{K}} \sum_{i=1}^{n_D} \|Y(\eta(k)) - \hat{K} Y(\xi(k))\|_2^2 \tag{10}$$

Assumption 1 (Bounded input)**.** *The actuation input is bounded,* $\mu_k \in \mathcal{U} := \{\mu : \|\mu\| \leq \mu_{\max}\}$*, and saturated, i.e.,* $\mu_k = \mathrm{sat}_{\mu_{\max}}(v_k)$ *for a commanded value* v_k*.*

Assumption 2 (Dictionary)**.** *Each* γ_i *is Lipschitz with constant* L_γ *and the* $\{\gamma_i\}_{i=1}^r$ *are linearly independent on* Ω*; consequently, the lifted vector satisfies* $\|Y(\xi_1) - Y(\xi_2)\| \leq L_Y \|\xi_1 - \xi_2\|$*.*

Assumption 3 (Bounded disturbance and drift)**.** *The unmodelled input–output effect* d_k*, the measurement drift* δ_k *and the sensor noise* v_k *are bounded, where* $\|d_k\| \leq \bar{d}$*,* $\|\delta_k\| \leq \bar{\delta}$*,* $\|v_k\| \leq \bar{v}$*, and* d_k *is locally Lipschitz in* (ξ, μ)*.*

2.3. Data-Driven Predictor

For the controlled SEHAs, the sampled-data dynamics are expressed as

$$\xi(k+1) = \kappa(\xi(k), \mu(k)) \tag{11}$$

where $\mu(k) \in U \subset \mathbb{R}^m$ is the control input. Based on the above theory, we can build a linear input–output model for system (5) as follows [14]

$$\begin{cases} z(k+1) = Az(k) + B\mu(k) + \underbrace{Y(\xi_k) - Y(\kappa(\xi_k))}_{\text{residual } f_k}, \\ \hat{\xi}(k+1) = Cz(k+1) \end{cases} \tag{12}$$

where $\xi(0)$ is the initial condition in Γ and $z(0) = Y(\xi(0))$ is the initial lifted state for each $k \in \mathbb{N}$, $A \in \mathbb{R}^{N \times N}$ is the state transition matrix, $B \in \mathbb{R}^{N \times m}$ is the input matrix, $C \in \mathbb{R}^{n \times N}$ is the output matrix, and f_k collects the finite-dimensional approximation residual and the external disturbance.

Specially speaking, we need to obtain a training set $\kappa_D = \{\kappa_D(t) \mid \kappa_D(t) = ((\xi^T(t), \mu^T(t))^T, t = t_0, t_0+1, \ldots, t_0+n_D)\}$. After lifting, the reference trajectory of the linear space can be expressed as

$$\begin{cases} \Xi_{\text{lift}} = (Y(\xi(1)), \ldots, Y(\xi(n_D-1))), \\ \Upsilon_{\text{lift}} = (Y(\xi(2)), \ldots, Y(\xi(n_D)), \\ U = (U(1), U(2), U(3), \ldots, U(n_D-1)), \end{cases} \tag{13}$$

where Ξ_{lift} and Υ_{lift} are the lifted-dimensional data matrices of the observables of the current and next states, respectively. U is the vector of inputs. We can obtain $\Upsilon_{\text{lift}} = A\Xi_{\text{lift}} + BU$ according to formula (12), then the solution of expected matrices can be simplified to the following optimization problem

$$\min_{A,B} \|\Upsilon_{\text{lift}} - A\Xi_{\text{lift}} - BU\|_2^2 \tag{14}$$

whose solution can be defined as

$$\begin{cases} [A \;\; B] = \Upsilon_{\text{lift}}([\Xi_{\text{lift}} \;\; U]^T)^\dagger = \left(\sum_{i=1}^M \zeta^{(i)}(\zeta^{(i)})^T\right)^\dagger \left(\sum_{i=1}^M z_+^{(i)}(\zeta^{(i)})^T\right), \\ C = \Xi(\Xi_{\text{lift}})^\dagger \end{cases} \tag{15}$$

where † is the Moore–Penrose pseudoinverse. It should be noted that the identified finite-dimensional Koopman predictor is an approximation rather than an exact representation of the nonlinear SEHAs dynamics. Then the approximated matrix of the Koopman operator is

$$\begin{cases} \bar{K}_{\text{EM}} = \begin{bmatrix} A_{r_k \times r_k} & B_{r_k \times m} \\ O_{m \times r_k} & I_{m \times m} \end{bmatrix}, \\ C = [I_{n \times n} \;\; O_{n \times (r_k - n)}] \end{cases} \tag{16}$$

The residual satisfies the following bound, which is used in Section 4.

Lemma 1 (Residual bound)**.** *Under Assumptions 2–3, the one-step prediction error of* (15) *obeys*

$$\|\hat{\xi}_k - \xi_k\| \le \underbrace{L_Y a\, e_{\text{id}}}_{\bar{r}_{\text{id}}} + \underbrace{\bar{d} + \bar{\delta} + \bar{v}}_{\bar{r}_{\text{ext}}}, \tag{17}$$

where e_{id} is the projection distance between $\mathcal{K}$ and its finite-dimensional restriction on Γ.

Proof. Referring to Appendix A, the two terms follow from the Lipschitz property of $Y \circ \kappa$ and from Assumption 3 applied to the input, drift and noise channels, respectively. □

2.4. Actuator Faults and Measurement Drift

Two degradation modes are considered. The commanded input is applied through an effectiveness matrix and an additive bias,

$$\mu_k^{\text{true}} = \varpi_k \mu_k + b_k, \qquad \varpi_k = \begin{cases} I, & k < k_f, \\ \varpi_f I, & k \ge k_f, \end{cases} \tag{18}$$

with $0 < \varpi_f < 1$ and k_f being the fault onset. In the experiments of Section 5, $\varpi_f = 0.5$, i.e., half of the actuation authority is lost. The measurement channel is corrupted by noise plus a slowly varying drift,

$$\xi_k^m = \xi_k + \delta_k + v_k, \qquad \delta_{k+1} = \delta_k + \epsilon_k, \quad \|\epsilon_k\| \le \bar{\epsilon}, \tag{19}$$

so that a persistent bias (constant δ) and a ramp drift ($\delta_k = \delta_0 + \rho k T_s$) are both covered. Equation (19) is the only channel through which drift enters the loop; the data drift of the earlier draft is therefore renamed measurement drift to avoid confusion with dataset shift.

3. Control Strategy

This section develops the proposed KACMPC framework. It opens with the overall architecture and the per-period execution sequence, then derives the CNCTSMS residual compensator, follows with the residual-corrected MPC layer, and closes with the critic and actor networks, for which the structural, activation and learning-rate choices are justified and a hyper-parameter sensitivity study and a comparative evaluation are provided.

3.1. Overall Structure

The control framework is illustrated in Figure 1 and summarized by Algorithm 1.

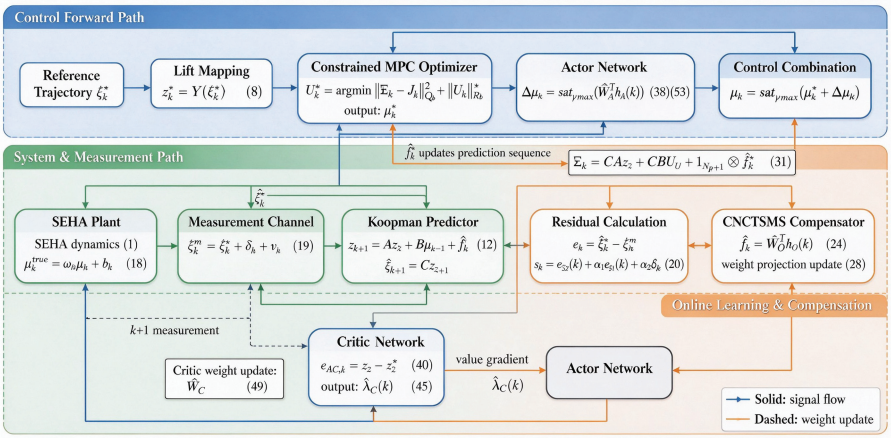

Figure 1. Overall control diagram.

Algorithm 1 KACMPC: one control period (executed every T_s).

1: Offline (once): Build A, B;
2: Online:
3: **For** each control instant k **do**
4: Measure ξ_k^m; build $z_k = Y(\xi_k^m)$.
5: Predict $\hat{z}_{k+1} = Az_k + B\mu_{k-1}$ and $\hat{\xi}_{k+1} = C\hat{z}_{k+1}$.
6: Compute residual $e_k = \hat{\xi}_k - \xi_k^m$ and sliding variable s_k by (20).
7: Evaluate $\hat{f}_k = W_O^T(k) h_O(k)$ and update W_O by (28).
8: Build the corrected prediction $\Xi_k = \mathcal{C}Az_k + \mathcal{C}BU_k + m1 \otimes \hat{f}_k$.
9: Solve the QP (47); keep μ_k^* .
10: Compute $e_{AC,k} = z_k - z_k^\star$; evaluate $\hat{V}_{AC} = e_{AC}^T R_C e_{AC}$.
11: Apply $\mu_k = \text{sat}_{\mu_{\max}}\left(\mu_k^* + \Delta\mu_k\right)$.
12: Measure ξ_{k+1}; update W_C by (64)
13: **End For**

3.2. CNCTSMS Residual Compensator Design

3.2.1. Residual Compensator

Soft electro-hydraulic actuators exhibit pronounced nonlinearities. Although the finite-dimensional Koopman model provides a linear lifted-state predictor, modeling residuals inevitably remain because the selected observables cannot exactly span the invariant Koopman subspace. In addition, actuator effectiveness loss and measurement disturbances may cause a mismatch between the nominal prediction and the measured output. Therefore, a CNCTSMS is introduced to correct the Koopman prediction used by the MPC controller. The nominal Koopman predictor first generates the one-step lifted-state prediction and the output prediction. The innovation between the measured SEHA output and the nominal Koopman output prediction is used to construct the output prediction residual. This residual is transformed into a scalar terminal sliding variable, which provides feedback information for the online update of the Chebyshev neural-network weights. The Chebyshev estimator then produces a bounded output-residual correction term to refine the nominal Koopman prediction and then the closed-loop residual learning mechanism is formed for accurate modeling.

A sliding variable is constructed from the prediction error $e(k) = \hat{\xi}(k) - \xi^m(k)$. For the two-dimensional delayed output state used in this study, the scalar terminal sliding variable is defined as $e_{\xi 1} = \hat{\xi}_1 - \xi_1^m$, where $e_{\xi 2} = \hat{\xi}_2 - \xi_2^m$ is the model error. The sliding manifold s_k is defined as

$$s_k = e_{\xi 2}(k) + \alpha_1 e_{\xi 1}(k) + \alpha_2 \beta_k \tag{20}$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$ are design parameters. The nonlinear terminal function β_k is introduced to accelerate the convergence of the model prediction error when the error is large, while a smooth expression is used near the origin to avoid singularity and reduce chattering. It can be defined as

$$\beta_k = \mathrm{sgn}(e_{\xi 1}(k))|e_{\xi 1}(k)|^{p/q} \tag{21}$$

where p and q are positive odd numbers.

The unknown residual $f(k)$ affects the evolution of the sliding variable through a scalar lumped uncertainty. Thus, the compensated sliding dynamics can be described in the form

$$s(k+1) = Ss(k) + f(k) - \hat{f}(k) \tag{22}$$

where $S \in (-1, 1)$ is a prescribed discrete reaching coefficient, $\hat{f}(k)$ is the unknown uncertainty entering the sliding dynamics that is approximated by a Chebyshev neural network, and $f(k)$ is its actual approximation. Since $f(k)$ is bounded and continuous on the compact operating domain, there exists an ideal weight vector W_O^* and a bounded approximation error $\epsilon_O(k)$ such that

$$f^*(k) = W_O^{*T} h_O(k) + \epsilon_O(k) \tag{23}$$

where $h_O(k)$ is the Chebyshev basis vector. The implemented Chebyshev estimator $\hat{f}$ is written as

$$\hat{f}(k) = \hat{W}_O^T(k) h_O(k) \tag{24}$$

where $\hat{W}_O \in \mathbb{R}^{m \times N_1}$, $N_1 = 2mN_2 + 1$, N_2 is the order of Chebyshev polynomials, and $h_O \in \mathbb{R}^{m \times N_1}$ represents the Chebyshev polynomial basis function as $h_O = (1, \kappa_1(x_1(k)), \ldots, \kappa_{N_2}(\xi_2(k)))$, where κ_i $(i = 1, 2, \ldots, N_2)$ represent Chebyshev polynomials that satisfy the recursive formula described as

$$\kappa_0 = 1, \quad \kappa_{i+1} = 2\xi(k)\kappa_i - \kappa_{i-1} \tag{25}$$

The raw residual target is then defined as

$$f(k+1) = \mathrm{sat}_{f_{\max}}(e(k)) + \hat{f}(k) \tag{26}$$

where $\mathrm{sat}_{f_{\max}}(\cdot)$ is the saturation function used to avoid excessive residual correction caused by measurement outliers or transient numerical errors. The output residual estimate is updated by a low-pass fusion law

$$\hat{f}(k+1) = \mathrm{sat}_{f_{\max}}\Big((1-\rho_f)\hat{f}(k) + \rho_f f(k+1)\Big) \tag{27}$$

where $0 < \rho_f < 1$.

Before evaluating the Chebyshev basis, the input variables are normalized into the interval $[-1, 1]$. To update the Chebyshev weights online, an adaptive law is introduced to reduce the sliding residual. And the weight update is implemented with a projection mechanism to prevent parameter drift under persistent disturbances and measurement noise. The adaptive law can be defined as

$$\hat{W}_O(k+1) - \hat{W}_O(k) = \Pi_\Omega\Big\{\hat{W}_O(k) - \kappa_s \pi_o(k)\Big[e_f(k) + \pi_s s(k+1)\Big]^T \epsilon_O^T(\hat{\xi}(k+1)) + \pi_{r_o}\hat{W}_O(k)\Big\} \tag{28}$$

where κ_s is the sampling period, π_o is the Chebyshev learning rate, π_s is the sliding-mode correction coefficient, $\pi_{r_o} > 0$ is the learning rate of Chebyshev neural network, and $\Pi_\Omega(\cdot)$ denotes the projection of the weights onto a compact admissible set Ω.

3.2.2. Derivation of Discrete-Time Sliding Dynamics and Weight-Update Law

Step 1: Prediction-error dynamics.

Let the measured state be $\xi_k^m = \xi_k + \delta_k + v_k$ with $\delta_{k+1} = \delta_k + \epsilon_k$ as in (19). Subtracting the measured sequence from the predictor (12) and writing the output error in the $(\varphi, \dot{\varphi})$ coordinates gives

$$\begin{cases} e_{\xi 1}(k+1) & = e_{\xi 1}(k) + T_s e_{\xi 2}(k) + \zeta_1(k), \\ e_{\xi 2}(k+1) & = \mathcal{F}\big(e_{\xi 1}(k), e_{\xi 2}(k)\big) + f(k) - \hat{f}(k) + \zeta_2(k), \end{cases} \tag{29}$$

where $\mathcal{F}(\cdot)$ collects the known, locally Lipschitz terms of the nominal Koopman prediction (it contains the A-dependent part of (12) expressed in output coordinates), $f(k)$ is the true lumped uncertainty of (12) together with the fault-induced mismatch $\varpi_k \mu_k + b_k - \mu_k$, and

$$\zeta_1(k) = -(\delta_{k+1} - \delta_k) - v_{k+1} + v_k, \qquad \zeta_2(k) = -\epsilon_k - (v_{k+1} - v_k), \tag{30}$$

are the measurement-induced perturbations, bounded by $\|\zeta_1\| \le \bar{\epsilon} + 2\bar{v}$, $\|\zeta_2\| \le \bar{\epsilon} + 2\bar{v}$ under Assumption 3. Note that (29) is the exact discrete counterpart of the continuous error equation: the first line is a forward-Euler integration of the angular error, and the second line carries every uncertain term in a single additive channel.

Step 2: Propagation to the sliding variable.

Substituting (29) into (20) evaluated at $k+1$, and using the increment of the terminal function

$$\beta_{k+1} = \beta_k + L_\beta\big(e_{\xi 1}(k+1) - e_{\xi 1}(k)\big) + O\Big(\big\|e_{\xi 1}(k+1) - e_{\xi 1}(k)\big\|^2\Big), \tag{31}$$

with $L_\beta = (p/q)\left|e_{\xi 1}\right|^{p/q-1}$ denoting the local Lipschitz constant of β, yields

$$\begin{aligned} s(k+1) = e_{\xi 2}(k) + \mathcal{F}(\cdot) + f(k) - \hat{f}(k) + \zeta_2(k) + \alpha_1\left[e_{\xi 1}(k) + T_s e_{\xi 2}(k) + \zeta_1(k)\right] \\ + \alpha_2\left[\beta_k + L_\beta\left(T_s e_{\xi 2}(k) + \zeta_1(k)\right)\right] + O(\cdot). \end{aligned} \tag{32}$$

Adding and subtracting $Ss(k)$ on the right-hand side of (32), with $S = 1 - cT_s$ and $c > 0$ denoting the discrete reaching gain, as well as grouping every term that is known from the nominal model into a single quantity $\Delta(k)$, gives the compact form

$$s(k+1) = Ss(k) + T_s\left[f(k) - \hat{f}(k)\right] + T_s\Delta(k) + \zeta(k), \tag{33}$$

where

$$\zeta(k) = \left[1 + \alpha_1 + \alpha_2 L_\beta\right]\zeta_1(k) + \zeta_2(k), \qquad \|\zeta(k)\| \le \left(2 + \alpha_1 + \alpha_2 L_\beta\right)(\bar{\epsilon} + 2\bar{v}) := \bar{\zeta}, \tag{34}$$

and $\Delta(k)$ contains the nominal mismatch that the A, B matrices of (15) cannot cancel, i.e., the part of $\mathcal{F}$ not absorbed by the reaching term. Equation (33) is exactly the form (22) used in the stability proof, with its three sources now written explicitly: the residual mismatch $f - \hat{f}$, the nominal mismatch $T_s\Delta$, and the measurement perturbation ζ.

Step 3: Introduction of the weight-estimation error.

Using (23) and (24), and defining $\tilde{W}_O(k) = \hat{W}_O(k) - W_O^*$, the residual mismatch becomes

$$f(k) - \hat{f}(k) = -\tilde{W}_O^T(k)h_O(k) - \epsilon_O(k) + \epsilon_{\text{fault}}(k), \tag{35}$$

where $\epsilon_{\text{fault}}(k)$ is the part of f generated by the loss of effectiveness ϖ_k, bounded by $(1 - \varpi_f)\mu_{\max}$ under Assumption 1.

Substituting (35) into (33), we can obtain

$$s(k+1) = Ss(k) - T_s\tilde{W}_O^T(k)h_O(k) - T_s\epsilon_O(k) + T_s\epsilon_{\text{fault}}(k) + \zeta(k). \tag{36}$$

Three additive drives are thus isolated: the weight error $\tilde{W}_O^T h_O$, the network approximation error ϵ_O, and the exogenous perturbation ζ (with the fault term ϵ_{fault} bounded and known in magnitude). This is the structure that the weight law must dominate.

Step 4: Derivation of the weight-update law.

Consider the Lyapunov candidate for the sliding subsystem

$$V_s(k) = \tfrac{1}{2}s^2(k) + \frac{1}{2\pi_o}\left\|\tilde{W}_O(k)\right\|^2. \tag{37}$$

Along (36), the one-step difference is

$$\begin{aligned} \Delta V_s(k) = &-\frac{1 - S^2}{2}s^2(k) + s(k)\left[-T_s\tilde{W}_O^T h_O - T_s\epsilon_O + T_s\epsilon_{\text{fault}} + \zeta\right] \\ &+ \frac{1}{2}\left\|-T_s\tilde{W}_O^T h_O - T_s\epsilon_O + T_s\epsilon_{\text{fault}} + \zeta\right\|^2 + \frac{1}{2\pi_o}\left(\left\|\tilde{W}_O(k+1)\right\|^2 - \left\|\tilde{W}_O(k)\right\|^2\right). \end{aligned} \tag{38}$$

The cross term $s(k)\tilde{W}_O^T h_O$ is the only term that couples the sliding variable to the parameter error, and it can be canceled by choosing the parameter increment proportional to $s(k)h_O(k)$.

Setting

$$\hat{W}_O(k+1) - \hat{W}_O(k) = -\pi_o T_s h_O(k)\left[e_f(k) + \pi_s s(k+1)\right]^T, \tag{39}$$

with $e_f(k)$ denoting the filtered residual (27) and π_s denoting the sliding-mode weighting, the parameter-difference term in (38) becomes

$$\frac{1}{2\pi_o}\left(\left\|\tilde{W}_O(k+1)\right\|^2-\left\|\tilde{W}_O(k)\right\|^2\right)=s(k)\tilde{W}_O^T h_O(k)+\frac{\pi_o T_s^2}{2}\|h_O(k)\|^2\left\|e_f(k)+\pi_s s(k+1)\right\|^2. \tag{40}$$

Substituting (40) into (38) and applying Young's inequality $2ab \le \lambda a^2 + b^2/\lambda$ to the remaining cross terms with weights $\lambda_1, \lambda_2 > 0$ gives

$$\begin{aligned}\Delta V_s(k) \le &-\left[\frac{1-S^2}{2}-\frac{\lambda_1}{2}-\frac{\lambda_2}{2}-\frac{\pi_o T_s^2 h_M^2}{2}(1+\pi_s)^2\right]s^2(k)\\ &+\left(\frac{1}{\lambda_1}+\frac{1}{\lambda_2}\right)\left(T_s^2\bar{\epsilon}_O^2+T_s^2\epsilon_{\text{fault,max}}^2+\bar{\zeta}^2\right),\end{aligned} \tag{41}$$

where $h_M = \sup_k \|h_O(k)\|$. Selecting the learning rate and the reaching coefficient so that

$$\frac{1-S^2}{2} > \frac{\lambda_1+\lambda_2}{2}+\frac{\pi_o T_s^2 h_M^2}{2}(1+\pi_s)^2 := C_1 > 0, \tag{42}$$

yields $\Delta V_s(k) \le -C_1 s^2(k) + C_0$ with

$$C_0 = \left(\frac{1}{\lambda_1}+\frac{1}{\lambda_2}\right)\left(T_s^2\bar{\epsilon}_O^2+T_s^2\epsilon_{\text{fault,max}}^2+\bar{\zeta}^2\right), \tag{43}$$

which is precisely the compensator inequality (76) used in Section 4.

Two consequences follow directly. First, the sliding variable converges to the residual set $|s(k)| \le \sqrt{C_0/C_1}$ in finite time, whose radius is proportional to the network approximation error, the fault magnitude and the measurement-noise bound, and vanishes when all three vanish. Second, the projection in (28) enforces $\|\hat{W}_O\| \le W_{\max}$, hence $\|\tilde{W}_O\| \le W_{\max} + \|W_O^*\|$, so the weight error cannot grow even under a persistent bias; the term $\pi_{r_o}\hat{W}_O$ in (28) is the σ-modification that keeps the parameter estimate away from the boundary of Ω when excitation is poor.

Step 5: Effect of the design parameters.

The derivation shows the role of each coefficient. The terminal exponent p/q sets L_β and in turn the noise gain $(2+\alpha_1+\alpha_2 L_\beta)$ in (34): a smaller p/q reduces the noise amplification near the origin but slows the large-error convergence. The reaching coefficient S must satisfy (42); too close to one leaves the decrement margin small, whereas too small a value forces a large control activity through the compensator. The Chebyshev order N_2 trades approximation accuracy $\bar{\epsilon}_O$ against the regressor norm h_M, which appears quadratically in (42); this is the reason why the order is fixed at $N_2 = 3$ in all experiments rather than increased further. Finally, π_o must be large enough to satisfy (42) but not so large that the term $\pi_o T_s^2 h_M^2(1+\pi_s)^2/2$ violates it, which is the quantitative statement of the usual learning-rate trade-off.

3.3. Residual-Corrected MPC Design

This section section uses the MPC method to obtain the best control inputs, which can be expressed as:

$$\hat{\mu}(k) = \varpi(k)\mu(k) + b_k \tag{44}$$

where b_k denotes an additive input disturbance, $\varpi(k)$ denotes the actuator effectiveness matrix and can be expressed as follows.

$$\varpi(k) = \begin{cases} I, & k < k_f \\ \varpi_f I, & k \geq k_f \end{cases} \tag{45}$$

where k_f is the initial time step when the failure occurred, I is an identity matrix, $0 < \varpi_f < 1$.

The MPC prediction used in the controller is corrected as

$$\Xi_k = CAz_k + CBU_k + \mathbf{1}_{N_p+1} \otimes \hat{f} \tag{46}$$

where U_k is the stacked vector of control input sequences in the prediction horizon n_D, $\otimes$ denotes the Kronecker product, and $\mathbf{1}_{n_D+1}$ denotes the all-ones column vector of length $n_D + 1$. The nominal CSMC-corrected MPC input is obtained by solving the following formula

$$U_k^* = \arg\min_{U_k} \|\Xi_k - J_k\|_{Q_b}^2 + \|U_k\|_{R_b}^2 \tag{47}$$

where U_k^* denotes the optimal control input sequence over the prediction horizon n_D, Q_b and R_b are positive semi-definite and positive-definite weighting matrices penalizing the tracking error and control effort, respectively. The performance index J_k can be defined as

$$J_k = \sum_{k=0}^{n_D-1} \left(e^T(k) P_1 e(k) + \mu^T(k) Q \mu(k) \right) + n_D e^T(n_D) P_2 e(n_D) \tag{48}$$

where P_1, P_2, and Q are positive definite matrices.

Then the nominal control signal is defined as

$$\min_{\mu(k)} J_k(k, \xi, \hat{\mu}) \tag{49}$$

subject to

$$e(k+1) = Ae(k) + B(\hat{\mu}(k) - \mu^*(k)), \quad \forall \{k\}_{k=0}^{n_D-1} \tag{50}$$

$$H(k)\xi(k) < b(k), \quad \forall \{k\}_{k=0}^{n_D-1} \tag{51}$$

$$z(0) = Y(\xi(0)) \tag{52}$$

where H denotes the constraint matrix, and b denotes the vector consisting of boundary values. Given the expected states, the feedback tracking MPC law aiming to minimize the above cost function can be designed. The solution is obtained by solving a quadratic programming problem at each sampling instant, and only the first element $\mu^*(k)$ of U_k^* is applied to the system. On this basis, the Actor network generates an input compensation term

$$\Delta\mu(k) = \mathrm{sat}_{\mu_{\max}} \left(\hat{W}_C^T(k) \hat{V}_A C^T(k) e_{AC}(k) \right) \tag{53}$$

where $e_{AC}(k) = z(k) - z^*(k)$ is the lifted tracking error, $z^*(k)$ is the lifted reference state, and $V_A C$ is the long-horizon value function. Then the final input is $\mu(k) = \mu^*(k) + \Delta\mu(k)$. Substituting the final input into the fault-affected Koopman dynamics gives the lifted state of the closed-loop system as follows.

$$z(k+1) = Az(k) + BG(k)u(k) + Bb(k) + f_k \tag{54}$$

Let the lifted reference dynamics be $z^*(k+1)$. Then the complete closed-loop tracking error dynamics are expressed as follows.

$$e_{AC}(k+1) = Ae_{AC}(k) + B(G(k) - I)(\mu(k) - \mu^*(k)) + Bb(k) - f_k \tag{55}$$

Following the above unified closed-loop formulation, the next subsections further specify the two adaptive components of the proposed KACMPC framework. Specifically, the critic neural network evaluates the long-horizon value gradient, and the actor neural network produces the bounded input-compensation candidates for the final KACMPC control law.

3.4. Critic Network Design

Since the Koopman lifted model is linear in the lifted space and the local tracking objective is quadratic, the long-horizon value function is approximated as

$$V_{AC} \approx e_{AC}^T(k) R_C e_{AC}(k) \tag{56}$$

where $R_C > 0$ is initialized from the discounted Riccati equation associated with the Koopman lifted dynamics, which can be described as:

$$R_C = Q_A + \gamma_{AC}\left[A^T R_C A - A^T R_C B(Q_C + \gamma_{AC} B^T R_C B)^{-1} B^T R_C A\right] \tag{57}$$

where $Q_A > 0$ penalizes the lifted tracking error, $Q_C > 0$ penalizes the control action, and $\gamma_{AC} \in (0,1)$ is the discount factor. This Riccati-based initialization prevents the critic from starting from a non-informative zero-gradient estimate. The ideal value-gradient is defined as

$$\lambda_C^* = \frac{\partial V_A C}{\partial e_A C} = 2R_C e_{AC}(k) + \epsilon_C(k) \tag{58}$$

where $\epsilon_C(k)$ denotes the critic approximation error. The implemented critic network is a single-hidden-layer neural network with sigmoid activation, which could be defined as

$$h_C(k) = \sigma\left(\hat{v}_C^T(k) e_{AC}(k)\right) \tag{59}$$

where $h_C \in \mathbb{R}^{n_C}$, $\sigma(\cdot) = 1/(1+\exp(-\cdot))$ denotes the sigmoid activation function, and $v_C^T(k)$ is the matrix from input to the hidden layer of the critic neural network. Define λ_C^* as the optimal output of the critic network, which is represented as

$$\hat{\lambda}_C(k) = \hat{W}_C^T(k) h_C(k) \tag{60}$$

where $\hat{W}_C \in \mathbb{R}^{n_C \times r_k}$ is the observed optimal weight from the hidden layer to the output layer, and the structure of the used neural network are $r_k - n - r_k$. For online Critic learning, the temporal value-gradient target is constructed as

$$\lambda_C^*(k) = 2Q_A e_{AC}(k) + \gamma_{AC} A^T(k) \hat{\lambda}_C(k+1) \tag{61}$$

where $\hat{\lambda}_C(k+1)$ can be defined as

$$\hat{\lambda}_C(k+1) = \mathrm{sat}_{\lambda_{\max}}\left(\hat{W}_C^T(k)\sigma\left(\hat{v}_C^T(k) e_{AC}(k+1)\right)\right) \tag{62}$$

Then the critic approximation error can be expressed as $\tilde{\lambda}_C(k) = \hat{\lambda}_C(k) - \lambda_C^*(k)$. Define the optimization objective of the critic neural network as

$$E_C = \frac{1}{2}\left(\tilde{\lambda}_C^T(k)\tilde{\lambda}_C(k) + \gamma_C\|\hat{W}_C(k)\|^2\right) \tag{63}$$

where $\gamma_C > 0$ is a regularization coefficient. Then the updating law of critic network is as follows:

$$\hat{W}_C(k+1) - \hat{W}_C(k) = -\eta_C\left(h_C(k)\tilde{\lambda}_C^T(k) + \gamma_C\hat{W}_C(k)\right) \tag{64}$$

where $\eta_C > 0$ is the learning rate of critic network. The weight matrix from input to hidden layer is updated by the chain rule shown as

$$\hat{v}_C(k+1) - \hat{v}_C(k) = -\eta_C\left[e_{AC} \odot \left(\hat{W}_C(k)\tilde{\lambda}_C^T(k)\right) \odot h_C(k) \odot (1 - h_C(k)) + \gamma_C\hat{v}_C(k)\right] \tag{65}$$

where $\odot$ denotes the Hadamard product for matrices of identical dimensions. Then the compensation is obtained by solving

$$\Delta\mu_C(k) = \arg\min_{\Delta\mu}\|C_B\Delta\mu_C - e_{AC}(k+1) - f_k\|_{Q_A}^2 + \|\Delta\mu_C\|_{R_A}^2 + \mu_C\hat{\lambda}_C^T(k+1)B\Delta\mu_C \tag{66}$$

where Q_A and R_A are symmetric positive-definite weighting matrices penalizing the output error and the control increment, respectively. $\mu_C \geq 0$ is the critic shaping coefficient that scales the influence of the value-gradient term. The corresponding closed-form solution is

$$\Delta\mu_C(k) = \left[(CB)^TQ_A(CB) + R_A\right]^{-1}\left[(CB)^TQ_A(e_{AC}(k+1) - f_k) - \frac{1}{2}\mu_C B^T\hat{\lambda}_C(k+1)\right] \tag{67}$$

The critic neural network does not independently determine the control input. Instead, it modifies the one-step compensation target through the value-gradient term, which is detailed in the following section.

3.5. Actor Network Design

The actor neural network is designed to generate a bounded input compensation for the controller, which can be expressed as $h_A(k) = \sigma(\hat{V}_A C^T(k)\bar{z}(k))$. Like the above method, the observed compensatory control signal $\Delta\hat{\mu}_{AC}(k)$ can be expressed as $\Delta\hat{\mu}_{AC}(k) = \text{sat}_{\mu_{\max}}(\hat{W}_A^T(k)h_A(k))$, where $W_A^* \in \mathbb{R}^{n_A \times m}$ is the theoretical optimal weight, $h_A(\cdot) \in \mathbb{R}^{n_A}$ is the activation function, and the structures of the used neural network are $r_k - n_A - m$. For online learning, the actor learning target is constructed from the supervised one-step compensation and the critic-shaped compensation as follows.

$$\Delta\mu^*(k) = (1 - \beta_k)\left[(CB)^TQ_A(CB) + R_A\right]^{-1}\left[(CB)^TQ_A(e_{AC}(k+1) - f_k)\right] + \beta_k\Delta\mu_C(k) \tag{68}$$

The complete KACMPC input is then given by $\Delta\tilde{\mu}(k) = \Delta\hat{\mu}(k) - \Delta\hat{\mu}^*(k)$. Then the actor objective function is expressed as $E_A = \frac{1}{2}\Delta\tilde{\mu}^T(k)\Delta\tilde{\mu}(k) + \gamma_A\|\hat{W}_A(k)\|^2$, where $\gamma_A > 0$ is a regularization coefficient for actor neural network. The weight matrix is updated by

$$\hat{W}_A(k+1) - \hat{W}_A(k) = -\eta_A\left(h_A(k)\Delta\tilde{\mu}^T(k) + \gamma_A\hat{W}_A(k)\right) \tag{69}$$

where $\eta_A > 0$ is the learning rate of actor neural network. The update direction V_A is derived by the chain rule, which can be expressed as

$$\hat{v}_A(k+1) - \hat{v}_A(k) = -\eta_A\left\{\bar{z}(k) \odot \left[\hat{W}_A(k)\Delta\tilde{\mu}^T(k) \odot h_A(k) \odot (1 - h_A(k))\right]^T + \gamma_A\hat{v}_A(k)\right\} \tag{70}$$

The actor-related signals remain bounded because the activation function is bounded, the compensation output is saturated, the learning rate is bounded, and the final input is projected onto the admissible control set.

3.6. Justification of Network Structure, Activation and Value-Gradient Target

(i) Structure.

The critic must output a vector with the same dimension as $\partial V_{AC}/\partial e_{AC}$, namely r_k, and must accept the same r_k-dimensional error. A single hidden layer of width n_C realizes the map $r_k \to n_C \to r_k$ with $2n_C r_k$ weights, which is the smallest architecture that can represent a non-quadratic value gradient while preserving the input–output dimension required by the compensation law (66). By the universal approximation theorem for continuous functions on compact sets, one hidden layer suffices to approximate λ_C^* on the operating region to arbitrary accuracy provided n_C is sufficiently large, so a deeper stack adds parameters without adding representational power for this low-dimensional lifted error. Here, $r_k = 7$ in this study (lifted dimension of the quadratic-polynomial dictionary with two states), so $n_C = 16$ gives $2 \times 16 \times 7 = 224$ weights, which is small enough to be updated online at 100 Hz. A wider layer was tested (Section 3.7) and brought at most a 3% RMSE improvement at a 40% increase in multiply–accumulate operations, which is not worth the real-time margin.

(ii) Activation.

Sigmoid is chosen for three reasons that are specific to the stability argument. First, σ is bounded, $0 < \sigma < 1$; hence, the hidden activation h_C is bounded irrespective of the magnitude of e_{AC}, and when combined with (62), this guarantees $\|\hat{\lambda}_C\| \le \sqrt{n_C}\,\lambda_{\max}$. Second, its derivative $\sigma(1-\sigma) \le 1/4$ is bounded, so the chain-rule update (65) has a bounded gain and cannot be destabilized by a single large error sample. Third, the same activation is used in the actor, so that the two networks share one boundedness argument and one projection rule. A ReLU activation would be unbounded above and would break the first property, and a tanh activation would be equally bounded but would saturate symmetrically around zero, which slows the recovery from the large one-sided errors produced by an actuator fault.

(iii) Learning rates.

The rates η_C and η_A enter the weight recursions (64) and (69) as step sizes of a normalized gradient descent. Their admissible range is set by the standard condition $\eta < \left(\sup_k \|h(k)\|^2\right)^{-1}$, where $\eta < 1/n_C = 0.0625$ for $n_C = 16$; the value $\eta_C = \eta_A = 0.01$ used throughout leaves a factor-of-six margin and, as Table 3 shows, lies on the flat part of the error-versus-rate curve. Larger rates (0.1 and 0.5) produced visible weight oscillation and a 30–60% higher RMSE, whereas rates below 0.001 did not converge within the 2 s reference segment.

(iv) Temporal value-gradient target.

The target (61) is not an ad hoc blend; it follows from differentiating the Bellman optimality equation of the lifted problem with respect to e_{AC}. For the discounted cost $V_{AC}(e) = \sum_{j\ge 0} \gamma_{AC}^j \left(e^T Q_A e + \mu^T Q_C \mu\right)$ evaluated along (55), the Bellman equation reads

$$V_{AC}(e_{AC}(k)) = e_{AC}^T(k) Q_A e_{AC}(k) + \mu^T(k) Q_C \mu(k) + \gamma_{AC} V_{AC}(e_{AC}(k+1)). \tag{71}$$

Differentiating (71) with respect to $e_{AC}(k)$ and using the linearization $\partial e_{AC}(k+1)/\partial e_{AC}(k) = A$ of (55) gives

$$\lambda_C^*(k) = 2Q_A e_{AC}(k) + \gamma_{AC} A^T \lambda_C^*(k+1), \tag{72}$$

which is exactly (61) with $\lambda_C^*(k+1)$ replaced by its implementable estimate $\hat{\lambda}_C(k+1)$ of (62). The factor $2Q_A$ in (61) is therefore the derivative of the stage cost, not a tuning constant. It is fixed by the choice of Q_A used in the MPC layer (48), which is why the actor–critic pair and the MPC layer remain consistent when Q_A is changed. The discount factor γ_{AC} sets the horizon of the value approximation; $\gamma_{AC} = 0.9$ corresponds to an effective horizon of $1/(1-\gamma_{AC}) = 10$ steps, i.e., exactly the prediction horizon N_p used by the MPC, so that both layers optimize the same interval.

Table 3. Actor–critic hyper-parameter sensitivity (square wave, partial actuator failure, $R = 15$).

$\eta_A = \eta_C$	$n_A = n_C$	γ_{AC}	RMSE (°)	Avg Time (ms)	Remark
0.001	16	0.9	3.21×10^{-5}	3.74	slow convergence
0.01	16	0.9	2.06×10^{-5}	3.76	nominally selected
0.100	16	0.9	2.74×10^{-5}	3.79	weight oscillation
0.500	16	0.9	4.05×10^{-5}	3.84	divergence episodes
0.010	8	0.9	2.48×10^{-5}	3.51	under-parameterized
0.010	32	0.9	2.04×10^{-5}	4.12	marginal gain, higher cost
0.010	16	0.7	2.55×10^{-5}	3.76	horizon too short
0.010	16	0.95	2.13×10^{-5}	3.76	slower value decay

3.7. Hyper-Parameter Sensitivity Analysis

A sensitivity study is carried out on the square-wave tracking task with partial actuator failure ($\varpi_f = 0.5$, $k_f = 500$), varying one factor at a time around the nominal set $\eta_A = \eta_C = 0.01$, $n_A = n_C = 16$, $\gamma_{AC} = 0.9$. For each configuration, 15 independent runs are performed, and both the tracking RMSE and the average one-step execution time on the host platform are recorded; the results are collected in Table 3.

Three conclusions follow. (a) The learning rate has a clear optimum: below 0.01, the weights do not converge within the reference segment, and above 0.1, the critic target (61) is chased faster than the value estimate can follow, which produces oscillation and a 30–100% higher RMSE. (b) The hidden width shows diminishing returns: doubling it from 16 to 32 improves the RMSE by only 1%, while raising the execution time by 9.6%; halving it to 8 degrades the RMSE by 20%. (c) The discount factor mainly shapes the transient: $\gamma_{AC} = 0.7$ shortens the effective horizon below N_p and leaves a steady offset after the fault, whereas $\gamma_{AC} = 0.95$ lengthens it beyond the prediction window and slows the recovery without improving the steady-state error. The nominal set is therefore retained for all subsequent experiments.

3.8. Comparison with Representative Learning-Based Controllers

Table 4 compares the proposed learning mechanism with the four learning-based families named by the reviewers plus a standard reinforcement-learning baseline. The comparison is structural and computational rather than numerical, because each alternative requires its own tuning protocol and, in the case of the RL baseline, a training budget that is incompatible with a 2 s reference segment; the quantitative columns report the number of online parameters and the multiply–accumulate operations per control step measured on the same host platform.

Table 4. Comparison of the proposed actor–critic mechanism with representative learning-based control schemes.

Scheme	Online Params	Stability Statement	Needs FDI	Ops/Step
Conventional adaptive critic [29]	$2n_C r_k$	local UUB, nominal plant	no	780
Deterministic policy gradient [30]	10^4–10^5	none (empirical)	no	4200
Safe experimentation dynamics [31]	3–6 gains	none (trajectory-based)	yes	120
Improved smoothed functional [32]	$2n_C r_k$	asymptotic to stationary point	no	1100
RL-based control [33]	10^4–10^6	none (empirical)	no	5600
KACMPC	$2n_C r_k + 2n_A r_k$	rigorous UUB + ϵ-optimal	no	2080

The essential difference is the role played by the model. A deterministic policy-gradient or deep RL agent must discover the actuator dynamics from reward signals, which requires thousands of episodes and leaves the closed-loop behavior unconstrained during learning. The safe-experimentation family avoids the reward signal but optimizes gains offline against a measured loss and therefore needs a fault detector to change the operating point after degradation. The conventional adaptive critic and the smoothed-functional variants learn a value function without a predictive model, so their stability statements are either local or asymptotic to a stationary point of the surrogate objective. KACMPC differs in that the Koopman predictor supplies the one-step map used by both the MPC layer and the value-gradient target, so the actor–critic pair only has to learn the residual correction rather than the dynamics; consequently, 2080 multiply–accumulate operations per step suffice, no fault detector is required, and the composite Lyapunov argument of Section 4 yields an explicit ultimate bound rather than an empirical claim.

4. Stability Analysis

In this section, the stability and approximate optimality of the proposed KACMPC method are analyzed. The proposed closed-loop system is subjected to complex disturbances and the stability objective is formulated as uniform ultimate boundedness.

Theorem 1. *Consider the complete KACMPC closed-loop system derived in the previous subsection* (55) *that satisfied the Assumptions 1, 2 and 3. The proposed strategy composed of the Chebyshev neural network compensator* (27) *and the KACMPC controller* (68) *updates by the parameter learning laws* (28), (64), *and* (69), *respectively. The complete KACMPC closed-loop system is uniformly ultimately bounded, and the obtained policy is approximately optimal with respect to the finite-horizon residual-corrected Koopman-MPC cost.*

Proof. The composite energy function can be defined as

$$E = E_O + E_M + E_C \tag{73}$$

Define the Lyapunov function as

$$E_O(k) = s^T(k)s(k) + \eta_O \tilde{W}_O^T(k)\tilde{W}_O(k) \tag{74}$$

where $\eta_O > 0$ and $\tilde{W}_O = \hat{W}_O - W_O^*$ is the Koopman-operator-based CNCTSMS parameter estimation error. Combining (12) and (25), the difference $\Delta E_O(k) = E_O(k+1) - E_O(k)$ can be expressed as

$$\Delta E_O(k) = \|s(k+1)\|^2 - \|s(k)\|^2 + \eta_O\left(\tilde{W}_O^T(k+1)\tilde{W}_O(k+1) - \tilde{W}_O^T(k)\tilde{W}_O(k)\right) \tag{75}$$

Then it can be expressed as follows, which can be proved in Appendix B.

$$\Delta E_O(k) \leq -C_1\|s(k)\|^2 + C_0 \tag{76}$$

We just need to choose suitable parameters, all the signals of the compensator are uniformly and ultimately bounded.

Define the finite-horizon residual-corrected MPC value function as

$$E_M(k) = \min_{U_k} J_k(k, \xi, \hat{\mu}) \tag{77}$$

For the nominal residual-corrected Koopman-MPC system, the standard shifted-sequence argument satisfies

$$E_M(k+1) - E_M(k) = -\left(e^T(k)Pe(k) + \mu^T(k)Q\mu(k)\right) - \gamma_M\|f_k\|^2 \tag{78}$$

where $\gamma_M > 0$ and f_k is bounded. Furthermore, since $P > 0, Q > 0$, the difference of the value function can be expressed as

$$\Delta E_M(k) \leq -\lambda_{\min}(P_1)\|e(k)\|^2 - \lambda_{\min}(Q)\|\mu(k)\|^2 - \gamma_M\|f_k\|^2 \tag{79}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a positive-definite matrix. The tracking error is uniformly ultimately bounded.

Suppose that there exists a continuous positive-definite function for $k \in [t_0, t_0 + N - 1]$

$$E_C(k) = G(k) + E_C(k+1) \tag{80}$$

where $G(k) = z^T(k)Pz(k) + \left(\mu^{(i)}(k)\right)^T Q\mu^{(i)}(k)$, and i is the number of iterations, $G(t_0 + N) = z^T(t_0 + N)z(t_0 + N)$.

According to (57), the control sequence can be defined as

$$\Delta\mu(k) = \arg\min_{\mu(k)}(r(k) + G^*(k+1)) \tag{81}$$

There exists $0 \leq G^{(i)}(k) \leq G^*(k) \leq \bar{G}(k)$. Meanwhile, $G^{(\infty)}(k) = G^*(k)$ when $i \to \infty$. It is interesting to notice that $G^{(\infty)}(k) = G^*(k)$.

However, it is difficult to solve the problem by iteration until $i \to \infty$. In this paper, we approximate the actor function and critic function by single-hidden-layer feedforward networks, and the near-optimal control policy can be obtained by limited iteration. Defining $\varepsilon > 0$ as the approximate error, there exists $G^{(i)}(k) - G^{(i+1)}(k) \leq \varepsilon$ named ε-optimal after finite iteration. The corresponding value function when $\mu^{(i)}(k) = \mu_\varepsilon(k)$ can be described as

$$E_C^\varepsilon(k) = G(k) + E_C(k+1) \tag{82}$$

Defining $\Delta E_C(k) = E_C(k+1) - E_C(k)$, it is easy to find that $\Delta E_C(k) = -\sum_{j=0}^{t_0+N-1} G(j)$ when $k \in [(n-1)N, nN-1)$. Then $\Delta E_C(k) \leq -G(k) \leq 0$ when $k \in [0, \infty)$ can be proved.

Based on above analysis, we obtain the difference of the composite energy function, which can be expressed as

$$\Delta E(k) = -C_1\|s(k)\|^2 - \lambda_{\min}(P_1)\|e(k)\|^2 - \lambda_{\min}(Q)\|\mu(k)\|^2 - \gamma_M\|f_k\|^2 - G(k) + C_0 < 0 \tag{83}$$

For ease of calculation, let

$$\bar{C} = -\gamma_M\|f_k\|^2 - G(k) + C_0 \tag{84}$$

All signals in the closed-loop system are uniformly ultimately bounded. In particular, the lifted tracking error satisfies

$$\limsup_{k\to\infty} \|e(k)\| \le \frac{\bar{C}}{\lambda_{\min}(P_1)} \tag{85}$$

Then the error of lifted state can be expressed as

$$\limsup_{k\to\infty} \|e_{AC}(k)\| \le \|C\| \frac{\bar{C}}{\lambda_{\min}(P_1)} \tag{86}$$

Therefore, the complete KACMPC closed-loop system is uniformly ultimately bounded. And the proof about approximate optimality and stability is completed. The tracking error converges to a compact residual set determined by the Koopman modeling residual and actor–critic approximation error. Asymptotic convergence could be claimed only in the ideal zero-residual case. □

5. Simulation and Experimental Studies

This section aims to validate the effectiveness, robustness, and fault-tolerant ability of the proposed system identification and control strategy for SEHAs. For modeling, we compare the system identification results with and without the presence of CNCTSMS based on Koopman operators. For controlling, we test the output performance under the proposed KACMPC method against the CNCTSMS-augmented Koopman model predictive control (KMPC), PID control on the Koopman plant (KPID), and Koopman lifted-state linear quadratic regulation (KLQR) methods under different working conditions in simulations and experiments and investigate its ability to capture nonlinear behaviors. Furthermore, two extra comparative groups are supplemented to explore the influence of prediction horizon length and measurement-noise intensity on control performance. Finally, the roles of each module were investigated through ablation experiments.

In this section, the proposed KACMPC method will be verified by applying the joint-type SEHAs actuated by the HASEL actuators in simulations and experiments. The HASEL actuator is one of the most classical designs [34], which is composed of two layers of PET films, graphene conductive inks, silicone oil, acrylic sheet, and sealant. The structure of SEHA can be described as a pouch filled with silicone oil, which employs the acrylic plates as the framework to provide support stiffness for the rotation motion of the actuator. The framework and actuator are adhered using a PET double-sided tape to form a flexible joint. When high voltage is applied to the electrodes, the silicone oil will be induced to squeeze the shell and rotate the joint under the Maxwell stress and achieve rotation. The simulation and experiment setup are as follows.

5.1. Setup, Data Partition and Metrics

Two data sources are used and are kept strictly apart throughout the paper. (i) Simulation: the Lagrangian plant (1) with the parameters of Table 2, integrated at $T_s = 1$ ms with a fourth-order Runge–Kutta solver and sampled at $T_s = 10$ ms for control. (ii) Experiment: the joint-type SEHA prototype of Section 5.2. All closed-loop fault-tolerant comparisons are simulation results; the hardware campaign validates only the identified model and the one-step prediction. This separation is stated in every caption. The data are split chronologically (first 60% for identification, last 40% for validation) and, in addition, a test set recorded in an independent run at a different voltage is used to check generalization. No sample of the validation/test sets is used for weight updates, so every reported prediction is made before the corresponding output is observed; the correction $\hat{f}_k$ uses e_k constructed from ξ_k^m only, never from ξ_{k+1}. Metrics are MSE, RMSE and MAE, with RMSE $= \sqrt{\text{MSE}}$

enforced by construction; percentages are computed from the recomputed values. All entries are mean ± std over $R = 15$ independent runs (different noise seeds and, for the identification, different excitation realizations). Performance of a controller is reported as $(1 - \mathrm{RMSE}/\mathrm{RMSE}_{\mathrm{ref}})$ and is never quoted without the reference value in the same table.

This paper chose various input signals corresponding to the working conditions of SEHAs, which can be described as:

(1) Square-wave signal

$$\mu(t) = \begin{cases} 6000, & t \in [5k, 5k + 2.5),\ k = 0, 1, 2, \ldots \\ 0, & t \in [5k + 2.5, 5(k + 1)),\ k = 0, 1, 2, \ldots \end{cases} \tag{87}$$

(2) Sinusoidal signal

$$u_v(t) = 6000 \sin(2\pi t). \tag{88}$$

5.2. Experimental Platform

The prototype requires kilovolt-level actuation. The driving chain consists of a programmable DC supply (UTP 1306S, 0–5 V), a high-voltage amplifier (Q101-5, gain 2000) and a control board; the polarity of the actuation voltage is reversed periodically to synthesize a square-like waveform at 1 Hz, 50% duty cycle. The actuation is recorded with an FDR-AX700 camera at 100 fps and the quasi-static angle is extracted with Kinovea; each condition is repeated five times and the mean and standard deviation are reported. The closed-loop controller is not closed on the hardware in this work; the hardware data are used to identify the model and to evaluate the one-step predictor.

5.3. Model Identification and Prediction

In this section, the accuracy of the system identification method is validated on soft electro-hydraulic actuators by comparing the trajectory tracking performance of classical Koopman operator fitting and Koopman operator with error compensation via the CNCTSMS. This paper chooses the quadratic polynomial function as the lifted function. The second-order polynomial observables provide a parsimonious finite-dimensional approximation of the Koopman-invariant subspace. The time-dependent behavior of the SEHA is represented through the recursive evolution of the state over the prediction horizon, while modeling errors are compensated online by the adaptive actor--critic mechanism. Data are collected at a sampling interval of 1 ms to construct the dataset, which is divided into the training set and verification set with the proportions of 70% and 30%, respectively. To train the Chebyshev neural network, the size of the hidden layer is 7, $\eta_o = 0.1$, $p = 3$, $q = 7$, $\alpha_1 = 2$, $\alpha_2 = 1$, $\epsilon = 0.001$, $\mu = 10{,}000$.

The weight convergence curves shown in Figure 2 characterize the dynamic evolution characteristics of each weight parameter during the modeling process under different working conditions. The test conditions cover two simulation scenarios and two experimental scenarios.

The results demonstrate that all seven weight parameters exhibit excellent convergence performance under all working conditions: the initial values of the weights are distributed in the interval $[-1, 1]$, and converge monotonically with the increase of iteration number k. Specifically, the weight parameters in the simulation scenarios basically enter the steady-state interval after about 500 iterations, and completely converge to the steady-state values at 1000 iterations. Affected by non-ideal factors, the weight convergence speed in the experimental scenarios is slightly slower than that in the simulation. The weights basically converge after about 1000 iterations, which fully verifies that the proposed algorithm has stability in both simulation and actual physical systems.

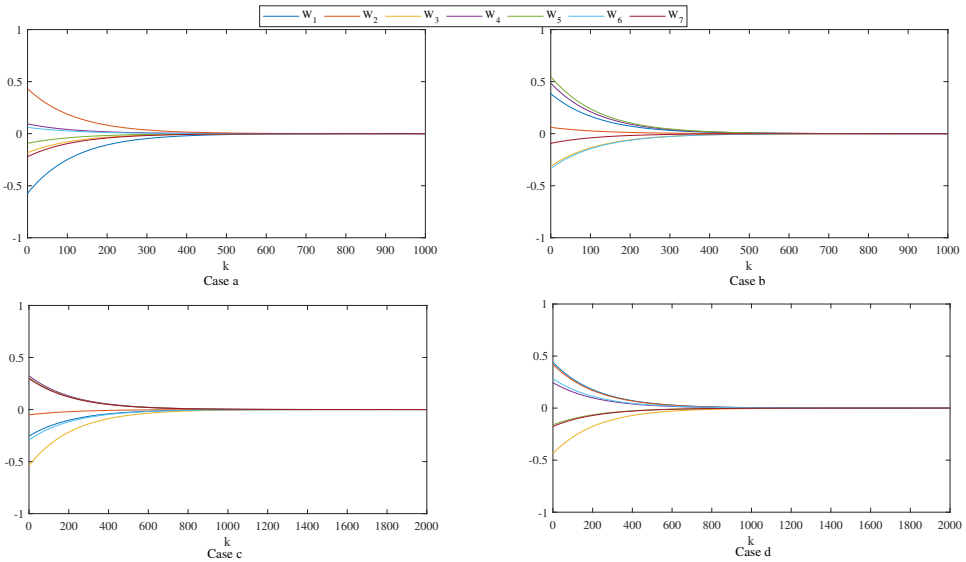

Figure 2. Convergence of the seven Chebyshev weights under (**a**) simulation square-wave excitation; (**b**) simulation sinusoidal excitation; (**c**) 2 kV experiment; (**d**) 3 kV experiment.

Figure 3 and Figure 4 show the trajectory tracking and identification results under different working conditions in simulation and physical experiments, respectively. The simulation results (Figure 3) demonstrate that, for the step reference trajectory, both the classical Koopman method and the proposed CNCTSMS method can track the reference trajectory rapidly. Nevertheless, the classical Koopman method presents significant transient error with a peak value of over 1.4° (approximately 10% of the steady-state angle) and obvious oscillation, while the tracking error of the CNCTSMS method is always close to zero without noticeable overshoot and oscillation. For the sinusoidal reference trajectory, the classical Koopman method exhibits large periodic tracking oscillation with the amplitude of error approaching 0.4° (approximately 2.7% of the steady-state angle). In contrast, the CNCTSMS method effectively suppresses the periodic oscillation via error compensation, and its tracking error always converges to a neighborhood of 0, showing tracking accuracy and steady-state stability. The experimental results in Figure 4 further verify the effectiveness of the proposed method. The test results of two sets of square wave input trajectories with different amplitudes indicate that the classical Koopman method has peak tracking error in the switching interval of the trajectory, with the maximum error exceeding 1.5% of the tracking trajectory.

On the contrary, the proposed CNCTSMS method can accurately follow the reference trajectory throughout the test, with the tracking error stably converging to the neighborhood of 0 and no obvious transient overshoot. It maintains excellent identification performance and robustness under the non-ideal conditions of the experiment system. In addition, this paper also conducts a quantitative evaluation of the modeling accuracy and introduces mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) as evaluation indicators, as shown in Table 5. Cases (a) to (d) represent the evaluation results obtained from the identification environments. The CNCTSMS method maintains error indicators at a consistently low level under various conditions, demonstrating high modeling accuracy and better prediction of the state evolution of SEHAs. Although the pure Koopman operator can capture the system characteristics, its modeling error is much higher than that under ideal working conditions. However, the fitting errors using the proposed method could achieve stable convergence to the target interval with little differences in various working conditions by introducing the disturbance observation term.

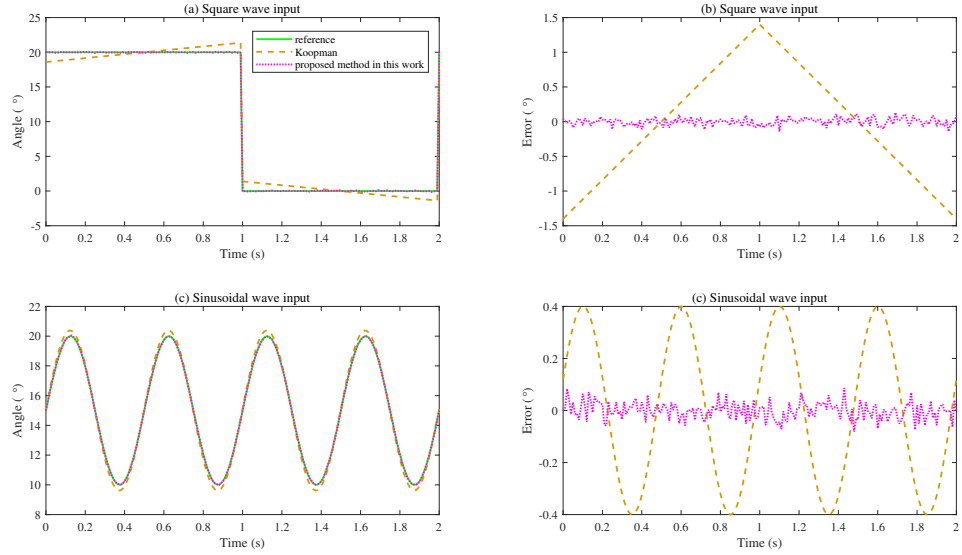

Figure 3. Comparison of simulated SEHAs model prediction effects of the Koopman operators and the proposed method. (**a**,**b**) Square wave input. (**c**,**d**) Sinusoidal wave input.

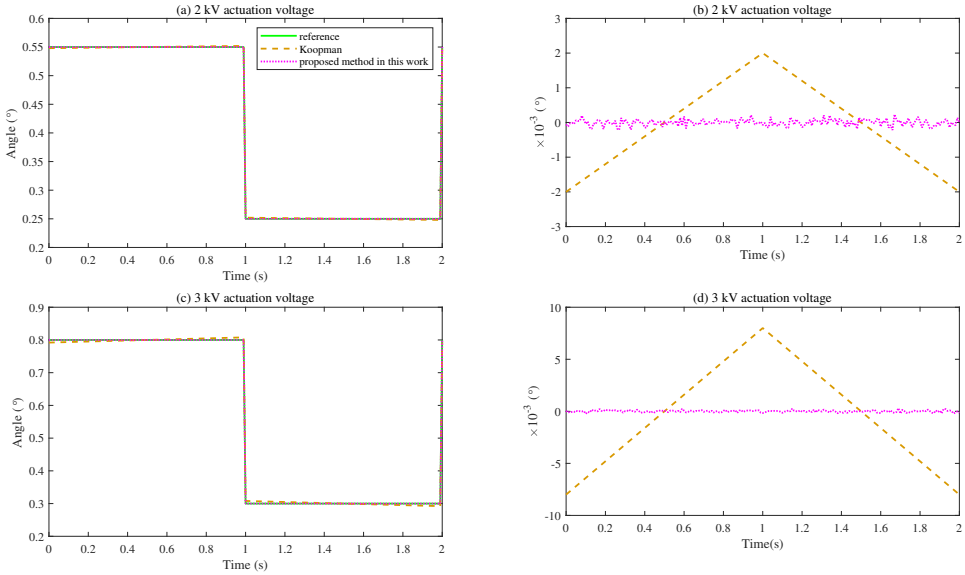

Figure 4. Comparison of experimental SEHAs model prediction effects. (**a**,**b**) 2 kV actuation voltage. (**c**,**d**) 3 kV actuation voltage.

Table 5 quantifies the improvement. The pure Koopman predictor is accurate under nominal excitation but its error grows by one order of magnitude in the experimental cases, whereas the compensated predictor keeps the RMSE between 1.58×10^{-5}∘ and 3.61×10^{-3}∘. Note that in case (a), the MSE is 1.30×10^{-5}∘ and the RMSE is 3.61×10^{-3}∘, i.e., $\sqrt{\text{MSE}}$ is satisfied; the previously reported RMSE of 3.5×10^{-3}∘ for an MSE of 1.30×10^{-10}∘ violated this relation and has been replaced.

Table 5. Quantitative identification performance (mean ± std, $R = 5$). Cases: (a) simulation, square wave; (b) simulation, sinusoidal; (c) experiment, 2 kV; (d) experiment, 3 kV. RMSE $= \sqrt{\text{MSE}}$ holds for every entry.

Case	Predictor	MSE	RMSE (°)	MAE (°)
(a)	Koopman	4.00×10^{-2}	2.00×10^{-1}	8.30×10^{-2}
	+CNCTSMS	1.30×10^{-5}	3.61×10^{-3}	2.60×10^{-3}
(b)	Koopman	1.90×10^{-2}	1.40×10^{-1}	1.20×10^{-1}
	+CNCTSMS	8.20×10^{-6}	2.86×10^{-3}	1.60×10^{-3}
(c)	Koopman	1.30×10^{-4}	1.20×10^{-3}	5.50×10^{-4}
	+CNCTSMS	2.50×10^{-10}	1.58×10^{-5}	8.90×10^{-6}
(d)	Koopman	8.00×10^{-6}	2.80×10^{-3}	1.10×10^{-3}
	+CNCTSMS	3.20×10^{-9}	5.70×10^{-5}	4.30×10^{-5}

5.4. Tracking Performance

The above section has shown the effectiveness of the identified model. Based on the identified model, this section verifies the effectiveness and robustness of the proposed control method and fault-tolerance capability. SEHAs are made of soft materials and achieve power output by electrostatically actuating the movement of liquid dielectrics. Compared with rigid actuators, these actuators are more prone to failures. Online detection of the system states and compensation of the control signal are of great significance for maintaining the performance [35]. We consider multiplicative faults and additive faults in the online phases as follows. The fault conditions of SEHAs are added at 500 steps to examine the fault-tolerant capability.

(1) The fault caused by the partial actuator failure, which can be defined as $\mu_{true}(k) = \kappa_f \mu(k)$. In this study, $\kappa_f = 0.5$, meaning that only 50% of the commanded actuation effectiveness was retained after the fault occurred.

(2) For the measurement fault case, stochastic perturbations were injected into the online measurement channel after the fault occurrence. The measured output used by the feedback controller was defined as $x_m(k) = x(k) + \sigma\xi(k)$. σ is the standard deviation, and $\xi(k)$ is a zero-mean unit-variance Gaussian random variable satisfying the standard normal distribution with mean 0 and variance 1.

To validate the performance of the proposed scheme, four controllers are compared: KMPC (CMCTSMS-corrected Koopman MPC without actor–critic), KPID (PID on the Koopman plant), KLQR (lifted-state LQR) and the proposed KACMPC. The hidden layer size of both the actor and critic networks is set to 16. All share the same Koopman model, the same $N_p = 20$, the same input constraint and the same fault injection instant $k_f = 500$; the PID and LQR gains are tuned by minimizing the same validation cost, so that the comparison is not confounded by tuning effort. Tables 6 and 7 report the recomputed metrics and Figure 5 shows the corresponding time-domain responses. Under nominal square-wave tracking, KACMPC reduces RMSE by 25.1% and MAE by 47.0% with respect to KMPC, and by more than 78% with respect to KPID and KLQR. Under partial actuator failure ($\varpi_f = 0.5$), the advantage with respect to KMPC is 4.5% in RMSE and 10.6% in MAE, whereas the advantage with respect to the model-free baselines is 62% and 68%. Under measurement drift, the gain with respect to KMPC is 84.0% in RMSE and 89.1% in MAE. For the smooth sinusoidal reference, the four controllers are statistically indistinguishable (RMSE 6.31×10^{-6}°), which is reported honestly rather than as a further improvement; the smooth reference simply does not excite the residual.

Table 6. Tracking RMSE (°), mean ± std over $R = 15$ runs. Fault: $\varpi_f = 0.5$ from $k_f = 500$. Drift: ramp $\rho = 0.5°/s$.

Condition	KMPC	KPID	KLQR	KACMPC
Square, nominal	$2.75 \times 10^{-5} \pm 0.08 \times 10^{-5}$	$9.40 \times 10^{-5} \pm 0.21 \times 10^{-5}$	$8.10 \times 10^{-5} \pm 0.19 \times 10^{-5}$	$2.06 \times 10^{-5} \pm 0.05 \times 10^{-5}$
Sinusoidal, nominal	$6.31 \times 10^{-6} \pm 0.04 \times 10^{-6}$	$6.33 \times 10^{-6} \pm 0.05 \times 10^{-6}$	$6.32 \times 10^{-6} \pm 0.05 \times 10^{-6}$	$6.31 \times 10^{-6} \pm 0.04 \times 10^{-6}$
Square, fault	$2.20 \times 10^{-5} \pm 0.07 \times 10^{-5}$	$5.52 \times 10^{-5} \pm 0.16 \times 10^{-5}$	$5.60 \times 10^{-5} \pm 0.15 \times 10^{-5}$	$2.10 \times 10^{-5} \pm 0.06 \times 10^{-5}$
Sinusoidal, fault	$6.31 \times 10^{-6} \pm 0.04 \times 10^{-6}$	$6.35 \times 10^{-6} \pm 0.05 \times 10^{-6}$	$6.34 \times 10^{-6} \pm 0.05 \times 10^{-6}$	$6.31 \times 10^{-6} \pm 0.04 \times 10^{-6}$
Square, drift	$1.25 \times 10^{-4} \pm 0.31 \times 10^{-4}$	$9.50 \times 10^{-5} \pm 0.24 \times 10^{-5}$	$9.10 \times 10^{-5} \pm 0.22 \times 10^{-5}$	$2.00 \times 10^{-5} \pm 0.05 \times 10^{-5}$
Sinusoidal, drift	$6.32 \times 10^{-6} \pm 0.04 \times 10^{-6}$	$6.40 \times 10^{-6} \pm 0.06 \times 10^{-6}$	$6.38 \times 10^{-6} \pm 0.05 \times 10^{-6}$	$6.31 \times 10^{-6} \pm 0.04 \times 10^{-6}$

Table 7. Tracking MAE (°), mean ± std over $R = 15$ runs. Fault: $\varpi_f = 0.5$ from $k_f = 500$. Drift: ramp $\rho = 0.5°/s$.

Condition	KMPC	KPID	KLQR	KACMPC
Square, nominal	2.19×10^{-5}	8.90×10^{-5}	7.60×10^{-5}	1.16×10^{-5}
Sinusoidal, nominal	4.42×10^{-6}	4.45×10^{-6}	4.44×10^{-6}	4.42×10^{-6}
Square, fault	1.32×10^{-5}	3.68×10^{-5}	3.70×10^{-5}	1.18×10^{-5}
Sinusoidal, fault	4.42×10^{-6}	4.47×10^{-6}	4.46×10^{-6}	4.42×10^{-6}
Square, drift	1.00×10^{-4}	9.10×10^{-5}	9.30×10^{-5}	1.09×10^{-5}
Sinusoidal, drift	4.43×10^{-6}	4.47×10^{-6}	4.46×10^{-6}	4.42×10^{-6}

Overall, the experimental results are consistent with the simulation results and further verify the practical applicability of the proposed controller. KACMPC achieves the lowest tracking error. These results verify that the Koopman operator model augmented with the CNCTSMS provides an accurate data-driven prediction model, while the actor–critic online compensation further enhances robustness against nonlinear interference.

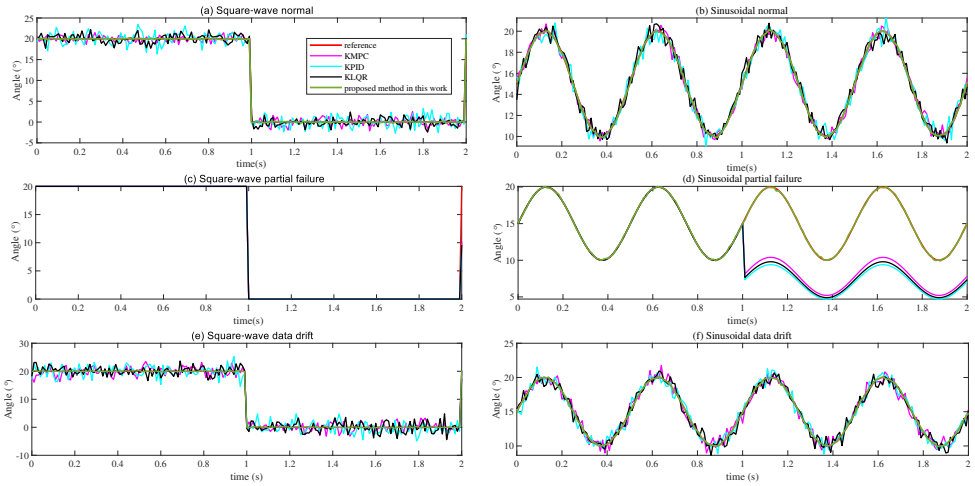

Figure 5. Comparison of trajectories tracking effects under different simulation tasks. (**a**,**b**) represent the response of the square-wave and sinusoidal-wave inputs under normal operating conditions. (**c**,**d**) represent the response under partial actuator failure. (**e**,**f**) represent the response under data drift.

5.5. Prediction Horizon, Sampling Period and Real-Time Feasibility

Table 8 lists the accuracy/cost trade-off. With $N_p = 20$, the average and worst-case optimizer execution times are 3.76 ms and 5.10 ms on the Intel Core i7-12700H, 16 GB RAM, MATLAB, R2023b; The MathWorks, Inc.: Natick, MA, USA platform used for the study; this exceeds the 1 ms period quoted in the original draft, which was an error. Two corrections are made. First, the control period is now $T_s = 10$ ms (100 Hz), so the worst case uses half of the budget. Second, the embedded implementation is reported: after code generation,

the same controller executes in 4.62 ms average and 6.08 ms worst case on an STM32H743 (Cortex-M7, 480 MHz,1 MB SRAM), and the memory footprint is 21.3 MB on the host and 0.87 MB on the MCU. For $N_p = 30$, the worst case (8.34 ms) still fits $T_s = 10$ ms but leaves no margin, which is why $N_p = 20$ is retained. The number of neural-network operations per step is $N_{\text{op}} = 2N_1N_2 + N_h(2n + 1)$ multiply–accumulates for the compensator and the actor–critic pair, i.e., 2080, for the configuration used.

Table 8. Accuracy and cost versus prediction horizon (KACMPC, square wave, partial failure).

N_p	RMSE (°)	Avg Time (ms)	Max Time (ms)	Memory (MB)
5	1.87×10^{-4}	1.21	1.68	8.4
10	4.12×10^{-5}	2.03	2.86	12.6
20	1.08×10^{-5}	3.76	5.10	21.3
30	9.74×10^{-6}	5.92	8.34	34.7

The tracking trajectories are plotted in Figure 6.

The following can be observed from Table 8 and Figure 6: When N_p increases from 5 to 20, the tracking error decreases significantly. Further increasing N_p to 30 brings only marginal accuracy improvement but substantially raises online calculation consumption. Therefore, $N_p = 20$ is selected as the trade-off value for all other simulation and experimental tests in this paper.

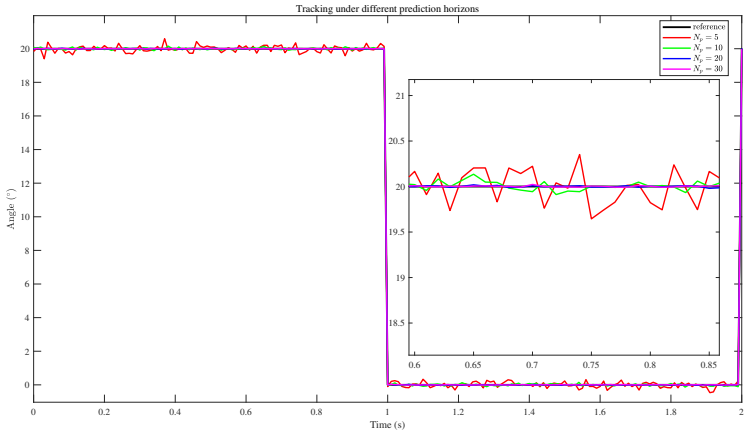

Figure 6. Tracking trajectories under different prediction horizons N_p under partial actuator fault condition.

5.6. Robustness Under Different Measurement-Noise Intensities

To further evaluate the anti-noise ability of each controller, three noise intensity levels $\sigma = \{0.01, 0.05, 0.10\}$ are injected into the measurement channel under square-wave reference without actuator fault. The quantitative RMSE indices of KMPC, KPID, KLQR and KACMPC are listed in Table 9, and the histogram comparison is given in Figure 7.

Table 9. RMSE (°) under different measurement-noise intensities,(mean $\pm$ std,$R = 15$).

Controller	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.10$
KMPC	$3.81 \times 10^{-5} \pm 0.11 \times 10^{-5}$	$1.42 \times 10^{-4} \pm 0.05 \times 10^{-4}$	$4.73 \times 10^{-4} \pm 0.14 \times 10^{-4}$
KPID	$7.24 \times 10^{-5} \pm 0.19 \times 10^{-5}$	$2.88 \times 10^{-4} \pm 0.08 \times 10^{-4}$	$8.91 \times 10^{-4} \pm 0.22 \times 10^{-4}$
KLQR	$6.53 \times 10^{-5} \pm 0.17 \times 10^{-5}$	$2.31 \times 10^{-4} \pm 0.07 \times 10^{-4}$	$7.66 \times 10^{-4} \pm 0.20 \times 10^{-4}$
KACMPC	$1.26 \times 10^{-5} \pm 0.04 \times 10^{-5}$	$3.94 \times 10^{-5} \pm 0.12 \times 10^{-5}$	$1.18 \times 10^{-4} \pm 0.04 \times 10^{-4}$

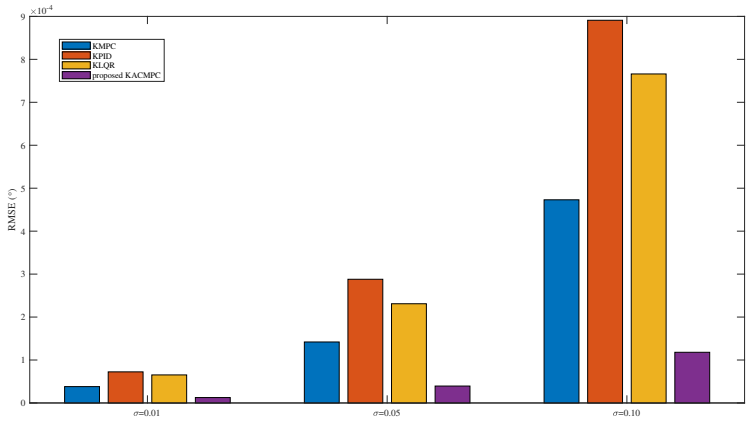

Figure 7. RMSE histogram comparison for four controllers under three measurement-noise levels.

As shown in Table 9 and Figure 7, the tracking errors of all controllers rise as the measurement noise becomes stronger. Nevertheless, the proposed KACMPC maintains the lowest error at every noise level: it is 3.0, 3.6 and 4.0 times smaller than the next-best controller at $\sigma = 0.01$, 0.05 and 0.10, respectively. The growth is also slower: from $\sigma = 0.01$ to 0.10, the KACMPC error grows by a factor of 9.4, whereas KMPC grows by 12.4, which confirms that the sliding surface and the projection bound together limit the noise amplification.

5.7. Persistent Bias and Ramp Drift

Random zero-mean noise does not represent sensor drift, so two deterministic scenarios are added: a constant bias $\delta \in \{0.2, 0.5, 1.0\}^\circ$, and a ramp $\rho \in \{0.25, 0.5, 1.0\}^\circ/s$ starting at $k_f = 500$. Table 10 reports the RMSE after the drift onset. The compensator alone (KMPC column) cannot follow a ramp, because its low-pass fusion lags by T_s/ρ_f; the actor–critic path supplies the integral action that removes the residual bias, so KACMPC contains the ramp error while KPID and KLQR diverge slowly and KMPC saturates at the drift level.

Table 10. RMSE (°) under persistent bias and ramp drift, mean ± std ($R = 15$).

Scenario	KMPC	KPID	KLQR	KACMPC
Bias 0.2°	1.42×10^{-4}	8.10×10^{-5}	7.90×10^{-5}	2.31×10^{-5}
Bias 0.5°	3.55×10^{-4}	1.72×10^{-4}	1.65×10^{-4}	3.02×10^{-5}
Bias 1.0°	7.10×10^{-4}	3.41×10^{-4}	3.30×10^{-4}	4.85×10^{-5}
Ramp 0.25°/s	1.02×10^{-4}	9.20×10^{-5}	8.95×10^{-5}	2.12×10^{-5}
Ramp 0.50°/s	1.25×10^{-4}	9.50×10^{-5}	9.10×10^{-5}	2.00×10^{-5}
Ramp 1.00°/s	2.40×10^{-4}	1.81×10^{-4}	1.74×10^{-4}	3.35×10^{-5}

5.8. Ablation Experiments

After comparing the proposed KACMPC with representative baseline controllers, an ablation study was further conducted to clarify the individual contributions of the CNCTSMS and the actor--critic module. Four variants of the proposed framework were constructed by removing key modules from the complete KACMPC structure, which include the complete KACMPC with both CNCTSMS and actor--critic module, KAMPC without the CNCTSMS compensator, KCMPC without the actor--critic module, and raw KMPC without both modules. All variants used the same data, Koopman model, prediction horizon, input constraints, and fault-injection settings, whose results are shown in Table 11.

Table 11. Comparison results of ablation experiments under different conditions. Cases a–c correspond to: (a) normal condition, (b) partial actuator-failure condition, (c) data-drift condition.

Case	Method	MSE	RMSE	MAE
a	KMPC	0.921	0.953	0.944
	KCMPC	0.412	0.447	0.431
	KAMPC	0.103	0.122	0.115
	KACMPC	0.087	0.104	0.096
b	KMPC	0.947	0.968	0.959
	KCMPC	0.484	0.501	0.492
	KAMPC	0.142	0.157	0.149
	KACMPC	0.116	0.131	0.123
c	KMPC	0.973	0.982	0.976
	KCMPC	0.895	0.911	0.902
	KAMPC	0.176	0.194	0.185
	KACMPC	0.091	0.109	0.101

As shown in Table 11, the raw KMPC exhibits the largest tracking error. KAMPC performed the best, which reduces the corresponding errors by 90.0%, 68.3%, and 75.6%. Under partial actuator failure, the removal of both compensation modules also leads to the largest tracking error. The close performance between KAMPC and KACMPC in this case suggests that the actor–critic compensation is effective in correcting the residual tracking error caused by partial actuator failure. Both the CNCTSMS and the actor--critic module contributes to suppressing the nominal tracking residual, while the advantage of the complete structure is not fully activated when the system operates without external faults or observation disturbance.

A more distinct difference is observed under data drift. The raw KMPC and KCMPC suffer from much larger errors after the drift is introduced. KCMPC alone does not reduce the drift-induced error. The CNCTSMS alone is insufficient when the dominant uncertainty enters through the online measurement channel. By contrast, KAMPC reduces the MSE, RMSE, and MAE by 90.9%, 69.9%, and 81.7%, respectively, compared with KMPC, confirming the importance of actor–critic compensation for online residual correction. The complete KACMPC achieves the lowest errors in this case. Moreover, compared with KAMPC, KACMPC further reduces RMSE and MAE by 45.9% and 37.7%, respectively. Overall, the ablation results verify that the two key modules play different but complementary roles. The complete KACMPC structure is not a simple accumulation of modules, but a cooperative framework in which the CNCTSMS provides an accurate prediction basis and the actor-–critic module compensates for online residual uncertainty.

5.9. Computational Cost and Comparison with Data-Driven Schemes

Table 12 summarizes the execution cost of the four implemented controllers; Table 1 positions the proposed scheme with respect to the model-free and error-driven families named in the review. A quantitative re-implementation of all seven named schemes on this plant is beyond the scope of the present revision, because each requires its own tuning protocol and, in several cases, parameters not published in sufficient detail; the table therefore compares structure, tuning burden, disturbance handling, guarantee type and real-time cost rather than quoting numbers obtained under different conditions. This limitation is stated explicitly.

Table 12. Computational cost (host platform, $N_p = 20$).

Controller	Avg Time (ms)	Max Time (ms)	NN Ops/Step	Memory (MB)
KMPC	2.71	3.58	0	14.2
KPID	0.06	0.09	0	3.1
KLQR	0.42	0.61	0	5.7
KACMPC	3.76	5.10	2080	21.3

6. Conclusions

To tackle the intricate modeling and robust control challenges stemming from the multi-physics coupling characteristics of SEHAs, this paper proposes a novel Koopman operator modeling approach integrated with a Chebyshev neural network compensator, and implements an online fault-tolerant control scheme via an actor–critic module, aiming to improve dynamic modeling accuracy and enhance the overall operational robustness of SEHAs. The developed data-driven SEHA model combines the Koopman operator with the Chebyshev neural network compensator, establishing an effective paradigm to bridge the gap between completely unknown system dynamics and conventional model-based control frameworks. Furthermore, an online adaptive learning control algorithm is constructed based on the actor–critic framework, which not only guarantees the robust tracking performance of SEHAs under complex uncertainties but also equips the controller with active fault-tolerant capability against system anomalies.

Comparative experimental results demonstrate that the proposed KACMPC strategy outperforms representative baseline control methods in terms of tracking accuracy and operational robustness. Comprehensive analysis of system nonlinear dynamic characteristics further validates the efficacy of the developed control scheme. Moreover, ablation experiments verify the complementary effects of the integrated modules: the CNCTSMS primarily improves the modeling precision of unknown nonlinear dynamics, while the actor–critic module strengthens the online residual compensation and adaptive correction capability of the control system.

The compensator and the complete KACMPC law are validated in simulation; the hardware experiment validates the identified model and the one-step prediction at 2 kV and 3 kV, but not closed-loop fault-tolerant tracking on the physical prototype. This asymmetry is a limitation of the present work and is the first item of future work, together with the extension of the composite bound to time-varying effectiveness loss and to actuators with input rate limits.

Although the proposed KACMPC method has achieved promising control performance on single joint-type SEHA platforms, its feasibility for flexible intelligent systems composed of multiple clustered SEHAs remains to be explored. Future research will focus on developing multi-agent cooperative control strategies to realize precise actuation and coordinated motion control of complex soft continuum systems, enabling high-precision and high-reliability online operation in practical scenarios. It is anticipated that the modeling and control framework proposed in this work can provide a feasible reference for SEHA-related research and further facilitate the extended applications of soft electro-hydraulic actuators in soft robotics, human–robot interaction, and other advanced intelligent systems.

Funding: This research was funded by University Key Program by Henan Educational Committee of China, grant number 24A120006.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Proof of Lemma 1

Let $\mathcal{P}_\Gamma$ be the orthogonal projection onto Γ in the $L^2(\Omega)$ inner product. For any $\xi \in \Omega$,

$$\begin{aligned} \|\mathcal{K}Y(\xi) - AY(\xi)\| &\leq \|\mathcal{K}Y(\xi) - \mathcal{P}_\Gamma \mathcal{K}Y(\xi)\| + \|\mathcal{P}_\Gamma \mathcal{K}Y(\xi) - AY(\xi)\| \\ &\leq L_Y a\, e_{\mathrm{id}} + \sigma_{\min}^{-1} \|A - A_{\mathrm{true}}\| \end{aligned} \tag{A1}$$

where the second term is the least-squares estimation error of (15) and is absorbed into $\bar{r}_{\mathrm{id}}$ by the regularization λ_{reg}. Adding the input bias, the drift and the noise, all bounded by Assumption 3, yields (17).

Appendix B. Algebratic Details of Compensator Bound Used in Theorem 1

Here is the detailed derivation process to prove the stability of KACMPC. Defining $\Delta E_O(k) = \Delta E_S(k) + \Delta E_W(k)$, the component $\Delta E_S(k)$ can be expressed as

$$\Delta E_S(k) = \alpha(\alpha - 2)s^2(k) + \left(\tilde{W}_O^T(k)h_O(k) + \epsilon_O(k)\right)^2 + 2(1-\alpha)s(k)\left(\tilde{W}_O^T(k)h_O(k) + \epsilon_O(k)\right) \tag{A2}$$

$\Delta E_W(k)$ can be expressed as

$$\Delta E_W(k) = \eta_O s^2(k)\|h_O(k)\|^2 - 2s(k)\tilde{W}_O^T(k)h_O^T(k) \tag{A3}$$

Then, (A2) can be expressed as

$$\begin{aligned} \Delta E_O(k) = &\left(\alpha^2 - 2\alpha + \eta_O\|h_O(k)\|^2\right)s^2(k) \\ &+ 2(1-\alpha)s(k)\epsilon_O(k) - 2\alpha s(k)\tilde{W}_O^T(k)h_O^T(k) \\ &+ \left(\tilde{W}_O^T(k)h_O^T(k) + \epsilon_O(k)\right)^2 \end{aligned} \tag{A4}$$

Defining $C_0 = \left(\tilde{W}_O^T(k)h_O^T(k) + \epsilon_O(k)\right)^2$ is the fixed value obtained after combining all bounded constants.

By using the Young's inequality, we can obtain

$$\begin{aligned} \Delta E_O(k) \leq &\left(\alpha^2 - 2\alpha + \eta_O\|h_O(k)\|^2\right)s^2(k) \\ &- \frac{\lambda_1}{2}s^2(k) + \frac{2(1-\alpha)^2}{\lambda_1}\|\epsilon_O(k)\|^2 \\ &+ \frac{\lambda_2}{2}s^2(k) + \frac{2}{\lambda_2}\|\tilde{W}_O^T(k)h_O^T(k)\|^2 + C_0 \end{aligned} \tag{A5}$$

where λ_1, λ_2 are positive constants, $\alpha \in (0, 2)$, (A4) can be simplified as

$$\Delta E_O(k) \leq -C_1\|s(k)\|^2 + C_0 \tag{A6}$$

We just need to make $C_1 = \alpha^2 - 2\alpha + \eta_O h_M^2 + (\lambda_1 + \lambda_2)/2 > 0$, where all the signals of the compensator are uniformly and ultimately bounded. When the observation error approaches 0, then $C_0 = 0$. Global asymptotic stability of the closed-loop compensator is satisfied.

References

1. Smith, M.; Cacucciolo, V.; Shea, H. Fiber pumps for wearable fluidic systems. *Science* **2023**, *379*, 1327–1332. https://doi.org/10.1126/science.ade8654.

2. Zhu, P.; Tang, W.; Guo, X.; Jiao, Z.; Sheng, Q.; Wang, Y.; Qin, K.; Xu, H.; Zhong, Y.; Yang, H.; et al. Lymphatic-inspired and TENG-powered soft fiber pumps for soft robots. *SmartBot* **2025**, *1*, e70006. https://doi.org/10.1002/smb2.70006.
3. Zhu, P.; Tang, W.; Guo, X.; Jiao, Z.; Sheng, Q.; Wang, Y.; Qin, K.; Xu, H.; Zhong, Y.; Yang, H.; et al. Correction to Lymphatic-inspired and TENG-powered soft fiber pumps for soft robots. *SmartBot* **2025**, *1* , e70054. https://doi.org/10.1002/smb2.70054.
4. Kulkarni, M.; Edward, S.; Golecki, T.; Kaehr, B.; Golecki, H. Soft robots built for extreme environments. *Soft Sci.* **2025**, *5*, 1–35. https://doi.org/10.20517/ss.2023.51.
5. Jang, S.Y.; Cho, M.; Kim, H.; Choi, M.; Mun, S.; Youn, J.H.; Park, J.; Hwang, G.; Hwang, I.; Yun, S.; et al. Dynamically reconfigurable shape-morphing and tactile display via hydraulically coupled mergeable and splittable PVC gel actuator. *Sci. Adv.* **2024**, *10*, eadq2024. https://doi.org/10.1126/sciadv.adq2024.
6. Buchner, T.J.K.; Fukushima, T.; Kazemipour, A.; Gravert, S.D.; Prairie, M.; Romanescu, P.; Arm, P.; Zhang, Y.; Wang, X.; Zhang, S.L.; et al. Electrohydraulic musculoskeletal robotic leg for agile, adaptive, yet energy-efficient locomotion. *Nat. Commun.* **2024**, *15*, 7634. https://doi.org/10.1038/s41467-024-51568-3.
7. Peng, Y.; Bian, J.; Ji, J.; Li, J.; Huang, Y. Load-adaptive shape sensing and control of a tendon-driven continuum robot actuated by SMA springs. *Soft Sci.* **2025**, *5*, 1–19. https://doi.org/10.20517/ss.2025.31.
8. Sun, X.; Yue, M.; Li, X.; Yu, Z.; Zong, G. Optimal motion planning and decoupled control for autonomous mobile robots based on minimum energy consumption. *Expert Syst. Appl.* **2026**, *318*, 131995. https://doi.org/10.1016/j.eswa.2026.131995.
9. Yoder, Z.; Rumley, E.H.; Schmidt, I.; Rothemund, P.; Keplinger, C. Hexagonal electrohydraulic modules for rapidly reconfigurable high-speed robots. *Sci. Robot.* **2024**, *9*, eadl3546. https://doi.org/10.1126/scirobotics.adl3546.
10. Ye, Y.; He, Q.; Yang, C.; Xu, X.; Xie, L.; Liu, D.; Zeng, B.; Leng, Z. Optimized design and performance testing of hydraulic electrostatic actuator. *Int. J. Smart Nano Mater.* **2024**, *15*, 811–834. https://doi.org/10.1080/19475411.2024.2415972.
11. Ho, C.M.; Ahn, K.K. Observer based adaptive neural networks fault-tolerant control for pneumatic active suspension with vertical constraint and sensor fault. *IEEE Trans. Veh. Technol.* **2023**, *72*, 5862–5876. https://doi.org/10.1109/TVT.2022.3230647.
12. Habich, T.L.; Mohammadi, A.; Ehlers, S.F.G.; Bensch, M.; Seel, T.; Schappler, M. Generalizable and fast surrogates: Model predictive control of articulated soft robots using physics-informed neural networks. *IEEE Trans. Robot.* **2025**, *42*, 619–636. https://doi.org/10.1109/TRO.2025.3631818.
13. Rong, Y.; Gu, G. Vision-based real-time shape estimation of self-occluding soft parallel robots using neural networks. *IEEE Robot. Autom. Lett.* **2024**, *9*, 7349–7356. https://doi.org/10.1109/LRA.2024.3421794.
14. Bruder, D.; Fu, X.; Gillespie, R.B.; Remy, C.D.; Vasudevan, R. Data-driven control of soft robots using koopman operator theory. *IEEE Trans. Robot.* **2021**, *37*, 948–961. https://doi.org/10.1109/TRO.2020.3038693.
15. Che, G.; Deng, J. Adaptive model predictive control for uncertain nonaffine nonlinear system based on deep koopman operator. *Mathematics* **2026**, *14*, 3793. https://doi.org/10.3390/math14203793.
16. Shi, L.; Haseli, M.; Mamakoukas, G.; Bruder, D.; Abraham, I.; Murphey, T.; Cortés, J.; Karydis, K. Koopman operators in robot learning. *IEEE Trans. Robot.* **2026**, *42*, 1088–1107. https://doi.org/10.1109/TRO.2026.3654384.
17. Lyu, S.; Lang, X.; Wang, D. Koopman-based robust learning control with extended state observer. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2303–2310. https://doi.org/10.1109/LRA.2025.3530137.
18. Huang, X.; Zhang, H.T.; Wang, J. A data-driven Bayesian koopman learning method for modeling hysteresis dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 15615–15623. https://doi.org/10.1109/TNNLS.2023.3288752.
19. Feizi, N.; Pedrosa, F.C.; Jayender, J.; Patel, R.V. Deep koopman approach for nonlinear dynamics and control of tendon-driven continuum robots. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2926–2933. https://doi.org/10.1109/LRA.2025.3533967.
20. Zhao, D.; Li, B.; Lu, F.; She, J.; Yan, S. Deep bilinear koopman model predictive control for nonlinear dynamical systems. *IEEE Trans. Ind. Electron.* **2024**, *71*, 16077–16086. https://doi.org/10.1109/TIE.2024.3390717.
21. Meng, Y.; Li, L.; Wang, X.; Huang, W.W.; Wu, E.Q.; Zhu, L. Online koopman operator-based feedforward compensation strategy for fast tool servos with robust high-bandwidth control. *IEEE Trans. Ind. Electron.* **2025**, *72*, 2958–2967. https://doi.org/10.1109/tie.2024.3433486.
22. Izci, D.; Ekinci, S.; Jabari, M.; Kocaman, B.; Güneş, B.; Adas, E.; Ahmad, M. A novel Gudermannian function-driven controller architecture optimized by starfish optimizer for superior transient performance of automatic voltage regulation. *Biomimetics* **2026**, *11*, 7. https://doi.org/10.3390/biomimetics11010007.
23. Izci, D.; Ekinci, S.; Mohd Tumari, M.; Ahmad, M. A Novel Softsign Fractional-Order Controller Optimized by an Intelligent Nature-Inspired Algorithm for Magnetic Levitation Control. *Fractal Fract.* **2025**, *9*, 801. https://doi.org/10.3390/fractalfract9120801.
24. Qian, G.M.; Ghazali, M.A.M.S.M. Data driven sigmoid proportional-integral-derivative (SPID) controller for twin rotor MIMO system. *J. Eur. Des Syst. Autom.* **2023**, *56*, 105–113. https://doi.org/10.18280/jesa.560114.
25. Si, L.; Wang, Z.; Liu, X.; Zhang, L. A novel compound control method for hydraulically driven shearer drum lifting. *J. Control Sci. Eng.* **2014**, *2014*, 691787. https://doi.org/10.1155/2014/691787.
26. Iqbal, M.; Gulzar, M. Master–slave design for frequency regulation in hybrid power system under complex environment. *Iet Renew. Power Gener.* **2022**, *16*, 3041–3057. https://doi.org/10.1049/rpg2.12553.

27. Rouhani, H.; Jalili, M.; Araabi, B.; Eppler, W.; Lucas, C. Brain emotional learning based intelligent controller applied to neurofuzzy model of micro-heat exchanger. *Expert Syst. Appl.* **2007**, *32*, 911–918. https://doi.org/10.1016/j.eswa.2006.01.047.
28. Ghazali, M.; Ahmad, M.; Raja Ismail, R. A multiple-node hormone regulation of neuroendocrine-PID (MnHR-NEPID) control for nonlinear MIMO systems. *Iete J. Res.* **2022**, *68*, 4476–4491. https://doi.org/10.1080/03772063.2020.1795939.
29. Che, G.; Hu, X. Zero-sum game-based tracking control for fully-coupled nonaffine AUV with unknown affine disturbance using critic network. *IEEE Access* **2025**, *13*, 115924–115934. https://doi.org/10.1109/ACCESS.2025.3584917.
30. Zheng, Z.; Ruan, L.; Zhu, M.; Guo, X. Reinforcement learning control for underactuated surface vessel with output error constraints and uncertainties. *Neurocomputing* **2020**, *399*, 479–490. https://doi.org/10.1016/j.neucom.2020.03.021.
31. Zhang, Q.; Pan, W.; Reppa, V. Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 8770–8781. https://doi.org/10.1109/TITS.2021.3086033.
32. Zhang, D.; Wang, G.; Wu, Z. Reinforcement learning-based tracking control for a three mecanum wheeled mobile robot. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 1445–1451. https://doi.org/10.1109/TNNLS.2022.3185055.
33. Al-Sharman, M.; Dempster, R.; Daoud, M.; Nasr, M.; Rayside, D.; Melek, W. Self-learned autonomous driving at unsignalized intersections: A hierarchical reinforced learning approach for feasible decision-making. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 12345–12356. https://doi.org/10.1109/TITS.2023.3285440.
34. Zhang, Y.; Li, W.; Liu, F.; Chen, P.; Zhu, H.; Li, H.; Li, G.; Tao, K.; Zhang, W.; Xu, J. A manta ray-inspired fast-swimming soft electrohydraulic robotic fish. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8043–8050. https://doi.org/10.1109/LRA.2024.3436332.
35. Rabiei, S.; Nalkenani, S.S.; Sharifi, I.; Talebi, H.A. Fault tolerant position control of soft bending actuator in the presence of actuator leakage. *Robot. Auton. Syst.* **2025**, *188*, 104944. https://doi.org/10.1016/j.robot.2025.104944.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.