Generalized Momenta-Based Koopman Formalism for Robust Control of Euler-Lagrangian Systems

Rajpal Singh[1], Aditya Singh[1], Chidre Shravista Kashyap[2], and Jishnu Keshavan[1]

Abstract—This paper presents a novel Koopman operator formulation for Euler–Lagrangian dynamics that employs an implicit generalized momentum-based state space representation, which decouples a known linear actuation channel from state-dependent dynamics and makes the system more amenable to linear Koopman modeling. By leveraging this structural separation, the proposed formulation only requires to learn the unactuated dynamics rather than the complete actuation-dependent system, thereby significantly reducing the number of learnable parameters, improving data efficiency, and lowering overall model complexity. In contrast, conventional explicit formulations inherently couple inputs with the state-dependent terms in a nonlinear manner, making them more suitable for bilinear Koopman models, which are more computationally expensive to train and deploy. Notably, the proposed scheme enables the formulation of linear models that achieve superior prediction performance compared to conventional bilinear models while remaining substantially more efficient. To realize this framework, we present two neural network architectures that construct Koopman embeddings from actuated or unactuated data, enabling flexible and efficient modeling across different tasks. Robustness is ensured through the integration of a linear Generalized Extended State Observer (GESO), which explicitly estimates disturbances and compensates for them in real-time. The combined momentum-based Koopman and GESO framework is validated through comprehensive trajectory tracking simulations and experiments on robotic manipulators, demonstrating superior accuracy, robustness, and learning efficiency relative to state-of-the-art alternatives.

Index Terms—Koopman operator, Euler–Lagrange systems, unactuated dynamics, linear Koopman models, generalized extended state observer (GESO), learning-based control

I. INTRODUCTION

CONTROL of nonlinear systems has long been a central challenge in robotics. Design of control methodologies that are general, scalable, and computationally efficient is particularly difficult owing to the complexity and high-dimensionality of such systems. Traditional model-based approaches often employ local linearization to make the problem tractable. However, local linearization techniques provide valid approximations only in the vicinity of nominal operating points and fail to represent system behavior precisely over the entire workspace.

In contrast, Koopman operator theory [1] offers a globally linear formulation for nonlinear dynamics by leveraging nonlinear observable/basis functions of the state of the system, enabling significantly enhanced prediction accuracy [2]. However, an exact representation of the Koopman operator often requires an infinite dimensional set of observables, which renders it infeasible for practical applications. To address this, several data-driven approaches such as Dynamic Mode Decomposition (DMD) [3], DMDc [4], Extended DMD (EDMD) [5], and Sparse Identification of Nonlinear Dynamics (SINDy) [6] have been proposed to approximate the Koopman operator with a finite set of basis functions. However, selecting a set of observables that is both sufficiently expressive and computationally tractable remains a significant challenge, particularly for complex dynamical systems. As such, recent advances have turned to learning-based methods [7], [8] to automate the discovery of an optimal set of basis functions.

Euler–Lagrangian systems constitute a core class of nonlinear dynamical systems widely encountered in robotics [9]. While Koopman-based formulations have shown promise, applying them directly to the modeling of the Euler-Lagrangian dynamical systems poses specific challenges. When represented *explicitly in terms of generalized positions and velocities*, the dynamics of Euler–Lagrangian systems is characterized by state-dependent control vector fields. Consequently, this dynamics is naturally more suited for learning bilinear Koopman representations [10], which can capture nonlinear interactions between state and control terms, rather than purely linear Koopman representations. Hence, many recent works have employed bilinear Koopman embedding to model Euler-Lagrangian systems [11], [12]. However, these bilinear models are significantly more complex to train, require larger datasets, and render the downstream control synthesis, especially for optimal control frameworks such as Model Predictive Control (MPC), more complicated and computationally demanding. To address these issues, the authors in [13] construct Koopman models at the actuator dynamics level rather than at the joint dynamics level. At the actuator level, the control inputs are naturally decoupled from the state-dependent dynamics, which makes the system dynamics more amenable to learning linear embeddings. However, this design choice requires modeling the influence of actuation indirectly by incorporating the actuator subsystem into the Koopman lifting so that the effects of actuation appear linearly. While this approach preserves the known mechanical structure, it complicates model training and introduces sensitivity to actuator dynamics and parameter uncertainty, which could otherwise be avoided by modeling directly at the joint dynamics level.

Motivated by this challenge, we propose a momentum-based Koopman operator framework tailored specifically for modeling Euler-Lagrangian system dynamics. Instead of using the general explicit state space representation, we employ an *implicit state space representation*, which uses generalized position and generalized momentum as generalized coordinates. This implicit state representation decouples a linear control channel from the remaining state-dependent nonlinear dynamics, making it more amenable to learning linear Koopman models. As such, the linear models obtained from the proposed formulation achieve superior prediction performance relative to bilinear models derived from explicit state representations. Further, the structure of the control matrix can be inferred a priori. As such, only the unactuated/passive part of the dynamics needs to be learned, drastically reducing the number of learnable parameters and improving data efficiency without sacrificing predictive power. We design two neural network architectures capable of learning Koopman embeddings from either actuated or purely unactuated data, enabling flexible modeling under diverse data collection scenarios. Another critical challenge in data-driven Koopman modeling is that the learned models are often sensitive to external disturbances and unmodeled dynamics, which prevents accurate prediction and control, limiting their robustness in real-world settings. As such, previous works have been proposed to improve the robustness of learned Koopman models. In particular, [14] and [15] employ higher-order Generalized Extended State Observer (GESO) [16] in order to deal with external disturbance. Inspired by the previous works, we integrate a linear GESO within our framework to estimate and compensate for external disturbances and unmodeled effects in real time. Unlike the nonlinear higher-order GESOs [14], [15] employed in previous studies, linear GESOs can be seamlessly incorporated within the linear controller frameworks without breaking their linear structure, thereby avoiding the additional complications such as saturation filters or stability adjustments. The benefits of the proposed scheme are demonstrated through extensive simulation and experimental studies, including comparison with leading alternative designs.

The main contributions are summarized as follows:

- We reformulate the Euler-Lagrangian dynamics in an implicit momentum-based representation that decouples the control channel from state-dependent dynamics, thereby yielding a structure naturally suited for learning linear Koopman models under actuation.
- The linear model learned from the proposed approach exhibits a superior prediction performance compared to the more computationally demanding bilinear models learned using the conventional explicit representation of the dynamics.
- By exploiting the known structure of the control matrix, our framework needs to learn only the passive dynamics rather than the full actuation-dependent dynamics, thereby substantially reducing the number of learnable parameters, improving data efficiency, and lowering model complexity.
- We design two neural network architectures that learn

Koopman embeddings from actuated or unactuated (passive) data, enabling flexible deployment under diverse data collection scenarios.
- We integrate a linear Generalized Extended State Observer (GESO) into the Koopman-MPC framework to explicitly estimate and compensate for external disturbances and unmodeled effects in real-time, while preserving the linear structure of downstream controllers.
- We validate the proposed approach through extensive simulation and experimental studies to demonstrate the superior tracking accuracy and disturbance rejection compared to state-of-the-art methods.

The remainder of the paper is organized as follows. Section II presents the preliminaries for Koopman operator theory. Sections III and IV present the proposed momentum-based Koopman formulation and neural architectures employed for learning the subsequent Koopman models. Sections V and VI detail GESO for disturbance estimation with its subsequent integration within a MPC framework. Section VII describes simulation and experimental results. Section VIII concludes the paper.

II. PRELIMINARIES

This section presents the fundamental concepts of Koopman operator theory. Consider a nonlinear autonomous system given by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{1}$$

where the state vector $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^n$ and the dynamics $\boldsymbol{f} : \mathbb{X} \to \mathbb{X}$ is assumed to be Lipschitz continuous over $\mathbb{X}$. The corresponding discrete-time dynamics can be written as $\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k)$, where $\boldsymbol{F} : \mathbb{X} \to \mathbb{X}$ denotes the system's flow map and $\boldsymbol{x}_k$ represents the state at k^{th} time step.

As per Koopman operator theory [1], the evolution of a set of observable functions $\sigma : \mathbb{X} \to \mathbb{C}$, within a Hilbert space, can be described by a linear, infinite-dimensional Koopman operator $\mathcal{K} : \mathbb{C} \to \mathbb{C}$ such that

$$\mathcal{K} \circ \sigma(\boldsymbol{x}_k) = \sigma(\boldsymbol{F}(\boldsymbol{x}_k)) = \sigma(\boldsymbol{x}_{k+1}),$$

where $\circ$ denotes functional composition.

The Koopman operator framework can be extended to systems with control inputs. Consider a control-affine system described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \boldsymbol{f}_i(\boldsymbol{x}) u_i, \tag{2}$$

where $u_i \in \mathbb{R}^m \ \forall \ i = 1, 2, ..., m$ denotes the control inputs. $\boldsymbol{f}_0 : \mathbb{X} \to \mathbb{X}$ and $\boldsymbol{f}_i : \mathbb{R}^m \to \mathbb{X} \ \forall \ i = 1, 2, ..., m$ represent the drift and state-dependent control vector fields, respectively. The corresponding Koopman representation takes the form:

$$\boldsymbol{z} = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x})]^\top,$$
$$\dot{\boldsymbol{z}} = \boldsymbol{A}_c \boldsymbol{z} + \boldsymbol{B}_c \boldsymbol{u}, \quad \boldsymbol{x} = \boldsymbol{C}^x \boldsymbol{z}, \tag{3}$$

where $\phi_i : \mathbb{X} \to \mathbb{C} \ \forall \ i = 1, 2, ..., N$ are the observable functions, and $\boldsymbol{A}_c \in \mathbb{R}^{N \times N}$, $\boldsymbol{B}_c \in \mathbb{R}^{N \times m}$ are the continuous-time Koopman model matrices. The matrix $\boldsymbol{C}^x \in \mathbb{R}^{n \times N}$ maps

the lifted state z back to the original state space x. Similarly, the corresponding bilinear formulation is given as

$$\dot{z} = A_c z + B_c (z \otimes u), \quad x = C^x z, \qquad (4)$$

where $B_c \in \mathbb{R}^{Nm \times m}$. $\otimes$ represents the Kronecker product. The corresponding discrete-time formulations are:

$$z_{k+1} = A z_k + B u_k, \quad x_{k+1} = C^x z_{k+1}, \qquad (5)$$

$$z_{k+1} = A z_k + B(z_k \otimes u_k), \quad x_{k+1} = C^x z_{k+1}. \qquad (6)$$

A more comprehensive treatment of the Koopman operators for control-affine systems can be found in [10], [17].

III. GENERALIZED MOMENTUM-BASED KOOPMAN FORMULATION

This section presents a momentum-based formulation of the Koopman operator for Euler-Lagrangian dynamical systems. The general Euler-Lagrangian dynamics is given as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \qquad (7)$$

where $q \in \mathbb{R}^n$ and $\tau \in \mathbb{R}^m$ refer to the position coordinates and control inputs, respectively. $M(q) \in \mathbb{R}^{n \times n}$, $C(q, \dot{q}) \in \mathbb{R}^n$ and $G(q) \in \mathbb{R}^n$ represent the mass matrix, the Coriolis terms and the gravity term, respectively. Traditionally, position (q) and velocity ($\dot{q}$) coordinates are chosen as generalized coordinates to explicitly represent the state space for the Euler-Lagrangian system. The evolution of the generalized velocities is given as

$$\ddot{q} = M(q)^{-1} \left(\tau - C(q, \dot{q})\dot{q} - G(q) \right).$$

Due to their interaction with the mass matrix, control vectors become state-dependent. Consequently, this explicit representation lends itself more naturally to bilinear formulation (4) [10]. Although expressive, bilinear Koopman models are computationally more expensive, require larger datasets, and often preclude integration with non-linear MPC frameworks [12].

Instead, we employ an implicit momentum-based formulation, where the state space vector is defined as:

$$x = \begin{bmatrix} q \\ M(q)\dot{q} \end{bmatrix} = \begin{bmatrix} x_u \\ x_a \end{bmatrix}, \qquad (8)$$

where x_u and x_a represent the actuated and the unactuated part of the dynamics, respectively. The evolution of the proposed state space vector is given as

$$\dot{x} = \begin{bmatrix} M^{-1} x_a \\ (\dot{M}(x_u) - C(x_u, M^{-1} x_a)) M^{-1} x_a - G(x_u) + \tau \end{bmatrix}$$

$$= \begin{bmatrix} f_u(x) \\ f_a(x) + \tau \end{bmatrix} = f(x) + \tilde{u}, \qquad (9)$$

where $f(x) = [f_u(x), f_a(x)]^\top$ and $\tilde{u} = [0_n, u]^\top$ where $0_n \in \mathbb{R}^n$ represents the zero vector and $u = \tau$. Hence, the proposed choice of state space completely decouples the linear control inputs from the state-dependent dynamics, making resultant dynamics more amenable for the formulation of linear Koopman models (3).

Having separated a linear control channel, we can apply the Control Coherent Koopman modeling formulation [13] to construct a linear Koopman model by writing the passive/unactuated part of the dynamics as:

$$\dot{x}_p = \dot{x} - \tilde{u}, \qquad (10)$$

Consider the Koopman lifted state to be $z = [x^\top, \phi(x)^\top]^\top$ where $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^N$ represents the observable functions. Then, as per [13], we can construct a Koopman matrix to relate the passive and the actuated dynamics as

$$\begin{bmatrix} \dot{x}_p \\ \dot{\phi}(x) \end{bmatrix} = \begin{bmatrix} A_{c,xx} & A_{c,x\phi} \\ A_{c,x\phi} & A_{c,\phi\phi}, \end{bmatrix} \begin{bmatrix} x \\ \phi(x) \end{bmatrix} \qquad (11)$$

where the Koopman matrix is expressed as block related to the evolution of state variables (x) and the observable functions ($\phi(x)$). Substituting (10), we get

$$\dot{z} = A_c z + B_c u, \ x = C^x z, \qquad (12)$$

where $A_c = \begin{bmatrix} A_{c,xx} & A_{c,x\phi} \\ A_{c,x\phi} & A_{c,\phi\phi} \end{bmatrix}$, $B_c = \begin{bmatrix} 0_{n \times m} \\ I_{n \times m} \\ 0_{N \times m} \end{bmatrix}$, $C^x = \begin{bmatrix} I_{n \times n} \\ 0_{n \times N} \end{bmatrix}$. $I_{n \times n}$ represents a $n \times n$ identity matrix. The corresponding discrete time Koopman model is

$$z_{k+1} = A z_k + B u_k, \ x_k = C^x z_k. \qquad (13)$$

It is to be noted that for Control Coherent Modeling Formulation in [13], Koopman models have to be constructed at the actuator dynamics level as a consequence of explicit state space representation. This indirect approach becomes necessary to capture the influence of actuation in a linear manner on system dynamics, as required by the assumptions to construct the formulation in (11). By contrast, the implicit state space representation naturally resolves this issue and enables the direct application of linear Koopman operator modeling at the joint dynamics level.

Remark 1: In the proposed formulation (13), the input matrix B is known a priori. Hence, only the Koopman operator matrix A governing the passive dynamics needs to be learned. This leads to significant improvement in learning efficiency, as the number of learnable parameters is reduced by Nm and Nm^2 compared to linear and bilinear Koopman formulations based on explicit state representations, respectively.

Remark 2: In the momentum-based Koopman model (13), the control input u directly influences the momentum states x_a within the same timestep, while its effect on the remaining lifted states z propagates only in the subsequent timestep. This inherent one-step transmission delay is consistent with physical causality analyses [13], [18], where actuation first impacts the actuated subsystem and subsequently propagates through the coupled dynamics.

IV. NEURAL ARCHITECTURES FOR LEARNING PASSIVE DYNAMICS

This section describes the neural network architectures developed to learn the passive dynamics of Euler-Lagrangian systems. We introduce two distinct architectures. The first is trained using actuated data, in which control inputs actively influence the evolution of the states; the second is trained on
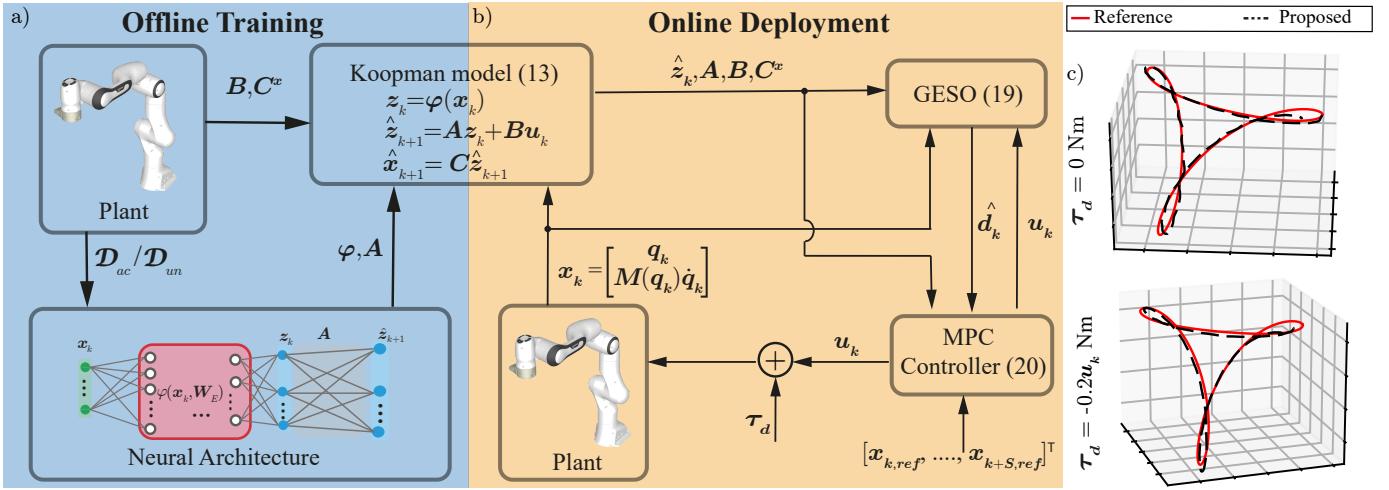
Fig. 1. Framework of the proposed momentum-based Koopman-MPC-GESO architecture. a) Model Training. b) Control framework. c) Trajectory tracking results.

unactuated data, where the system evolves without external actuation. The datasets used for training are defined as follows:

$$\boldsymbol{\mathcal{D}}_{ac} = (\boldsymbol{X}_i, \boldsymbol{Y}_i, \boldsymbol{U}_i)_{i=1}^p, \; \boldsymbol{\mathcal{D}}_{un} = (\boldsymbol{X}_i, \boldsymbol{Y}_i)_{i=1}^p, \quad (14)$$

where $\boldsymbol{X}_i = [\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_{w-1}^i]$ and $\boldsymbol{Y}_i = [\boldsymbol{x}_2^i, \boldsymbol{x}_3^i, \ldots, \boldsymbol{x}_w^i]$ are the snapshot matrices capturing consecutive state observations, and $\boldsymbol{U}_i = [\boldsymbol{u}_1^i, \boldsymbol{u}_2^i, \ldots, \boldsymbol{u}_{w-1}^i]$ denotes the corresponding control inputs for the actuated dataset. p and w represent the total number of trajectories in the dataset and the number of snapshots per trajectory, respectively.

Both networks share a common loss structure designed to ensure accurate lifting and prediction, within the Koopman space. The constituent components of the loss are defined as follows:

$$L_{pred} = \|\boldsymbol{x}_{k+1} - \boldsymbol{C}^x \hat{\boldsymbol{z}}_{k+1}\|_2, \; L_{lift} = \|\boldsymbol{z}_{k+1} - \hat{\boldsymbol{z}}_{k+1}\|_2, \quad (15)$$

where $\boldsymbol{z}_k$ and $\hat{\boldsymbol{z}}_k$ denote the ground truth and predicted values of the lifted state. The difference between the two networks lies in how $\hat{\boldsymbol{z}}_{k+1}$ is computed:

- For the **unactuated dataset**, where no control input is applied, $\hat{\boldsymbol{z}}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k$.
- For the **actuated dataset**, the evolution also depends on the control input, so $\hat{\boldsymbol{z}}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k + \boldsymbol{B}\boldsymbol{u}_k$. Note that, in this case, $\boldsymbol{B}$ is not a learnable parameter but a known matrix.

The total training loss is given as

$$L = \alpha_1 L_{pred} + \alpha_2 L_{lift} + \gamma_1 \|\boldsymbol{W}\|_1 + \gamma_2 \|\boldsymbol{W}\|_2, \quad (16)$$

where α_1 and α_2 are weighting coefficients for the prediction and lifting consistency losses, respectively. $\|\boldsymbol{W}\|_1$ and $\|\boldsymbol{W}\|_2$ represent the L_1 and L_2 norms of the network's trainable parameters, included to encourage sparsity and limit parameter magnitude. Hyperparameters γ_1 and γ_2 control the strength of the regularization terms.

Remark 3: Datasets collected under unactuated conditions may not achieve the same learning performance as actuated datasets, primarily due to reduced variability and limited excitation of the system dynamics. In particular, the absence of control inputs constrains the richness of the observed trajectories, which can limit the learning performance. Nonetheless, learning from unactuated data remains valuable when actuated data collection is costly, unsafe, or otherwise impractical.

V. Linear Generalized Extended State Observer

In practical scenarios, robotic systems are invariably subjected to external disturbances and unmodeled dynamics, possibly arising from the finite-dimensional approximation of the Koopman operator, that can degrade control performance. To enhance robustness against such uncertainties, we incorporate a Generalized Extended State Observer (GESO) [16] into the control framework. GESO enables real-time estimation of disturbances, which can then be actively compensated through the control input channel, thereby improving tracking accuracy and robustness.

To this end, consider the Euler-Lagrangain dynamics (7) in presence of external disturbance, $\boldsymbol{\tau}_d$:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_d, \quad (17)$$

Under the effect of the external disturbance and unmodeled dynamics, the Koopman dynamics (12) can now be written as:

$$\dot{\boldsymbol{x}} = \boldsymbol{C}^x(\boldsymbol{A}_c \boldsymbol{z} + \boldsymbol{B}_c \boldsymbol{u}) + \boldsymbol{d}, \quad (18)$$

where $\boldsymbol{d} = \boldsymbol{C}^x \boldsymbol{B}_c \boldsymbol{\tau}_d + \boldsymbol{\Delta}$ is the lumped disturbance term which represents the combined effect of model uncertainty $\boldsymbol{\Delta}$ and external disturbance $\boldsymbol{\tau}_d$. Subsequently, the GESO dynamics can be formulated as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{C}^x(\boldsymbol{A}_c \boldsymbol{z} + \boldsymbol{B}_c \boldsymbol{u}) + k_1(\boldsymbol{x} - \hat{\boldsymbol{x}}) + \hat{\boldsymbol{d}},$$
$$\dot{\hat{\boldsymbol{d}}} = k_2(\boldsymbol{x} - \hat{\boldsymbol{x}}), \quad (19)$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{d}}$ represent the actual state of the system and the lumped disturbance estimated by linear GESO, respectively. Once the lumped disturbance is estimated, the updated model can be fed to the controller for disturbance compensation.
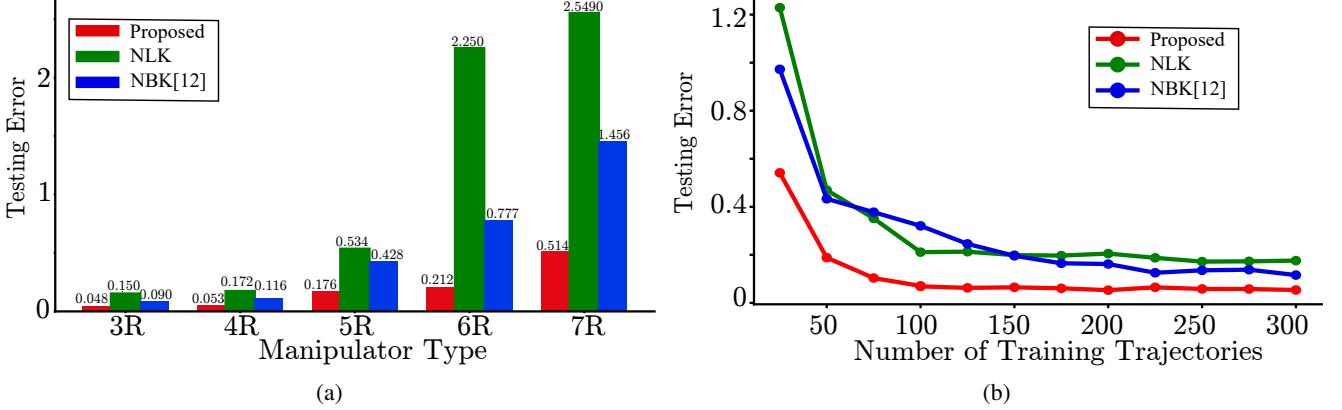
Fig. 2. Prediction performance comparison for the proposed scheme with Nominal Linear Koopman (NLK) and Nominal Bilinear Koopman (NBK) [12]. Testing errors across a) different manipulators, and b) different sizes of training datasets for the 4R manipulator.

VI. MODEL PREDICTIVE CONTROL

The proposed Koopman architecture is paired with linear MPC. Linear MPC with a prediction (and control) horizon s is defined as an optimization problem of the form

$$\min_{Z,U} \sum_{k=0}^{s} \begin{bmatrix} \boldsymbol{C}^x \boldsymbol{z}_k - \boldsymbol{x}_{ref,k} \\ \boldsymbol{u}_k \end{bmatrix}^T \boldsymbol{R} \begin{bmatrix} \boldsymbol{C}^x \boldsymbol{z}_k - \boldsymbol{x}_{ref,k} \\ \boldsymbol{u}_k \end{bmatrix} \quad (20)$$

$$\text{s.t } \boldsymbol{z}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{C}^{x\top}\hat{\boldsymbol{d}}_k dt, \ \boldsymbol{z}_0 = \boldsymbol{\varphi}(\boldsymbol{x}_0),$$
$$\boldsymbol{x}^- \leq \boldsymbol{C}^x \boldsymbol{z}_k \leq \boldsymbol{x}^+, \ \boldsymbol{u}^- \leq \boldsymbol{u}_k \leq \boldsymbol{u}^+, \ k = 0, ..., s-1,$$

where $\boldsymbol{x}_{ref,k}$ refers to the reference state at k^{th} timestep. $\boldsymbol{R}$ represents the penalty matrix, while $[\boldsymbol{x}^-, \boldsymbol{x}^+]$ and $[\boldsymbol{u}^-, \boldsymbol{u}^+]$ represent the state and input constraints, respectively. The full proposed momentum-based Koopman-MPC-GESO architecture is shown in Fig. 1.

Remark 4: The formulation of GESO as a linear observer allows it to be directly embedded within the MPC architecture while preserving the controller linearity. This integration allows disturbance estimation and compensation to be seamlessly integrated within the optimization solver, such that the state and input constraints are now enforced consistently within the MPC framework. By contrast, the method in [14] employs a nonlinear GESO, which cannot be incorporated into the linear MPC formulation without compromising the tracking performance. In particular, the observer is placed outside the control loop, forcing disturbances to be compensated in an ad-hoc manner. This separation not only undermines the controller's ability to enforce constraints systematically but also requires incorporating additional mechanisms, such as external saturation filters, to maintain safety, thus increasing complexity and potentially reducing overall robustness.

VII. RESULTS

This section presents a comprehensive evaluation of the proposed architecture through simulation and experimental studies on serial manipulators. The assessment considers both open-loop prediction accuracy and closed-loop control performance, thereby providing a rigorous demonstration of the effectiveness and robustness of the proposed framework.

A. Prediction Performance Comparison

This subsection evaluates the open-loop prediction performance of the proposed scheme. Comparisons are conducted against two baseline approaches, the Nominal Linear Koopman (NLK) model and the Nominal Bilinear Koopman (NBK) [12] model. Both baselines employ explicit state space representations, which is in contrast with the current study that relies on an implicit state representation. The datasets are collected at a simulation frequency of 100 Hz with 1000 snapshots in each trajectory.

Testing errors are used as comparison metrics. The testing errors are standardized to ensure fair comparison between the proposed scheme ($\boldsymbol{x} = [\boldsymbol{q}, \boldsymbol{M}\dot{\boldsymbol{q}}]^\top$) and the baseline nominal scheme ($\boldsymbol{x} = [\boldsymbol{q}, \dot{\boldsymbol{q}}]^\top$), which employ different state space coordinates. The corresponding performance comparison across different ranges of manipulators and sizes of datasets is shown in Fig. 2. The proposed scheme outperforms both NLK and, more importantly, NBK [12] across all the evaluations. This result is particularly noteworthy because bilinear formulations are typically regarded as better suited for Euler-Lagrangian systems due to the intrinsic coupling between states and inputs. Nevertheless, the momentum-based formulation achieves superior performance by structurally separating the linear input channel from the passive nonlinear dynamics, thereby making the system more amenable to linear Koopman modeling. This structural advantage not only reduces modeling complexity but also enables higher prediction accuracy than bilinear approaches, reconciling the dual objectives of modeling accuracy and computational efficiency in a way that bilinear approaches cannot. Further, it can be observed (Fig. 2b) that the linear models only require 100 trajectories to achieve the optimal performance, with the proposed scheme consistently outperforming the baselines across all dataset sizes. While NBK [12] requires a much larger dataset with about 250 trajectories to achieve convergence in terms of prediction capabilities while still offering considerably inferior performance.

Hence, the proposed formulation enables learning of linear Koopman models which offer superior performance as compared to bilinear Koopman models based on explicit state
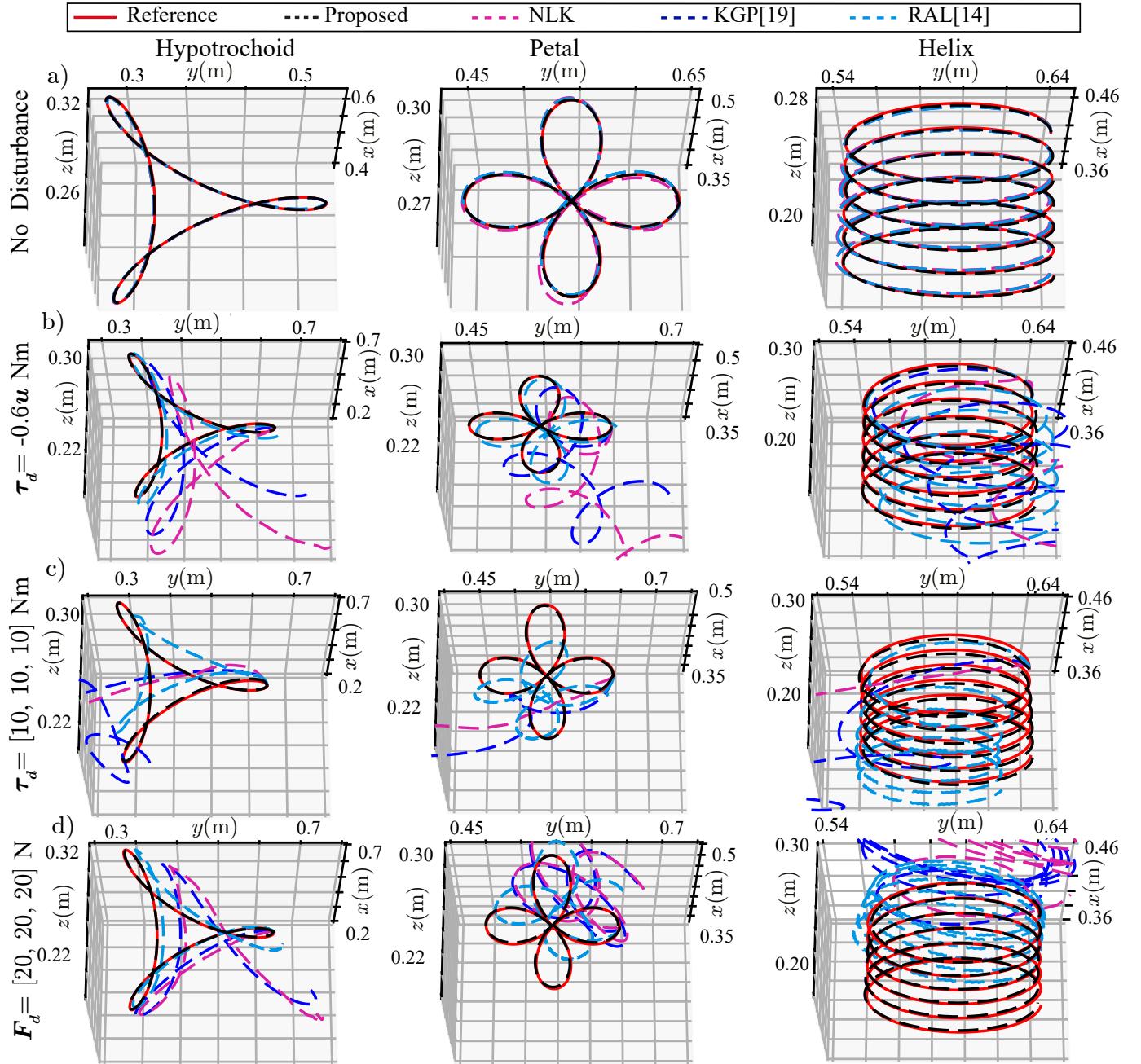
Fig. 3. Tracking performance comparisons for a 3R serial manipulator with a) No disturbance. b) Actuator fault ($\boldsymbol{\tau_d} = 0.6\boldsymbol{u}$ Nm). c) Constant disturbance torque ($\boldsymbol{\tau_d} = [10, 10, 10]$ Nm) d) Constant end effector load($\boldsymbol{F_d} = [20, 20, 20]$ N).

representation while offering higher data efficiency.

B. Closed Loop Control Performance Comparison

This subsection evaluates the closed-loop tracking performance of the proposed scheme through extensive simulations on a serial manipulator. Specifically, we consider a 3R serial manipulator operating in a 3D workspace, with physical parameters defined as $m_i = 0.6$ kg, $l_i = 0.33$ m, $I_i = \text{diag}\left[0, \frac{m_i l_i^2}{12}, \frac{m_i l_i^2}{12}\right]$ kgm^2 for $i = 1, 2, 3$, where m_i, l_i, and I_i denote the mass, length, and inertia of the ith link, respectively. The simulations are run at a frequency of 100 Hz. The parameters for GESO are $k_1 = 40$ and $k_2 = 800$. To

demonstrate the efficacy of the proposed architecture, we conduct comparison studies against state-of-the-art alternatives, including the NLK formulation, Koopman models enhanced with Gaussian Processes (KGP) [19], and the Robust Active Learning (RAL) algorithm [14], which constructs an offline model via active learning and augments online control using higher-order GESOs. For fairness, MPC is employed as the base controller across NLK, KGP [19], and RAL [14]. In the previous subsection, we have already demonstrated superior prediction performance over the bilinear models. Further, the computation burden of bilinear models paired with MPC leading to their infeasibility for real-time applications has
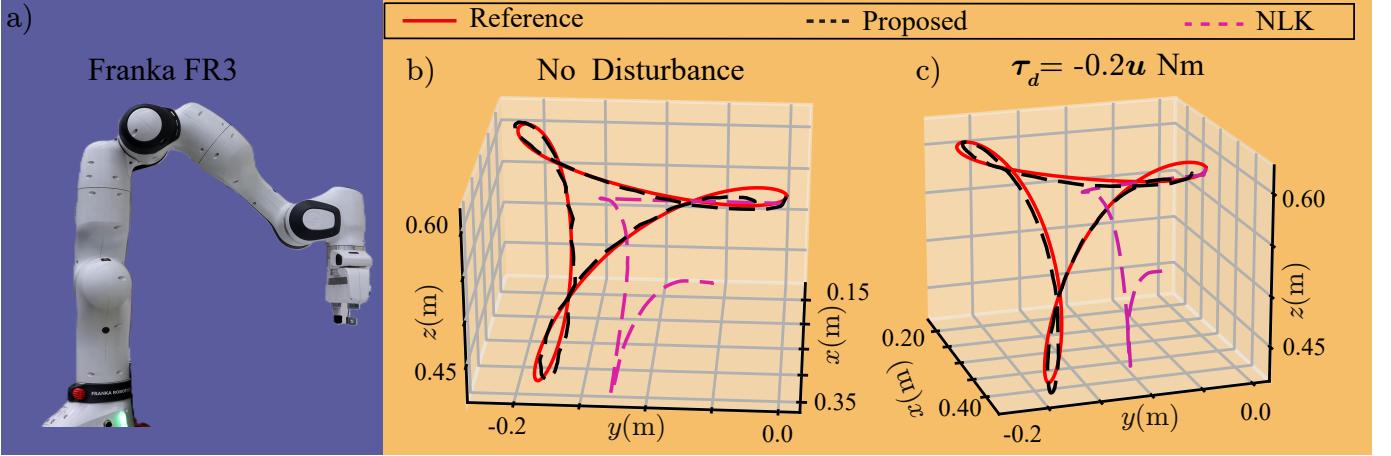
Fig. 4. Experimental results for trajectory tracking with a) Franka Research 3 manipulator b) without and c) with external disturbances.

already been studied extensively [11], [17]. Hence, we restrict ourselves to comparisons with linear models.

Initially, tracking performance evaluation is conducted for a scenario where the controller is tasked with following various reference trajectories in joint space in the absence of any external disturbances. The corresponding results are shown in Fig. 2a and Table I, which demonstrate the ability of the proposed architecture to accurately trace the reference path. It is noticed that the proposed momentum-based Koopman architecture provides better tracking performance compared to the nominal Koopman model. The superior performance relative to the nominal Koopman model can be attributable to its more precise model. Further, the proposed scheme gives similar or slightly better performance as compared to KGP [19] and RAL schemes [14].

Subsequently, we consider the tracking evaluation for trajectory tracking under the influence of various types of disturbance. In particular, we consider three types of disturbances namely, actuator faults (d_1) with $\tau_d = -0.6u$ Nm, constant torque disturbance (d_2) with $\tau_d = [10, 10, 10]$ Nm, and constant end effector load (d_3) of $F_d = [20, 20, 20]$ N. The corresponding results are shown in Figs. 3b - 3d and Table I. As can be seen, even under such extreme external disturbances, the proposed framework is able to track the desired trajectory effectively. This robustness is achieved by the incorporated GESO, which compensates for disturbances in real-time. In contrast, the baseline NLK architecture fails to mitigate the effects of these disturbances, resulting in deteriorated tracking performance and significant deviation from the desired path. Further, the proposed scheme also significantly outperforms KGP [19] and RAL [14] schemes, both of which are equipped with capabilities, a Gaussian process, and higher-order GESOs, respectively, to deal with disturbances. The relatively poor performance of RAL can be attributed to the limited prediction accuracy of the offline Koopman model obtained through active learning, which is inferior to the proposed scheme. Furthermore, since RAL [14] employs higher-order GESOs, these observers cannot be directly embedded within an MPC framework. Instead, they must be appended

to the MPC output, necessitating external saturation filters for safety, which acts as an additional hindrance to the overall effectiveness of RAL [14]. In addition, when paired with a high-gain base controller, higher-order GESOs in RAL [14] induce control chattering which introduces discontinuous corrective actions that further degrade closed-loop performance.

Table I: RMSE errors (m) for a variety of disturbances (d_0 : No Disturbance, d_1 : $\tau_d = -0.6u$ Nm, d_2 : $\tau_d = [10, 10, 10]$ Nm, $d_3 : F_d = [20, 20, 20]$ N.)

Disturbance	Proposed	NLK	KGP [19]	RAL [14]
Hypotrochoid				
d_0	**0.0028**	0.0138	0.0040	0.0030
d_1	**0.0029**	0.3507	0.1133	0.0117
d_2	**0.0045**	1.3218	0.5736	0.0739
d_3	**0.0034**	0.4573	0.1863	0.0764
Petal				
d_0	**0.0021**	0.0175	0.0062	0.0030
d_1	**0.0024**	0.892	0.1116	0.0121
d_2	**0.0040**	3.405	0.5973	0.0914
d_3	**0.0034**	0.9810	0.1578	0.0669
Helix				
d_0	0.0043	0.1164	0.0032	**0.0031**
d_1	**0.0048**	2.4147	0.1792	0.0162
d_1	**0.0079**	6.9205	0.7451	0.1090
d_2	**0.0038**	1.568	0.2736	0.0682

In summary, the proposed momentum-based Koopman framework achieves superior prediction accuracy and closed-loop tracking performance compared to the state-of-the-art alternatives. Further, the integration of GESOs enables the proposed architecture to maintain robust trajectory tracking even under substantial external disturbances.

C. Experimental Results

This subsection evaluates the closed-loop prediction performance of the proposed scheme on FRANKA Research 3 (Fig. 4a) manipulator arm. For learning the Koopman model, 500 trajectories are generated randomly within the workspace of the robot, each consisting of 1000 data points sampled

at 200 Hz. The parameters for GESO (19) are $k_1 = 100$ and $k_2 = 2000$. The effect of the proposed algorithms is evaluated by implementing the control architecture to trace a hypotrochoid path. The corresponding results are shown in Fig. 4.

As seen from Fig. 4b and c, the robot is able to trace the desired path accurately both in the presence and absence of the external disturbances, while the NLK model is unable to trace the desired paths in either case. The resulting RMSE errors for the proposed scheme are 0.024 m and 0.035 m for the case with and without disturbance, respectively. The corresponding errors for NLK are 0.077 m and 0.091 m. The poor tracking performance of the NLK stems from a combination of the poor prediction model and a lack of effective disturbance compensation. Hence, the experimental results demonstrate accurate real-time trajectory tracking of the proposed scheme with the added benefit of mitigating disturbances.

VIII. Conclusion

This paper presents a momentum-based Koopman operator formulation for Euler-Lagrangian dynamics that employs an implicit generalized momentum–based state space. This representation decouples a known linear actuation channel from the state-dependent dynamics, making the dynamics more amenable to linear Koopman modeling in contrast to explicit state space representations, which lend themselves more naturally to bilinear Koopman formulations. By leveraging this structural separation, the proposed approach requires to learns only the unactuated dynamics rather than the full actuation-dependent system, thereby significantly reducing the number of learnable parameters, improving data efficiency, and lowering overall model complexity. To realize this framework, we introduced two neural network architectures capable of constructing Koopman embeddings from either actuated or unactuated data, enabling flexible deployment under diverse data availability scenarios. Linear Generalized Extended State Observer (GESO) is integrated with proposed scheme to enhance robustness to external disturbances and unmodeled dynamics. Unlike higher-order nonlinear observers employed in previous studies, linear GESOs can be seamlessly incorporated into linear control frameworks, preserving their structure while enhancing stability and robustness. Extensive simulation and experimental studies are conducted to show that the momentum-based Koopman formulation achieves superior predictive accuracy compared to explicit state space representation-based linear and bilinear Koopman models, while the integrated GESO ensures reliable performance under a wide range of disturbances.

References

[1] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[2] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[3] J. H. Tu, *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

[4] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[5] M. Williams, I. Kevrekidis, and C. Rowley, "Extending dynamic mode decomposition: A data-driven approximation of the koopman operator," in *APS Division of Fluid Dynamics Meeting Abstracts*, pp. H17–003, 2014.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[7] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 10, p. 103111, 2017.

[8] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22445–22451, 2019.

[9] J. J. Craig, *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[10] D. Goswami and D. A. Paley, "Global bilinearization and controllability of control-affine nonlinear systems: A koopman spectral approach," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 6107–6112, IEEE, 2017.

[11] C. K. Sah, R. Singh, and J. Keshavan, "Real-time constrained tracking control of redundant manipulators using a koopman-zeroing neural network framework," *IEEE Robotics and Automation Letters*, 2024.

[12] D. Bruder, X. Fu, and R. Vasudevan, "Advantages of bilinear koopman realizations for the modeling and control of systems with unknown dynamics," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4369–4376, 2021.

[13] H. H. Asada and J. A. Solano-Castellanos, "Control-coherent koopman modeling: A physical modeling approach," *arXiv preprint arXiv:2403.16306*, 2024.

[14] S. Lyu, X. Lang, and D. Wang, "Koopman-based robust learning control with extended state observer," *IEEE Robotics and Automation Letters*, 2025.

[15] H. Chen and C. Lv, "Incorporating eso into deep koopman operator modeling for control of autonomous vehicles," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 5, pp. 1854–1864, 2024.

[16] J. Han, "From pid to active disturbance rejection control," *IEEE transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[17] R. Singh, C. K. Sah, and J. Keshavan, "Adaptive koopman embedding for robust control of nonlinear dynamical systems," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649251341907, 2025.

[18] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System dynamics: modeling and simulation of mechatronic systems*. J. Wiley, 2006.

[19] Y. Lu, W. Yao, Y. Xiao, and X. Xu, "Vector field-guided learning predictive control for motion planning of mobile robots with unknown dynamics," *arXiv preprint arXiv:2405.08283*, 2024.