2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Koopman Operator-Based Finite Impulse Response Filter for Nonlinear Systems*

Zhichao Pan[1], Biao Huang[2], and Fei Liu[1]

Abstract— **This paper proposes a novel Koopman operator-based finite impulse response (KFIR) filter for nonlinear dynamic systems. This filter is generalized from the minimum variance unbiased (MVU) FIR filter for linear systems by using a global linear approximation of the nonlinear dynamics obtained from Koopman operator theory and the extended dynamic mode decomposition (EDMD) algorithm. Based on the recursive linear model, a reduced-order FIR filtering structure is proposed, and the optimal gain is derived to minimize the trace of the estimation error covariance. Unlike traditional methods, the KFIR filter requires no prior knowledge of the initial state and fully utilizes the data of a moving horizon. Simulation results show that the proposed filter has excellent robustness against unexpected modeling uncertainties and inaccurate noise information, making it suitable for real applications.**

I. INTRODUCTION

Nonlinear dynamic systems are prevalent across various fields, including engineering, physics, economics, and biology. Accurate state estimation is crucial for controlling and monitoring such systems. However, developing effective filtering algorithms for nonlinear systems is a challenging task. While there is no universal mathematical framework capable of addressing all nonlinear problems, most existing nonlinear techniques, such as the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [1], fall under the infinite impulse response (IIR) structure. These methods leverage all past measurements, leading to a gradual accumulation of modeling and computational errors over time. Moreover, their success heavily relies on accurate model and noise assumptions. The utilization of inadequate models can result in compromised or even unstable estimation performance. Furthermore, the accumulation of errors frequently triggers a decline in performance or even divergence in the case of nonlinear IIR filters.

In contrast to the IIR structure, filters with a finite impulse response (FIR) architecture utilize only a finite set of recent data to estimate the current state, effectively preventing the accumulation of errors. Furthermore, FIR filters directly employ all recent data to optimize the current state, enhancing their robustness and degree of freedom. These filters possess advantageous engineering characteristics such as bounded input/bounded output (BIBO) stability, resilience against transient model uncertainties, and round-off errors [2], rendering them highly competitive for various applications. Due to their simplicity, stability, and robustness, FIR filters have been widely used for state estimation in linear dynamic systems [3]–[5]. However, designing FIR filters for nonlinear systems is challenging, and only a few attempts have been made to date. In [6], the unbiased FIR (UFIR) filter was extended to nonlinear systems and then applied to indoor robot localization [6]. In [7], an alternative nonlinear FIR filter that can manage the horizon length was proposed. These methods are based on the local linearization of the nonlinear system, which actually still requires previous estimation results.

In recent years, the Koopman operator has received significant attention in the field of nonlinear dynamic systems. The Koopman operator [8], [9] is an infinite-dimensional linear operator that describes the evolution of a system's observables over time. By utilizing the Koopman operator, a nonlinear system can be transformed into a linear system in an infinite-dimensional Hilbert space. With the advancement of theoretical studies [10] and data-driven techniques such as the dynamic mode decomposition (DMD) algorithm [11], [12] and deep learning [13] to find finite-dimensional approximations, the Koopman operator theory has emerged as a powerful tool for analyzing nonlinear systems. The use of Koopman theory has shown promising results in various applications, including control [14], forecasting, and identification [15] of nonlinear systems.

As a result of the well-established research on state estimation of linear systems, it is an attractive idea to apply the Koopman operator to nonlinear state estimation. In [16], a Koopman operator-based Kalman filter (KKF) was proposed, which used a linear regression model to approximate the Koopman operator from data. This method was later improved by using a bilinear approximation of the Koopman operator in [17]. However, data-driven models may have difficulty to capture time-varying system parameters, noises, and faults, which could impact the estimation accuracy. To overcome the non-Gaussian problem and outlier issues in the lifted state space, a robust generalized maximum-likelihood Koopman operator-based Kalman filter was introduced in [18] using the Student's t-distribution. However, all of these Koopman-based filters discussed above are IIR filters, which require prior knowledge of the initial state, and there is an

*This work was supported by The National Natural Science Foundation of China under Grant 61833007, the 111 Project under Grant B23008, and the China Scholarship Council (CSC). (Corresponding author: Fei Liu.)

[1]Zhichao Pan, Fei Liu are with the Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Institute of Automation, Jiangnan University, Wuxi, 214122, China 7191905019@stu.jiangnan.edu.cn; fliu@jiangnan.edu.cn

[2]Biao Huang is with department of chemical and materials engineering, University of Alberta, Edmonton, Alberta, T6G 1H9, Canada bhuang@ualberta.ca

error accumulation problem.

In this paper, we introduce a novel Koopman operator-based finite impulse response (KFIR) filter for state estimation of nonlinear dynamic systems. The proposed filter is a generalized form of the minimum variance unbiased (MVU) FIR filter designed for linear systems [3]. By utilizing the Koopman operator theory and the extended DMD (EDMD) algorithm, the nonlinear system is transformed and approximated into a linear model in a finite-dimensional space. Next, we propose a reduced-order FIR filtering structure and derive the optimal gain that minimizes the trace of the error covariance. The proposed KFIR filter does not require any data outside the moving horizon. A Gaussian-Newton method is adopted to estimate the unknown initial state. Compared to EKF and KKF, the proposed filter has better robustness against unexpected modeling uncertainties and inaccurate noise information.

The remainder of the paper is organized as follows. Section II formulates the state estimation problem for nonlinear systems. Section III constructs the Koopman-based linear system and its recursive model. Section IV proposes the FIR filter structure and derives the optimal gain. Simulation results to evaluate the performance of the proposed filter are presented in Section V. Finally, Section VI concludes the paper and discusses future research directions.

II. PROBLEM FORMULATION

Consider the following nonlinear dynamic systems:

$$\boldsymbol{x}_{k+1} = \mathbf{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k), \tag{1a}$$

$$\boldsymbol{y}_k = \mathbf{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k), \tag{1b}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$, $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$, and $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ refer to the system state, input, measurement, and noise respectively; nonlinear mappings $\mathbf{f} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$ and $\mathbf{g} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$ are single-valued and analytic. To provide a more general description and simplify the notation, both the process noise and the measurement noise are included in one vector $\boldsymbol{v}_k$. Additionally, it is assumed that $\boldsymbol{v}_k$ is a random variable with zero mean and known covariance, i.e.,

$$\text{mean}\{\boldsymbol{v}_k\} = \mathbf{0}, \qquad \text{cov}\{\boldsymbol{v}_k\} = \mathbf{Q}_v. \tag{2}$$

The objective is to design a filter to estimate the system state $\boldsymbol{x}_k$ by fully using a moving horizon of ℓ inputs and measurements $\{(\boldsymbol{u}_m, \boldsymbol{y}_m), ..., (\boldsymbol{u}_{k-1}, \boldsymbol{y}_{k-1}), (\boldsymbol{u}_k, \boldsymbol{y}_k)\}$, where $m = k - \ell + 1$. In contrast, data before time instant m will be ignored in estimating $\boldsymbol{x}_k$.

To solve the problem, a linear recursive model based on the Koopman operator through offline analysis is established in Section.III. Building upon this foundation, Section.IV presents the development of an online FIR filter.

III. LINEAR RECURSIVE MODEL BASED ON THE KOOPMAN OPERATOR

A. Koopman operator

Definition 1 (Koopman operator). Consider a nonlinear autonomous system $\boldsymbol{x}_{k+1} = \mathbf{f}(\boldsymbol{x}_k)$, where $\boldsymbol{x}_k \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$. The Koopman operator $\mathbf{K} : \mathbb{H} \to \mathbb{H}$ is defined as a linear operator who makes

$$\mathbf{K}\boldsymbol{\varphi} \triangleq \boldsymbol{\varphi} \circ \mathbf{f}, \quad \forall \varphi_i : \mathbb{X} \to \mathbb{R}^1 (or\ \mathbb{C}^1), \tag{3}$$

where $\boldsymbol{\varphi} = [\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \cdots]^\mathsf{T} \in \mathbb{H}$ is a vector of basis functions (typically infinite-dimensional).

Based on the above definition, a infinite-dimensional linear system can be represented as

$$\boldsymbol{\varphi}(\boldsymbol{x}_{k+1}) = \boldsymbol{\varphi}(\mathbf{f}(\boldsymbol{x}_k)) = \mathbf{K}\boldsymbol{\varphi}(\boldsymbol{x}_k). \tag{4}$$

According to the Koopman operator theorem [8], [10], the above linear system can fully describe the nonlinear dynamic $\boldsymbol{x}_{k+1} = \mathbf{f}(\boldsymbol{x}_k)$, without produce errors. Consider the input $\boldsymbol{u}_k$ and noise $\boldsymbol{v}_k$, we can define an extended state $\boldsymbol{\zeta}_k = \left[\boldsymbol{x}_k^\mathsf{T}, \boldsymbol{u}_k^\mathsf{T}, \boldsymbol{v}_k^\mathsf{T}\right]^\mathsf{T}$. Therefore, the system (1a) can be convert to

$$\boldsymbol{\varphi}(\boldsymbol{\zeta}_{k+1}) = \boldsymbol{\varphi}\big(\mathbf{f}(\boldsymbol{\zeta}_k), \mathbf{S}\boldsymbol{u}_k, \mathbf{S}\boldsymbol{v}_k\big) = \mathbf{K}\boldsymbol{\varphi}(\boldsymbol{\zeta}_k), \tag{5}$$

where $\mathbf{S}$ denote the left-shift operator, i.e., $\mathbf{S}\boldsymbol{u}_k = \boldsymbol{u}_{k+1}$. Readers can refer [10] for more details on Koopman operator theory.

The Koopman operator maps basis functions to their corresponding future states. However, for real systems, the analytic form of $\mathbf{K}$ is usually infinite-dimensional. To make it computationally feasible, a common approach is to use a finite-dimensional approximation in the form of a linear time-invariant (LTI) system, which can be expressed as

$$\boldsymbol{z}_{k+1} = \mathbf{A}\boldsymbol{z}_k + \mathbf{B}\boldsymbol{u}_k + \mathbf{E}\boldsymbol{v}_k + \mathbf{G}\boldsymbol{\delta}_k, \tag{6a}$$

$$\boldsymbol{y}_k = \mathbf{C}\boldsymbol{z}_k + \mathbf{D}\boldsymbol{u}_k + \mathbf{F}\boldsymbol{v}_k + \mathbf{H}\boldsymbol{\delta}_k, \tag{6b}$$

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{\psi}(\boldsymbol{x}_k) \end{bmatrix}, \quad \boldsymbol{x}_k = \mathbf{T}\boldsymbol{z}_k. \tag{6c}$$

In this system, $\boldsymbol{z}_k \in \mathbb{R}^{N_x}$ represents the lifted state, where N_x is the number of the lifted state variables. The vector $\boldsymbol{\psi}(\boldsymbol{x}_k) \in \mathbb{R}^{n_\psi}$ is a set of finite-dimensional basis functions that are used to approximate the Koopman operator, it is a simplified version of $\boldsymbol{\varphi}(\boldsymbol{\zeta}_k)$. The approximation error $\boldsymbol{\delta}_k \in \mathbb{R}^{N_x+n_y}$ consists of two parts: $\boldsymbol{\delta}_k = \left[\boldsymbol{\delta}_k^{x\mathsf{T}}, \boldsymbol{\delta}_k^{y\mathsf{T}}\right]^\mathsf{T}$, where $\boldsymbol{\delta}_k^x \in \mathbb{R}^{N_x}$ and $\boldsymbol{\delta}_k^y \in \mathbb{R}^{n_y}$. The matrices $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$, $\mathbf{B} \in \mathbb{R}^{N_x \times n_u}$, $\mathbf{C} \in \mathbb{R}^{n_y \times N_x}$, $\mathbf{F} \in \mathbb{R}^{n_y \times N_x}$, $\mathbf{G} = [\mathbf{I},\ \mathbf{0}] \in \mathbb{R}^{N_x \times (N_x+n_y)}$, and $\mathbf{H} = [\mathbf{0},\ \mathbf{I}] \in \mathbb{R}^{n_y \times (N_x+n_y)}$ are the lifted system matrices, respectively. $\mathbf{T} = [\mathbf{I},\ \mathbf{0}] \in \mathbb{R}^{n_x \times N_x}$ is used to transform $\boldsymbol{z}_k$ back to $\boldsymbol{x}_k$.

Remark 1. To ensure differentiability of the Koopman operator's basis functions, which is required for derivative calculations in Section IV-B, here the optional types of basis functions are restricted to smooth ones, e.g., polynomials, sines, and Gaussians.

To facilitate filter design, a global linearization between $\boldsymbol{z}_k$ and $\boldsymbol{x}_k$ is obtained as

$$\boldsymbol{z}_k = \mathbf{L}\boldsymbol{x}_k + \mathbf{c}_z + \boldsymbol{\varepsilon}_k, \tag{7}$$

where $\mathbf{L} \in \mathbb{R}^{N_x \times n_x}$ is the linear transformation matrix, and $\mathbf{c}_z$ and $\boldsymbol{\varepsilon}_k \in \mathbb{R}^{N_x}$ are the bias and error vectors, respectively.

Matrices in (6a), (6b), and (7) can be determined by using the extended dynamic mode decomposition (EDMD) algorithm [12]:

$$\mathbf{A},\mathbf{B},\mathbf{E} = arg\min_{\mathbf{A},\mathbf{B},\mathbf{E}} \|\tilde{\mathbf{Z}}_+^{\mathrm{D}} - \mathbf{A}\tilde{\mathbf{Z}}^{\mathrm{D}} - \mathbf{B}\tilde{\mathbf{U}}^{\mathrm{D}} - \mathbf{E}\tilde{\mathbf{V}}^{\mathrm{D}}\|_{\mathrm{F}}^2, \tag{8a}$$

$$\mathbf{C},\mathbf{D},\mathbf{F} = arg\min_{\mathbf{C},\mathbf{D},\mathbf{F}} \|\tilde{\mathbf{Y}}^{\mathrm{D}} - \mathbf{C}\tilde{\mathbf{Z}}^{\mathrm{D}} - \mathbf{D}\tilde{\mathbf{U}}^{\mathrm{D}} - \mathbf{F}\tilde{\mathbf{V}}^{\mathrm{D}}\|_{\mathrm{F}}^2, \tag{8b}$$

$$\mathbf{L},\mathbf{c}_z = arg\min_{\mathbf{L},\mathbf{c}_z} \|\tilde{\mathbf{Z}}^{\mathrm{D}} - \mathbf{L}\tilde{\mathbf{X}}^{\mathrm{D}} - \mathbf{c}_z\mathbf{1}\|_{\mathrm{F}}^2, \tag{8c}$$

where $\tilde{\mathbf{X}}^{\mathrm{D}} = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{\mathrm{M}}]$, $\tilde{\mathbf{Y}}^{\mathrm{D}} = [\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_{\mathrm{M}}]$, $\tilde{\mathbf{U}}^{\mathrm{D}} = [\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{\mathrm{M}}]$, and $\tilde{\mathbf{V}}^{\mathrm{D}} = [\boldsymbol{v}_0, \boldsymbol{v}_1, ..., \boldsymbol{v}_{\mathrm{M}}]$ represent the data stack matrices of $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ respectively, which are generated from the real nonlinear system (1). M denotes the data length. $\tilde{\mathbf{Z}}^{\mathrm{D}} = [\boldsymbol{z}_0, \boldsymbol{z}_1, ..., \boldsymbol{z}_{\mathrm{M}}]$, and $\tilde{\mathbf{Z}}_+^{\mathrm{D}} = [\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{\mathrm{M}+1}]$, are the data stack matrices of $\boldsymbol{z}$ and the next step of $\boldsymbol{z}$, respectively, which are obtained by substituting $\tilde{\mathbf{X}}^{\mathrm{D}}$ into the lifting function (6c). The analytical solution to (8) is

$$[\mathbf{A},\mathbf{B},\mathbf{E}] = \tilde{\mathbf{Z}}_+^{\mathrm{D}} \begin{bmatrix} \tilde{\mathbf{Z}}^{\mathrm{D}} \\ \tilde{\mathbf{U}}^{\mathrm{D}} \\ \tilde{\mathbf{V}}^{\mathrm{D}} \end{bmatrix}^{\dagger}, \quad [\mathbf{C},\mathbf{D},\mathbf{F}] = \tilde{\mathbf{Y}}^{\mathrm{D}} \begin{bmatrix} \tilde{\mathbf{Z}}^{\mathrm{D}} \\ \tilde{\mathbf{U}}^{\mathrm{D}} \\ \tilde{\mathbf{V}}^{\mathrm{D}} \end{bmatrix}^{\dagger}, \tag{9a}$$

$$[\mathbf{L},\mathbf{c}_z] = \tilde{\mathbf{Z}}^{\mathrm{D}} \begin{bmatrix} \tilde{\mathbf{X}}^{\mathrm{D}} \\ \mathbf{1} \end{bmatrix}^{\dagger}, \tag{9b}$$

where † denotes the pseudo-inverse.

Meanwhile, the variances of the errors $\boldsymbol{\delta}_k^x$, $\boldsymbol{\delta}_k^y$, and $\boldsymbol{\varepsilon}_k$ can also be estimated using the available data set:

$$\begin{aligned} &\mathrm{cov}\{\boldsymbol{\delta}_k\} = \mathbf{R}_\delta = \mathrm{diag}\{\mathbf{R}_\delta^x, \mathbf{R}_\delta^y\}, \\ &\text{with } \mathbf{R}_\delta^x \approx \mathrm{cov}\{\tilde{\mathbf{Z}}_+^{\mathrm{D}} - \mathbf{A}\tilde{\mathbf{Z}}^{\mathrm{D}} - \mathbf{B}\tilde{\mathbf{U}}^{\mathrm{D}} - \mathbf{E}\tilde{\mathbf{V}}^{\mathrm{D}}\}, \\ &\qquad \mathbf{R}_\delta^y \approx \mathrm{cov}\{\tilde{\mathbf{Y}}^{\mathrm{D}} - \mathbf{C}\tilde{\mathbf{Z}}^{\mathrm{D}} - \mathbf{D}\tilde{\mathbf{U}}^{\mathrm{D}} - \mathbf{F}\tilde{\mathbf{V}}^{\mathrm{D}}\}, \end{aligned} \tag{10a}$$

$$\mathrm{cov}\{\boldsymbol{\varepsilon}_k\} = \mathbf{R}_\varepsilon \approx \mathrm{cov}\{\tilde{\mathbf{Z}}^{\mathrm{D}} - \mathbf{L}\tilde{\mathbf{X}}^{\mathrm{D}} - \mathbf{c}_z\mathbf{1}\}. \tag{10b}$$

To obtain the statistical characteristics accurately, the data set used in (10) should be distinguished from that of (8).

Remark 2. The data stack can be obtained by simulating the nonlinear system model (1), with the simulation involving either a single trajectory or multiple trajectories.

B. The linear recursive model

The system (6) on a horizon of ℓ points, with recursively computed forward-in-time solutions, is derived as follows

$$\boldsymbol{z}_{[k,m]} = \mathbf{A}_\ell \boldsymbol{z}_m + \mathbf{B}_\ell \boldsymbol{u}_{[k,m]} + \mathbf{E}_\ell \boldsymbol{v}_{[k,m]} + \mathbf{G}_\ell \boldsymbol{\delta}_{[k,m]}, \tag{11a}$$

$$\boldsymbol{y}_{[k,m]} = \mathbf{C}_\ell \boldsymbol{z}_m + \mathbf{D}_\ell \boldsymbol{u}_{[k,m]} + \mathbf{F}_\ell \boldsymbol{v}_{[k,m]} + \mathbf{H}_\ell \boldsymbol{\delta}_{[k,m]}, \tag{11b}$$

where $\boldsymbol{z}_{[k,m]} = \left[\boldsymbol{z}_k^{\mathsf{T}}, ..., \boldsymbol{z}_m^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{\ell N_x}$, $\boldsymbol{y}_{[k,m]} = \left[\boldsymbol{y}_k^{\mathsf{T}}, ..., \boldsymbol{y}_m^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{\ell n_y}$, $\boldsymbol{u}_{[k,m]} = \left[\boldsymbol{u}_k^{\mathsf{T}}, ..., \boldsymbol{u}_m^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{\ell n_u}$, $\boldsymbol{v}_{[k,m]} = \left[\boldsymbol{v}_k^{\mathsf{T}}, ..., \boldsymbol{v}_m^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{\ell n_w}$, $\boldsymbol{v}_{[k,m]} = \left[\boldsymbol{v}_k^{\mathsf{T}}, ..., \boldsymbol{v}_m^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{\ell n_v}$. The extended system matrices $\mathbf{A}_\ell \in \mathbb{R}^{\ell N_x \times \ell N_x}$, $\mathbf{B}_\ell \in \mathbb{R}^{\ell N_x \times \ell n_u}$, $\mathbf{E}_\ell \in \mathbb{R}^{\ell N_x \times \ell n_v}$, $\mathbf{C}_\ell \in \mathbb{R}^{\ell n_y \times \ell N_x}$, $\mathbf{D}_\ell \in \mathbb{R}^{\ell n_y \times \ell n_u}$, and $\mathbf{F}_\ell \in \mathbb{R}^{\ell n_y \times \ell n_v}$ are time-invariant, and are specified as

$$\begin{aligned} \mathbf{A}_\ell &= \left[\left(\mathbf{A}^{\ell-1}\right)^{\mathsf{T}}, \ \left(\mathbf{A}^{\ell-2}\right)^{\mathsf{T}}, \ \ldots, \ \mathbf{A}^{\mathsf{T}}, \ \mathbf{I}\right]^{\mathsf{T}}, \\ \mathbf{B}_\ell &= \begin{bmatrix} \mathbf{0} & \mathbf{B} & \cdots & \mathbf{A}^{\ell-2}\mathbf{B} & \mathbf{A}^{\ell-1}\mathbf{B} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{A}^{\ell-3}\mathbf{B} & \mathbf{A}^{\ell-2}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{E}_\ell &= \begin{bmatrix} \mathbf{0} & \mathbf{E} & \cdots & \mathbf{A}^{\ell-2}\mathbf{E} & \mathbf{A}^{\ell-1}\mathbf{E} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{A}^{\ell-3}\mathbf{E} & \mathbf{A}^{\ell-2}\mathbf{E} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{E} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{G}_\ell &= \begin{bmatrix} \mathbf{0} & \mathbf{G} & \cdots & \mathbf{A}^{\ell-2}\mathbf{G} & \mathbf{A}^{\ell-1}\mathbf{G} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{A}^{\ell-3}\mathbf{G} & \mathbf{A}^{\ell-2}\mathbf{G} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{G} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{C}_\ell &= \mathrm{diag}\big\{\mathbf{C},\mathbf{C},...,\mathbf{C}\big\}\mathbf{A}_\ell, \\ \mathbf{D}_\ell &= \mathrm{diag}\big\{\mathbf{C},\mathbf{C},...,\mathbf{C}\big\}\mathbf{B}_\ell + \mathrm{diag}\big\{\mathbf{D},\mathbf{D},...,\mathbf{D}\big\}, \\ \mathbf{F}_\ell &= \mathrm{diag}\big\{\mathbf{C},\mathbf{C},...,\mathbf{C}\big\}\mathbf{E}_\ell + \mathrm{diag}\big\{\mathbf{F},\mathbf{F},...,\mathbf{F}\big\}, \\ \mathbf{H}_\ell &= \mathrm{diag}\big\{\mathbf{C},\mathbf{C},...,\mathbf{C}\big\}\mathbf{G}_\ell + \mathrm{diag}\big\{\mathbf{H},\mathbf{H},...,\mathbf{H}\big\}, \end{aligned} \tag{12}$$

The variances of uncertainties $\boldsymbol{v}_{[k,m]}$ and $\boldsymbol{\delta}_{[k,m]}$ are

$$\mathrm{cov}\{\boldsymbol{v}_{[k,m]}\} = \mathbf{Q}_V = \mathrm{diag}\{\mathbf{Q}_v, \mathbf{Q}_v, ..., \mathbf{Q}_v\}, \tag{13a}$$

$$\mathrm{cov}\{\boldsymbol{\delta}_{[k,m]}\} = \mathbf{R}_\Delta = \mathrm{diag}\{\mathbf{R}_\delta, \mathbf{R}_\delta, ..., \mathbf{R}_\delta\}. \tag{13b}$$

Moreover, the ℓ-steps prediction from $\boldsymbol{z}_m$ to $\boldsymbol{z}_k$ can also be derived from (11a) as

$$\boldsymbol{z}_k = \bar{\mathbf{A}}_\ell \boldsymbol{z}_m + \bar{\mathbf{B}}_\ell \boldsymbol{u}_{[k,m]} + \bar{\mathbf{E}}_\ell \boldsymbol{v}_{[k,m]} + \bar{\mathbf{G}}_\ell \boldsymbol{\delta}_{[k,m]}, \tag{14}$$

where $\bar{\mathbf{A}}_\ell$, $\bar{\mathbf{B}}_\ell$, $\bar{\mathbf{E}}_\ell$, and $\bar{\mathbf{G}}_\ell$ refer to the first N_x rows of $\mathbf{A}_\ell$, $\mathbf{B}_\ell$, $\mathbf{E}_\ell$, and $\mathbf{G}_\ell$, respectively. This equation will be used in the filter design.

IV. FILTER DESIGN

A. FIR filter structure

To estimate the value of $\boldsymbol{x}_k$ based on the extended system (11), we propose the use of a reduced-order filter with a finite impulse response (FIR) structure:

$$\hat{\boldsymbol{x}}_k = \boldsymbol{\Lambda}_k \boldsymbol{y}_{[k,m]} + \boldsymbol{\mu}_k, \tag{15}$$

where $\boldsymbol{\Lambda}_k \in \mathbb{R}^{n_x \times \ell n_y}$ is the filter gain, and $\boldsymbol{\mu}_k \in \mathbb{R}^{n_x}$ is a compensation vector that ensures $\hat{\boldsymbol{x}}_k$ is unbiased.

Define the estimation error as $\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k$. The optimal gain is designed by minimizing the trace of the error covariance $\boldsymbol{P}_k = \mathrm{cov}\{\boldsymbol{e}_k\}$:

$$\boldsymbol{\Lambda}_k^*, \boldsymbol{\mu}_k^* = arg\min_{\boldsymbol{\Lambda}_k, \boldsymbol{\mu}_k} \mathrm{tr}\{\boldsymbol{P}_k\}. \tag{16}$$

To solve the optimal gain, the estimation error should be analyzed. By substituting (15) and (14) into the estimation error expression, one can obtain:

$$\begin{aligned}\boldsymbol{e}_k &= \mathbf{T}\boldsymbol{z}_k - \hat{\boldsymbol{x}}_k \\ &= \left(\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k\mathbf{C}_\ell\right)\boldsymbol{z}_m + \left(\mathbf{T}\bar{\mathbf{B}}_\ell - \boldsymbol{\Lambda}_k\mathbf{D}_\ell\right)\boldsymbol{u}_{[k,m]} \\ &\quad + \left(\mathbf{T}\bar{\mathbf{E}}_\ell - \boldsymbol{\Lambda}_k\mathbf{F}_\ell\right)\boldsymbol{v}_{[k,m]} + \left(\mathbf{T}\bar{\mathbf{G}}_\ell - \boldsymbol{\Lambda}_k\mathbf{H}_\ell\right)\boldsymbol{\delta}_{[k,m]} \\ &\quad - \boldsymbol{\mu}_k. \end{aligned} \tag{17}$$

It implies that the estimation error relates to the unknown initial lifted state $\boldsymbol{z}_m$. To this end, existing linear FIR filters can compute the covariance of $\boldsymbol{z}_m$ [19], or use the unbiased condition $\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k\mathbf{C}_\ell$ to eliminate the influence of $\boldsymbol{z}_m$ [3]. However, both methods are too conservative for the Koopman-based linear system (6) since the definition of (6c) provides additional information about $\boldsymbol{z}_m$. In other words, the degree of freedom of $\boldsymbol{z}_m$ is only equal to that of $\boldsymbol{x}_m$. If $\boldsymbol{x}_m$ is determined, then $\boldsymbol{z}_m$ is also determined. For this purpose, the constraint (6c) should be taken into account when dealing with $\boldsymbol{z}_m$.

B. Initial state estimation

We propose a novel method based on weighted nonlinear least squares to estimate the unknown initial lifted state $\boldsymbol{z}_m$, aiming to improve estimation accuracy by utilizing the constraint information in the lift function (6c). The optimization problem is formulated as minimizing the fitting error:

$$\hat{\boldsymbol{x}}_m^* = arg\min_{\hat{\boldsymbol{x}}_m} \|\boldsymbol{r}(\hat{\boldsymbol{x}}_m)\|^2_{\mathbf{M}^{-1}}, \tag{18}$$

where the residuals $\boldsymbol{r}(\hat{\boldsymbol{x}}_m)$ and the penalty matrix $\mathbf{M}$ are specified as

$$\boldsymbol{r}(\hat{\boldsymbol{x}}_m) = \boldsymbol{y}_{[k,m]} - \mathbf{C}_\ell\begin{bmatrix}\hat{\boldsymbol{x}}_m \\ \boldsymbol{\psi}(\hat{\boldsymbol{x}}_m)\end{bmatrix} - \mathbf{D}_\ell\boldsymbol{u}_{[k,m]}, \tag{19a}$$

$$\mathbf{M} = \mathbf{F}_\ell\mathbf{Q}_V\mathbf{F}_\ell^\mathsf{T} + \mathbf{H}_\ell\mathbf{R}_\Delta\mathbf{H}_\ell^\mathsf{T}. \tag{19b}$$

The optimization problem (18) belongs to the category of weighted nonlinear least squares thus has no analytical solution. Therefore, we use the Gauss-Newton algorithm to solve it iteratively:

$$\hat{\boldsymbol{x}}_m^{t+1} = \hat{\boldsymbol{x}}_m^t + \boldsymbol{K}_m^t\boldsymbol{r}(\hat{\boldsymbol{x}}_m^t), \quad t \in \{0, 1, ..., t_s\} \tag{20}$$

where t_s is the maximum number of iterations, the step size $\boldsymbol{K}_m^t$ is computed as

$$\boldsymbol{K}_m^t = \left[(\mathbf{C}_\ell\boldsymbol{L}_m^t)^\mathsf{T}\mathbf{M}^{-1}(\mathbf{C}_\ell\boldsymbol{L}_m^t)\right]^{-1}(\mathbf{C}_\ell\boldsymbol{L}_m^t)^\mathsf{T}\mathbf{M}^{-1}, \tag{21a}$$

$$\boldsymbol{L}_m^t = \begin{bmatrix}\mathbf{I} \\ \boldsymbol{J}_m^t\end{bmatrix}, \qquad \boldsymbol{J}_m^t = \left.\frac{\partial\boldsymbol{\psi}(\boldsymbol{x})}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_m^t}. \tag{21b}$$

To begin with, the global linearization equation (7) is utilized. By combining (7) with (11b), an unbiased smoother can be employed:

$$\hat{\boldsymbol{x}}_m^0 = \boldsymbol{K}_m^0\left(\boldsymbol{y}_{[k,m]} - \mathbf{C}_\ell\mathbf{c}_z - \mathbf{D}_\ell\boldsymbol{u}_{[k,m]}\right), \tag{22}$$

where

$$\boldsymbol{K}_m^0 = \left[(\mathbf{C}_\ell\mathbf{J})^\mathsf{T}\bar{\mathbf{M}}^{-1}(\mathbf{C}_\ell\mathbf{J})\right]^{-1}(\mathbf{C}_\ell\mathbf{J})^\mathsf{T}\bar{\mathbf{M}}^{-1}, \tag{23a}$$

$$\bar{\mathbf{M}} = \mathbf{M} + \mathbf{C}_\ell\mathbf{R}_\varepsilon\mathbf{C}_\ell^\mathsf{T}. \tag{23b}$$

Finally, the iteration process will stop when either the estimate converges or when the iteration step exceeds t_s.

Remark 3. The initial state estimation utilizes only the data within the moving horizon. While it is possible to obtain $\hat{\boldsymbol{x}}_m$ from the previous estimation step, doing so would result in a filter that no longer belongs to a FIR structure.

C. Optimal gain

After obtaining $\hat{\boldsymbol{x}}_m^*$, the lifted state estimate $\hat{\boldsymbol{z}}_m^* = \left[(\hat{\boldsymbol{x}}_m^*)^\mathsf{T}, \boldsymbol{\psi}^\mathsf{T}(\hat{\boldsymbol{x}}_m^*)\right]^\mathsf{T}$ can be computed. However, the estimation error $\boldsymbol{z}_m - \hat{\boldsymbol{z}}_m^*$ is unknown. To reduce the influence of the error, the following local linearization is used to describe the dependence between $\boldsymbol{z}_m$ and $\boldsymbol{x}_m$:

$$\boldsymbol{z}_m = \hat{\boldsymbol{z}}_m^* + \boldsymbol{L}_m^*(\boldsymbol{x}_m - \hat{\boldsymbol{x}}_m^*) + \boldsymbol{\sigma}_m, \tag{24}$$

where $\boldsymbol{L}_m^*$ is the result of $\boldsymbol{L}_m^t$ in (21b) when $\hat{\boldsymbol{x}}_m^t = \hat{\boldsymbol{x}}_m^*$, and $\boldsymbol{\sigma}_m \in \mathbb{R}^{N_x}$ represents the linearization error. In this way, the error $\boldsymbol{z}_m - \hat{\boldsymbol{z}}_m^*$ is divided into two parts. By substituting (17) into (24), the first part $\boldsymbol{L}_m^*(\boldsymbol{x}_m - \hat{\boldsymbol{x}}_m^*)$ can be eliminated by using the new unbiased condition

$$\left(\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k\mathbf{C}_\ell\right)\boldsymbol{L}_m^* = \mathbf{0}. \tag{25}$$

Compared the direct use of $\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k\mathbf{C}_\ell$, the constraint (25) is time-varying and more relaxed. It is still a difficult task to obtain the covariance of the local linearization error $\boldsymbol{\sigma}_m$, so we approximate it as a decay of the covariance of $\mathbf{R}_\varepsilon$, which is the global linearization error:

$$\mathrm{cov}\{\boldsymbol{\sigma}_m\} = \mathbf{R}_\sigma \approx \alpha\mathbf{R}_\varepsilon, \qquad \alpha \in [0, 1]. \tag{26}$$

Now, by using $\boldsymbol{\mu}_k$ to balance known bias in (17), the estimation error covariance becomes

$$\begin{aligned}\boldsymbol{P}_k &= \left(\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k\mathbf{C}_\ell\right)\mathbf{R}_\sigma\left(\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k\mathbf{C}_\ell\right)^\mathsf{T} \\ &\quad + \left(\mathbf{T}\bar{\mathbf{E}}_\ell - \boldsymbol{\Lambda}_k\mathbf{G}_\ell\right)\mathbf{Q}_V\left(\mathbf{T}\bar{\mathbf{E}}_\ell - \boldsymbol{\Lambda}_k\mathbf{G}_\ell\right)^\mathsf{T} \\ &\quad + \left(\mathbf{T}\bar{\mathbf{G}}_\ell - \boldsymbol{\Lambda}_k\mathbf{H}_\ell\right)\mathbf{R}_\Delta\left(\mathbf{T}\bar{\mathbf{G}}_\ell - \boldsymbol{\Lambda}_k\mathbf{H}_\ell\right)^\mathsf{T}. \end{aligned} \tag{27}$$

Then, the optimization problem (16) can be rewritten as the following form:

$$\boldsymbol{\Lambda}_k^* = arg\min_{\boldsymbol{\Lambda}_k} \mathrm{tr}\{\boldsymbol{P}_k\}, \quad \text{s.t. } (25). \tag{28}$$

Finally, the solution is given by the following Theorem.

Theorem 1. Given system (11), the optimal FIR filter minimizing the trace of the estimation error covariance is expressed as

$$\hat{\boldsymbol{x}}_k^* = \boldsymbol{\Lambda}_k^*\boldsymbol{y}_{[k,m]} + \boldsymbol{\mu}_k^*, \tag{29}$$

with

$$[\boldsymbol{\Lambda}_k^*, \diamond] = \begin{bmatrix}\mathbf{T}\boldsymbol{\Pi} & \mathbf{T}\mathbf{A}_\ell\boldsymbol{L}_m^*\end{bmatrix}\begin{bmatrix}\boldsymbol{\Omega} & \mathbf{C}_\ell\boldsymbol{L}_m^* \\ (\mathbf{C}_\ell\boldsymbol{L}_m^*)^\mathsf{T} & \mathbf{0}\end{bmatrix}^\dagger, \tag{30a}$$

$$\boldsymbol{\mu}_k^* = \left(\mathbf{T}\bar{\mathbf{A}}_\ell - \boldsymbol{\Lambda}_k^*\mathbf{C}_\ell\right)\hat{\boldsymbol{z}}_m^* + \left(\mathbf{T}\bar{\mathbf{B}}_\ell - \boldsymbol{\Lambda}_k^*\mathbf{D}_\ell\right)\boldsymbol{u}_{[k,m]}, \tag{30b}$$

where $\diamond$ can be set free, and matrices $\mathbf{\Pi}$ and $\mathbf{\Omega}$ are specified as

$$\mathbf{\Pi} = \bar{\mathbf{A}}_\ell \mathbf{R}_\sigma \mathbf{C}_\ell^\mathsf{T} + \bar{\mathbf{E}}_\ell \mathbf{Q}_V \mathbf{F}_\ell^\mathsf{T} + \bar{\mathbf{G}}_\ell \mathbf{R}_\Delta \mathbf{H}_\ell^\mathsf{T} \tag{31a}$$

$$\mathbf{\Omega} = \mathbf{C}_\ell \mathbf{R}_\sigma \mathbf{C}_\ell^\mathsf{T} + \mathbf{F}_\ell \mathbf{Q}_V \mathbf{F}_\ell^\mathsf{T} + \mathbf{H}_\ell \mathbf{R}_\Delta \mathbf{H}_\ell^\mathsf{T}. \tag{31b}$$

Proof. For convenience, partition the matrices $\boldsymbol{\Lambda}_k$ and $\mathbf{T}$ by rows into $\boldsymbol{\Lambda}_k^\mathsf{T} = [\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, ..., \boldsymbol{\lambda}_{n_x}]$, $\mathbf{T}^\mathsf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_{n_x}]$, respectively. After that, the minimization problem (28) is reduced to

$$\begin{aligned} \boldsymbol{\Lambda}_k^* = arg \min_{\boldsymbol{\lambda}_1,...,\boldsymbol{\lambda}_{n_x}} \sum_{i=1}^{n_x} J_i, \\ \text{s.t. } (\boldsymbol{L}_m^*)^\mathsf{T} \mathbf{C}_\ell^\mathsf{T} \boldsymbol{\lambda}_i = (\boldsymbol{L}_m^*)^\mathsf{T} \bar{\mathbf{A}}_\ell^\mathsf{T} \mathbf{t}_i, \quad i = 1, 2, ..., n_x, \end{aligned} \tag{32}$$

where each J_i is given as

$$\begin{aligned} J_i &= \left(\mathbf{t}_i^\mathsf{T} \bar{\mathbf{A}}_\ell - \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{C}_\ell\right) \mathbf{R}_\sigma \left(\mathbf{t}_i^\mathsf{T} \bar{\mathbf{A}}_\ell - \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{C}_\ell\right)^\mathsf{T} \\ &\quad + \left(\mathbf{t}_i^\mathsf{T} \bar{\mathbf{E}}_\ell - \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{G}_\ell\right) \mathbf{Q}_V \left(\mathbf{t}_i^\mathsf{T} \bar{\mathbf{E}}_\ell - \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{G}_\ell\right)^\mathsf{T} \\ &\quad + \left(\mathbf{t}_i^\mathsf{T} \bar{\mathbf{G}}_\ell - \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{H}_\ell\right) \mathbf{R}_\Delta \left(\mathbf{t}_i^\mathsf{T} \bar{\mathbf{G}}_\ell - \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{H}_\ell\right)^\mathsf{T} \\ &= \boldsymbol{\lambda}_i^\mathsf{T} \mathbf{\Omega} \boldsymbol{\lambda}_i - 2\mathbf{t}_i^\mathsf{T} \mathbf{\Pi} \boldsymbol{\lambda}_i + \mathbf{t}_i^\mathsf{T} \mathbf{\Phi} \mathbf{t}_i, \end{aligned} \tag{33}$$

where $\mathbf{\Phi} = \bar{\mathbf{A}}_\ell \mathbf{R}_\sigma \bar{\mathbf{A}}_\ell^\mathsf{T} + \bar{\mathbf{E}}_\ell \mathbf{Q}_V \bar{\mathbf{E}}_\ell^\mathsf{T} + \bar{\mathbf{G}}_\ell \mathbf{R}_\Delta \bar{\mathbf{G}}_\ell^\mathsf{T}$. It implies that the i-th sub-objective function J_i only depends on $\boldsymbol{h}_i$ and is only subject to the i-th constraint $(\boldsymbol{L}_m^*)^\mathsf{T} \mathbf{C}_\ell^\mathsf{T} \boldsymbol{\lambda}_i = (\boldsymbol{L}_m^*)^\mathsf{T} \bar{\mathbf{A}}_\ell^\mathsf{T} \mathbf{t}_i$. This simplifies the minimization problem (32) to n_x independent minimization problems:

$$\boldsymbol{\lambda}_i^* = arg \min_{\boldsymbol{\lambda}_i} J_i, \quad \text{s.t. } (\boldsymbol{L}_m^*)^\mathsf{T} \mathbf{C}_\ell^\mathsf{T} \boldsymbol{\lambda}_i = (\boldsymbol{L}_m^*)^\mathsf{T} \bar{\mathbf{A}}_\ell^\mathsf{T} \mathbf{t}_i. \tag{34}$$

It can be found that (34) belongs to the equality constrained convex quadratic programming problem. By using a Lagrange multiplier $\boldsymbol{\theta}_i \in \mathbb{R}^{n_x}$ and searching for the extremum of the Lagrangian, the solution to (34) is given by the following linear equation

$$\begin{bmatrix} \mathbf{\Omega} & \mathbf{C}_\ell \boldsymbol{L}_m^* \\ (\boldsymbol{L}_m^*)^\mathsf{T} \mathbf{C}_\ell^\mathsf{T} & \mathbf{0}_{n_x \times n_x} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_i \\ \boldsymbol{\theta}_i \end{bmatrix} = \begin{bmatrix} \mathbf{\Pi}^\mathsf{T} \mathbf{t}_i \\ (\boldsymbol{L}_m^*)^\mathsf{T} \bar{\mathbf{A}}_\ell^\mathsf{T} \mathbf{t}_i \end{bmatrix}. \tag{35}$$

All of the above linear equations for $i = 1, 2, ..., d_x$ share the same coefficient matrix. Hence, the optimal gain $\boldsymbol{\Lambda}^*$ is a value of $\boldsymbol{\Lambda}$ that satisfies

$$\begin{bmatrix} \mathbf{\Omega} & \mathbf{C}_\ell \boldsymbol{L}_m^* \\ (\boldsymbol{L}_m^*)^\mathsf{T} \mathbf{C}_\ell^\mathsf{T} & \mathbf{0}_{n_x \times n_x} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}^\mathsf{T} \\ \mathbf{\Theta} \end{bmatrix} = \begin{bmatrix} (\mathbf{T}\mathbf{\Pi})^\mathsf{T} \\ (\mathbf{T}\bar{\mathbf{A}}_\ell \boldsymbol{L}_m^*)^\mathsf{T} \end{bmatrix}, \tag{36}$$

where $\mathbf{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{d_x}]$. By solving the above equation, the result of the optimal $\boldsymbol{\Lambda}_k$ is shown in (30a). After that, $\boldsymbol{\mu}_k^*$ in (30b) is obtained to have $\hat{\boldsymbol{x}}_k^*$ unbiased. □

Remark 4. If system (1) is linear, the lifted state $\boldsymbol{z}_k = \boldsymbol{x}_k$, the system matrices $\mathbf{G} = \mathbf{H} = \mathbf{0}$, $\mathbf{T} = \mathbf{I}$, the linear transformation matrices $\mathbf{L} = \boldsymbol{L}_m^t = \boldsymbol{L}_m^* = \mathbf{I}$, and the errors $\boldsymbol{\delta}_k = \boldsymbol{\varepsilon}_k = \boldsymbol{\sigma}_m = \mathbf{0}$. In that case, the proposed filter will be reduced to the MVU FIR filter [3].

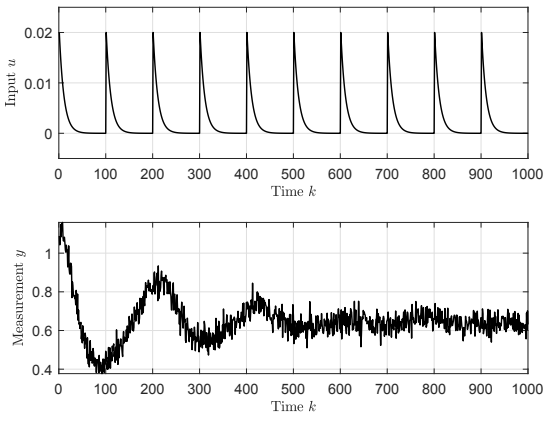

Fig. 1. Input and measurement sequences (Normal case).

V. SIMULATION

A. System description

Consider a discrete-time nonlinear prey-predator model that is adopted from [20]. The dynamic equation and measurement function for this system are given by

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} x_{1,k} + \mathrm{T}_s\big(\mathrm{a}x_{1,k} - \mathrm{b}x_{1,k}^2 - \mathrm{c}x_{1,k}x_{2,k} + v_{1,k}\big) \\ x_{2,k} + \mathrm{T}_s\big(\mathrm{d}x_{2,k} + \mathrm{e}x_{1,k}x_{2,k} + u_k + v_{2,k}\big) \end{bmatrix}, \tag{37a}$$

$$\boldsymbol{y}_k = \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{x}_k + v_{3,k}. \tag{37b}$$

In this system, the two states, $x_{1,k}$ and $x_{2,k}$, represent the populations of prey and predators, respectively. An input, u_k, is included to modify the evolution of the predator population. The system parameters are set to $a = 0.25$, $b = 0.2$, $c = 0.95$, $d = 0.55$, and $e = 1.1$, while the sampling interval is $\mathrm{T}_s = 0.1$. The system noise $\boldsymbol{v}_k$ follows a Gaussian distribution $\boldsymbol{v}_k \sim \mathcal{N}\left(\mathbf{0}, \mathrm{diag}(0.01^2, 0.01^2, 0.04^2)\right)$. In the simulation, the true state trajectory starts from the initial value $\boldsymbol{x}_0 = [0.83, 0.28]^\mathsf{T}$ and lasts more than 1000 samples. The input signal is designed as $u_k = 0.02e^{-\frac{\mathrm{mod}(k,100)}{10}}$, where $\mathrm{mod}(\cdot)$ is the modulo operation. The designed input signal and the resulting measurement sequences are shown in Fig.1.

B. Linear model based on the Koopman operator

To obtain the linear model (6), we generate 300 state trajectories, each with 1000 samples. The initial states and inputs are generated by uniform distributions within their bounds given by $x_{1,0} \in [0.2, 0.9]$, $x_{2,0} \in [0.05, 0.5]$, and $u_k \in [-0.02, 0.02]$. The training data set is shown in Fig.2, each gray line is a state trajectory. The basis functions are selected as $\boldsymbol{\psi}(\boldsymbol{x}) = \big[x_1^2,\ x_1x_2,\ x_2^2,\ x_1^3,\ x_1^2x_2,\ x_1x_2^2,\ x_2^3,\ x_1^4,\ x_1^3x_2,\ x_1^2x_2^2,\ x_1x_2^3,\ x_2^4,\ x_1^5,\ x_1^4x_2^1,\ x_1^3x_2^2,\ x_1^2x_2^3,\ x_1x_2^4,\ x_2^5\big]^\mathsf{T}$ so that the dimension of $\boldsymbol{z}_k$ is $N_x = 20$. After obtaining the linear model (6), another 200 state trajectories are used to test the model and get the covariance of the approximation error $\boldsymbol{\delta}_k$. Using the acquired linear model, the long-term prediction outcome is illustrated in Fig.2. The prediction trajectory

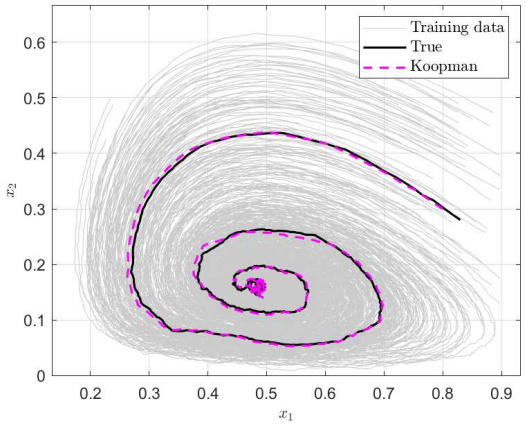

Fig. 2. Training data set and long-term prediction.

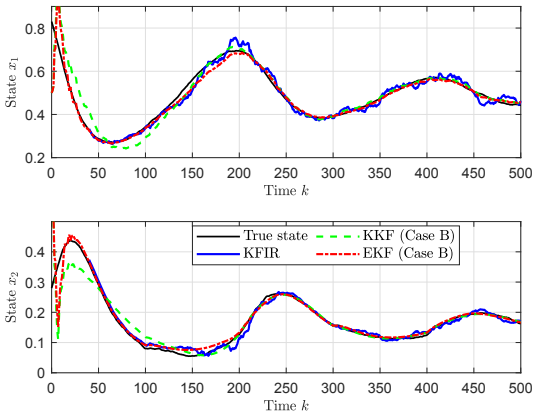

Fig. 3. Estimation results (Normal case).

originates from $[0.8, 0.3]^\mathsf{T}$ and is driven by the same inputs as the true trajectory. Evidently, the predicted trajectory derived from the Koopman operator (dashed magenta line) closely follows the actual state trajectory (black line).

C. Estimation results

By applying the proposed KFIR to the above system, the estimation results are shown in Fig.3. The horizon length of KFIR here is set to $\ell = 40$. In the initial state estimation, $\alpha = 0.1$ and $t_s = 1$ are used. As a comparison, the results based on the EKF [1] and the KKF [16] are also displayed. Both EKF and KKF are IIR filters, and they require a guess $\hat{\boldsymbol{x}}_0$ of the initial state $\boldsymbol{x}_0$. We consider three different scenarios to compare their performances:

- **Case A**: The initial guess is accurately equal to the true value, and the initial error covariance is set to zero, i.e., $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0$ and $\hat{\boldsymbol{P}}_0 = \mathbf{0}$.
- **Case B**: The initial guess is imprecise, but the initial error covariance effectively reflects the uncertainty in the guessed value. We set $\hat{\boldsymbol{x}}_0 = [0.5,\ 0.5]^\mathsf{T}$ and $\hat{\boldsymbol{P}}_0 = 0.1\mathbf{I}$.
- **Case C**: The initial guess is imprecise, and the initial error covariance cannot effectively reflect the uncertainty in the guessed value. We set $\hat{\boldsymbol{x}}_0 = [0.5,\ 0.5]^\mathsf{T}$ and $\hat{\boldsymbol{P}}_0 = \mathbf{I}$.

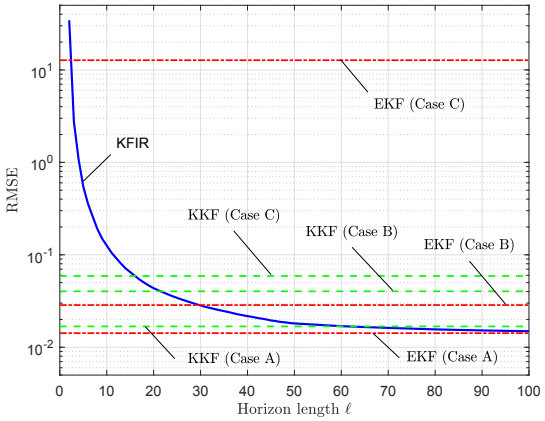

Fig. 4. Estimation performance comparison (Normal case).

Fig. 3 also shows the results of EKF and KKF under Case B. Both methods converge to the true state after some iterations. On average, computing one state estimate costs 1.2287×10^{-2} (s), 2.3349×10^{-5}(s), and 7.6491×10^{-3} (s) for KFIR, KKF, and EKF, respectively.

The complete comparison of their estimation accuracy is presented in Fig. 4. The root mean square error (RMSE) is used as the accuracy criterion. In the case of the KKF, the RMSE is computed using the formula $\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{K_{max}}-\ell}\sum_{k=\ell}^{\mathrm{K_{max}}} \|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\|^2}$, where $\mathrm{K_{max}}$ is the total simulation steps. For other filters, the RMSEs are calculated as $\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{K_{max}}}\sum_{k=1}^{\mathrm{K_{max}}} \|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\|^2}$. It is observed that EKF provides the best estimation results when the initial state is perfectly known (Case A). However, in the case where the initial error covariance is set to a bad one (Case C), EKF appears to be non-converging and generates the worst results. The performance of the proposed KFIR does not require an initial guess of the state, but it is dependent on the horizon length. As shown in Fig. 4, the RMSE of KFIR decreases with the horizon length, and if the horizon length is long enough, the estimation results of KFIR surpass those of KKF under any case and approach those of EKF with a perfect initial guess.

D. Robust test

To further evaluate the robustness of the KFIR filter, a fault is introduced into the system. Specifically, the system parameters are changed to $a = 0.15$, $b = 0.1$, $c = 1.15$, $d = 0.35$, $e = 1.3$, and $\mathrm{T}_s = 0.12$ between time steps $k \in [300, 500]$. Moreover, the noise covariance is assumed to be known inaccurate, i.e., $\mathbf{Q}_v$ in filter settings is known as $\mathrm{diag}(\gamma^2 0.01^2, \gamma^2 0.01^2, 1/\gamma^2 0.04^2)$, where $\gamma = 5$. The remaining settings are the same as in the previous subsection.

The estimation results produced by KFIR, KKF, and EKF are presented in Fig.5. It is observed that all filters have negligible errors in the first 300 samples. During the fault

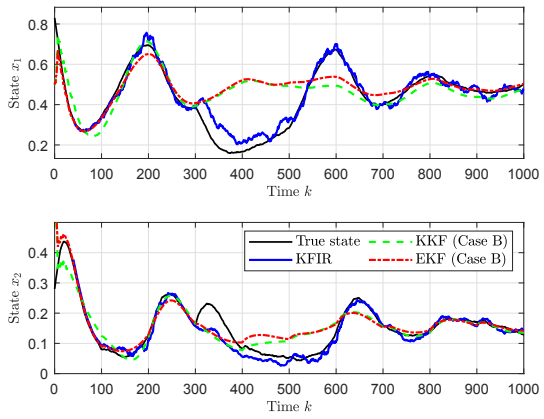

Fig. 5. Estimation results (Fault case).

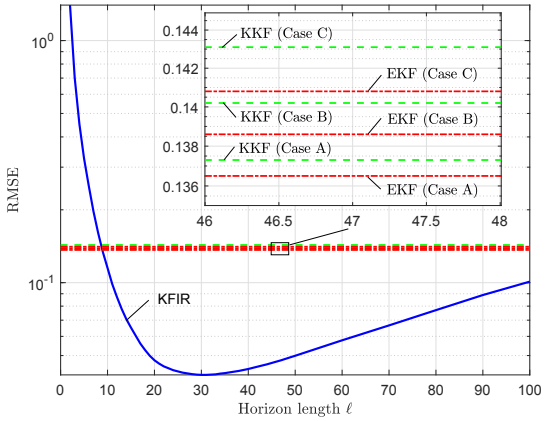

Fig. 6. Estimation performance comparison (Fault case).

period, KFIR provides the most reliable estimation results. Moreover, KFIR converges to the true state much faster than the other two filters after the system returns to normal.

We also conducted a comparison of the estimation performance of the KFIR filter with different horizon lengths, and the results are presented in Fig.6. It can be observed that the RMSE of KFIR is concave with respect to the horizon length and has a minimal value of $\ell = 31$. When $\ell > 31$, increasing ℓ reduces the estimation accuracy. This is because increasing the window length causes the FIR filter to approach an IIR one. Furthermore, Fig. 6 also shows the estimation results of EKF and KKF. In this case, both EKF and KKF perform similarly due to the fault and noise covariance mismatch that are the main factors affecting the estimation accuracy.

VI. CONCLUSIONS

In this paper, a novel Koopman operator-based FIR filter that minimizes the trace of the state estimation error covariance for nonlinear dynamic systems is proposed. This filter is a generalized version of the MVU FIR filter, originally designed for linear systems. Compared to IIR filters, the proposed filter does not require an initial state guess and converges to optimal estimates using any initial state value. Simulation results demonstrate that the proposed filter has better robustness against faults and inaccurate noise information than EKF and KKF. Some possible directions for future research include considering the approximation error as bounded noise to further improve robustness, as well as exploring more efficient iterative realization of the batch algorithm to reduce computational cost.

REFERENCES

[1] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[2] Y. S. Shmaliy, "Linear optimal fir estimation of discrete time-invariant state-space models," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3086–3096, 2010.

[3] S. Zhao, Y. S. Shmaliy, B. Huang, and F. Liu, "Minimum variance unbiased fir filter for discrete time-variant systems," *Automatica*, vol. 53, pp. 355–361, 2015.

[4] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, "Unbiased finite impluse response filtering: An iterative alternative to kalman filtering ignoring noise and initial conditions," *IEEE Control Systems Magazine*, vol. 37, no. 5, pp. 70–89, 2017.

[5] S. Zhao, Y. Ma, and B. Huang, "Robust fir state estimation of dynamic processes corrupted by outliers," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 139–147, 2018.

[6] Y. S. Shmaliy, "Suboptimal fir filtering of nonlinear models in additive white gaussian noise," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5519–5527, 2012.

[7] J. M. Pak, C. K. Ahn, M. T. Lim, and M. K. Song, "Horizon group shift fir filter: Alternative nonlinear filter using finite recent measurements," *Measurement*, vol. 57, pp. 33–45, 2014.

[8] 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.

[9] B. O. Koopman and J. v. Neumann, "Dynamical systems of continuous spectra," *Proceedings of the National Academy of Sciences*, vol. 18, no. 3, pp. 255–263, 1932.

[10] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern koopman theory for dynamical systems," *arXiv preprint arXiv:2102.12086*, 2021.

[11] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[12] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[13] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

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

[15] A. Mauroy and J. Goncalves, "Linear identification of nonlinear systems: A lifting technique based on the koopman operator," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6500–6505.

[16] A. Surana and A. Banaszuk, "Linear observer synthesis for nonlinear systems using koopman operator framework," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 716–723, 2016.

[17] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6492–6499.

[18] M. Netto and L. Mili, "Robust koopman operator-based kalman filter for power systems dynamic state estimation," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2018, pp. 1–5.

[19] Y. S. Shmaliy and O. Ibarra-Manzano, "Time-variant linear optimal finite impulse response estimator for discrete state-space models," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 2, pp. 95–104, 2012.

[20] C. Combastel, "An extended zonotopic and gaussian kalman filter (ezgkf) merging set-membership and stochastic paradigms: Toward non-linear filtering and fault detection," *Annual Reviews in Control*, vol. 42, pp. 232–243, 2016.