2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Data-Driven Koopman Operator-Based Error-State Kalman Filter for Enhanced State Estimation of Quadrotors in Agile Flight

Peng Huang[1], Ketong Zheng[2], and Gerhard Fettweis[1,2]

Abstract— **Highly dynamic maneuvers pose a challenge to conventional state estimators of quadrotors in rapidly tracking the pose. This paper proposes a data-driven Koopman operator-based error-state Kalman filter (K-ESKF) to enhance pose estimation in agile flight. Our method uses the Koopman operator theory to transform the full-state nonlinear quadrotor dynamics into a lifted bilinear control system driven by accelerations and angular rates. A deep neural network (DNN) is used to represent the Koopman observable functions. Our proposed K-ESKF extends the propagation step of a standard error-state Kalman filter (ESKF) using the lifted bilinear control system. An open-source quadrotor dataset, NeuroBEM, is used for training and evaluating the DNN and for testing the K-ESKF. The learned Koopman bilinear system demonstrates a 60% less attitude errors compared to the first-order Euler method in terms of model accuracy. Using real trajectories from the dataset, our proposed K-ESKF can estimate the pose as accurately as the ESKF during normal flight. More importantly, our proposed approach outperforms the ESKF by achieving about 50% less attitude and velocity estimation errors in a highly agile flight. During drastic attitude and velocity changes, the K-ESKF can still estimate the pose while the ESKF loses tracking.**

I. INTRODUCTION

Dealing with nonlinearity is a challenge in estimation and optimization within robotics. The Kalman filter [1] is an optimal recursive estimator that achieves the Cramér-Rao Lower Bound for linear time-invariant systems (LTIs) [2]. Several efforts have been made to deal with nonlinearity based on the dynamic model, such as the extended Kalman filter (EKF) [2], the unscented Kalman filter (UKF) [3], and the particle filter [4]. Koopman operator theory provides a unique approach to nonlinear systems, allowing data-driven modeling without the need for explicit system dynamics. It also simplifies the design of controllers and estimators by transforming the nonlinear system into a lifted linear system. The Koopman operator [5] extends the original state space with infinite observables, where nonlinear system evolves linearly in this infinite dimension space.

The challenge is to find a finite set of observables that well approximate the Koopman operator. Williams [6] proposed the extended dynamic mode decomposition (EDMD) method to approximate the Koopman operator using a finite set of predefined observables, such as Hermite polynomials, radial basis functions (RBFs), and discontinuous spectral elements. Deep learning has been shown to be effective in approximating the nonlinear mapping and can be used to identify a set of functions that span the space of observables [7]–[9]. Folkestad et al. [10] used deep neural networks (DNN) to directly approximate the linear transformation with predefined observables. In our previous work [11], we combined DNN and EDMD, where the DNN is used only to train observables and the EDMD is used to find the linear transition matrix.

Recently, there has been an increasing interest in applying the Koopman operator for estimation and control with control-affine systems, which can represent most of the dynamical systems, including autonomous ground vehicles and quadrotors. According to Koopman Canonical Transform (KCT), the nonlinear control affine system can be approximated by a lifted koopman bilinear system, which can be regulated effectively through e.g., optimal control [9], [11], [12] and model predictive control [13], [14].

Koopman operator theory has also gained attention for its applications in state estimation. Surana et al. [15] developed a Koopman operator-based approach for observer synthesis in discrete-time autonomous nonlinear systems. The approach computes the Koopman spectrum through a Galerkin-weighted residual approach, also known as a kernel-based EDMD. Performance was evaluated on a 2D system with a closed-form Koopman operator, which does not require the Galerkin-based approach. In pioneering research [16], Surana extended this approach to control-affine systems, proposed an observer design using the Koopman observer form (KOF) in continuous time and discussed the nonlinear observability criterion. While [15], [16] demonstrated the advantages of the KOF over the EKF for 2D systems with a lifted state space of dimension three, the simplicity of the examples does not show the full potential of the KOF in real applications. Netto and Miliani [17] proposed a generalized maximum likelihood Koopman operator-based Kalman filter (GM-KKF) to estimate the rotor angle and speed of synchronous generators. In particular, the GM-KKF exhibited faster convergence compared to Koopman operator-based Kalman filters (KKFs) using a batch-mode regression formulation. Syed et al. [18] presented a KKF for fault detection in superconducting high-frequency cavities and showed that the KKF can achieve a speedup of three orders of magnitude compared to the UKF while maintaining the same detection capabilities. A KKF was applied to distributed systems [19], and an extension was introduced to incorporate saturation in innovations to account

*The work is financed on the basis of the budget passed by the Saxon State Parliament.

[1]Barkhausen Institut, 01062 Dresden, Germany.
[2]Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, 01069 Dresden, Germany.
peng.huang@barkhauseninstitut.org
{ketong.zheng, gerhard.fettweis}@tu-dresden.de

for measurement outliers [20]. Another application can be found in the estimation of the wrench acting on a 3D elastic rod [21]. Overall, the KKF shows better performance than the conventional estimator such as EKF in terms of robustness, accuracy, and convergence.

However, the systems in the above work [15], [17]–[20] do not have an input term, which makes the computation of the Koopman operator rather simple and limits its application in robotics. Furthermore, the choice of lifting functions/observables has an impact on the approximation of the Koopman operator with a nonlinear system. Most work has chosen a kernel-based approach that acquires the Koopman tuple decomposition (or EDMD) [15], [16], [19]–[21]. Depending on the application, a predefined set of observables can be used [17], [18]. Another issue to consider is that the computational complexity of the KKF increases exponentially with the increasing dimension of the observables.

The paper proposes a data-driven Koopman operator-based error-state Kalman filter (K-ESKF) for accurate and robust pose and inertial measurement unit (IMU) bias estimation in quadrotor systems. Instead of using a kernel-based approach or predefined observables, we present a DNN-based Koopman bilinear modeling approach to accurately model the full-state dynamics of quadrotors. The DNN represents the parameterized observable functions with input of quaternions. An open-source drone dataset, NeuroBEM [22], has been used for the training and evaluation of the DNN and test of the K-ESKF. Our lifted bilinear model shows 60% less attitude errors than the first-order Euler method and can improve velocity accuracy with higher DNN output dimensions. Our main contribution is to extend the propagation update of the state variables in the error-state Kalman filter with the Koopman bilinear control system. This extension can ease the computational burden introduced by the high dimension of the lifted state in the full realization of the Kalman filter. Our proposed K-ESKF can estimate the pose and IMU biases not only in normal operation, but also in highly agile motions where the standard ESKF suffers from loss of tracking.

The remainder of this paper is organized as follows: Section II presents preliminaries. In Section III, we detail the application of the K-ESKF to a quadrotor. The focus then shifts to Section IV, where we discuss our experimental results. Finally, Section V concludes our work.

Notation: Scalars are denoted by non-bold x. Vectors and matrices are set in bold font $\boldsymbol{x}$ and $\boldsymbol{X}$. Let $[\cdot]^T$ denote transpose. The subscript $k \in \mathbb{N}_+$ denotes a discrete time instant. $\mathcal{H}$ represents the Hilbert space. Measurements are denoted by a tilde, $\tilde{\boldsymbol{y}}$, and estimates are denoted by a hat $\hat{\boldsymbol{x}}$. $\|\cdot\|_F$ denotes the Frobenius norm. The left subscripts ${}_W\boldsymbol{x}$ and ${}_B\boldsymbol{x}$ denote vector $\boldsymbol{x}$ expressed in the world and the body coordinates, respectively. Unit quaternions $\boldsymbol{q} = [q_w, q_x, q_y, q_z]^T$ are used to represent the attitude, e.g., $\boldsymbol{q}_{WB}$ is the attitude of the quadrotor body with respect to the world frame. We use $\otimes$ to denote quaternion multiplication. Let $[\boldsymbol{x}_k]$ define a batch of data, where $k = 1, \cdots, N$.

II. Preliminary

A. The Koopman Operator Review

We consider the dynamics of a discrete-time nonlinear autonomous system:

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

with $\boldsymbol{x}_k \in X \subset \mathbb{R}^n$, where X is the state space.

The preliminaries of the Koopman theory can be found in [6], [23], [24]. The Koopman operator $\mathcal{K}$ acts on observable functions $g_i(\boldsymbol{x})$ and evolves them over time. Mathematically, for the vector-valued observable $\boldsymbol{g}(\boldsymbol{x}_k) = \begin{bmatrix} g_1(\boldsymbol{x}_k) & \cdots & g_p(\boldsymbol{x}_k) \end{bmatrix}^T \in \mathcal{G} \subset \mathcal{H} : \mathbb{R}^n \mapsto \mathbb{C}^p$, where $\mathcal{G}$ is the observable space, the Koopman operator transforms $\boldsymbol{g}$ as follows:

$$\mathcal{K}\boldsymbol{g}(\boldsymbol{x}_k) = \boldsymbol{g} \circ \boldsymbol{f}(\boldsymbol{x}_k) = \boldsymbol{g}(\boldsymbol{x}_{k+1}), \tag{2}$$

with $\circ$ denoting the composition operator. It is noted that the dimension p should be infinite in theory $p \to \infty$, however, in practice, p will be finite. Assuming that the observable function g_i is spanned by a set of Koopman eigenfunctions $\{\varphi_i : \mathbb{R}^n \mapsto \mathbb{C}\}_{i=1}^p$. Therefore, $\boldsymbol{g}$ can be decomposed into the eigenfunctions [6]

$$\boldsymbol{g}(\boldsymbol{x}_k) = \boldsymbol{V}\boldsymbol{\varphi}(\boldsymbol{x}_k), \tag{3}$$

where $\boldsymbol{\varphi}(\boldsymbol{x}_k) = \begin{bmatrix} \varphi_1(\boldsymbol{x_k}) & \cdots & \varphi_p(\boldsymbol{x_k}) \end{bmatrix}^T$ and $\boldsymbol{V}$ denotes the matrix of Koopman modes, where the ith column of $\boldsymbol{V}$ describes the evolution of the ith Koopman eigenfunction.

The Koopman operator maps the functions of the state space into the eigenfunctions, yielding:

$$[\mathcal{K}\boldsymbol{\varphi}](\boldsymbol{x}_k) = \boldsymbol{\Lambda}\boldsymbol{\varphi}(\boldsymbol{x}_k) \tag{4}$$

where the Koopman eigenvalues lies in the diagonal of the matrix $\boldsymbol{\Lambda}$. Therefore, within the space of the observable functions, the original nonlinear dynamics evolve linearly as follows:

$$\boldsymbol{g}(\boldsymbol{x}_{k+1}) = [\mathcal{K}\boldsymbol{g}](\boldsymbol{x}_k) = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{\varphi}(\boldsymbol{x}_k) = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}\boldsymbol{g}(\boldsymbol{x}_k). \tag{5}$$

In this work, the observables $\boldsymbol{g}(\boldsymbol{x})$ are represented by the parameterized functions learned in a DNN. The learned observables together with the original state form the lifted state space. Then it becomes an optimization problem to find an optimal solution of $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$ as the lifted transition matrix.

B. Koopman Lifted Bilinear Modeling

Next we extend the Koopman linear model with control input and consider an input-affine nonlinear autonomous system [16]:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \sum_{j=1}^{m} \boldsymbol{b}_j(\boldsymbol{x}_k) u_{j,k}, \tag{6}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k), \tag{7}$$

where the input functions $\boldsymbol{b}_j$ are the j-th nonlinear vector-valued terms depending on the state. The input vector is given by $\boldsymbol{u}_k = \begin{bmatrix} u_{1,k} & \cdots & u_{m,k} \end{bmatrix}^T \in \mathcal{U} \subset \mathbb{R}^m$ at time

instance k. Here $\boldsymbol{y}_k$ is the measurement at time k, and $\boldsymbol{h}$ is nonlinear measurement function.

Defining the Koopman lifted state space by $\boldsymbol{z}_k = \boldsymbol{g}(\boldsymbol{x}_k)$ and assuming that $\boldsymbol{b}_j$ lies inside the lifted state space, we transform the system (6) into a bilinear control form [16]

$$\boldsymbol{z}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k + \sum_{j=1}^{m} \boldsymbol{B}_{u_j} \boldsymbol{z}_k u_{j,k}, \tag{8}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{z}_k, \tag{9}$$

where $\boldsymbol{A} \in \mathbb{R}^{p\times p}$ is a matrix that captures the linear part of the dynamics, and $\boldsymbol{B}_{u_j} \in \mathbb{R}^{p\times p}$ are matrices related to the control terms u_j. If the measurement equation (7) is nonlinear with respect to the state, we can apply the Koopman operator to $\boldsymbol{h}$ using KCT, resulting in a constant lifted measurement matrix $\boldsymbol{C}$. In practice, it is not possible to have infinite dimensional observables. Therefore, an adequate set of observables will have an impact on the accuracy of the Koopman bilinear model. Note that the elements in the matrices $\boldsymbol{A}$, $\boldsymbol{B}_{u_j}$, and the observables $\boldsymbol{g}(\boldsymbol{x})$ are assumed to be in the space of $\mathbb{R}$. Theoretically, they can be complex [6] and for dealing with their complex elements, see [6], [15].

C. Quadrotor Model Dynamics

The kinematics of a quadrotor describes the position $\boldsymbol{p}_W$, the velocity $\boldsymbol{v}_W$ and the attitude $\boldsymbol{q}_{WB}$ of a rigid body in three-dimensional space. The system is driven by accelerometer and gyroscope measurements, which provide information about acceleration and angular velocity within the body frame. The state is represented by

$$\boldsymbol{x} = \begin{bmatrix} {}_W\boldsymbol{p} & {}_W\boldsymbol{v} & \boldsymbol{q}_{WB} & \boldsymbol{b}_a & \boldsymbol{b}_g \end{bmatrix}^T. \tag{10}$$

The input is given by

$$\boldsymbol{u} = \begin{bmatrix} {}_B\tilde{\boldsymbol{a}} & {}_B\tilde{\boldsymbol{\omega}} \end{bmatrix}^T. \tag{11}$$

The IMU measurements are subject to corruption by both white noise ($\boldsymbol{w}_a$, $\boldsymbol{w}_g$) and bias ($\boldsymbol{b}_a$, $\boldsymbol{b}_g$)

$${}_B\tilde{\boldsymbol{a}} = {}_B\boldsymbol{a} + \boldsymbol{b}_a + \boldsymbol{w}_a, \tag{12}$$

$${}_B\tilde{\boldsymbol{\omega}} = {}_B\boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{w}_g. \tag{13}$$

The acceleration and angular velocity biases are modeled as random walk processes with their derivatives represented by white Gaussian processes $\boldsymbol{w}_{b_a}$ and $\boldsymbol{w}_{b_g}$. The dynamics of the quadrotor and IMU biases are described as follows [22], [25], [26]

$$\begin{aligned} \dot{\boldsymbol{q}}_{WB} &= \boldsymbol{q}_{WB} \otimes \begin{bmatrix} 0 \\ {}_B\boldsymbol{\omega}/2 \end{bmatrix}, \\ {}_W\dot{\boldsymbol{v}} &= \boldsymbol{R}(\boldsymbol{q}_{WB})(\boldsymbol{f}_{prop} + \boldsymbol{f}_{ext}) + {}_W\boldsymbol{g}, \\ {}_W\dot{\boldsymbol{p}} &= {}_W\boldsymbol{v}, \quad \dot{\boldsymbol{b}}_a = \boldsymbol{w}_{b_a}, \quad \dot{\boldsymbol{b}}_g = \boldsymbol{w}_{b_g}, \end{aligned} \tag{14}$$

where $\boldsymbol{R}(\boldsymbol{q}_{WB})$ represents the rotation matrix of the body frame with respect to the world frame, ${}_W\boldsymbol{g} = [0, 0, -9.81\,\mathrm{m/s^2}]^T$ is the gravitational acceleration in the world frame, $\boldsymbol{f}_{prop}$ is the collective specific force generated by the propellers, $\boldsymbol{f}_{ext}$ is the external specific force including aerodynamic drag and other external force. The purpose of this paper is to estimate the attitude of the quadrotor rather than to estimate the forces. Here we assume that the acceleration measurement after removing the bias is equal to the acceleration in the body frame, i.e. ${}_B\boldsymbol{a} = {}_B\tilde{\boldsymbol{a}} - \boldsymbol{b}_a = \boldsymbol{f}_{prop} + \boldsymbol{f}_{ext}$. And the gravity is constant since we consider indoor quadrotor applications in a rather small space.

III. Methodology

A. Data-Driven Bilinearization of Quadrotor Dynamics

The nonlinearity in (14) only exists in the velocity and quaternion dynamics. The quaternion dynamics depends on the quaternion and the input of angular velocities, and it fulfills the input-affine form (6). This property allows the application of the Koopman operator to transfer the quaternion dynamics into the bilinear control form as (8) driven by the angular velocity. In addition, the nonlinearity in the velocity equation is also caused by the quaternion in terms of the rotation matrix $\boldsymbol{R}(\boldsymbol{q}_{WB})$. Therefore, the Koopman observable functions can only depend on the quaternion.

To efficiently explore the meansurement data and learn proper koopman observation function, we present a fully connected deep neural network as parameterized functions being Koopman observables, as shown in Fig. 1. The input of the network is a batch of quaternions $[\boldsymbol{q}_k], k = 1, ..., N$, where the subscript is omitted for simplicity. The parameterized function is denoted by $\boldsymbol{\phi}_k = \boldsymbol{\phi}(\boldsymbol{q}_k) \in \mathbb{R}^{p-4}$. We concatenate the original state and the network output $\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{q}_k & \boldsymbol{\phi}_k \end{bmatrix}^T \in \mathbb{R}^p$, which allows the reconstruction of quaternions without a decoding layer. The quaternion bilinear form is formulated as

$$\hat{\boldsymbol{z}}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k + \textstyle\sum_{j=1}^{3} \boldsymbol{B}_{\omega_j} \boldsymbol{z}_k \omega_{j,k}, \tag{15}$$

$$\boldsymbol{A}, \boldsymbol{B}_{\omega_j} = \underset{\boldsymbol{A},\ \boldsymbol{B}_{\omega_j}}{\operatorname{argmin}} \ \textstyle\sum_{k=1}^{N} \left\| \boldsymbol{z}_{k+1} - \hat{\boldsymbol{z}}_{k+1} \right\|_F. \tag{16}$$

The solution can be obtained in a straightforward manner using least squares fitting. The bilinear matrices $\boldsymbol{A}$ and $\boldsymbol{B}_{\omega_j}$ are updated after each iteration of the DNN. In a similar manner, the velocity dynamics is formulated as

$${}_W\hat{\boldsymbol{v}}_{k+1} = {}_W\boldsymbol{v}_k + \textstyle\sum_{j=1}^{3} \boldsymbol{B}_{a_j} \boldsymbol{z}_{kB} a_{j,k} + {}_W\boldsymbol{g}\Delta t, \tag{17}$$

$$\boldsymbol{B}_{a_j} = \underset{\boldsymbol{B}_{a,j}}{\operatorname{argmin}} \ \textstyle\sum_{k=1}^{N} \left\| {}_W\boldsymbol{v}_{k+1} - {}_W\hat{\boldsymbol{v}}_{k+1} \right\|_F. \tag{18}$$

The loss function is defined as follows:

$$\begin{aligned} \mathcal{L} = \frac{1}{N} \textstyle\sum_{k=1}^{N} (&|e_{\boldsymbol{q}_{k+1}}| + \|\boldsymbol{\phi}_{k+1}(\boldsymbol{q}_{k+1}) - \hat{\boldsymbol{\phi}}_{k+1}(\hat{\boldsymbol{q}}_{k+1})\|_F \\ &+ \|{}_W\boldsymbol{v}_{k+1} - {}_W\hat{\boldsymbol{v}}_{k+1}\|_F) \end{aligned} \tag{19}$$

Note that the quaternion error is characterized by a smoother measure of quaternion distance (adapted from [27])

$$e_{\boldsymbol{q}_k} = 1 - (\boldsymbol{q}_k \cdot \hat{\boldsymbol{q}}_k)^2 \in (0, 1). \tag{20}$$

B. Koopman Operator-based ESKF

Before presenting the K-ESKF, let us briefly revisit the ESKF. The system state $\boldsymbol{x} = \bar{\boldsymbol{x}} + \delta\boldsymbol{x}$ compromises a nominal state $\bar{\boldsymbol{x}}$, and an error-state $\delta\boldsymbol{x}$.

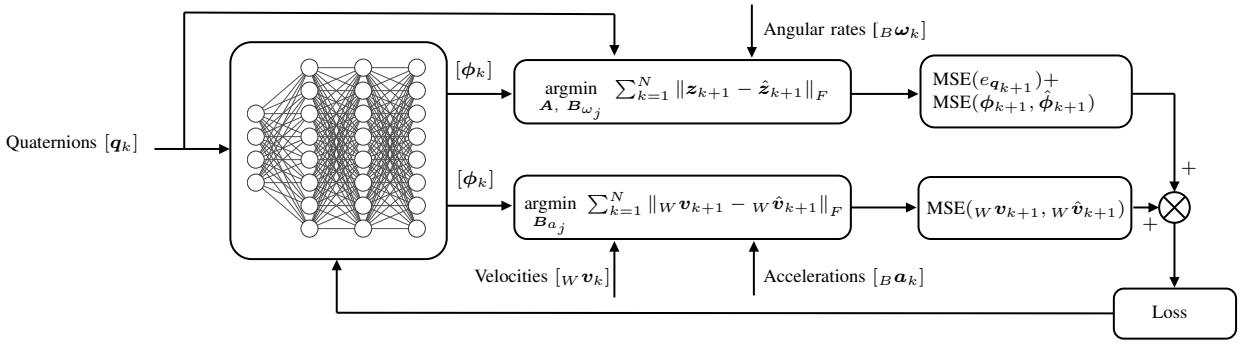


Fig. 1: Simultaneous bilinearization of quadrotor dynamics and learning of the observables of quaternions using a DNN.

The benefit of this error-state formulation is that the error-state is small-signal and is more closely approximated to be linear [25], [28]. The dimension of the system (14) is reduced by 1 and becomes 15 due to the fact that the error-state of the quaternion is the local perturbation $\delta\boldsymbol{\theta} \in \mathbb{R}^3$:

$$\begin{aligned} \dot{\delta\boldsymbol{\theta}} &= -[{}_B\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_g]_\times \delta\boldsymbol{\theta} - \delta\boldsymbol{b}_g - \boldsymbol{w}_g, \\ \dot{\delta\boldsymbol{v}} &= -\boldsymbol{R}_{WB}([{}_B\tilde{\boldsymbol{a}} - \boldsymbol{b}_a]_\times \delta\boldsymbol{\theta} + \delta\boldsymbol{b}_a + \boldsymbol{w}_a), \\ \dot{\delta\boldsymbol{p}} &= \delta\boldsymbol{v}, \quad \dot{\delta\boldsymbol{b}}_a = \boldsymbol{w}_{b_a}, \quad \dot{\delta\boldsymbol{b}}_g = \boldsymbol{w}_{b_g}. \end{aligned} \tag{21}$$

The measurement update uses the position residual, which is a linear measurement model. Due to space constraints, we will not introduce the complete ESKF formulation. For details, refer to [25], [29].

We have represented the observables of quaternions using the parameterized functions defined by the DNN and bilinearized the quaternion and velocity dynamics equations in the previous section. In the K-ESKF, we extend the propagation of the quaternion and velocity using the Koopman bilinear control form (15) and (17). The quaternion can be retrieved by $\boldsymbol{q}_k = [\boldsymbol{I}_{4\times4} \mid \boldsymbol{0}_{4\times(p-4)}]\boldsymbol{z}_k$. The other computations remain the same as in a standard ESKF.

Remark 1: The difference between the standard ESKF and our K-ESKF is the propagation equations for the quaternion and velocity. The standard ESKF usually uses the first-order Euler method. Our approach is a minimal realization of the Koopman operator-based model on the Kalman filtering. A full realization is to apply a linear Kalman filter on the bilinear model in the both propagation and measurement update, as well as the error-state covariance. However, it poses challenges on the computational efficiency due to the increased lifted state space.

Remark 2: Using the full realization, the resulting Kalman filter is still time-varying, depending on the nominal state and the input. An ideal case is that the lifted system is LTI and the input and measurement noise are stationary. In this case, the limiting solution of the Kalman filter can be solved in advance via the Ricatti equation, giving a computational complexity of $\mathcal{O}(p^2)$, depending on lifted state dimension. As a reference, the computing complexity of an EKF is $\mathcal{O}(n^3)$ [30], depending on the original state dimension. Therefore, the computation of a full realization KKF is more complex than that of an EKF due to $p \gg n$, for example, in our application, we select $p = 32$. However, in our approach, the core computation of the Kalman filter, i.e., the Kalman gain, keeps the same complexity as the standard ESKF.

Remark 3: Another reason for choosing the minimal realization is that our focus is on how well the bilinearization of the quadrotor dynamics can be approximated via the Koopman operator in highly dynamic scenarios, which pose challenges to the traditional propagation method.

IV. Experimental Results

This section evaluates the performance of the K-ESKF on a quadrotor using an open-source dataset. Our investigation focuses on three key aspects: 1. How is the model prediction accuracy of the Koopman bilinear control system compared to conventional integration using the first-order Euler method? 2. What is the relationship between the dimension of the observables and the model accuracy? 3. How does the K-ESKF perform compare to the ESKF for quadrotor pose and IMU bias estimation under different maneuvers?

A. Quadrotor Dataset and DNN Setup

We use the NeuroBEM [22] from the University of Zurich, an open-source quadrotor dataset. This dataset is chosen for its comprehensive set of flight maneuvers and diverse scenarios, including 2D/3D circles, lemniscate patterns, linear oscillations, melon-like, and random point flight maneuvers. These maneuvers cover a wide range of maneuvers from low speed to highly agile and high speed trajectories. The dataset provides detailed information on the poses captured with high precision by a motion capture system and includes onboard IMU measurements. Out of the 1.8 million data points available, we allocate 50% for training and 10% for validation. Additionally, two flights are reserved for showcasing in Section IV-C: wobbly circle and melon-shaped trajectories.

For the training of our neural network and subsequent bilinearization, we use the ground truth for quaternions and velocities, and processed accelerations and angular rates at a sampling rate of 400 Hz. The architecture of the DNN involves three hidden layers, each comprising units to match the output dimension. The output layer is designed to define parameterized functions as observable functions. The dimension of the output layer is an open question, which will be investigated in Section IV-B. The training process involves batches of 50,000 data points. We use the Adam optimizer with a learning rate of 0.001. The training spans 500 epochs.

The hyperbolic tangent function ($\tanh$) is chosen as the activation function.

B. Model Accuracy vs Observable Dimension

The model accuracy of attitude and velocity is evaluated by comparing our deep Koopman bilinear modeling with results from the first-order Euler integration results from multiple-step prediction. For the experiments, we change the dimension of the output layer of the DNN from 4 to 36 in increments of 4.

To evaluate the attitude accuracy, the attitude root mean square errors (RMSEs) are computed in Fig. 2a. The attitude RMSEs are 60% less than those derived from the first-order integration approach and are consistently stable, about 4 degrees, across different dimensions of the parameterized functions. Figure 2b shows that the velocity accuracy depends on the dimension of the DNN output. The velocity accuracy improves with higher dimensions and converges after a dimension of 24. As a benchmark, the velocity obtained using the true attitude provides the best results. More realistic results are given by the velocities computed using the attitudes by the integration method, which, however, yield a velocity error of 14 m/s.

The accuracy of the attitude model can be explained by the fact that the quaternion dynamics involves the multiplication between the quaternion and the angular rate, which is inherently close to bilinear model. The combination of the quaternion and the angular rate has been proven to be effective in approximating the quaternion dynamics [12]. The difference is that the input is included in the lifted state space, whereas our observable functions do not contain the angular velocity. Conversely, the accuracy of the velocity dynamics, which is influenced by the coordinate transformation of the body acceleration to the world frame, benefits from more observables. More observable functions on the quaternion enable a more accurate approximation of the rotation matrix.

Therefore, the selection of the DNN output dimension depends on the requirement of the velocity accuracy. From Fig. 2b, a dimension greater than 10 is recommended for this application and dataset. Considering the convergence of velocity accuracy after dimension 24, a reasonable choice is 28. This not only achieves optimal model accuracy but also keeps the DNN as compact as possible. In summary, the results confirm the effectiveness of our proposed approach and demonstrate stable and accurate prediction, especially in the quaternion dynamics.

C. Estimation Results

This subsection evaluates our proposed K-ESKF using flight measurements from the NeuroBEM dataset, and compares it with the standard ESKF.

1) Prerequisites for the K-ESKF and the standard ESKF: The initial conditions for the error-state include a standard deviation (STD) of $0.2\,\mathrm{m}$ for position, $1\,\mathrm{m/s}$ for velocity, $8\,\mathrm{deg}$ for attitude (converted to quaternion), $1\,\mathrm{m/s^2}$ for acceleration bias, and $0.1\,\mathrm{rad/s}$ for angular rate bias. The position measurement noise is set at $0.2\,\mathrm{m}$. Although the

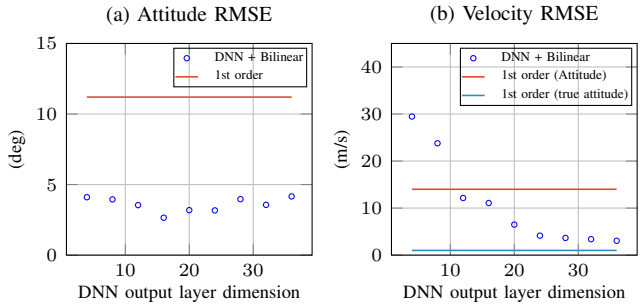


Fig. 2: Attitude and velocity model accuracy with different DNN output layer dimension. Comparison of RMSEs between the bilinear model and the 1st-order integration.

NeuroBEM does not provide position measurements from any sensors, we use the ground truth with additive white Gaussian noise which is used in the measurement update. In practice, indoor localization approaches such as vision- or lidar-based approaches can be used on quadrotors.

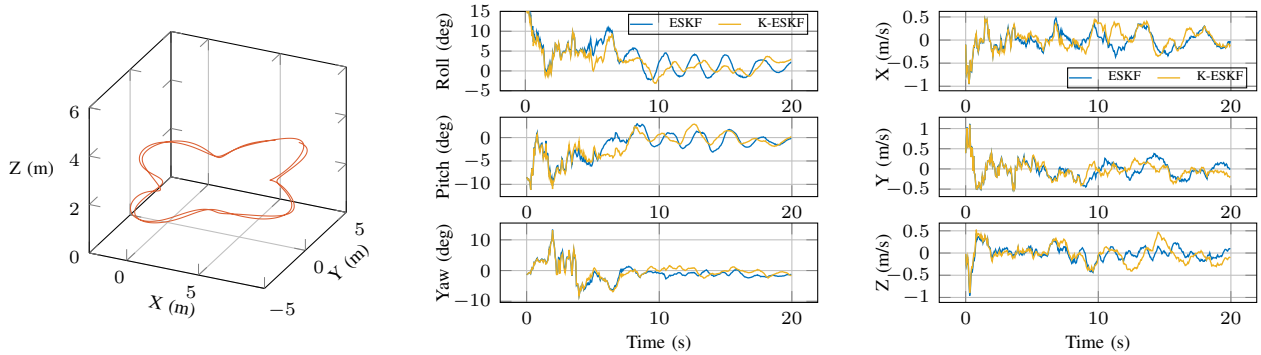
The noise densities for the acceleration and angular rate are $1 \times 10^{-1}\,\mathrm{m/(s^2\sqrt{Hz})}$ and $1 \times 10^{-3}\,\mathrm{rad/(s\sqrt{Hz})}$, respectively. The noise densities for acceleration and angular rate biases are $4 \times 10^{-5}\,\mathrm{m/(s^3\sqrt{Hz})}$ and $4 \times 10^{-8}\,\mathrm{rad/(s^2\sqrt{Hz})}$, respectively. The initialization of position, velocity, and attitude uses the ground truth with additive Gaussian noise of STD of $2\,\mathrm{m}$, $0.5\,\mathrm{m/s}$, and $8\,\mathrm{deg}$, while the initial IMU biases are zero. We manually add an acceleration bias of $0.2\,\mathrm{m/s^2}$ and an angular rate bias of $0.05\,\mathrm{rad/s}$ to the IMU data. The prediction step runs at $400\,\mathrm{Hz}$, while the measurement update operates at $100\,\mathrm{Hz}$. According to the findings from Sec. IV-B, we have chosen 28 as the DNN output dimension.

Note that the standard ESKF uses the first-order integration method. Although the Runge-Kutta method (RK4) could be used, but the accuracy improvement is negligible due to the high prediction sampling rate of $400\,\mathrm{Hz}$.

2) Comparison of Estimation Results: We have chosen two different trajectories from the NeuroBEM dataset: wobbly circle and melon shape trajectories.

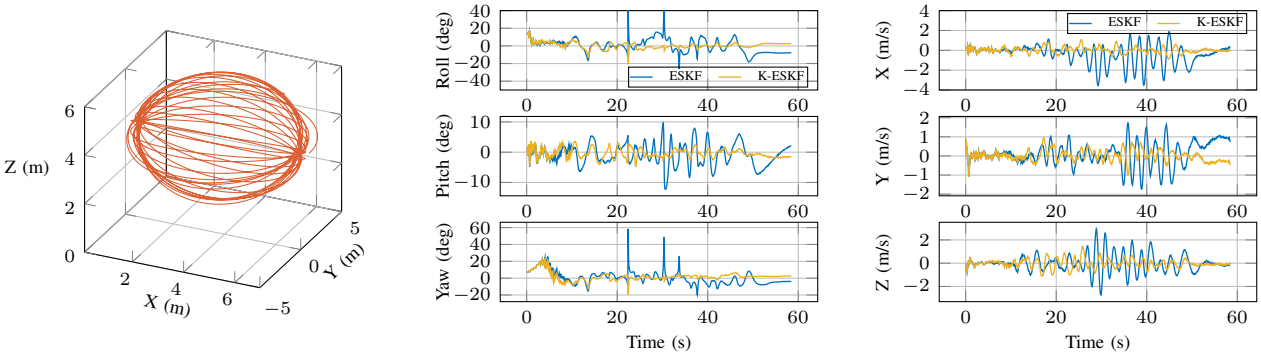
Trajectory 1: Wobbly circle. Figure 3a presents the ground truth of the wobbly-circle trajectory. Both the K-ESKF and the standard ESKF are able to estimate the attitude with equivalent accuracy as shown in Fig. 3b. The velocity estimates from both approaches have the same level of accuracy. In this trajectory, the operations of the angular rate and the acceleration are in the normal range such that both methods can deliver the same performance.

Trajectory 2: Melon shape. Figure 4a provides the ground truth of the melon-shaped trajectory, another impressive trajectory provided by the NeuroBEM. The acceleration and angular rate are operating at very wide range, with a maximal angular rate of $433\,\mathrm{deg/s}$ and a maximal acceleration of $35.6\,\mathrm{m/s^2}$. Significant differences in the attitude estimates can be found in the peaks of the ESKF curve in Fig. 4b. The peaks indicate several transitions between $-\pi$ to π rad even in the roll angle. The ESKF cannot track the roll and yaw angles fast enough, while the K-ESKF tracks the attitude more accurately. In addition, Fig. 4c shows that the K-ESKF outperforms the ESKF in tracking the velocity.



(a) True trajectory. (b) Euler angles estimate errors (from quaternions). (c) Velocity estimate errors.

Fig. 3: Trajectory 1: Wobbly circle. The K-ESKF estimates the attitude and the velocity at the same level as the ESKF.



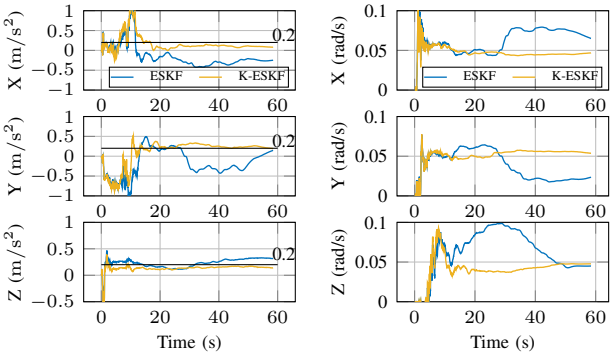
(a) True trajectory. (b) Euler angles estimate errors (from quaternions). (c) Velocity estimate errors.

Fig. 4: Trajectory 2: Melon shape. The K-ESKF outperforms the ESKF in both tracking the attitude and the velocity.

TABLE I: Trajectory 2: RMSE for Position, Velocity, and Euler Angles.

	ESKF	**K-ESKF**
Position (m)	[0.25, 0.19, 0.15]	[0.08, 0.08, 0.10]
Velocity (m/s)	[1.04, 0.75, 0.64]	[0.32, 0.30, 0.41]
Attitude (deg)	[8.00, 3.52, 7.26]	[3.58, 2.14, 4.00]

Table I summarizes the RMSEs of the position, velocity and attitude estimates. Using the K-ESKF, the attitude error and the velocity error are reduced by about 50%. This aggressive trajectory highlights that the K-ESKF not only estimates the pose as accurately as the standard ESKF in normal flights but also achieves better results in highly agile scenarios. The reason is that the K-ESKF can estimate the IMU biases more stably than the ESKF, as shown in Fig. 5.



(a) Acceleration bias estimates. (b) Angular rate bias estimates.

Fig. 5: Trajectory 2: Melon shape. The K-ESKF estimates the IMU bias more robust than the ESKF. As a reference, the acceleration bias is $0.2\,\mathrm{m/s^2}$ and the angular rate bias is $0.05\,\mathrm{rad/s}$.

V. CONCLUSIONS

The paper proposes a novel approach, the Koopman operator-based error-state Kalman filter (K-ESKF), for accurate pose and IMU bias estimation in quadrotors, especially during highly agile maneuvers. Using the data-driven Koopman bilinear modeling, we can effectively capture the full-state dynamics of a quadrotor, especially in quaternion. Through the evaluation using the NeuroBEM quadrotor dataset, our method demonstrates better performance compared to the first-order Euler method. As the DNN output dimension increases, we observe an exponential reduction in the velocity prediction error. Using two trajectories from the dataset, the comparative analysis with the standard ESKF highlights that our approach is not only suitable for normal flights but also accurately tracks the pose during highly agile maneuvers. Evaluation using the melon-shaped flight confirms that the K-ESKF significantly outperforms the ESKF, achieving an approximately 50% reduction in attitude and velocity errors due to its ability to accurately track IMU biases. These results highlight the robustness and effectiveness of the proposed K-ESKF for quadrotors in highly agile scenarios.

Future work could extend the K-ESKF to the error-state covariance update by training the Koopman observation functions on the error state, rather than on the quaternion.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[2] B. D. O. Anderson and J. B. Moore, "Optimal filtering," *Dover Publications*, vol. 1, no. 1, p. 367, 1979, iSBN: 0136381227.

[3] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.

[4] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1, pp. 99–141, 2001.

[5] 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.

[6] 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.

[7] Y. Han, W. Hao, and U. Vaidya, "Deep learning of Koopman representation for control," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1890–1895.

[8] R. Wang, Y. Han, and U. Vaidya, "Deep Koopman data-driven control framework for autonomous racing," in *Proc. Int. Conf. Robot. Autom.(ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–6.

[9] H. Shi and M. Q.-H. Meng, "Deep Koopman operator with control for nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7700–7707, 2022.

[10] C. Folkestad and J. W. Burdick, "Koopman NMPC: Koopman-based learning and nonlinear model predictive control of control-affine systems," *CoRR*, vol. abs/2105.08036, 2021. [Online]. Available: https://arxiv.org/abs/2105.08036

[11] K. Zheng, P. Huang, and G. Fettweis, "Optimal control of quadrotor attitude system using data-driven approximation of Koopman operator," in *The 22nd World Congress of the International Federation of Automatic Control (IFAC World Congress)*, Yokohama, Japan, Jul 2023.

[12] V. Zinage and E. Bakolas, "Koopman operator based modeling and control of rigid body motion represented by dual quaternions," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3997–4002.

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

[14] M. Kanai and M. Yamakita, "Lifted bilinear model-based linear model predictive control with scalability," in *The 22nd World Congress of the International Federation of Automatic Control (IFAC World Congress)*, Yokohama, Japan, Jul 2023.

[15] 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.

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

[17] M. Netto and L. Mili, "A robust data-driven Koopman Kalman filter for power systems dynamic state estimation," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7228–7237, Nov. 2018.

[18] W. H. Syed, A. Eichler, A. Nawaz, B. Sharan, and H. Werner, "Koopman-based Kalman filter for fault detection for the superconducting radio frequency cavities of the european xfel," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6855–6860.

[19] K. Wang, M. Liu, W. He, C. Zuo, and F. Wang, "Koopman Kalman particle filter for dynamic state estimation of distribution system," *IEEE Access*, vol. 10, pp. 111 688–111 703, 2022.

[20] K. Wang, M. Liu, Y. Man, C. Zuo, and W. He, "Innovation-saturated Koopman Kalman filter for distribution system dynamic state estimation against measurement outliers," *Energy Reports*, vol. 9, pp. 888–897, Oct. 2023.

[21] L. Zeng, S. Sadati, and C. Bergeles, "Koopman operator-based extended kalman filter for cosserat rod wrench estimation," in *2023 International Symposium on Medical Robotics (ISMR)*, Atlanta, GA, USA, Apr. 2023, pp. 1–7.

[22] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza, "NeuroBEM: Hybrid aerodynamic quadrotor model," in *Robotics: Science and Systems XVII*, Jul. 2021.

[23] J. N. Kutz, J. L. Proctor, and S. L. Brunton, "Koopman theory for partial differential equations," *arXiv preprint arXiv:1607.07076*, 2016.

[24] A. Mauroy, I. Mezić, and Y. Susuki, *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, ser. Lecture Notes in Control and Information Sciences. Springer International Publishing, 2020.

[25] J. Solà, "Quaternion kinematics for the error-state Kalman filter," *CoRR*, vol. abs/1711.02508, 2017.

[26] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, "VIMO: Simultaneous visual inertial model-based odometry and force estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.

[27] D. Q. Huynh, "Metrics for 3D rotations: comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, Oct. 2009.

[28] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, "Extended Kalman filter vs. error state Kalman filter for aircraft attitude estimation," in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6615.

[29] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[30] A. Valade, P. Acco, P. Grabolosa, and J.-Y. Fourniols, "A study about Kalman filters applied to embedded sensors," *Sensors*, vol. 17, no. 12, p. 2810, 2017.