2024 Tenth Indian Control Conference (ICC)
9 - 11 December, 2024
IISER Bhopal, India

Koopman framework-based observer design for actuator fault identification in Quadrotor

Ravi Kiran Akumalla, Amit Shukla, Tushar Jain

Abstract— **This paper investigates the implementation of the Koopman framework to quadrotor by numerical simulations. Initially, the nonlinear dynamics of a quadrotor is modelled in MATLAB/SIMULINK. The attitude and altitude control of the model is achieved using individual PID controllers. From this closed-loop model, the attitude data and angular torque data are collected. Using this data, a linear approximation of the attitude dynamics of the quadrotor is found by applying the Koopman framework. This identified Koopman linear model is leveraged as a predictor within the model predictive controller (MPC) to achieve inner loop attitude tracking of the quadrotor. Then, a loss of actuator effectiveness (LOE) fault is introduced in one of the actuators, which is successfully identified by designing a linear observer based on the identified Koopman linear model. Finally, the simulation results are provided, describing the attitude control and actuator fault identification in quadrotor using the Koopman framework.**

I. INTRODUCTION

With increasing computation capabilities, the recent application of the Koopman framework in the area of robotic control systems has seen a notable rise [1]. The nonlinear system, which is of finite dimensions, is theoretically converted into a linear system of infinite dimensions, using the Koopman framework. However, a linear approximation is found by lifting it to finite number of higher dimensions for practical applications[2][3]. This lifting is accomplished by observable functions, which are the functions of states and inputs of the dynamical system. The identified linear model, which embodies the nonlinear system dynamics, is used for control applications[4].

Koopman first introduced the idea of the framework[5], which eventually gained more attention as a data-driven technique in the field of control systems [6][7]. In [8], a linear model of a wind farm was estimated using the Koopman data-driven framework, and the MPC controller was effectively applied to the discovered linear model. Similarly, the unknown dynamics of an under-actuated ship system are identified by Koopman data-driven approach and efficiently implemented MPC using the identified Koopman predictor [9]. S.P Nandanoori, et.al., in their work[3] studied the nonlinear multi agent systems by obtaining its finite dimensional approximation using koopman framework. In [4][10], linear observers were designed to nonlinear systems while in another study [11], the fault detection and isolation were performed on a benchmark model of a two-link manipulator leveraging the Koopman framework. Indeed, there are many other successful applications of the Koopman framework in the literature [12][13]. This inspiration drove us to implement it to a highly nonlinear quadrotor.

The Koopman operator-based control of quadrotor is also explored in the literature. In [14] the dynamics of the quadrotor are described on SE(3) based on Koopman theory by deriving the observable functions in a systematic approach. In [1], the Koopman-based active learning strategy is applied to stabilize a freely falling quadrotor. In [15], the quadrotor attitude is optimally controlled, using deep neural networks for learning the observable functions along with the Koopman linear model. Recently, the Koopman model of quadrotor is identified in noisy environments, and optimal reference trajectory tracking is obtained[16]. Some other recent works for Koopman framework-based quadrotor control can be referred to in [17][18][19]. But till our knowledge, the Koopman framework for quadrotor with faults is not investigated. So in this study, this data-driven framework is explored for fault identification in quadrotor.

We used a new set of physically motivated observable functions for obtaining the Koopman predictor of attitude dynamics of the quadrotor and leveraging it in MPC for tracking the desired attitude. We then designed a linear observer using this Koopman predictor for actuator fault identification in quadrotor. The multiplicative fault, specifically loss of actuator effectiveness, is considered in this work to show the effectiveness of the designed Koopman-based linear observer. The following are the main contributions:

- Selection of physically motivated observable functions from the nonlinear attitude dynamics of quadrotor for Koopman framework-based attitude control.
- Designing an observer based on identified Koopman predictor for actuator fault identification in quadrotor.

The organisation of the paper starts with discussing the Koopman framework, Koopman Model Predictive Control (KMPC), along with the modelling of the quadrotor control by PID in Section II. The methodology of selecting the observable functions, finding the linear Koopman predictor and its application in KMPC for attitude control, and observer design for fault identification is explained in Section III. Section IV discusses the simulation results, while Section

This work was not supported by any organization

R.K.Akumalla is with the School of Mechanical and Materials Engineering, Indian Institute of Technology, Mandi, HP, India ravi30011997@gmail.com

A.Shukla is with the Faculty of School of Mechanical and Materials Engineering, Indian Institute of Technology Mandi, HP, India amitshukla@iitmandi.ac.in

T.Jain is with the Faculty of School of Computing and Electrical Engineering, Indian Institute of Technology Mandi, HP, India tushar@iitmandi.ac.in

V concludes discussing the future works.

II. Preliminaries

A. Notations

The symbol $I_{n\times n}$ denotes an n-order identity matrix. The symbol $0_{m\times n}$ represents a zero matrix with order $m\times n$. If we assume that $a\in R^n$, then a^T represents its transpose and $a^\dagger$ represents its pseudo-inverse. The P-norm is represented by $||a||_P^2 = a^T Pa$, where $P\in R^{n\times n} > 0$ is a diagonal matrix.

B. Koopman Framework

Nonlinear systems can be handled with the Koopman operator by employing a globally linear description[2]. The Koopman operator is described in brief here; for a more thorough explanation, see [20]. Suppose a nonlinear system that has a state $s\in R^n$ and input $u\in R^r$ is given as:

$$\dot{s} = f(s,u) \tag{1}$$

The states of the nonlinear system (1) evolve according to the nonlinear map $f: R^n\times R^r \to R^n$. Additionally, a collection of scalar-valued functions referred to as observable functions is defined as $h\in\mathscr{H}$. These functions, which are a part of the infinite-dimensional Hilbert space $\mathscr{H}$, are the functions of states and inputs. Now, the Koopman operator $\mathscr{K}:\mathscr{H}\to\mathscr{H}$ is given as

$$\mathscr{K}h(s,u) = h(f(s,u)) \tag{2}$$

This operator, which is of infinite dimensions, linearly propagates the observable functions h with respect to time, in contrast to the nonlinear system (1). For practical purposes, the Koopman operator K is approximated to finite dimensions. An explanation of the same is illustrated in Fig. 1. From a given nonlinear dynamical system $\dot{s} = f(s,u)$, $m+1$ time domain samples of the state data $s\in R^n$ and input data $u\in R^r$ are collected at a specific sampling time and divided in to the sets $U\in R^{r\times m}$, $S\in R^{n\times m}$ and $T\in R^{n\times m}$. Here U is the input data matrix and T is the left-shifted data matrix of S by one instant such that $t(k) = s(k+1)$ with $t(k)\in T$ and $s(k+1)\in S$. These data are then passed into observable functions, represented by $h = [h_1, h_2, ..., h_p]^T$; $h_i: R^n\to R$; $i = \{1, 2.., p\}$, which can be selected either as physics-informed observable functions from the dynamics or by estimating through machine learning techniques. After applying each sample to the observable functions, the data matrices S and T are lifted to Q_S and Q_T with each element as $q_{s(k)} = h(s(k))$; $s(k)$ being the k^{th} sample. Finally, an optimization problem,

$$\begin{aligned} &\min_{A,B} ||Q_T - AQ_S - BU||_F \\ &\min_{C} ||S - CQ_S||_F \end{aligned} \tag{3}$$

as shown in (3), is solved to get the approximated linear model with state and input matrices respectively as $A\in R^{p\times p}$, $B\in R^{p\times r}$ and output matrix $C\in R^{n\times p}$. This model maps the original states from the lifted states resulting in the Koopman linear predictor.

C. Koopman Model Predictive Control

Upon obtaining the Koopman predictor, it is integrated as the prediction model within the MPC framework[21], thereby establishing a Koopman-based linear MPC (KMPC) for nonlinear system.

$$\begin{aligned} \min_{u(k:k+N-1)} \quad & \sum_{j=k}^{k+N-1} ||s(j) - s_{ref}||_{\boldsymbol{Q}}^2 + ||u(j)||_{\boldsymbol{R}}^2 \\ \text{s.t.} \quad & q(j+1) = Aq(j) + Bu(j) \\ & s(j) = Cq(j) \\ & u_{min} \le u(j) \le u_{max} \end{aligned} \tag{4}$$

The KMPC formulation is shown in (4).Here, $s(j)$ represents the current state and s_{ref} represents the desired state. The minimum energy input u is designed to minimize the error between current and desired state while satisfying the constraints. The terms Q and R are tuning factors corresponding to the Q-norm and R-norm, respectively. Matrices A, B and C are part of the Koopman linear predictor and u_{min}, u_{max} define the minimum and maximum constraints for input to be designed.

D. Modelling of quadrotor

The nonlinear dynamics of the quadrotor are modelled in the SIMULINK for data collection to Koopman framework. The overall nonlinear dynamics of the quadrotor are given in (5). Refer [22] [23] for detailed derivation of the dynamics. From Fig. 2, two reference frames can be noticed. One is earth fixed frame $E = [e_x, e_y, e_z]^T$ while the other is body fixed frame $B = [b_x, b_y, b_z]^T$. $x, y,$ and z stand for the quadrotor's cartesian position with respect to the earth fixed frame. The variables $\phi, \theta,$ and ψ show the quadrotor's roll, pitch, and yaw orientation along the axes e_x, e_y, and e_z. $\omega_1, \omega_2, \omega_3,$ and ω_4 show its propeller angular velocities.

$$\begin{aligned} \ddot{x} &= (\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta)\frac{u_1}{\mu} \\ \ddot{y} &= (-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi)\frac{u_1}{\mu} \\ \ddot{z} &= -a_g + (\cos\phi\cos\theta)\frac{u_1}{\mu} \\ \ddot{\phi} &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{J_R}{I_x}\dot{\theta}\omega_r + \frac{1}{I_x}u_2 \\ \ddot{\theta} &= \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{J_R}{I_y}\dot{\phi}\omega_r + \frac{1}{I_y}u_3 \\ \ddot{\psi} &= \frac{I_x - I_y}{I_z}\dot{\theta}\dot{\phi} + \frac{1}{I_z}u_4 \end{aligned} \tag{5}$$

$\ddot{x}, \ddot{y}, \ddot{z}$ stand for linear accelerations, and $\dot{x}, \dot{y}, \dot{z}$ denote linear velocities. $\dot{\phi}, \dot{\theta}, \dot{\psi}$ stand for angular velocities and $\ddot{\phi}, \ddot{\theta}, \ddot{\psi}$ denotes the angular accelerations of the quadrotor along e_x, e_y, e_z. The input vector is defined as $u = [u_1, u_2, u_3, u_4]^T$, where u_1,u_2,u_3,u_4 respectively denotes total lift force, roll torque, pitch torque, and yaw torque. ω_r represents the relative angular velocity. The relation of thrust and torques

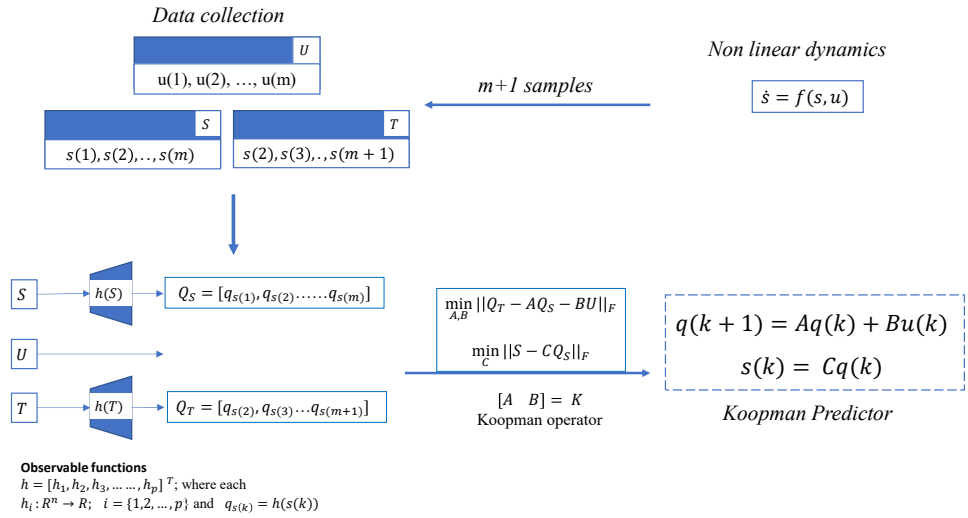


Fig. 1. Koopman framework

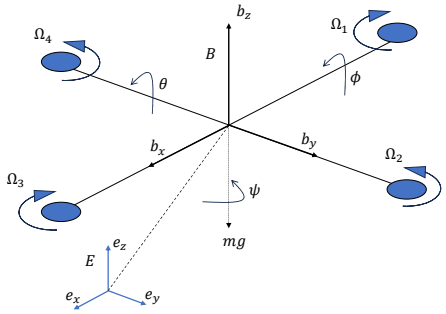


Fig. 2. Schematic diagram of quadrotor

TABLE I
QUADROTOR PARAMETERS

Symbol	Parameter of quadrotor	Value
μ	mass	0.65kg
l	arm length	0.23m
b	thrust factor	$3.13x10^{-5}$Ns^2
d	drag factor	$7.5x10^{-7}$Ns^2
I_x, I_y	Inertial moment: x,y-axis	7.5×10^{-3}kg.m^2
I_z	Inertial moment: z-axis	1.3×10^{-2}kg.m^2
J_r	Moment of inertia of propellers	6.5×10^{-5}kg.m^2
a_g	acceleration due to gravity	9.8 m/s^2

Units: kg =kilogram, m =metre, Ns^2 =Newton.seconds^2, kg.m^2 = kilogram.metre^2, m/s^2 =metre per seconds^2.

with angular velocity is given as (6)

$$\begin{aligned} u_1 &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 &= lb(-\omega_2^2 + \omega_4^2) \\ u_3 &= lb(-\omega_1^2 + \omega_3^2) \\ u_4 &= d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \\ \omega_r &= -\omega_1 + \omega_2 - \omega_3 + \omega_4 \end{aligned} \tag{6}$$

Considering these nonlinear dynamics and parameters as in Table I, the quadrotor model is built in SIMULINK with $[x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ as overall state of the quadrotor. The desired altitude (z_{des}) and attitude $(\phi_{des}, \theta_{des}, \psi_{des})$ tracking as shown in Fig. 3. are implemented using PID controller as described in[23]. The lifting force u_1 is designed

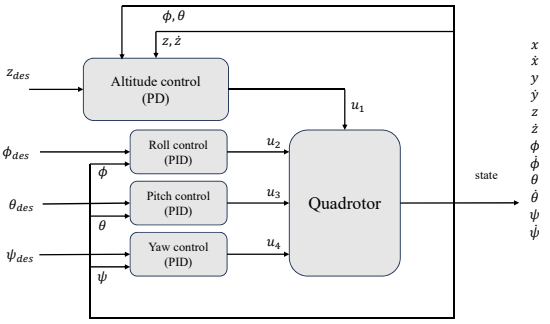


Fig. 3. Implementation of PID controller to quadrotor

as:

$$\begin{aligned} u_1 &= \frac{\mu(a_g + \ddot{z}_{des})}{cos(\phi)cos(\theta)} \\ \text{where } \ddot{z}_{des} &= k_p(z_{des} - z) + k_d(\dot{z}_{des} - \dot{z}) \end{aligned} \tag{7}$$

The roll, pitch and yaw control is implemented by designing angular torques u_2, u_3, u_4 as

$$\begin{aligned} u_{2,3,4} &= \text{PID}(error)_{\phi,\theta,\psi} \\ error_\phi &= \phi_{des} - \phi \\ error_\theta &= \theta_{des} - \theta \\ error_\psi &= \psi_{des} - \psi \end{aligned} \tag{8}$$

The k_p, k_i, k_d - PID parameters for altitude control and roll, pitch, yaw control are adopted as described in [23].

III. METHODOLOGY

A. Selection of Observable functions

The observable functions in this work are selected as $h = [\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}, \psi\dot{\theta}, \dot{\phi}\dot{\psi}, \dot{\theta}\dot{\phi}]^T$, which are the nonlinear terms in the attitude dynamics. This idea of selecting physically motivated observable functions from dynamics is adopted from the literature [24][25].

B. Linear Koopman Predictor

The state data (more precisely, the attitude data) and input data (angular torques) are gathered for 50 sec at a sampling time of 0.1sec, following the implementation of altitude and attitude control of the quadrotor using PID in SIMULINK. These data are organized into the state data matrices $S \in R^{6\times500}$, $T \in R^{6\times500}$ and input data matrix $U \in R^{3\times500}$. Now, using the observable functions selected in the previous section, the lifted state matrices are obtained. Given that nine observable functions exist, the lifted state matrices obtained are $Q_S \in R^{9\times500}$ and $Q_T \in R^{9\times500}$. Leveraging the lifted state matrices, the optimization problem (3) is solved analytically[24][25] as shown in (9) to obtain the Koopman predictor (K) matrices $A \in R^{9\times9}, B \in R^{9\times3}$. As the original states are already augmented in the lifted states., the matrix $C \in R^{3\times9}$ can be given as (10).

$$
\begin{aligned}
K &= \begin{bmatrix} A & B \end{bmatrix} = M * G^{\dagger} \\
\text{where} \quad M &= \frac{1}{m+1}\sum_{k=1}^{m} q_{s(k+1)} * \begin{bmatrix} q_{s(k)} \\ u(k) \end{bmatrix}^{T} \\
G &= \frac{1}{m+1}\sum_{k=1}^{m} q_{s(k)} * q_{s(k)}^{T}
\end{aligned} \tag{9}
$$

$$
C = \begin{bmatrix} I_{3\times3} & 0_{3\times6} \end{bmatrix} \tag{10}
$$

C. Inner loop attitude control by KMPC

The attitude control and altitude control of the quadrotor are performed using KMPC and PID respectively as shown in Fig. 4. As discussed in (4), the KMPC is implemented for inner loop attitude tracking, by leveraging the A, B, C matrices in its architecture. Based on the parameters of the quadrotor, the minimum and maximum constraints on the angular torque inputs are selected as $[-1,-1,-1]^T$ and $[1,1,1]^T$. The YALMIP toolbox [26] is used for implementation of KMPC.

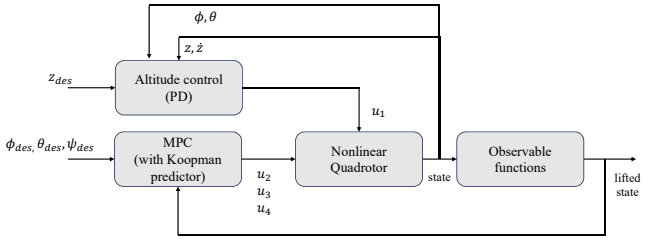


Fig. 4. Architecture of KMPC with PID

D. Fault Identification

The loss of actuator effectiveness (LOE) fault is induced in one of the actuators of the quadrotor model by decreasing the propeller efficiency. This multiplicative fault[27][28] can be modelled as $f_i\omega_i^2$, where the efficiency variable $f_i \in [0,1]$ and $i \in \{1,2,3,4\}$. $f_i = 1$ implies no fault, $f_i = 0$ implies total loss, and $f_i < 1$ implies partial loss of effectiveness of actuator.

A linear observer as shown in Fig. 5, is designed based on the obtained Koopman predictor to identify the fault.

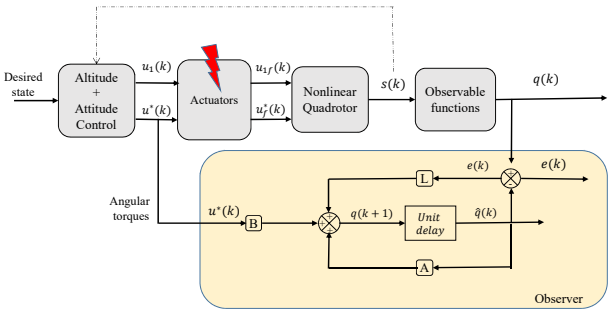


Fig. 5. Koopman based observer

For instance, assume the Koopman predictor obtained (9) is in good coherence with (1). Now, the attitude dynamics of nonlinear quadrotor with observable functions is represented as:

$$
q(k+1) = Aq(k) + Bu_f^*(k) \tag{11}
$$

Based on (11), a linear observer is designed for the nonlinear quadrotor as:

$$
\hat{q}(k+1) = A\hat{q}(k) + Bu^*(k) + L(q(k) - \hat{q}(k)) \tag{12}
$$

where $q(k)$ is original lifted state, $\hat{q}(k)$ is the estimate of the lifted state, L is the observer gain matrix, $u^*(k) = [u_2(k), u_3(k), u_4(k)]^T$ is the calculated angular torque input by controller and $u_f^*(k) = [u_{2f}(k).u_{3f}(k), u_{4f}(k)]^T$ is the faulty angular torque to the quadrotor. Now the estimation error $e(k)$ is given as:

$$
e(k) = q(k) - \hat{q}(k) \tag{13}
$$

which can be further extended as

$$
\begin{aligned}
e(k+1) &= q(k+1) - \hat{q}(k+1) \\
&= (A-L)e(k) + B(u_f^*(k) - u^*(k))
\end{aligned} \tag{14}
$$

From (14) it is clear that if eigenvalues of $A-L$ are less than unity, the estimation error converges to zero asymptotically. Using *place* command in MATLAB, the matrix L is selected such that $eig(A-L) < 1$. Now, from (14) the LOE of torque input $(u^* - u_f^* = LOE_u)$ is estimated as:

$$
u^*(k) - u_f^*(k) = -B^{\dagger}[e(k+1) - (A-L)e(k)] \tag{15}
$$

If the estimated $LOE_u = [LOE_{u2}, LOE_{u3}, LOE_{u4}]^T$ matches with actual loss of input due to fault in the SIMULINK model, our assumption (11) about the coherence of obtained Koopman predictor with original system will also be proved. Further from equation (6), the loss of roll and pitch torque can be deduced as:

$$
\begin{aligned}
LOE_{u2} &= u_2 - u_{2f} = lb[-(1-f_2)\omega_2^2 + (1-f_4)\omega_4^2] \\
LOE_{u3} &= u_3 - u_{3f} = lb[-(1-f_1)\omega_1^2 + (1-f_3)\omega_3^2]
\end{aligned} \tag{16}
$$

Assuming only one faulty actuator at a given time, by just observing the loss of roll torque (LOE_{u2}) or pitch torque

(LOE_{u3}), the faulty motor can be identified as, if

$$\begin{aligned} LOE_{u2} > 0 &\implies \omega_4 \text{ is faulty} \\ LOE_{u2} < 0 &\implies \omega_2 \text{ is faulty} \\ LOE_{u3} > 0 &\implies \omega_3 \text{ is faulty} \\ LOE_{u3} < 0 &\implies \omega_1 \text{ is faulty.} \end{aligned} \tag{17}$$

IV. Results and Discussion

To start with the Koopman framework, the state data and input data are collected from the SIMULINK model of the quadrotor controlled by PID controller, by giving the desired roll and pitch angles as sine wave of amplitude 0.01 and bias 1 and 0.5 respectively. The desired yaw angle is taken as zero. Using this state and input data, the linear predictor matrices A, B are obtained using Koopman framework as explained in section III-B. The Fig. 6 shows the prediction error between the states of nonlinear quadrotor and Koopman predictor for the same input. As the error is in the order of 10^{-3}, the obtained linear predictor is in good coherence with the nonlinear quadrotor. This matrices are then used in KMPC architecture as described in section III-C to track the desired angles. From Fig. 7,8 it can be observed that the KMPC is tracking varying angles of roll and pitch respectively.

A tiny spike can be seen at around 150^{th} sample in the roll angle and around 95^{th} sample in the pitch angle. This is a result of the computed A matrix which is coupling one angle to another. Therefore, a sudden alteration in one angle has an impact on the other. In order to prevent this, when calculating the Koopman predictor a constraint on proper structure of the matrices has to be taken into account, which will be an interesting direction of future research.

Then, a linear observer is designed based on the obtained Koopman predictor as discussed in section III-D. To test its effectiveness in fault identification, two cases are considered. In case 1, the efficiency of motor 1 is decreased to 55% at $t = 50sec$. From Fig. 9, it can be observed that the estimated loss of pitch torque converges to a negative value, $LOE_{u3} = -0.3 < 0$ implying that motor 1 is faulty. Also the estimated loss of pitch torque is matching with true loss. Similarly, in case 2, the efficiency of motor 4 is decreased to 64%. From Fig. 10, it can be observed that the estimated loss of roll torque is positive, $LOE_{u2} = 0.2 > 0$ implying that motor 4 is faulty and it is matching with true loss. Although the other loss torque is getting stimulated as soon as fault occurs, it can be observed that it is converging to zero later.

The ability of the designed Koopman based observer to differentiate faults from disturbances is also observed by introducing Gaussian white noise in the attitude dynamics of the quadrotor. To test it, the efficiency of motor 3 and 4 are reduced to 65% and 80% respectively. From Fig. 11, it can be observed that the disturbances are not having any effect during fault identification.

V. Conclusion and Future works

The quadrotor attitude tracking and actuator fault identification is performed using the Koopman framework. The Koopman linear model is identified for the attitude dynamics of the quadrotor by selecting physically motivated observable functions and found in good coherence with the original nonlinear dynamics. Then a linear observer is designed based on this identified Koopman model for actuator fault identification in quadrotor. The LOE fault is injected in one of the actuators and identified successfully. Also, the loss of input torque is estimated with the designed observer and found to match with the true loss. In future works, the work will be extended to the fault tolerant control of the Koopman-based quadrotor. Also, a proper structure constraint for the estimation of the Koopman predictor will be investigated.

References

[1] I. Abraham and T. D. Murphey, "Active learning of dynamics for data-driven control using koopman operators," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1071–1083, 2019.

[2] P. Bevanda, S. Sosnowski, and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," *Annual Reviews in Control*, vol. 52, pp. 197–212, 2021.

[3] S. P. Nandanoori, S. Pal, S. Sinha, S. Kundu, K. Agarwal, and S. Choudhury, "Data-driven distributed learning of multi-agent systems: A koopman operator approach," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 5059–5066.

[4] 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.

[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] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman operator in systems and control*. Springer, 2020.

[7] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[8] B. Sharan, A. Dittmer, and H. Werner, "Real-time model predictive control for wind farms: a koopman dynamic mode decomposition approach," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 1006–1011.

[9] S. Li, Z. Xu, J. Liu, and C. Xu, "Learning-based extended dynamic mode decomposition for addressing path-following problem of underactuated ships with unknown dynamics," *International Journal of Control, Automation and Systems*, pp. 1–14, 2022.

[10] 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.

[11] M. Bakhtiaridoust, M. Yadegar, and N. Meskin, "Data-driven fault detection and isolation of nonlinear systems using deep learning for koopman operator," *ISA transactions*, vol. 134, pp. 200–211, 2023.

[12] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "Deep neural networks with koopman operators for modeling and control of autonomous vehicles," *IEEE transactions on intelligent vehicles*, vol. 8, no. 1, pp. 135–146, 2022.

[13] J. Wang, B. Xu, J. Lai, Y. Wang, C. Hu, H. Li, and A. Song, "An improved koopman-mpc framework for data-driven modeling and control of soft actuators," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 616–623, 2022.

[14] V. Zinage and E. Bakolas, "Koopman operator based modeling for quadrotor control on se (3)," *IEEE Control Systems Letters*, vol. 6, pp. 752–757, 2021.

[15] K. Zheng, P. Huang, and G. P. Fettweis, "Optimal control of quadrotor attitude system using data-driven approximation of koopman operator," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 834–840, 2023.

[16] Y. Oh, M. H. Lee, and J. Moon, "Koopman-based control system for quadrotors in noisy environments," *IEEE Access*, vol. 12, pp. 71 675–71 684, 2024.

[17] S. Martini, A. Rizzo, M. Stefanovic, P. Livreri, M. J. Rutherford, and K. P. Valavanis, "Koopman operator based modeling and control of quadrotors," in *IUTAM Symposium on Optimal Guidance and Control for Autonomous Systems*. Springer, 2023, pp. 253–266.

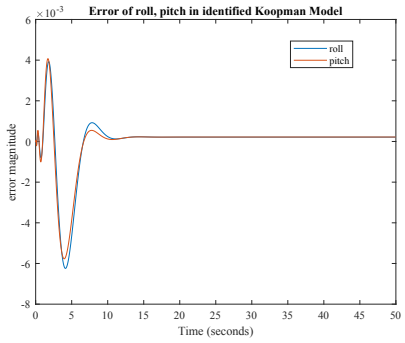


Fig. 6. Prediction error

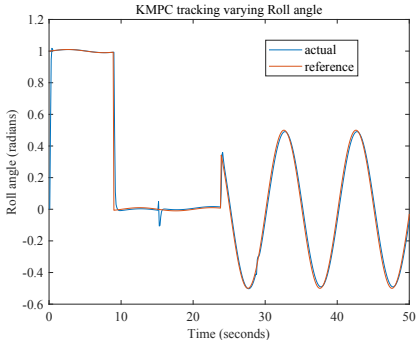


Fig. 7. Roll angle

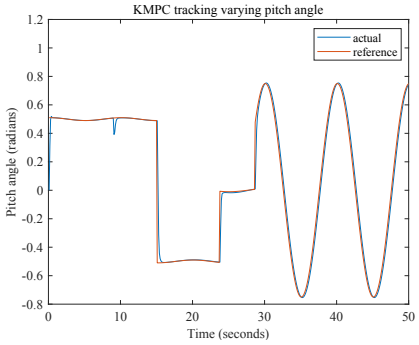


Fig. 8. Pitch angle

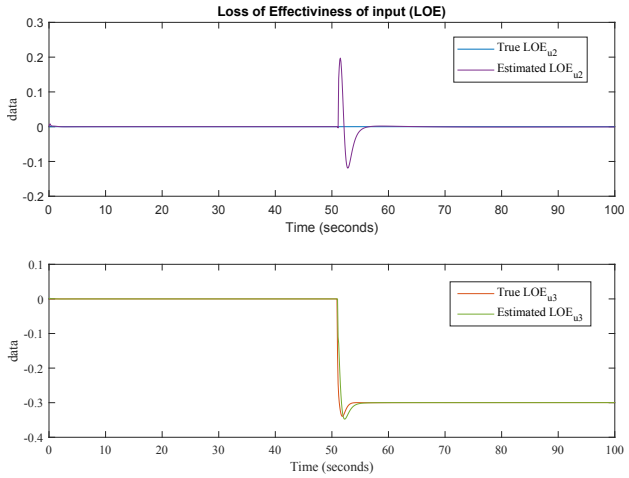


Fig. 9. LOE of roll and pitch torque with faulty motor 1

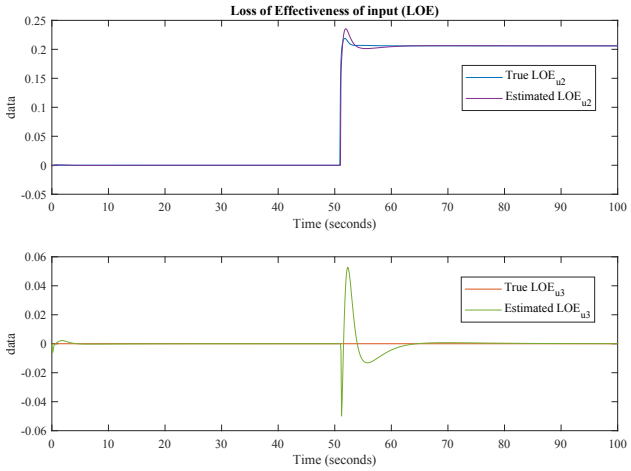


Fig. 10. LOE of roll and pitch torque with faulty motor 4

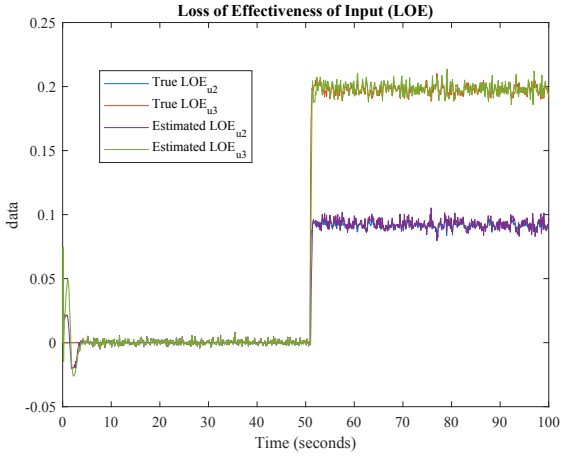


Fig. 11. Loss of effectiveness of roll and pitch torque with disturbance

[18] S. Martini, K. P. Valavanis, M. Stefanovic, A. Rizzo, and M. J. Rutherford, "Koopman-based reduced order controller design for quadrotors," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2024, pp. 370–375.

[19] Z. M. Manaa, A. M. Abdallah, M. A. Abido, and S. S. A. Ali, "Koopman-lqr controller for quadrotor uavs from data," *arXiv preprint arXiv:2406.17973*, 2024.

[20] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, p. e0150171, 2016.

[21] M. Korda, Y. Susuki, and I. Mezić, "Power grid transient stabilization using koopman model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 297–302, 2018.

[22] H. Mo and G. Farid, "Nonlinear and adaptive intelligent control techniques for quadrotor uav–a survey," *Asian Journal of Control*, vol. 21, no. 2, pp. 989–1008, 2019.

[23] M. Karahan and C. Kasnakoglu, "Modeling and simulation of quadrotor uav using pid controller," in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2019, pp. 1–4.

[24] 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)*. IEEE, 2021, pp. 6855–6860.

[25] P. S. Cisneros, A. Datar, P. Göttsch, and H. Werner, "Data-driven quasi-lpv model predictive control using koopman operator techniques," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6062–6068, 2020.

[26] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[27] A. Baldini, R. Felicetti, A. Freddi, A. Monteriù, and M. Tempesta, "Estimation of actuator faults in quadrotor vehicles: from theory to validation with experimental flight data," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1249–1256.

[28] A. Eltrabyly, D. Ichalal, and S. Mammar, "Fault-tolerant model predictive control trajectory tracking for a quadcopter with 4 faulty actuators," *IFAC-PapersOnLine*, vol. 54, no. 4, pp. 141–146, 2021.