Mechatronics 88 (2022) 102907

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

A hybrid disturbance observer for delivery drone's oscillation suppression[☆],[☆☆]

Zhu Chen[a], Chang Liu[a], Hao Su[b], Xiao Liang[c,*], Minghui Zheng[a,*]

[a] Mechanical and Aerospace Engineering Department, University at Buffalo, Buffalo, NY, United States of America
[b] Mechanical and Aerospace Engineering Department, North Carolina State University, United States of America
[c] Civil, Structural, and Environmental Engineering Department, University at Buffalo, Buffalo, NY, United States of America

ARTICLE INFO

Keywords:
DOB design
Learning filter
Neural network
Delivery drones

ABSTRACT

This paper proposes a new hybrid disturbance observer (DOB) to help suppress disturbance to the control systems. The proposed hybrid DOB consists of three main parts: (1) an actual system, (2) a simulated system, and (3) a learning filter that connects the actual and simulated systems. The simulated system aims to replicate the actual system response, where it leverages a neural network model to predict the input disturbance and generate the predicted system response. Such system response is used to generate a learning signal through a learning filter; this learning signal is then added to the feedforward loop of the estimation framework to enhance the disturbance estimate and its suppression performance for the actual system. The proposed hybrid DOB is designed to advance the standard DOB structure with a learning-based feedforward compensation. While the proposed method does not modify the baseline controller, it is well suited to systems whose baseline controllers are difficult or impossible to be changed. Considering the delivery drones are subject to oscillations when dropping payloads, experimental tests with multiple payload dropping scenarios have been conducted using both the hybrid and standard DOB, where the compared results validate the effectiveness and advantages of the proposed hybrid DOB.

1. Introduction

Disturbance observer (DOB) is a powerful technique for disturbance rejection and it has been applied to various systems including hard disk drives, manipulators, road vehicles, and drones [1–4]. A standard DOB scheme usually consists of two components which are a stable plant inverse and a low-pass filter [5]. The plant inverse outputs an internal negative feedback to have the disturbance cancellation; the low-pass filter on the other hand, is synthesized for robustness purpose. While the standard DOB-based control traditionally targets for the linear time-invariant (LTI) and single-input–single-output systems, its limits have been pushed to suit more generalized systems such as non-minimum phase systems [6,7], multiple-input–multiple-output systems [8,9], and nonlinear systems [10].

In the last few years, advanced control techniques have been combined with DOB for performance enhancement. For example, fuzzy logic DOBs [11], sliding mode DOBs [12,13], and adaptive DOBs [14,15] are introduced to either improve the disturbance estimation or increase the system robustness to disturbance and uncertainties. In [16], a learning-based DOB framework is designed by leveraging iterative learning control (ILC) scheme and H-infinity synthesis theory, and this design is adaptive to generalized systems. In [17], an internal mode frame-based DOB is designed for delivery drones to reduce disturbance caused by the payloads. Besides the model-based DOBs, machine learning on the other hand, promises model-free methods due to its data-driven mechanism. In recent years, neural network (NN) based DOB has been introduced to have predictions and enhanced disturbance estimations [18–20], and it has been employed in various applications including missile control [21] and autonomous underwater vehicles [22].

It is well acknowledged that model-based DOBs can achieve high performance due to the explicit expression of system dynamics. However, demanding performance usually requires an accurate model with expensive tuning design, and this is infeasible for systems that the baseline controller or DOB parameters are not allowed to be modified. Data driven-based DOBs on the other hand, are positively recognized for

[☆] This work was partially supported by the USA National Science Foundation [Grant No: 2046481].

[☆☆] This paper was recommended for publication by Associate Editor Cong Wang.

[*] Corresponding authors.
E-mail addresses: zhuchen@buffalo.edu (Z. Chen), cliu57@buffalo.edu (C. Liu), hsu4@ncsu.edu (H. Su), liangx@buffalo.edu (X. Liang), mhzheng@buffalo.edu (M. Zheng).

https://doi.org/10.1016/j.mechatronics.2022.102907
Received 1 June 2022; Received in revised form 24 August 2022; Accepted 11 October 2022
Available online 5 November 2022

replacing complicated dynamic models. The prediction further enables generating feedforward signals to compensate the estimation.

To suppress the disturbance to systems whose baseline controllers are impossible to be modified, we propose a hybrid DOB and consider the disturbance rejection with a drone dropping payload scenarios. To reduce the modeling complexity, the dynamic changes of the drone caused by the detaching of the payload from the drone is not modeled, and the oscillations caused by this dropping motion is considered as the external disturbance to the drone. The hybrid DOB includes a long short-term memory (LSTM) network to map from the payload information to the input disturbance (equivalent disturbance injected to the system), and the prediction is utilized to generate a learning signal through a dynamic filter. The learning signal is added to the estimation loop with the goal to achieve satisfactory performance regarding disturbance suppression. Mostly, the studies of drones with payload focus on the perturbation caused by the payload carrying instead of the dropping process [23,24]. The related studies of hybrid DOB design with applications to drones [25,26] are rare and the existing works are limited to numerical studies. This paper is based on our previous conference paper [26] with the following improvements: in this paper, (1) the LSTM network maps directly from the payload information to the input disturbance; (2) the system stability and robustness have been rigorously discussed; (3) the learning filter design is formulated into an optimization problem, and this optimization formulation method reflects the novelty contribution; (4) experimental tests are conducted for the validation.

Even though the fundamental problem in this payload dropping can be considered as a weight and inertia change of the drone, such that adaptive controller or tuning the gain of the PID controller might be able to handle the oscillations caused by this sudden inertia change, it is worth mentioning that, modifying the controller requires an accurate model and extensive tuning efforts. Besides, many commercial drones are not open for baseline controller modification once they are manufactured. Therefore, to provide an alternative method for disturbance suppression without modifying the baseline controller, our proposed method models the payload dropping as an external disturbance to the drone to avoid the re-modeling process and utilizes the neural network model to help improve the disturbance observer. The proposed DOB is an add-on which does not modify the original feedback controller, allowing more flexibility for the controller and DOB design. The major contributions of this paper lie in the design of a new hybrid DOB framework which consists of an actual system, a simulated system, and a learning filter. The learning filter design has been formulated into an optimization problem, and the systematic design guidelines are provided. The advantages of such a DOB can be summarized as follows. (1) For the drone delivery problem, an alternative DOB-based method is used to suppress the disturbance without modifying the baseline controller and system stability. This is particularly useful for systems whose baseline controller cannot be modified. (2) The influence to the drone caused by the payload dropping motion is treated as the external disturbance to avoid complicate dynamic modeling process, and this disturbance can be predicted by leveraging neural networks. (3) The learning signal compensates for errors associated with both the baseline controller and the DOB inaccuracies, allowing more design flexibility for the two dynamic components. (4) The hybrid DOB can achieve better system performance than the standard DOB regarding the drone's oscillation suppression, and experimental tests have been conducted to validate the effectiveness of the proposed method.

The remainder of the paper is organized as follows: Section 2 formulates the drone's payload dropping problem, where the hybrid DOB framework is presented, as well as the system stability and robustness are analyzed; Section 3 formulates the learning filter design into an optimization problem with solutions discussed; Section 4 demonstrates the validation process including the testing platform and 2 testing scenarios; Section 5 concludes the paper.

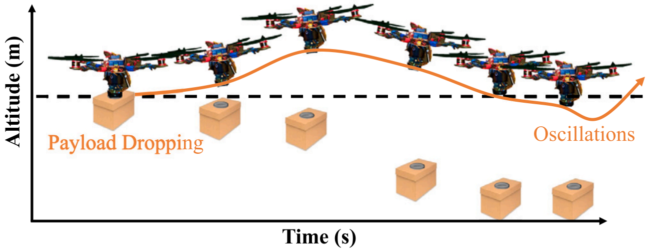


Fig. 1. Oscillations during payload dropping.

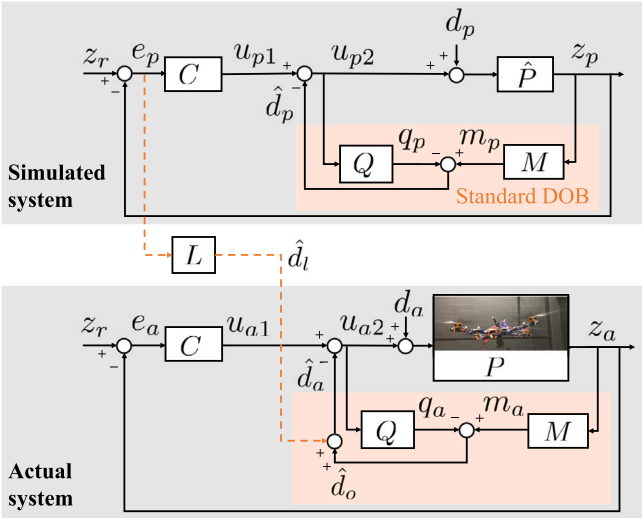


Fig. 2. Hybrid DOB framework.

2. Overview of the proposed hybrid DOB

2.1. Overall scheme

Large oscillation usually happens when a drone drops a payload. Consider a drone is delivering a package, and at the final phase, it hovers at a certain altitude and then drops the package to the ground as shown in Fig. 1. This sudden dropping action would cause undesired oscillations mainly in the vertical direction to the drone which needs to be suppressed to guarantee a safe and stable flight.

To suppress this oscillations, we propose a hybrid DOB as shown in Fig. 2. The hybrid DOB structure consists of three main parts: (1) an actual delivery system (in the bottom figure), (2) a simulated delivery system (in the upper figure), and (3) a learning filter that connects the actual and simulated systems. We will first introduce the notations and definitions used in the hybrid DOB structure: (1) the actual system consists of the drone with the dynamics denoted as P, the baseline feedback controller C, a standard DOB which consists of a model inverse parameter M, and a low-pass filter Q. And z_r is the trajectory reference in the vertical direction, z_a is the actual tracking output of the drone in the vertical direction, e_a is the trajectory tracking error, u_{a1} is the output of C, u_{a2} is the input of Q, q_a and m_a are the outputs of Q and M, respectively, d_a is the external disturbance to the drone system caused by the payload dropping action, $\hat{d}_o$ is the intermediate estimate from the DOB, and $\hat{d}_a$ is the output of the hybrid DOB. (2) The simulated system includes a nominal model of the drone dynamics $\hat{P}$, and the same C, M, Q parameters used in the actual system. And q_p and m_p are the outputs of Q and M, respectively, z_r is the same reference trajectory, z_p is the tracking output, e_p is the trajectory tracking error, u_{p1} is the output of C, u_{p2} is the input of Q,

d_p is the external disturbance, and $\hat{d}_p$ is the estimated disturbance. (3) The learning filter L takes e_p as its input, and generates the output $\hat{d}_l$ which is named as the learning signal. The learning signal is added to the feedforward estimation loop of the actual system to compensate for the estimate $\hat{d}_o$. The learning filter is designed based on the system dynamics and therefore, it dynamically brides the simulated and actual systems and transfer the knowledge from the simulated system to the actual system. In this study, the tracking error from the simulated system is utilized to synthesize the learning signal and added to the actual system for performance enhancement. The major differences between the simulated system and the actual system lie in that (i) the simulated system includes the nominal model $\hat{P}$ which approximates the actual model P in the actual system; (ii) a learning signal $\hat{d}_l$ is added to the estimation loop in the actual system, while there is no learning signal added in the simulated system; (iii) the simulated system is performed offline and the tracking error data can be collected in advance, while the implementation of actual system is in real time. The simulated system aims to replicate the actual system for the case when there is no learning signal added to the actual system.

2.2. Compensated disturbance estimation

This section presents more details on disturbance estimation using the proposed hybrid DOB. The necessity of the estimate compensation is first explained, and then the compensation process is illustrated.

The estimate $\hat{d}_a$ consists of two components, that is, the $\hat{d}_o$ which is from the standard DOB and the learning signal $\hat{d}_l$. In practice, the information $\hat{d}_o$ is not sufficient for systems' oscillation suppression and it is necessary to compensate for the estimation. Consider that in the actual system, when there is no $\hat{d}_l$, then $\hat{d}_o = \hat{d}_a$, and the estimate $\hat{d}_a$ can be related to the actual disturbance d_a as $\hat{d}_a = \Gamma\{d_a\}$, where Γ lumps P, C, M and Q. Generally, the parameter M is designed to inverse the plant P, while Q is a low-pass filter to reduce the high-frequency noise. If $\hat{d}_a$ accurately equals to d_a, then the disturbance d_a can be completely cancelled by adding $\hat{d}_a$ to the control loop. Ideally, if M correctly inverses P, that is $M = QP^{-1}$, and Q is designed to be 1, then we have $\hat{d}_a = d_a$. However, in practice, this is challenging to achieve due to several reasons: (1) M is designed based on the nominal model $\hat{P}$, and the modeling uncertainties inevitably exist between the nominal model and the actual model, and hence, M will not correctly inverse P; (2) the inversion-based design M can introduce delays to the system if $\hat{P}$ is a strict causal system, because $M = \hat{P}^{-1}$ is non-causal and needs to be modified to be causal; (3) the input signals of the DOB can be noisy and a non-unit Q is necessary to filtrate the noises. These design compromises can lead to an undesired estimate $\hat{d}_a$. On the other hand, demanding disturbance suppression performance requires an accurate model and extensive tuning efforts for C, M, Q, which is usually formidable, and for some systems, tuning the C, M, Q parameters is even impossible. Therefore, the disturbance estimation $\hat{d}_o$ (no $\hat{d}_l$ added) is subject to errors caused by not-well tuned controllers and inaccurate DOB designs. To improve the estimate, a learning signal compensated estimation mechanism is designed to improve the estimate $\hat{d}_o$. The learning signal is generated offline, and the learning filter can be non-causal system which avoids introducing delays to the signals. Essentially the learning mechanism is a feedforward method, and it does not necessarily require very accurate C, M, Q parameters since the learning signal aims to compensate for the performance error caused by those parameter inaccuracies.

2.3. Stability and robustness analysis

In this section, we will theoretically analyze how the DOB affects the closed-loop stability and robustness to modeling uncertainties. For better reading, we present it in the theory-proof format.

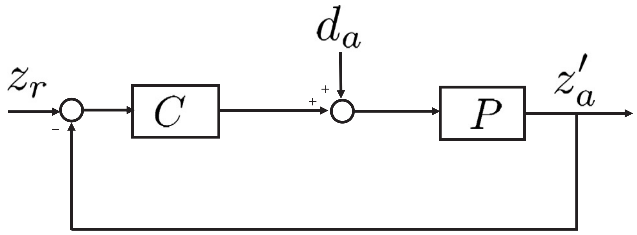


Fig. 3. Original closed-loop system without DOB.

Theory: The proposed hybrid DOB in Fig. 2 (1) does not affect the stability of the original closed-loop system which is without DOB, and (2) has proved robustness to a bounded modeling uncertainties if the following three conditions hold.

- The nominal model $\hat{P}$ is minimum phase.
- The original system without DOB (Fig. 6) is stable.
- The Q-filter is designed such that $\Delta Q = 0$, where Δ is the unmodeled dynamics which is stable and bounded by 1, and

$$P = \hat{P}(1 + \Delta) \tag{1}$$

Proof. We first prove that the DOB is an add-on to the control system and does not affect the original system stability. Considering that the feedforward signal in the hybrid DOB, i.e., $\hat{d}_l$, does not affect the stability and robustness of the feedback system, to shorten the notations, we remove it during our analysis without loss of generality. With the actual system diagram in Fig. 2, the dynamics can be derived as follows:

$$z_a = P\{d_a + u_{a2}\} \tag{2}$$

$$u_{a1} = C\{z_r - z_a\} \tag{3}$$

$$\hat{d}_a = \hat{d}_o = M\{z_a\} - Q\{u_{a2}\} \tag{4}$$

$$u_{a2} = u_{a1} - \hat{d}_a. \tag{5}$$

Plug (3) and (4) into (5) to have

$$u_{a2} = (1 - Q)^{-1}(C\{z_r - z_a\} - M\{z_a\}) \tag{6}$$

and plug (6) into (2) and re-organize the equation to have

$$z_a = [1 - Q + P(C + M)]^{-1}[(1 - Q)P\{d_a\} + PC\{z_r\}]. \tag{7}$$

Design

$$M = Q\hat{P}^{-1} \tag{8}$$

and plug (8) and (1) into (7) to have

$$z_a = [1 + \Delta Q + PC]^{-1}[(1 - Q)P\{d_a\} + PC\{z_r\}]. \tag{9}$$

For the actual system in Fig. 2, if the DOB is removed, the closed-loop system dynamics with the same controller C, plant P, inputs z_r and d_a is given in Fig. 3, where z_a' is the corresponding output. It is straightforward to have

$$z_a' = (1 + PC)^{-1}(P\{d_a\} + PC\{z_r\}). \tag{10}$$

Therefore, if the condition $\Delta Q = 0$ is satisfied, system (9) and (10) share the same characteristic equation which is related to $1 + PC = 0$. This indicates that the DOB will not affect the original system stability if the mentioned conditions are satisfied.

Proof. Secondly, we will prove that the proposed DOB is robust to a bounded modeling uncertainty. To investigate the robustness, without loss of generality, we remove the external input signals to the actual system in Fig. 2, resulting an equivalent closed-loop system with DOB incorporated given in Fig. 4, where u_{a3} is the output of Δ.

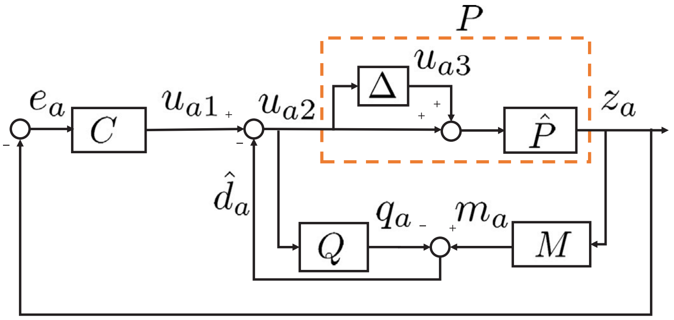


Fig. 4. Closed-loop system with DOB incorporated.

With Fig. 4 we have

$$z_a = \hat{P}\{u_{a2} + u_{a3}\} \tag{11}$$

and

$$\hat{d}_a = m_a - q_a = M\{z_a\} - Q\{u_{a2}\}. \tag{12}$$

Plug (8) and (11) into (12) to have

$$\begin{aligned}\hat{d}_a &= M\{z_a\} - Q\{u_{a2}\}\\ &= Q\hat{P}^{-1}\hat{P}\{u_{a2} + u_{a3}\} - Q\{u_{a2}\}\\ &= Q\{u_{a3}\}.\end{aligned} \tag{13}$$

Then

$$u_{a1} = C\{e_a\} = C\{0 - z_a\} \tag{14}$$

and with (14), (11), and (13), we have

$$\begin{aligned}u_{a2} &= u_{a1} - \hat{d}_a\\ &= -C\{z_a\} - Q\{u_{a3}\}\\ &= -C\hat{P}\{u_{a2} + u_{a3}\} - Q\{u_{a3}\}\end{aligned} \tag{15}$$

and

$$u_{a2} = H\{u_{a3}\} \tag{16}$$

where

$$H = -(Q + C\hat{P})(1 + C\hat{P})^{-1}. \tag{17}$$

From now on the input and output of Δ is related by H. If the Δ block is removed from the original closed-loop system in Fig. 4, an equivalent simplified closed-loop system can be constructed as given in Fig. 5, where u_{a3} and u_{a2} serves as the output and input of Δ, and H is placed into the feedback loop. Based on the small gain theory [27], the closed-loop system in Fig. 5 is robust if the above three conditions are satisfied.

In this study, the nominal model $\hat{P}$ is minimum phase, and a controller is designed to stabilize the system in Fig. 6. The third condition shows that $\Delta Q = 0$ for all frequency range. This assumes that the Δ has gains close to 0 at low-frequency range, and in this study, Q parameter is designed as a low-pass filter with gains close to 0 at high-frequency range to satisfy the third condition. Therefore, the proposed method can guarantee the system stability and robustness to the bounded modeling uncertainty.

3. Optimization-based learning filter design

In this section, we present details on how to systematically design the learning filter with theoretically proved improvements. To do so, we will first augment a system which includes several components to a unified state-space realization; then based on this, we will decouple the to-be-designed learning filter from the augmented system; thirdly, the learning filter design problem is re-formulated into a feedback controller design problem, which can be further formulated into an optimization problem. We will introduce these procedures step by step in the following subsections.

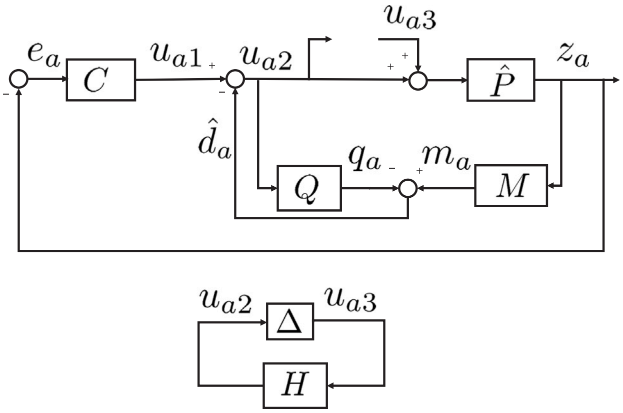


Fig. 5. Equivalent simplified closed-loop system.

3.1. System augmentation

In this subsection, derivations are presented to relate e_a and e_p in Fig. 2. The drone's closed-loop dynamics of the position tracking in the vertical direction is of the interest in this study, and due to that the bandwidth of the attitude control loop is much higher than that of the position control loop, the position-loop dynamics is approximated as an LTI system $\hat{P}$. As mentioned above, the simulated system is designed in the way to replicate the actual system for the case when no learning signal is used. The disturbance d_p in the simulated system is a prediction of the actual disturbance d_a. Though the prediction error naturally exists, it is reasonable to assume that both the modeling and the prediction is accurate for the learning filter design and derivation purposes. That is, without loss of generality, the following assumptions are used just for the learning filter design and derivation purposes.

Assumption 1. $d_a = d_p$ & $\hat{P} = P$

Denote systems $\hat{P}$, M, Q, C, L with the following forms

$$\begin{aligned}&\hat{P} \sim \left[\begin{array}{c|c} A_P & B_P \\ \hline C_P & 0 \end{array}\right] \quad M \sim \left[\begin{array}{c|c} A_M & B_M \\ \hline C_M & D_M \end{array}\right] \quad Q \sim \left[\begin{array}{c|c} A_Q & B_Q \\ \hline C_Q & 0 \end{array}\right]\\ &C \sim \left[\begin{array}{c|c} A_C & B_C \\ \hline C_C & D_C \end{array}\right] \quad L \sim \left[\begin{array}{c|c} A_L & B_L \\ \hline C_L & D_L \end{array}\right]\end{aligned} \tag{18}$$

where '$\sim$' means denoted by, and $A_{\{\cdot\}}$, $B_{\{\cdot\}}$, $C_{\{\cdot\}}$, and $D_{\{\cdot\}}$ are state matrices, input matrices, output matrices, and feedforward matrices, respectively.

For the simulated system, denote x_{P1}, x_{M1}, x_{Q1}, x_{C1} as the state variables of the system $\hat{P}$, M, Q, C, respectively; denote k as the discrete-time index, and the following state-space realizations are listed:

$$\begin{aligned}\hat{P}:\quad & x_{P1}(k+1) = A_P x_{P1}(k) + B_P(u_{p2}(k) + d_p(k))\\ & z_p(k) = C_P x_{P1}(k)\\ M:\quad & x_{M1}(k+1) = A_M x_{M1}(k) + B_M z_p(k)\\ & m_p(k) = C_M x_{M1}(k) + D_M z_p(k)\\ Q:\quad & x_{Q1}(k+1) = A_Q x_{Q1}(k) + B_Q u_{p2}(k)\\ & q_p(k) = C_Q x_{Q1}(k)\\ C:\quad & x_{C1}(k+1) = A_C x_{C1}(k) + B_C e_p(k)\\ & u_{p1} = C_C x_{C1}(k) + D_C e_p(k).\end{aligned} \tag{19}$$

where $e_p = z_r - z_p$. For the actual system, denote $x_{Pa}, x_{Ma}, x_{Qa}, x_{Ca}, x_L$ as the state variables of the system P, M, Q, C, L respectively, and the following state-space realizations are listed:

$$\begin{aligned}
P:\ & x_{Pa}(k+1) = A_P x_{Pa}(k) + B_P(u_{a2}(k) + d_a(k))\\
& z_a(k) = C_P x_{Pa}(k)\\
M:\ & x_{Ma}(k+1) = A_M x_{Ma}(k) + B_M z_a(k)\\
& m_a(k) = C_M x_{Ma}(k) + D_M z_a(k)\\
Q:\ & x_{Qa}(k+1) = A_Q x_{Qa}(k) + B_Q u_{a2}(k)\\
& q_a(k) = C_Q x_{Qa}(k)\\
C:\ & x_{Ca}(k+1) = A_C x_{Ca}(k) + B_C e_a(k)\\
& u_{a1} = C_C x_{Ca}(k) + D_C e_a(k)\\
L:\ & x_L(k+1) = A_L x_L(k) + B_L e_p(k)\\
& \hat{d}_l(k) = C_L x_L(k) + D_L e_p(k).
\end{aligned} \tag{20}$$

where $e_a = z_r - z_a$. In order to relate e_a and e_p, we first define new variables $\tilde{x}_P = x_{Pa} - x_{P1}$, $\tilde{x}_M = x_{Ma} - x_{M1}$, $\tilde{x}_Q = x_{Qa} - x_{Q1}$, $\tilde{x}_C = x_{Ca} - x_{C1}$, and then we have

$$e_a(k) - e_p(k) = -(z_a(k) - z_p(k)) = -C_P \tilde{x}_P(k) \tag{21}$$

and

$$\begin{aligned}
u_{a2}(k) - u_{p2}(k) =& [u_{a1}(k) - (m_a(k) - q_a(k) + \hat{d}_l(k))]\\
&- [u_{p1} - (m_p(k) - q_p(k))]\\
=&(u_{a1}(k) - u_{p1}(k)) - (m_a(k) - m_p(k))\\
&+ (q_a(k) - q_p(k)) - \hat{d}_l(k)\\
=&(C_C\tilde{x}_C(k) + D_C(e_a(k) - e_p(k)))\\
&- (C_M\tilde{x}_M(k) + D_M(z_a(k) - z_p(k)))\\
&+ (C_Q\tilde{x}_Q(k)) - (C_L x_L(k) + D_L e_p(k)).
\end{aligned} \tag{22}$$

Next, a new dynamic system T is constructed to have $e_a = T\{e_p\}$, that is, with an input e_p, the output of T is e_a. Choose e_p and e_a as the input and output of T, and denote $\mathbf{x}_T$ as the state variables of T, where $\mathbf{x}_T = [\tilde{x}_P,\ \tilde{x}_M,\ \tilde{x}_Q,\ \tilde{x}_C,\ x_L]^T$. With (19), (20) and some basic mathematical operations, the following equations can be written

$$\begin{aligned}
&\tilde{x}_P(k+1) = A_P\tilde{x}_P(k) + B_P(u_{a2}(k) - u_{p2}(k))\\
&\tilde{x}_M(k+1) = A_M\tilde{x}_M(k) + B_M(z_a(k) - z_p(k))\\
&\tilde{x}_Q(k+1) = A_Q\tilde{x}_Q(k) + B_Q(u_{a2}(k) - u_{p2}(k))\\
&\tilde{x}_C(k+1) = A_C\tilde{x}_C(k) + B_C(e_a(k) - e_p(k))\\
&x_L(k+1) = A_L x_L(k) + B_L e_p(k).
\end{aligned} \tag{23}$$

Denote A_T, B_T, C_T, D_T as the system matrix, input matrix, output matrix, and feedforward matrix of T. Then by plugging (21) and (22) into (23), the state-space realization of T is given in (24), where $D_T = 1$.

$$\begin{aligned}
&\mathbf{x}_T(k+1)\\
&= \underbrace{\begin{bmatrix} A_P - B_PC_P(D_C + D_M) & -B_PC_M & B_PC_Q & B_PC_C & -B_PC_L\\ B_MC_P & A_M & 0 & 0 & 0\\ -B_QC_P(D_C + D_M) & -B_QC_M & A_Q + B_QC_Q & B_QC_C & -B_QC_L\\ -B_CC_P & 0 & 0 & A_C & 0\\ 0 & 0 & 0 & 0 & A_L \end{bmatrix}}_{A_T} \mathbf{x}_T(k)\\
&+ \underbrace{\begin{bmatrix} -B_PD_L\\ 0\\ -B_QD_L\\ 0\\ B_L \end{bmatrix}}_{B_T} e_p(k)\\
&e_a(k) = \underbrace{[-C_P\ 0\ 0\ 0\ 0]}_{C_T}\mathbf{x}_T(k) + e_p(k).
\end{aligned} \tag{24}$$

From now on, the augment system T which relates e_p and e_a is constructed.

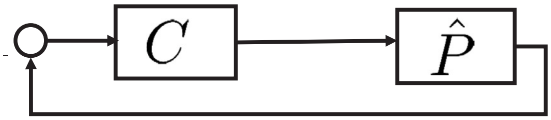


Fig. 6. Nominal closed-loop system without DOB.

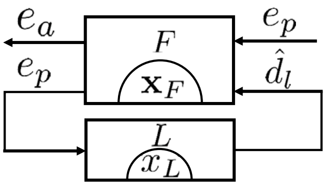


Fig. 7. Constructed system. The to-be-designed L is placed into the feedback loop. The closed-loop system from e_p to e_a is T.

3.2. Decoupling of the learning filter

Eq. (24) indicates that T lumps $\hat{P}$, M, Q, C, L. To better formulate the learning filter design problem, this subsection decouples the unknown dynamics L from the known dynamics $\hat{P}$, M, Q, and C.

Denote a new state variable $\mathbf{x}_F = [\tilde{x}_P,\ \tilde{x}_M,\ \tilde{x}_Q,\ \tilde{x}_C]^T$, and with (22) and (23), the state-space realization in (25) can be written,

$$\begin{aligned}
\mathbf{x}_F(k+1) =& \underbrace{\begin{bmatrix} A_P - B_PC_P(D_C + D_M) & -B_PC_M & B_PC_Q & B_PC_C\\ B_MC_P & A_M & 0 & 0\\ -B_QC_P(D_C + D_M) & -B_QC_M & A_Q + B_QC_Q & B_QC_C\\ -B_CC_P & 0 & 0 & A_C \end{bmatrix}}_{A_F} \mathbf{x}_F(k)\\
&+ \underbrace{\begin{bmatrix} 0\\ 0\\ 0\\ 0 \end{bmatrix}}_{B_{F_1}} e_p(k) + \underbrace{\begin{bmatrix} -B_P\\ 0\\ -B_Q\\ 0 \end{bmatrix}}_{B_{F_2}} \hat{d}_l(k)\\
e_a(k) =& \underbrace{[-C_p\ 0\ 0\ 0]}_{C_{F_1}}\mathbf{x}_F(k) + e_p(k) + 0\cdot\hat{d}_l(k)\\
e_p(k) =& \underbrace{[0\ 0\ 0\ 0]}_{C_{F_2}}\mathbf{x}_F(k) + e_p(k) + 0\cdot\hat{d}_l(k).
\end{aligned} \tag{25}$$

where a new defined system F is

$$F \sim \left[\begin{array}{c|cc} A_F & B_{F_1} & B_{F_2}\\ \hline C_{F_1} & D_{F_{11}} & D_{F_{12}}\\ C_{F_2} & D_{F_{21}} & D_{F_{22}} \end{array}\right] = \left[\begin{array}{c|cc} A_F & B_{F_1} & B_{F_2}\\ \hline C_{F_1} & 1 & 0\\ C_{F_2} & 1 & 0 \end{array}\right] \tag{26}$$

and (25) indicates that system F has two inputs e_p, $\hat{d}_l$ and two outputs e_a, e_p, and the closed-loop system from e_p to e_a remains as T. Therefore, the learning filter L can be decoupled from T, as shown in Fig. 7, where it shows that L is placed into the feedback loop of the system, and F only lumps the known dynamics $\hat{P}$, M, Q, C.

3.3. Optimization-based design principle and practical design guideline

A design guideline for the learning filter is to have $\|e_a\|_2 \leq \|e_p\|_2$, and this can be achieved by reaching a minimum norm of T, which is equivalent to reach

$$\begin{aligned}
&\min_{L}\ \gamma\\
&\text{s.t. } |\lambda_i(A_T)| < 1,\ \forall\ i\\
&\qquad \bar{\sigma}\{D_T + C_T(\eta I - A_T)^{-1}B_T\} < \gamma,\ \forall\ |\eta| > 1
\end{aligned} \tag{27}$$

where $\lambda_i(A_T)$ is the ith eigenvalue of A_T, $\bar{\sigma}\{\cdot\}$ denotes the maximum singular value of a matrix, γ is a real number, and I is a identity matrix. Further, (27) can be transferred to the following convex optimization problem [28]

$$\min_{R,\ S\in R^{n\times n},\ \gamma}\quad \gamma \tag{28a}$$

$$\begin{bmatrix} \mathbb{L}_1(R,\gamma) & 0 & 0 \\ 0 & \mathbb{L}_2(S,\gamma) & 0 \\ 0 & 0 & -\mathbb{L}_3(R,S)-\delta I \end{bmatrix} < 0, \tag{28b}$$

where δ is small and positive, and

$$L_1(R,\gamma) =$$

$$\left[\begin{array}{c|c} N_R & \mathbf{0} \\ \hline \mathbf{0} & I \end{array}\right]^T \left[\begin{array}{cc|c} A_F R + R A_F^T & R C_{F1}^T & B_{F1} \\ C_{F1} R & -\gamma I & D_{F11} \\ \hline B_{F1}^T & D_{F11}^T & -\gamma I \end{array}\right] \left[\begin{array}{c|c} N_R & \mathbf{0} \\ \hline \mathbf{0} & I \end{array}\right] < 0 \tag{29}$$

$$L_2(S,\gamma) =$$

$$\left[\begin{array}{c|c} N_S & \mathbf{0} \\ \hline \mathbf{0} & I \end{array}\right]^T \left[\begin{array}{cc|c} A_F S + S A_F^T & S B_{F1}^T & C_{F1}^T \\ B_{F1}^T S & -\gamma I & D_{F11}^T \\ \hline C_{F1} & D_{F11} & -\gamma I \end{array}\right] \left[\begin{array}{c|c} N_S & 0 \\ \hline 0 & I \end{array}\right] < 0 \tag{30}$$

$$L_3(R,S) = \begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \tag{31}$$

where N_R and N_S denote the bases of the null spaces of $(B_{F2}^T,\ D_{F12}^T)$ and $(C_{F2}^T,\ D_{F21}^T)$, respectively. The system F is obtained as shown in (26), and therefore N_R and N_S are determined. *Matlab* software can be used to efficiently search suitable symmetric matrices R and S which satisfy (29)~(31) to solve the problem in (28). The feedback controller L can be constructed with the following steps. After R and S are obtained, compute two full-column-rank matrices U and V which satisfy

$$UV^T = I - RS \tag{32}$$

and then an adequate X_{cl} can be obtained as the unique solution of the following equations

$$\left[\begin{array}{c|c} S & I \\ \hline V^T & 0 \end{array}\right] = X_{cl} \left[\begin{array}{c|c} I & R \\ \hline 0 & U^T \end{array}\right]. \tag{33}$$

Then the positive definite matrix X_{cl} can be used to form a Bounded Real Lemma inequality, and by solving this inequality a solution $[A_L,\ B_L;\ C_L,\ D_L]$ can be obtained to construct the learning filter L. More details can be referred to the chapter 7 in [28].

Note that the optimization-based learning filter design is not new, the novelty here is to provide a systematic way to first dynamically connect the two errors e_a and e_p and then formulates the design problem into the optimization problem.

4. Validation

This section demonstrates the validation of the proposed hybrid DOB. A drone platform is first built, and then two testing scenarios are carried out with results discussed in detail.

4.1. Platform and parameters setup

As shown in Fig. 8, the test platform mainly includes an assembled quadrotor drone with its specifications given in Table 1. Reflective markers are attached to the drone, and the VICON motion capture system with infrared cameras are used to track the reflective markers to estimate the position and orientation of the drone. In the test, a bottle of water is used as the payload and it is rigidly hooked under the drone, and there is no swing during the test. The gravity center of the drone and the payload is nearly aligned in the vertical direction, and only the disturbance in the vertical direction is considered.

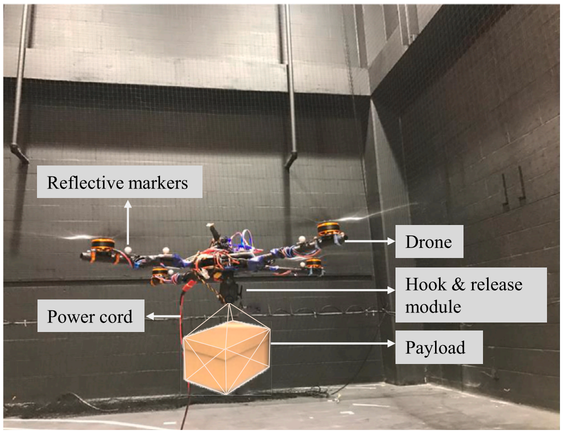


Fig. 8. Experimental platform: the drone is connected with a thin light power cable and its influence to the results are negligible. The cable is placed at the same position for each test. A bottle of water is used as the payload in the real test and the payload weight is changed by adding/reducing water in the bottle.

The electronics equip the drone system to receive the data from the VICON system, execute control commands including driving the rotor and releasing the servo, and store the flight data in real time. The VICON system outputs high-accuracy data at a frequency of 100 hz. The hook and release device is mounted at the bottom of the drone. A motor is controlled to hook and release the payload.

The quadrotor is a highly nonlinear system [29] and robust controllers or adaptive controllers are able to stabilize the drone, but it requires accurate models and extensive system identification efforts. Also, tuning the feedback controller is not feasible for system whose baseline controller is not allowed to be modified. On the other hand, there existing considerable amount of work in the tracking control of drones which uses PID controllers that are designed based on a linearized model around hover conditions [30–33]. When the roll and pitch angles are reasonably small, a standard PID controller usually performs adequately. In our case, a PID controller is designed to stabilize the drone in the first place. A very accurate model and PID controller are not required. Instead, the generated learning signal is able to compensate for those inaccuracies, which provides an alternative way to handle this problem.

In this study, the drone plant absorbs the baseline controller and the closed-loop dynamics of the position tracking is identified as an LTI system $\hat{P}$ using recursive least square-based algorithm [34]. Basic DOB parameters Q and M are implemented for the drone, where M aims to inverse $\hat{P}$, and Q is a low pass filter. Similarly, extensive tuning work for the M and Q parameters are unnecessary. To provide some insights of the dynamic parameters, the transfer functions and bode plot tool is used to show the frequency properties of the dynamics parameters. The transfer functions of M, Q, P, L, and T are given as

$$\begin{aligned} M(z^{-1}) &= \frac{1 - 1.408z^{-1} + 0.4215z^{-2}}{0.4958z^{-1} - 0.4827z^{-2}} \\ Q(z^{-1}) &= \frac{0.4425}{z - 0.3679} \\ \hat{P}(z^{-1}) &= \frac{0.4958z^{-1} - 0.4827z^{-2}}{1 - 1.408z^{-1} + 0.4215z^{-2}} \\ L(z^{-1}) &= \frac{-0.6953 + 0.9792z^{-1} - 0.2931z^{-2}}{0.4958 - 0.486z^{-1} + 0.003252z^{-2}} \\ T(z^{-1}) &= \frac{1 - 2.953z^{-1} + 3.055z^{-2} - 1.248z^{-3} + 0.1462z^{-4}}{1 - 2.605z^{-1} + 2.228z^{-2} - 0.6356z^{-3} + 0.01215z^{-4}} \end{aligned} \tag{34}$$

where z^{-1} is the discrete-time operator. Note that the original T is a 12th order system, whose order can be equivalently reduced to 4th order, and this will not affect the design of the L. The bode plots of M and Q are given in Fig. 9, where it shows that Q has a cutoff frequency of 10 hz, and the magnitude at the low-frequency range has been reduced below 0 db to lower the DOB's sensitivity to noises in practice. The bode plots of $\hat{P}$, L, and T are given in Fig. 10, and it shows that at the low-frequency range (< 6.4 hz), the magnitude of T

Table 1
Main specifications of the assembled drone.

Frame brand	Tarot 650
Wheelbase diameter	650 mm
Motor	JMT MT3508 brushless motor
Electronic speed controller	Readttosky (30A)
Propeller brand	QWINOUT
Propeller size	length: 15 inch, pitch: 5.5
Flight controller	Pixhawk
Companion computer	Raspberry pi
Payload releasing servo brand	Hitech
PWM servo motor driver brand	MakerFocus
Drone weight without payload	1350 g

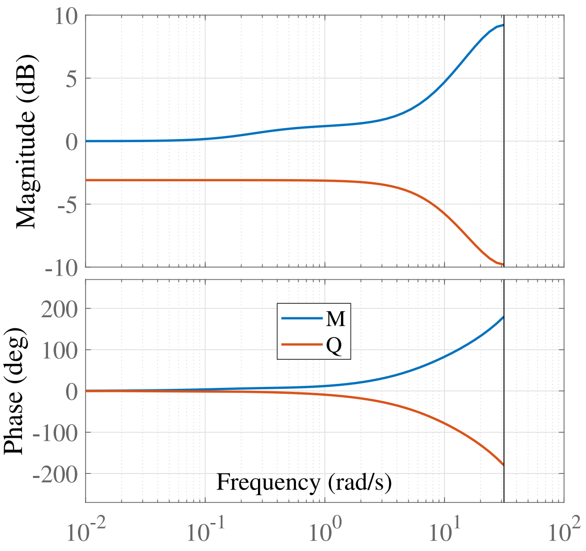


Fig. 9. Bode plots of M and Q.

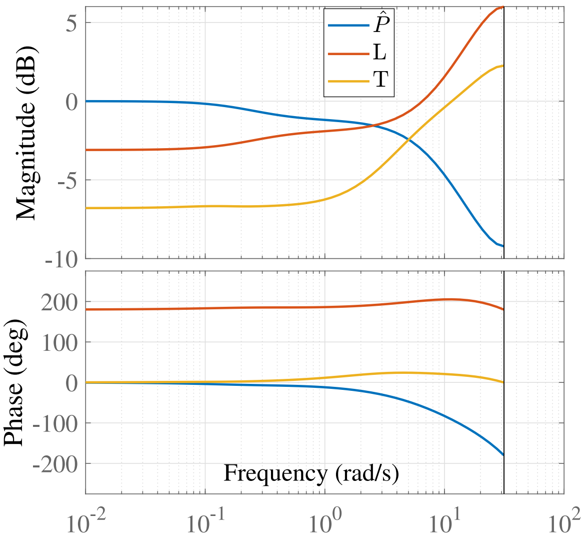


Fig. 10. Bode plots of $\hat{P}$, L and T.

is smaller than 1, and this is equivalent to meet the condition (27) in the time domain analysis. The learning filter L has a very resembling magnitude curve as the M (the two curves are not identical in shape), and this indicates that the optimization solver generates a solution L which mainly based on the inversion of $\hat{P}$.

The prediction d_p in the simulated system is obtained prior to the real test. The neural network model is leveraged as a standard tool to

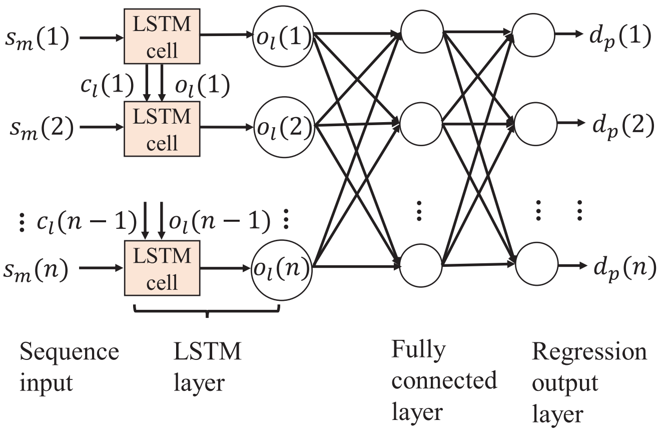


Fig. 11. LSTM network structure.

have a basic prediction. Consider that the signals used in the control system are time series, we select the long short-term memory (LSTM) neural network model to have regression due to that LSTM can learn the long term dependencies of the data. The LSTM model maps from the payload weight to the equivalent disturbance and the model structure is given in Fig. 11, where the network model includes a sequence input layer, an LSTM layer, a fully connected layer, and a regression output layer; with sequence length n, $[s_m(1), s_m(2), \cdots, s_m(n)]^T$ denotes the input sequence, $[d_p(1), d_p(2), \ldots, d_p(n)]^T$ denotes the output sequence; the LSTM layer includes n LSTM cells which the cell structure can be referred to [26], and $[c_l(1), c_l(2), \ldots, c_l(n-1)]^T$ denotes the cell state, and $[o_l(1), o_l(2), \ldots, o_l(n-1)]^T$ denotes the cell output. The LSTM cells are nonlinear and chain-connected such that the LSTM layer is able to learn nonlinear and long-term dependencies. The payload weight information is discretized into a time series and is used as the input of the network model.

To train the neural network model, the training dataset is collected with the following steps: (1) the drone takes off from the ground carrying a payload with mass m, and then it hovers at an altitude of 1 meter; (2) then the drone drops the payload, and the altitude variation in the output channel around the hovering point is recorded for a defined time duration, and the data will be used as the model's output (label) of the training; (3) the scalar m is manually transformed into a time series and the data will be used as the model's input (feature) of the training. By using payload with different weight, 85 data samples in total are collected. Consider that the network aims to predict the disturbance only in the vertical direction, and the drone's payload capacity is limited to 190 grams, a small dataset size is regarded as sufficient to train a neural network model for the prediction task.

In the training, 80% of the data is used as the training dataset and the remaining as the validating dataset. The batch gradient descent algorithm [35] is used with 'Adam' optimizer. The learning rate is set to be 0.01 without a learning decay; the gradient threshold is set to 1 to prevent gradients exploding. The training process is given in Fig. 12, where RMSE means the root mean square error. It shows that the loss reduces quickly in the beginning and then converges. Note that the up and downs within the first 10 iterations could be related to over learning or the model's non-convexity property, and similar loss trend can be observed in the figure 8 in paper [36]. In batch Gradient Descent, one iteration utilizes all the dataset for training, and the average of the gradients of all the training examples is calculated and used to update the weight parameters and bias. The training process shows that the loss converges after 20 iterations and the RMSE value is kept small, which shows that the weight parameters are suitable after 20 iterations. Though RMSE is used as the performance metric for the regression, the model mismatch between the NN and the actual

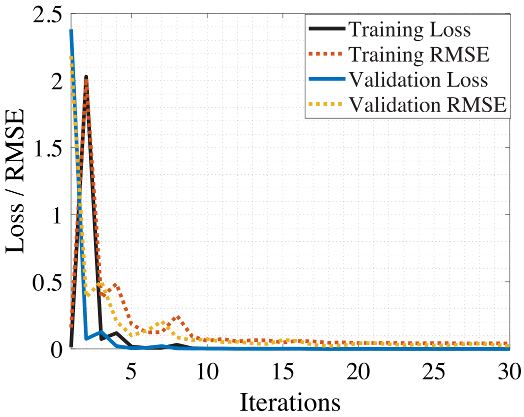


Fig. 12. LSTM network training process.

dynamics is not quantified and discussed in this study. Moreover, this paper intends not to use large datasets to train a perfect NN with great efforts, instead, we show that the proposed method can achieve desired performance with network model uncertainties exist.

4.2. Experimental tests

Two test scenarios are carried out. In scenario 1, the carried payload weight is 95 grams; in scenario 2, the carried payload weight is 145 grams and the other settings are the same as in scenario 1. For each scenario, the drone first takes off carrying the payload and reaches the altitude of 1 meter; then it hovers for a while and starts to release the payload, and at the same time, the drone will oscillate and the proposed hybrid DOB initiates to suppress the oscillation. The learning signal will be generated before the dropping action takes place, and the DOB is running in real-time. The detailed implementation procedure of the proposed DOB can be referred to Algorithm 1.

```
Algorithm 1 Hybrid DOB
Inputs:
1. Drone model $\hat{P}$, learning filter $L$, standard DOB ($M$ and $Q$);
2. LSTM network;
Initialization: Set the hovering reference $z_r = 1$ meter for the drone
if the payload is not yet dropped then
    1. with the payload weight value $m$, the disturbance prediction ($d_p$) is generated;
    2. then $d_p$ is added to the simulated system to generate the tracking error $e_p$;
    3. then $e_p$ goes through the learning filter $L$ and the learning signal $\hat{d}_l$ is generated and sent to the drone;
else
    1. The $\hat{d}_l$ is added to the estimate $\hat{d}_o$ at the instance when the payload dropping command is received;
    2. The flying data during the payload dropping process is recorded for the designed flying duration.
end
```

A flying duration of 7.7 s is recorded after the payload dropping command is received, and the flight data are saved on board for analysis. To show the effectiveness of the proposed DOB, in each scenario, the drone will repeat the payload dropping operation (1) without using a DOB, (2) with using a standard DOB, (3) with using the hybrid DOB, and the results are recorded for comparison purposes. Note that the DOB parameters M and Q are the same in the standard DOB and hybrid DOB design.

Scenario 1: payload is 95 g: With the payload weight, the NN predicts the input disturbance, and the prediction goes through the simulated system to generate the tracking error. With the learning filter, the learning signal is generated as shown in Fig. 15. The drone's oscillation variation without DOB, with a standard DOB, and with the hybrid DOB are given in Fig. 13. The 2-norm of the altitude variation from reference in the 3 cases are 0.8984, 0.2574, 0.1468, respectively. The results show that without a DOB, the oscillation caused by the dropping action is large, and a standard DOB can suppress the oscillations tremendously, where the hybrid DOB can further reduce the oscillations to a certain degree. The predicted disturbance d_p from the NN, along with the estimated disturbance $\hat{d}_p$ and $\hat{d}_a$ are given in Fig. 16. Though prediction error could exist and has not been quantified, the proposed method is expected to handle a certain amount of uncertainties as explained above.

Scenario 2: payload is 145 g: With this larger payload weight, the learning signal is generated as shown in Fig. 15. The drone's oscillation variation without DOB, with a standard DOB, and with the hybrid DOB are given in Fig. 14. The 2-norm of the altitude variation from reference in the 3 cases are 1.3410, 0.2742, 0.1971, respectively. The results indicate that similar conclusions as in scenario 1 can be made and the proposed DOB is effective to reject disturbance with larger quantities. The predicted disturbance d_p from the NN, along with the estimated disturbance $\hat{d}_p$ and $\hat{d}_a$ are given in Fig. 16.

Compared to the system performance of the standard DOB, the hybrid DOB is better, and the 2-norm of the tracking error is reduced by 43% in scenario 1 and 28% in scenario 2 over the standard DOB. Figs. 13 and 14 indicate that without a DOB, it would take a very long time to reach the steady state. Longer time frame may not be able to compare the transient performances obviously in the figures due to large scale of the x-axis, and here only 7.7 s duration is provided. To provide some insights of the steady state error, in scenario 1, the steady state errors of the one without DOB, the one with standard DOB, the one with the hybrid DOB are 0.08, 0.02, 0.02, respectively. In scenario 2, the steady state errors of the one without DOB, the one with standard DOB, the one with the hybrid DOB are 0.10, 0.02, 0.02, respectively. The advantages of the hybrid DOB over the standard DOB is twofold: (1) the hybrid DOB has better disturbance suppression performance over the standard DOB regarding the transient performance; (2) the hybrid DOB provides more design flexibility to C, M, Q parameters, that is, in the cases that the baseline controller C and the DOB parameters Q, M are not well designed, the hybrid DOB serves as a feasible method to improve the system performance without tuning the baseline controller and DOB parameters, which this is useful for systems that tuning is not convenient. Note that the drone prototype developed has a physical payload limitation, but it is suitable as a platform to validate the proposed DOB algorithm and the theoretical contributions. However, this algorithm is adaptive to real delivery drone applications with larger payload capacities. Another point to make is that most systems can be linearized at equilibrium points and therefore the proposed method can be applied. However, the proposed method has limitations when it comes to time-variant systems.

5. Conclusions

Regarding the delivery drone's payload dropping scenarios, this paper proposes a new hybrid DOB to suppress its oscillations. The hybrid DOB leverages the neural network prediction and the dynamic filters to generate a learning signal to enhance the disturbance estimate. The method is tested with actual experimental tests and those results validate that the learning mechanism is effective to suppress oscillations for delivery drones. The proposed method can serve as an alternative DOB design which can provide more flexibility for baseline controller and DOB parameter design in specific applications. Note that we do not highlight the neural network part, but leverage it as a standard predicting technique. The proposed method is applicable to general and linearized LTI systems. The future work will explicitly explore the system robustness to predicting uncertainties, as well as

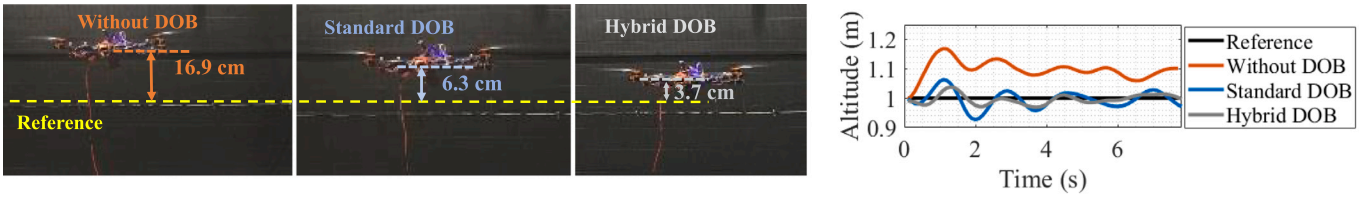


Fig. 13. Scenario 1: the maximum oscillation variation during the payload dropping process for the three control cases, and they are 16.9 cm, 6.3 cm, and 3.7 cm, respectively. 'Reference' in the figure stands for the hovering altitude before the dropping motion. The experimental video is available via this link.

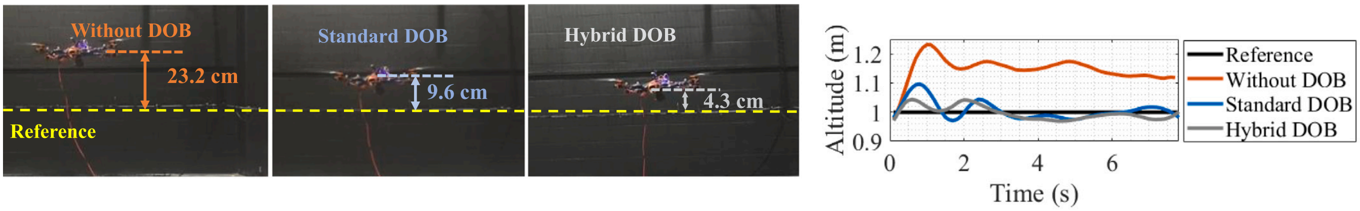


Fig. 14. Scenario 2: the maximum oscillation variation during the payload dropping process for the three control cases, and they are 23.2 cm, 9.6 cm, and 4.3 cm, respectively. 'Reference' in the figure stands for the hovering altitude before the dropping motion. The experimental video is available via this link.

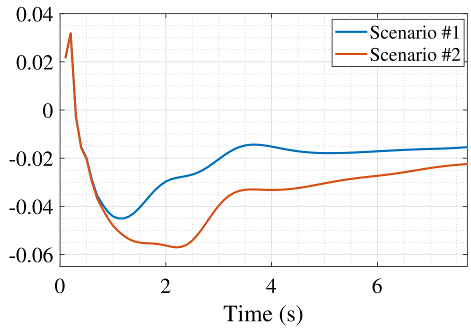


Fig. 15. Learning signals.

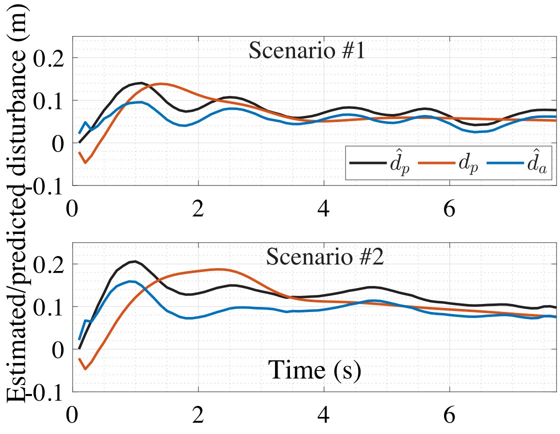


Fig. 16. Estimated/predicted disturbance in the two scenarios.

test the dropping scenarios using some commercial drones with larger payload limit.

CRediT authorship contribution statement

Zhu Chen: Conceptualization, Methodology, Algorithm development, Simulation, Experimental verification, Writing – original and revised manuscript. **Chang Liu:** Experimental verification. **Hao Su:** Conceptualization, Writing – review & editing. **Xiao Liang:** Supervision, Conceptualization, Writing – review & editing. **Minghui Zheng:** Supervision, Conceptualization, Methodology, Algorithm development, Simulation, Writing – review & editing.

Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.mechatronics.2022.102907. Co-author Minghui Zheng is currently serving as an associate editor of this journal.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.mechatronics.2022.102907.

References

[1] Zhou S, Zheng M, Chen X, Tomizuka M. Control of dual-stage hdds with enhanced repetitive disturbance rejection. In: ASME 2017 conference on information storage and processing systems. 58103, 2017, p. V001T03A002.

[2] Zhao Z, He X, Ahn CK. Boundary disturbance observer-based control of a vibrating single-link flexible manipulator. IEEE Trans Syst Man Cybern: Systems 2019.

[3] Chen J, Shuai Z, Zhang H, Zhao W. Path following control of autonomous four-wheel-independent-drive electric vehicles via second-order sliding mode and nonlinear disturbance observer techniques. IEEE Trans Ind Electron 2020;68(3):2460–9.

[4] Zhao Y, Cao Y, Fan Y. Disturbance observer-based attitude control for a quadrotor. In: 2017 4th international conference on information, cybernetics and computational social systems. IEEE; 2017, p. 355–60.

[5] Chen W-H, Yang J, Guo L, Li S. Disturbance-observer-based control and related methods—An overview. IEEE Trans Ind Electron 2015;63(2):1083–95.

[6] Seo H-T, Kim K-S, Kim S. Generalized design of disturbance observer for non-minimum phase system using an H-infinity approach. In: 2013 13th international conference on control, automation and systems. IEEE; 2013, p. 1–4.

[7] Lyu X, Zheng M, Zhang F. H-infinity based disturbance observer design for non-minimum phase systems with application to UAV attitude control. In: 2018 American control conference. IEEE; 2018, p. 3683–9.

[8] Zheng M, Zhou S, Tomizuka M. A design methodology for disturbance observer with application to precision motion control: An H-infinity based approach. In: 2017 American control conference. IEEE; 2017, p. 3524–9.

[9] Zhou S, Zheng M, Zhang F, Tomizuka M. Synthesized disturbance observer for vehicle lateral disturbance rejection. In: 2018 American control conference. IEEE; 2018, p. 398–403.

[10] Chen W-H. Disturbance observer based control for nonlinear systems. IEEE/ASME Trans Mechatronics 2004;9(4):706–10.

[11] Li J-F, Jahanshahi H, Kacar S, Chu Y-M, Gómez-Aguilar J, Alotaibi ND, et al. On the variable-order fractional memristor oscillator: Data security applications and synchronization using a type-2 fuzzy disturbance observer-based robust control. Chaos Solitons Fractals 2021;145:110681.

[12] Wang F, He L. FPGA-based predictive speed control for PMSM system using integral sliding-mode disturbance observer. IEEE Trans Ind Electron 2020;68(2):972–81.

[13] Sun T, Cheng L, Wang W, Pan Y. Semiglobal exponential control of Euler–Lagrange systems using a sliding-mode disturbance observer. Automatica 2020;112:108677.

[14] Wang X, Zhang G, Sun Y, Cao J, Wan L, Sheng M, et al. AUV near-wall-following control based on adaptive disturbance observer. Ocean Eng 2019;190:106429.

[15] Guerrero J, Torres J, Creuze V, Chemori A. Adaptive disturbance observer for trajectory tracking control of underwater vehicles. Ocean Eng 2020;200:107080.

[16] Zheng M, Lyu X, Liang X, Zhang F. A generalized design method for learning-based disturbance observer. IEEE/ASME Trans Mechatronics 2021;26(1):45–54.

[17] Li H, Wang Y, Shi Z, Chang M. An internal model frame-based disturbance attenuation control scheme for quad-rotors transporting unknown payloads. Trans Inst Meas Control 2019;41(14):3991–4000.

[18] Zhu L, Jiang C-s, Zhang C-y. Adaptive trajectory linearization control for aerospace vehicle based on RBFNN disturbance observer. Acta Aeronaut Astronaut Sinica-Seri A and B 2007;28(3):673.

[19] Wang D, Zong Q, Tian B, Shao S, Zhang X, Zhao X. Neural network disturbance observer-based distributed finite-time formation tracking control for multiple unmanned helicopters. ISA Trans 2018;73:208–26.

[20] He W, Sun Y, Yan Z, Yang C, Li Z, Kaynak O. Disturbance observer-based neural network control of cooperative multiple manipulators with input saturation. IEEE Trans Neural Netw Learn Syst 2019;31(5):1735–46.

[21] Zhao B, Xu S, Guo J, Jiang R, Zhou J. Integrated strapdown missile guidance and control based on neural network disturbance observer. Aerosp Sci Technol 2019;84:170–81.

[22] Wang T, Lu W, Yan Z, Liu D. Dob-Net: Actively rejecting unknown excessive time-varying disturbances. In: 2020 IEEE international conference on robotics and automation. IEEE; 2020, p. 1881–7.

[23] Min B-C, Hong J-H, Matson ET. Adaptive robust control (ARC) for an altitude control of a quadrotor type UAV carrying an unknown payloads. In: 2011 11th international conference on control, automation and systems. IEEE; 2011, p. 1147–51.

[24] Pratama GNP, Masngut I, Cahyadi AI, Herdjunanto S. Robustness of PD control for transporting quadrotor with payload uncertainties. In: 2017 3rd international conference on science and technology-computer. IEEE; 2017, p. 11–5.

[25] Matassini T, Shin H-S, Tsourdos A, Innocenti M. Adaptive control with neural networks-based disturbance observer for a spherical UAV. IFAC-PapersOnLine 2016;49(17):308–13.

[26] Chen Z, Liang X, Zheng M. Including image-based perception in disturbance observer for warehouse drones. In: Dynamic systems and control conference. 84287, American Society of Mechanical Engineers; 2020, V002T39A005.

[27] Liu K-Z, Yao Y. Robust control: theory and applications. John Wiley & Sons; 2016.

[28] Gahinet P, Apkarian P. A linear matrix inequality approach to H-infinity control. Internat J Robust Nonlinear Control 1994;4(4):421–48.

[29] Flores G, Lozano R. Lyapunov-based controller using singular perturbation theory: An application on a mini-UAV. In: 2013 American control conference. IEEE; 2013, p. 1596–601.

[30] Liang X, Zheng M, Zhang F. A scalable model-based learning algorithm with application to UAVs. IEEE Control Syst Lett 2018;2(4):839–44.

[31] Hoffmann G, Waslander S, Tomlin C. Quadrotor helicopter trajectory tracking control. In: AIAA guidance, navigation and control conference and exhibit. 2008, p. 7410.

[32] Mellinger D, Kumar V. Minimum snap trajectory generation and control for quadrotors. In: Robotics and automation (ICRA), 2011 IEEE international conference on. IEEE; 2011, p. 2520–5.

[33] Mellinger D, Lindsey Q, Shomin M, Kumar V. Design, modeling, estimation and control for aerial grasping and manipulation. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2011, p. 2668–73.

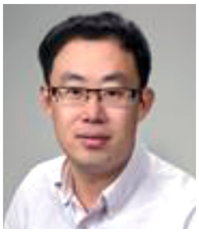
[34] Ding F, Wang X, Chen Q, Xiao Y. Recursive least squares parameter estimation for a class of output nonlinear systems based on the model decomposition. Circuits Systems Signal Process 2016;35(9):3323–38.

[35] Hinton G, Srivastava N, Swersky K. Neural networks for machine learning lecture 6A overview of mini-batch gradient descent. Cited on 2012;14(8).

[36] Wang F, Zai Y, Zhao J, Fang S. Field application of deep learning for flow rate prediction with downhole temperature and pressure. In: International petroleum technology conference. OnePetro; 2021.

Zhu Chen received the B.E. and M.S. degrees, in 2015 and 2018 respectively, from Huazhong University of Science and Technology, Wuhan, China. He is currently a Ph.D. student in the Department of Mechanical and Aerospace Engineering at University at Buffalo, NY, USA. His research interests include advanced learning and control with applications to robotic systems.

Chang Liu received the B.S. degree in Energy and Power Engineering, in 2018, from Huazhong University of Science and Technology, Wuhan, China, and the M.S. degree in Mechanical and Aerospace Engineering, in 2021, from the University at Buffalo, NY, USA. His research interests include planning, control, robotics, and deep learning.



Hao Su received the B.S. degree from the Harbin Institute of Technology, Harbin, China, the M.S. degree from the State University of New York University at Buffalo, Buffalo, NY, USA, and the Ph.D. degree from Worcester Polytechnic Institute, Worcester, MA, USA. He is an Associate Professor with the Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC, USA. He was an Irwin Zahn Endowed Assistant Professor with the City University of New York. He was a Research Scientist with Philips Research North America and then a Postdoctoral Fellow with Harvard University and Wyss Institute for Biologically Inspired Engineering. He holds patents on surgical robotics and socially assistive robots. Dr. Su was the recipient of the National Science Foundation CAREER Award, the National Institutes of Health R01 Award, the Best Student Paper Award of ASME Dynamic Systems Control Division Mechatronics Technical Committee, the Toyota Mobility Challenge Discover Award, the Best Medical Robotics Paper Runner-Up Award in the IEEE International Conference on Robotics and Automation (ICRA), and the Philips Innovation Transfer Award. He was also the recipient of the Advanced Simulation & Training Award from the Link Foundation and the Dr. Richard Schlesinger Award from the American Society for Quality. He serves as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS, ASME Journal of Mechanisms and Robotics, ICRA, and IEEE/RSJ International Conference on Intelligent Robots and Systems.

Xiao Liang received the B.S. degree from Hunan University, Changsha, China, in 2010, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA, in 2011 and 2016, respectively, all in civil engineering. He joined the Department of Civil, Structural and Environmental Engineering, University at Buffalo, NY, USA, in 2018. His research interests include health monitoring and autonomous inspection of infrastructure systems through advanced data analytics, model-based and machine learning.

Minghui Zheng received B.S. degree in engineering mechanics in 2008 and M.S. degree in control science and engineering in 2011, both from Beihang University, China, and the Ph.D. degree in mechanical engineering from University of California, Berkeley, CA, USA, in 2017. She joined the University at Buffalo, Buffalo, NY, USA, in 2017, where she is currently an Assistant Professor of mechanical and aerospace engineering. Her research interests include learning, planning, and control of multiple robotic systems, such as collaborative manipulators for remanufacturing and drones for disaster resilience. Dr. Zheng was the recipient of the NSF CAREER Award in 2021.