IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 November 2022, accepted 1 March 2023, date of publication 9 March 2023, date of current version 15 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3254540

RESEARCH ARTICLE

Real-Time Observability-Aware Inertia Parameter Estimation for Quadrotors

GEONWOO KWON, JAEGAK LEE, AND HYUNSEOK YANG
Department of Mechanical Engineering, Yonsei University, Yonsei, Seodaemun, Seoul 03722, South Korea
Corresponding author: Hyunseok Yang (hsyang@yonsei.ac.kr)

ABSTRACT This study focuses on the system identification of a quadrotor under an unknown payload with a time-optimal trajectory. A model-based control scheme should be utilized for a quadrotor, which does not necessarily guarantee robustness to model uncertainty. Thus, accurate system identification is required during flight. However, inertia parameter identification for the control scheme is vulnerable to sensor noise without a trajectory that produces rich data. We utilized Kalman filter (KF), which estimates the angular velocity, to reduce noise. In the process model of KF, the variance attributed to model uncertainty is derived, and the derived variance plays a pivotal role in adjusting Kalman gain. Recursive least squares (RLS) was utilized to identify the inertia parameter. However, all inertia parameters cannot always be observed with a time-optimal trajectory. Thus, this study proposes a criterion that distinguishes between observable and unobservable parameters and the correction law depending on the criteria. The correction law prevents RLS from correcting the unobservable parameters. We call this method the observability-aware RLS with KF. This study compared RLS, RLS with a low-pass filter (LPF), and observability-aware RLS with KF. While RLS with LPF shows a sharp increase or decrease in moment of inertia (MOI), center of mass (COM) offset, and sensor location, ours does not. RLS without any filtering has an inaccurate estimation of MOI and the height of the sensor location. However, our approach is sufficiently accurate to be applied to model-based control.

INDEX TERMS Inertia parameter estimation, Kalman filter, observability, recursive least square, unknown payload.

I. INTRODUCTION

In the last decade, quadrotors have become popular owing to their capability to hover in place and move freely in a 3-dimensional Euclidean space. Owing to this capability, quadrotors can be applied to transportation [1], [2], terrain mapping [3], [4], inspection [5], [6], and construction [7], [8]. Among these applications, the transportation task by a quadrotor is the main focus of this study.

To complete the transportation task successfully, model-based control schemes such as disturbance observer (DOB), model predictive control (MPC), and model reference adaptive control (MRAC), which secure robustness to disturbance and stability, should be implemented. However, the model-based controller does not necessarily guarantee the robustness of the dynamic model uncertainty of a quadrotor under an unknown payload. Therefore, considering accurate model parameters in the control scheme is highly important.

To reflect the model uncertainty of the control scheme, MRAC using various estimation techniques [9], [10], [11], [12], [13], [14], [16], [17], [18] has been proposed. Inertia parameter estimation of an aerial manipulator without using a force/torque sensor in each joint through the adaptation law was proposed in [9] and [10]. However, prior knowledge regarding the payload geometry was required in [9]. The payload was treated as a point mass, and the MOI was obtained by the parallel axis theorem in [10], making it difficult for the control scheme to deal with a payload with a non-uniform mass density distribution. The authors of [11] utilized an extended Kalman filter (EKF) to fuse inertial measurement unit (IMU) sensor and rotor speed processed through a LPF and compensated for the variation in COM

The associate editor coordinating the review of this manuscript and approving it for publication was Ming Xu.

location on the control scheme. However, mapping from the rotor speed to the angular acceleration is required in advance.

Unlike in [9], [10], and [11], the necessity of prior knowledge or assumptions about the payload was eliminated in [12], [13], [14], [15], [16], [17], and [18]. Reference [12] assumed that the collective thrust input is linearly proportional to the resultant force produced by the four rotors and estimated the ratio of the former to the latter divided by mass through the adaptation law. Then, the control scheme compensated for the estimated ratio. A multiple KF through ultrasound sonar and 9-axis IMU sensor to estimate the mass and z-axis MOI was devised, and altitude and yaw control were performed depending on the load in [13]. Although the necessity of prior knowledge was removed in [12] and [13], they did not secure the stability of the attitude control because they excluded the estimation of the changed COM location. In contrast, Reference [14] estimated and compensated for the changed COM location in the attitude dynamics. The assumption orientational dynamic response is faster than the translational one removed the need for prior knowledge about MOI. However, this assumption is not always valid because the quadrotor's acceleration tracking performance depends on the size of the MOI [15].

Compared to [14], the coupling between the translational dynamic response and orientational response is considered through an adaptation law that estimates the mass and MOI simultaneously [16]. However, the location of COM should be consistent with the center of the geometry. Although [17] compensated for the model uncertainty through a radial base function neural network, the variation in COM location was not reflected in the orientational dynamic model. Unlike multiple studies [9], [10], [11], [12], [13], [14] and [16], [17], the estimation of the variation of COM, MOI, and mass through the adaptation law was presented in [18]. The shortcoming of reference [18] is that the sharp increase or decrease in COM variation estimation occurs owing to motor saturation, and the convergence rate of MOI parameter estimation is too slow to be reflected in the control scheme. Hence, research on system identification should be conducted to develop a model-based control scheme for a quadrotor with an unknown payload.

Considerable effort has been devoted to system identification for a long time. Inertia parameter estimation was performed using maximum likelihood estimation (MLE) [19]. They determined the dynamic parameters that minimize the residual of the IMU sensor, camera, and dynamic model. However, the residual related to the dynamic model makes the computational load of estimation heavy because it computes Runge–Kutta fourth order and Levenberg–Marquardt simultaneously. Compared to MLE, the filtering approach is sufficiently fast to process the sensor data.

Compared to MLE, least squares (LS) [20], [21] allows for inertia parameter identification during flight. In [20], the consideration of angular momentum conservation reduced the effects of sensor noise. This enhances the accuracy of estimating the product of inertia. However, it could not secure the stability of orientational dynamics and altitude because it did not infer mass and the variation in COM position. In contrast, [21] only estimated the changed COM and compensated for that in the attitude PID controller until the quadrotor reached the hovering state. Even though the convergence rate is sufficiently fast for attitude control to compensate for the parameter, the estimation of COM variation is determined depending on the initial value setup for MOI. Thus, it is difficult to guarantee the stability of attitude control at all times.

Unlike the LS method, filtering approaches such as EKF, unscented Kalman filter (UKF) [22], [23], [24], [25], [26], [27], [28], square root unscented Kalman filter (SRUKF) [29], and complementary filter (CF) [30] have been conducted. While multiple studies [22], [23], [24], [25], [29], [30] did not address the estimation of the changed COM, which is a significant element in model-based control [21], all inertia parameters such as the changed COM, MOI, and mass by fusing rotor speed sensors were inferred in [26], [27], and [28]. Thrust force and drag moment coefficients were identified in [27] and [28]. However, incorrect identification of these two parameters may lead to an inaccurate estimation of the inertia parameter. In contrast, [26] considered the inertia parameter estimation, including mass, varied COM, and MOI, given the coefficients. By tracking the Lissajous trajectory, the identification was performed; however, the convergence rate was slow, and there was bias about the estimated inertia parameter due to the noise effect of the sensor.

An excitation trajectory should be applied to the quadrotor to amplify the signal-to-noise ratio (SNR) and gain rich data. Research on trajectory generation was inspired by the system identification of manipulators [31], [32], [33], [34], [35]. A trajectory composed of a Fourier series, whose coefficients minimize the fisher information matrix (FIM), was utilized in [36] and [37]. The disadvantage of [36] and [37] is that the complexity of LS increases with the size of the matrix. Hence, it is inadequate to apply the estimation method to model-based control directly. In contrast, [38] treated the condition number of FIM as the residual of model predictive control (MPC) and produced rich data. However, the method applies to satellites and cannot be applied to quadrotors constrained by gravity. The maximization of the empirically expanded local observability Gramian (E2LOG) [39] based on the index of unobservability [40] made the data richer. The trajectory obtained by minimizing the condition number or maximizing E2LOG tends to have a high amplitude and unnecessary motion. In other words, the trajectory increases energy inefficiency compared to the time-optimal trajectory.

It is necessary to estimate the accurate inertia parameter by reducing the effect of sensor noise, even while tracking the time-optimal trajectory. This study devises KF to reduce the sensitivity to noise, which estimates the angular rate by adjusting the proportion of confidence in the process model to the one in the measurement model. Specifically, the variation of the estimated angular velocity is reduced when more confidence is placed in the process model than in the measurement model. In [26], [27], and [28], which considers the inertia parameter as a state variable, the variance of the angular velocity attributed to model uncertainty in the process model was not dealt with. This causes the user to set the initial parameter value close to the ground truth. Otherwise, the estimation of the inertial parameter can be biased. This study derives the variance caused by the model uncertainty and reflects the variance in KF. We used RLS to identify the inertia parameter. However, the time-optimal trajectory does not always enable the RLS to observe all inertia parameters. Thus, this paper presents the criteria that distinguish between the observable and the unobservable inertia parameter and proposes a correction law that constructs an observation matrix for RLS to correct only the observable one. The correction law makes RLS aware of the observability. The main contributions of this study are as follows:

- For robust and precise control, an inertia parameter estimator including mass, MOI, COM, and sensor location under the triple integral method, which belongs to the time optimal trajectory generator, is designed.
- In KF, the variance caused by model uncertainty is considered in the process model, unlike multiple works [26], [27], [28]. The variance plays a key role in the adjustment of Kalman gain.
- The correction law is proposed to prevent the inertia parameter estimator, RLS, from correcting unobservable parameters.
- Unlike multiple works [26], [27], [28], [29], [30], [36], [37], [38], [39], our approach enables the estimator to be triggered at the start of a flight. The inertia parameter convergence rate can deteriorate when the estimator begins to estimate from the ground station.

The remainder of this paper is organized as follows. The dynamic model of a quadrotor under an unknown payload and the relationship between the observation matrix and the dynamics are described in Section II. Section III presents the design of KF and the observability-aware RLS. Section IV presents the simulation environment, in which the noise level, estimator setup, and flight data are described. Section V discusses the simulation results and analyses. Finally, the conclusions and future work are presented in Section VI.

II. INERTIA PARAMETER IDENTIFICATION

This section introduces a quadrotor dynamic model under payload by the Newton-Eulerian equation, which is divided into translational and orientational dynamic models. Then, it addresses the relationship between attitude dynamics and observation matrix $\boldsymbol{H}_1$ and translational dynamics and observation matrix $\boldsymbol{H}_2$.

A. QUADROTOR DYNAMIC MODEL

Before delving into the observability-aware inertia parameter estimation method, we will present the quadrotor dynamic model. The four motor speeds determined the collective thrust and moment. The off-diagonal terms of the inertia tensor were assumed to be zero.

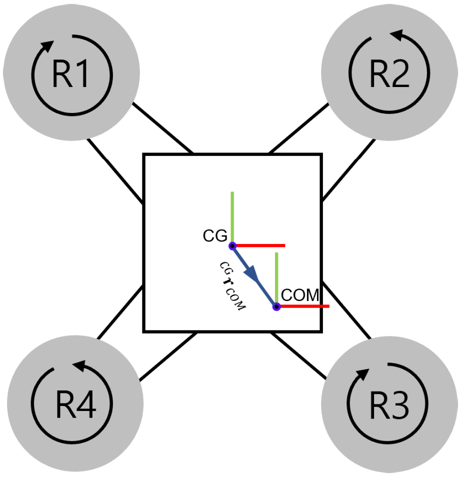

FIGURE 1. The quadrotor rotor configuration and the frame infromation.

The index of the rotor is shown in Fig. 1. As shown in Figure 1, rotors R1 and R3 rotate clockwise, whereas R2 and R3 rotate counterclockwise. ${}^{CG}\boldsymbol{r}_{COM}$ indicates the location of COM expressed in the frame center of geometry (CG) which is the centroid of the four rotors' axis in the horizontal plane. It is impossible to obtain the relative height between the CG frame and the COM frame because the height of the CG is arbitrarily chosen.

The thrust generated by each rotor in the CG frame was proportional to the square of the rotor speed. The x and y components of the moment generated by each rotor i in the CG frame or COM frame are proportional to the arm length and force produced by each rotor. The rotor thrust and moments along the x-, y-, and z-axes of each rotor can be written as follows:

$$F_{Ri} = C_T \omega_{Ri}^2, \tag{1}$$

$$M_{x,Ri} = \frac{l\sqrt{2}}{2} C_T F_{Ri} = \frac{l\sqrt{2}}{2} C_T \omega_{Ri}^2, \tag{2}$$

$$M_{y,Ri} = \frac{l\sqrt{2}}{2} C_T F_{Ri} = \frac{l\sqrt{2}}{2} C_T \omega_{Ri}^2, \tag{3}$$

$$M_{z,Ri} = C_M \omega_{Ri}^2. \tag{4}$$

In (1)–(4), C_T, C_M, and l represent the lift coefficient, moment coefficient, and arm length, respectively, of the quadrotor. The arm length is the displacement from the CG frame to the center of each rotor in the horizontal plane. ω_{Ri} denotes the i-th rotor speed, In the CG frame, the forward dynamic model can be expressed as follows:

$$\begin{bmatrix} T \\ \boldsymbol{M} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{l\sqrt{2}}{2} & \frac{l\sqrt{2}}{2} & -\frac{l\sqrt{2}}{2} & -\frac{l\sqrt{2}}{2} \\ \frac{l\sqrt{2}}{2} & -\frac{l\sqrt{2}}{2} & -\frac{l\sqrt{2}}{2} & \frac{l\sqrt{2}}{2} \\ \frac{C_M}{C_T} & -\frac{C_M}{C_T} & \frac{C_M}{C_T} & -\frac{C_M}{C_T} \end{bmatrix} \begin{bmatrix} F_{R1} \\ F_{R2} \\ F_{R3} \\ F_{R4} \end{bmatrix}. \tag{5}$$

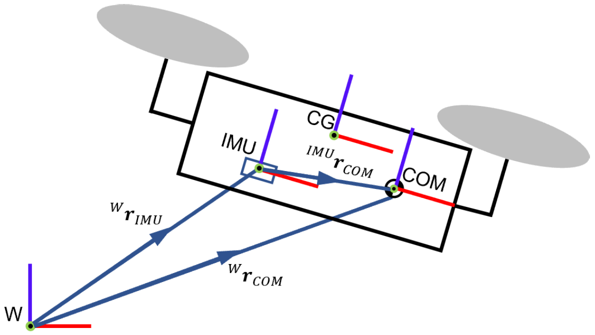

FIGURE 2. The sensor, COM, CG, and World frame of quadrotor system information.

In (5), T and $\boldsymbol{M}$ are the total thrust and the moment expressed in CG frame, respectively.

Figure 2 shows the world, IMU, and COM frames. The superscript denotes the reference frame, and the subscript denotes the frame to be expressed. Superscript W denotes the world. For example, $^{W}\boldsymbol{r}_{IMU}$ denotes the IMU frame location in the world frame.

Using the Newton–Euler equation, the quadrotor dynamic and kinematic models can be modeled as

$$m^{W}\dot{\boldsymbol{v}}_{COM} = \boldsymbol{Ad}(^{W}\boldsymbol{q}_{CG})\begin{bmatrix}0\\0\\T\end{bmatrix} - m\boldsymbol{g}, \tag{6}$$

$$^{W}\dot{\boldsymbol{r}}_{COM} = {}^{W}\boldsymbol{v}_{COM}, \tag{7}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega} = \boldsymbol{M} + {}^{COM}\boldsymbol{r}_{COM\to CG}\times, \begin{bmatrix}0\\0\\T\end{bmatrix}, \tag{8}$$

$$^{W}\dot{\boldsymbol{q}}_{CG} = \frac{1}{2}{}^{W}\boldsymbol{q}_{CG} * \boldsymbol{\omega}. \tag{9}$$

Equations (6) and (8) show the translational and attitude dynamics. Equations (7) and (9) represent translational and rotational kinematics, respectively. m, $\boldsymbol{J}$ and $^{COM}\boldsymbol{r}_{COM\to CG}$ denote the mass and MOI of the quadrotor and the location of the CG expressed in the COM frame, which is called the COM offset, and are of interest in this study. $^{W}\boldsymbol{q}_{CG}$ is the quaternion and contains the meaning of rotation. $\boldsymbol{Ad}$ is the adjoint operator. $\boldsymbol{Ad}(^{W}\boldsymbol{q}_{CG})$ is the rotation matrix that transforms the body frame into the world frame. Operator $*$ is the multiplication of the quaternion. $\boldsymbol{g}$ denotes the gravity vector expressed in the global frame. $^{W}\dot{\boldsymbol{v}}_{COM}$ is the acceleration of COM in the world frame. Collective thrust T is expressed in the CG frame. The attitude dynamics has a fictitious moment, $\boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega}$ owing to the change of the CG frame and the moment generated by the collective thrust and COM offset, $^{COM}\boldsymbol{r}_{COM\to CG}\times\begin{bmatrix}0 & 0 & T\end{bmatrix}^{T}$.

B. ATTITUDE DYNAMICS AND H_1

By rearranging Equation (8), we can rearrange the attitude dynamics as follows:

$$\begin{bmatrix}M_x\\M_y\\M_z\end{bmatrix} = \boldsymbol{H}_1\boldsymbol{\theta}_1 \tag{10}$$

where

$$\boldsymbol{H}_1 =. \begin{bmatrix}\dot{p} & -qr & qr & 0 & T\\ pr & \dot{q} & -pr & -T & 0\\ -pq & pq & \dot{r} & 0 & 0\end{bmatrix}$$

and $\boldsymbol{\theta}_1 =. \begin{bmatrix}J_{xx} & J_{yy} & J_{zz} & x_{off} & y_{off}\end{bmatrix}^{T}$.

In Equation (10), $\boldsymbol{\theta}_1$ denotes the MOI and COM information for the inference. Specifically, x_{off} and y_{off} are the x and y coordinates of $^{COM}\boldsymbol{r}_{COM\to CG}$. J_{xx}, J_{yy}, and J_{zz} denote the x-, y-, and z-axes inertia, respectively. $\dot{p}, \dot{q}, \dot{r}, p, q$, and r denote the angular acceleration along the x-, y-, and z-axes and angular velocity along the x-, y-, and z-axes, respectively. In Equation (10), $\boldsymbol{H}_1$ is the observation matrix for MOI and COM.

C. TRANSLATIONAL DYNAMICS AND H_2

The IMU sensor model can be described as follows:

$$\boldsymbol{a}_{IMU} = {}^{CG}\boldsymbol{R}_{W}(^{W}\ddot{\boldsymbol{r}}_{IMU} + \boldsymbol{g}). \tag{11}$$

The representation of COM based on Figure 1 can be written as follows:

$$^{W}\boldsymbol{r}_{COM} = {}^{W}\boldsymbol{r}_{IMU} + {}^{W}\boldsymbol{R}_{CG}{}^{IMU}\boldsymbol{r}_{COM}. \tag{12}$$

By differentiating Equation (12), the velocity relation can be written as

$$\begin{aligned}^{W}\dot{\boldsymbol{r}}_{COM} &= {}^{W}\dot{\boldsymbol{r}}_{IMU} + \frac{d}{dt}(^{W}\boldsymbol{R}_{CG}{}^{IMU}\boldsymbol{r}_{COM})\\ &= {}^{W}\dot{\boldsymbol{r}}_{IMU} + {}^{W}\dot{\boldsymbol{R}}_{CG}{}^{IMU}\boldsymbol{r}_{COM}\\ &+ {}^{W}\boldsymbol{R}_{CG}{}^{IMU}\dot{\boldsymbol{r}}_{COM}.\end{aligned} \tag{13}$$

In Equation 13, because the IMU sensor location is fixed for the COM frame, the last term $^{IMU}\dot{\boldsymbol{r}}_{COM}$ vanishes. The derivative of the rotation matrix $^{W}\dot{\boldsymbol{R}}_{CG}{}^{IMU}$ can be derived as follows:

$$\begin{aligned}^{W}\dot{\boldsymbol{R}}_{CG}{}^{IMU} &= \frac{d}{dt}\begin{bmatrix}^{W}\boldsymbol{x}_{CG} & ^{W}\boldsymbol{y}_{CG} & ^{W}\boldsymbol{z}_{CG}\end{bmatrix}\\ &= [^{W}\boldsymbol{\omega}_{CG}]_{\times}{}^{W}\boldsymbol{R}_{CG}, .\end{aligned} \tag{14}$$

$$\text{where}\quad [\boldsymbol{\omega}]_{\times} = \begin{bmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{bmatrix} \in \mathfrak{so}(3).$$

Equation (14) shows the derivative of the rotation matrix. The bracket operator $[\cdot]_{\times}$ maps a vector $\boldsymbol{\omega} \in \mathbb{R}^3$ in the Euclidean space to skew the symmetric matrix $[\boldsymbol{\omega}]_{\times} \in \mathfrak{so}(3)$. The angular velocity in the bracket operator is expressed in the world frame. Because the angular velocity $^{W}\boldsymbol{\omega}$ is in the Euclidean space, the angular velocity can be transformed through the rotation matrix $^{W}\boldsymbol{R}_{CG}$ and the bracket has the following property according to [41].

$$[\boldsymbol{R}^{T}\boldsymbol{\omega}]_{\times} = \boldsymbol{R}^{T}[\boldsymbol{\omega}]_{\times}\boldsymbol{R} \tag{15}$$

By substituting Equation (14) into (15), the time derivative of the rotation matrix can be reduced to:

$$^{W}\dot{\boldsymbol{R}}_{CG} = {}^{W}\boldsymbol{R}_{CG}[^{CG}\boldsymbol{\omega}]_{\times}. \tag{16}$$

Using equation (16), Equation (13) can be simplified as follows:

$$ {}^{W}\dot{\boldsymbol{r}}_{COM} = {}^{W}\dot{\boldsymbol{r}}_{IMU} + {}^{W}\boldsymbol{R}_{CG}[{}^{CG}\boldsymbol{\omega}]_\times {}^{IMU}\boldsymbol{r}_{COM}. \tag{17}$$

Similarly, the linear acceleration relation can be obtained by differentiating Equation (17).

$$\begin{aligned} {}^{W}\ddot{\boldsymbol{r}}_{COM} &= {}^{W}\ddot{\boldsymbol{r}}_{IMU} + {}^{W}\boldsymbol{R}_{CG}[{}^{CG}\boldsymbol{\omega}]^2_\times {}^{IMU}\boldsymbol{r}_{COM} \\ &\quad + {}^{W}\boldsymbol{R}_{CG}[{}^{CG}\boldsymbol{\alpha}]_\times {}^{IMU}\boldsymbol{r}_{COM}. \end{aligned} \tag{18}$$

In Equation (18), ${}^{CG}\boldsymbol{\alpha}$ denotes angular acceleration. Using the IMU sensor model (11) and the two-frame acceleration relation (18), the equation for IMU acceleration-related observation matrix can be expressed as follows:

$$ m\boldsymbol{a}_{IMU} - ([{}^{CG}\boldsymbol{\omega}]^2_\times + [{}^{CG}\boldsymbol{\alpha}]_\times)(m{}^{COM}\boldsymbol{r}_{IMU}) = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}. \tag{19}$$

Equation (19) can be rearranged as follows:

$$\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = \boldsymbol{H}_2\boldsymbol{\theta}_2 \tag{20}$$

where $\boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{h}_2(1) & -pr - \dot{q} \\ \boldsymbol{h}_2(2) & -pr + \dot{p} \\ \boldsymbol{h}_2(3) & p^2 + q^2 \end{bmatrix}$

$$\boldsymbol{h}_2(1) = a_x + {}^{COM}x_{IMU}(q^2 + r^2) + {}^{COM}y_{IMU}(-pq + \dot{r}),$$

$$\boldsymbol{h}_2(2) = a_y - {}^{COM}x_{IMU}(pq + \dot{r}) + {}^{COM}y_{IMU}(p^2 + r^2),$$

$$\boldsymbol{h}_2(3) = a_z + {}^{COM}x_{IMU}(-pr + \dot{q}) - {}^{COM}y_{IMU}(qr + \dot{p}),$$

$$\boldsymbol{\theta}_2 = \begin{bmatrix} m \\ m{}^{COM}z_{IMU} \end{bmatrix},$$

$${}^{COM}x_{IMU} = -\left(x_{offset} + {}^{IMU}x_{COM}\right),$$

and $\quad {}^{COM}y_{IMU} = -\left(y_{offset} + {}^{IMU}y_{COM}\right).$

The observation matrix $\boldsymbol{H}_2$ for estimating the mass and sensor location is shown in Equation (20).

III. DESIGN OBSERVABILITY-AWARE RLS WITH KF

The overall pipeline for the algorithm of the observability-aware RLS with KF comprises KF, data processing, and RLS in Figure 3. The main role of KF is to reduce the noise of the angular rate by adjusting the confidence ratio in the process model to that in the measurement model. The arguments for KF are the control input (collective thrust T and moment $\boldsymbol{M}$), the previous angular velocity parameters (values and covariance), and the estimated MOI parameter. Here, the control input is obtained from Equation (5) and the relationship between the rotor speed and tuple of collective thrust and moment generated by the four rotors. KF returns angular velocity parameters for data processing.

When the non-zero z component of gravity-rejected acceleration processed by a low-pass filter is detected for the first time, the data processing and RLS algorithms are activated. In Figure 3, $a_{z,criteria}$ denotes the acceleration of the altitude motion criteria. The role of the detection is to prevent RLS from correcting the inertia parameters when the quadrotor is at the ground. The data processing algorithm stores the angular rate at the previous time step, obtains the angular velocity at the current time step, and returns the processed observation matrix. The observation matrix is based on (10). The algorithm inserted the data into the corresponding column by distinguishing the angular rate from the corresponding standard deviation. This prevents the RLS from correcting the unobservable inertia parameters.

RLS method plays a key role in estimating MOI and COM. The arguments of RLS are the observation matrix, previous MOI, and COM parameters (values and covariance); it then returns the corrected MOI and COM parameters. We describe the function of each component (KF, data-processing algorithm, and RLS).

A. KALMAN FILTER

KF comprises a process model and a measurement model. The process model computes the temporary belief of the angular velocity using a dynamic model and control input. The measurement model corrects the temporary belief by utilizing the data measured via the IMU sensor. In the process model, the variance of the inverse normal distribution should be derived approximately because the variance of the inverse MOI $\hat{\boldsymbol{J}}_i^{-1}$ is different from that of MOI $\hat{\boldsymbol{J}}_i$. Although the inverse distribution no longer belongs to Gaussian distribution, it can approximately be regarded as Gaussian distribution near the ground truth. The temporary belief of the angular velocity $\overline{\boldsymbol{\omega}}_{i+1}$ can be obtained through a dynamic model, but the dynamic model involves noise and dynamic parameter uncertainty. The noise caused by the rotor thrust is negligible if there is no wind disturbance. Thus, the parameter uncertainty is mostly related to the variance of the temporary belief $\boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i+1}}$. Because COM offset values are always observable, their convergence rate is sufficiently fast to neglect their uncertainty. In contrast, MOI parameter does not guarantee observability at all times, which means that it is impossible to obtain the MOI parameter when there is insufficient oscillation during flight. In this regard, the uncertainty of MOI should be considered in the temporary variance of the process model. First, the true angular velocity based on the orientational dynamics can be written as

$$\boldsymbol{\omega}_{i+1} = \boldsymbol{\omega}_i - \boldsymbol{J}^{-1}(\boldsymbol{u}_i - \boldsymbol{J} \times \boldsymbol{J}\boldsymbol{\omega}_i)dt \tag{21}$$

where $\quad \boldsymbol{u}_i = \boldsymbol{M}_i - \hat{\boldsymbol{r}}_{off} \times \boldsymbol{T}_i. \quad$ and $\boldsymbol{T}_i = \begin{bmatrix} 0 & 0 & T_i \end{bmatrix}^T.$

In (21), $\boldsymbol{\omega}_i$, $\boldsymbol{\omega}_{i+1}$, and $\boldsymbol{J}$ denote the true angular velocity at the i-th and (i+1)-th time steps, the true MOI. $\boldsymbol{u}_i$ represents the equivalent moment at the time step i. The control input data T_i and $\boldsymbol{M}_i$ are acquired by the motor speed ω_{Rj} for $j = 1, 2, 3, 4$. Assuming that the noise effects of the control inputs (moment $\boldsymbol{M}_i$ and collective thrust T_i) are negligible, the temporary

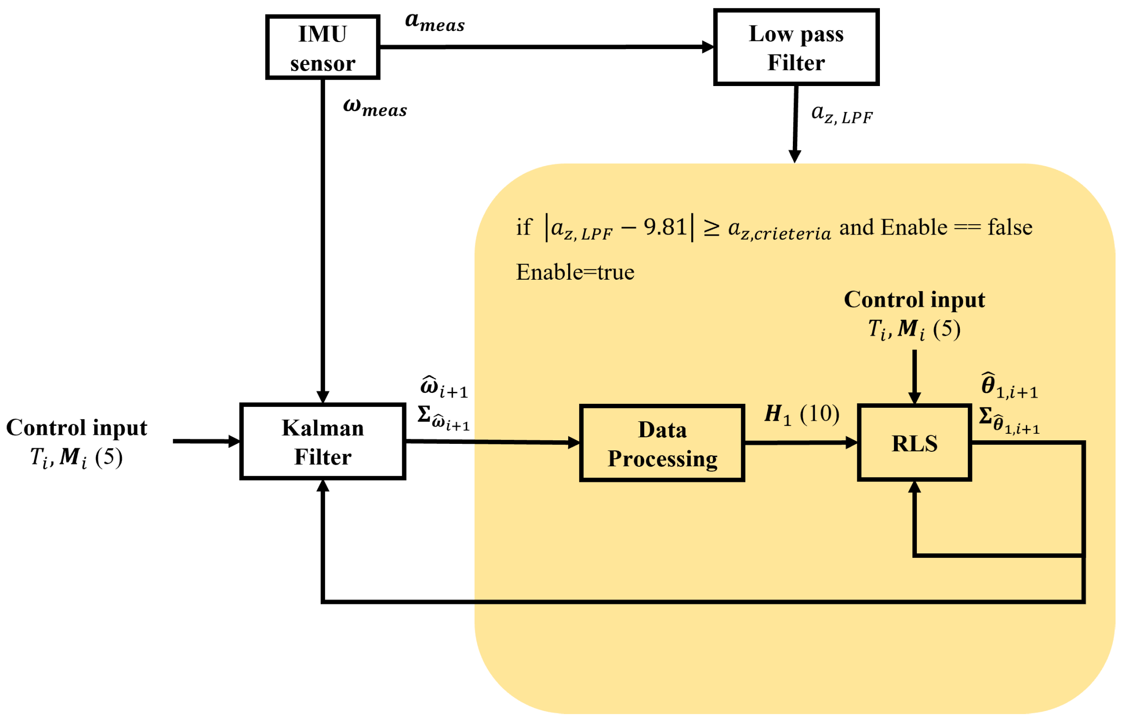
IMU sensor
a_{meas}
Low pass Filter
ω_{meas}
$a_{z,LPF}$
if $|a_{z,LPF} - 9.81| \geq a_{z,crieteria}$ and Enable == false
Enable=true
Control input
$T_i, \boldsymbol{M}_i$ (5)
$\hat{\boldsymbol{\theta}}_{1,i+1}$
$\boldsymbol{\Sigma}_{\hat{\theta}_{1,i+1}}$
$\hat{\boldsymbol{\omega}}_{i+1}$
$\boldsymbol{\Sigma}_{\hat{\omega}_{i+1}}$
Control input
$T_i, \boldsymbol{M}_i$ (5)
Kalman Filter
Data Processing
$\boldsymbol{H}_1$ (10)
RLS

FIGURE 3. The pipeline of Observability-aware KF.

belief is

$$\overline{\boldsymbol{\omega}}_{i+1} = \hat{\boldsymbol{\omega}}_i + \hat{\boldsymbol{J}}_i^{-1}(\boldsymbol{u}_i - \hat{\boldsymbol{J}}_i \times \hat{\boldsymbol{J}}_i\hat{\boldsymbol{\omega}}_i)dt. \tag{22}$$

The error in angular velocity between the temporary belief and ground truth can be obtained by subtracting (21) from (22).

$$\begin{aligned}\tilde{\boldsymbol{\omega}}_{i+1} &= \tilde{\boldsymbol{\omega}}_i + [(\boldsymbol{J}^{-1} - \hat{\boldsymbol{J}}_i^{-1})\boldsymbol{u}_i \\ &\quad + \boldsymbol{J}^{-1}(\boldsymbol{J} \times \boldsymbol{J}\boldsymbol{\omega}_i) - \hat{\boldsymbol{J}}_i^{-1}(\hat{\boldsymbol{J}}_i \times \hat{\boldsymbol{J}}_i\hat{\boldsymbol{\omega}}_i)]dt\end{aligned} \tag{23}$$

where $\quad \tilde{\boldsymbol{\omega}}_i = \hat{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i, \quad \tilde{\boldsymbol{\omega}}_{i+1} = \hat{\boldsymbol{\omega}}_{+1i} - \boldsymbol{\omega}_{i+1}.$

In (23), the effect of the fictitious moment $\boldsymbol{J} \times \boldsymbol{J}\boldsymbol{\omega}_i$ can be neglected, because the moment $\boldsymbol{M}_i$ is greater than the fictitious moment. Thus, Equation (23) can be approximated as

$$\tilde{\boldsymbol{\omega}}_{i+1} \approx \tilde{\boldsymbol{\omega}}_i + [(\boldsymbol{J}^{-1} - \hat{\boldsymbol{J}}_i^{-1})\boldsymbol{u}_i]dt. \tag{24}$$

By the definition of covariance, the covariance of the temporary belief can be derived as

$$\mathbb{E}[\tilde{\boldsymbol{\omega}}_{i+1}\tilde{\boldsymbol{\omega}}_{i+1}^T] = \mathbb{E}[\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i^T] + \boldsymbol{\Sigma}_{add} \tag{25}$$

,

where $\quad \boldsymbol{\Sigma}_{add}. = dt^2\mathbb{E}[(\boldsymbol{J}^{-1}), -\hat{\boldsymbol{J}}_i^{-1})^2\boldsymbol{u}_i\boldsymbol{u}_i^T].$

In (25), the last term $\boldsymbol{\Sigma}_{add}$ contains an inverse normal distribution, because it has $\mathbb{E}[(\boldsymbol{J}^{-1} - \hat{\boldsymbol{J}}_i^{-1})^2]$. The variance of the inverse normal distribution for each component can be written as

$$\begin{aligned}\mathbb{E}\left[\frac{1}{\boldsymbol{J}(k)} - \frac{1}{\hat{\boldsymbol{J}}(k)}\right]^2 &= \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{\boldsymbol{J}(k)} - \frac{1}{\hat{\boldsymbol{J}}_i(k)}\right]^2 \\ &= \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\left[\frac{\hat{\boldsymbol{J}}_i(k) - \boldsymbol{J}(k)}{\hat{\boldsymbol{J}}_i(k)\boldsymbol{J}(k)}\right]^2 \\ &= \frac{1}{\boldsymbol{J}(k)^2}\mathbb{E}\left[\frac{\hat{\boldsymbol{J}}(k) - \boldsymbol{J}(k)}{\hat{\boldsymbol{J}}(k)}\right]^2\end{aligned} \tag{26}$$

where $k = 1, 2, 3,$
$\boldsymbol{J}(1) = J_{xx}, \quad \boldsymbol{J}(2) = J_{yy}, \quad$ and $\quad \boldsymbol{J}(3) = J_{zz}.$

In (26), index k denotes the component of the MOI parameter. When the estimated MOI parameter is slightly biased, the variance can be approximately expressed as

$$\mathbb{E}\left[\frac{1}{\boldsymbol{J}(k)} - \frac{1}{\hat{\boldsymbol{J}}(k)}\right]^2 \approx \frac{1}{\boldsymbol{J}(k)^4}\mathbb{E}\left[\hat{\boldsymbol{J}}(k) - \boldsymbol{J}(k)\right]^2. \tag{27}$$

The variance in (26) can be represented as an inequality as follows:

$$\mathbb{E}\left[\frac{1}{\boldsymbol{J}(k)} - \frac{1}{\hat{\boldsymbol{J}}(k)}\right]^2 \leq \frac{1}{\boldsymbol{J}_{min}(k)^4}\mathbb{E}\left[\hat{\boldsymbol{J}}(k) - \boldsymbol{J}(k)\right]^2. \tag{28}$$

In (28), $\boldsymbol{J}_{min}(k)$ for k = 1, 2, and 3 is the lower bound of MOI values arbitrarily set by the user. The last term in (25) can be replaced by (27) and (28).

$$\boldsymbol{\Sigma}_{add}(k,k)=\begin{cases}\dfrac{dt^2u(k)^2}{\boldsymbol{J}_{min}(k)^4}\boldsymbol{\Sigma}_{\hat{\boldsymbol{J}}}(k,k),\ \hat{\boldsymbol{J}}(k)\le\boldsymbol{J}_{min}(k)\\[2ex]\dfrac{dt^2u(k)^2}{\hat{\boldsymbol{J}}(k)^4}\boldsymbol{\Sigma}_{\hat{\boldsymbol{J}}}(k,k),\ \hat{\boldsymbol{J}}(k)>\boldsymbol{J}_{min}(k).\end{cases}\tag{29}$$

In (29), when the estimated MOI value is less than the minimum value that cannot occur, the variance is set to the maximum variance. The upper bound of inequality (28) was chosen to reflect this. Otherwise, the variance was computed using the estimated MOI value. As the estimated parameter has a small bias, the variance of MOI $\boldsymbol{\Sigma}_{\hat{\boldsymbol{J}}}(k,k)$ decreases. In addition, when the equivalent moment $\boldsymbol{u}(k)$ is close to zero, the additive variance becomes nearly zero because the uncertainty of the temporary belief also originates from motion.

Algorithm 1 presents the process model of KF. The equivalent moment $\boldsymbol{u}_i$ is determined by the moment, COM offset, and the thrust. The temporary belief is calculated using (22). Using condition (29), the additive variance containing the model uncertainty of MOI is computed. The temporary variance is the summation of the previous variance and additive variance.

Algorithm 2 shows the measurement model of KF. The IMU sensor corrected the temporary belief in the angular rate. The correction depends on Kalman gain $\boldsymbol{K}_{kalman}$. Here, to avoid excessive error in temporary belief, the residual for each component $r(k)$ and constraint c are compared, and the weight of the measurement $\boldsymbol{R}_{meas}(k,k)$ is adjusted. When there is too much discrepancy between the temporary belief and IMU sensor data, the measurement weight becomes large.

The main role of KF is to reduce the effect of noise on angular velocity by adjusting the confidence in the process and measurement models. Because the control input has less noise than IMU sensor noise, it is possible to reduce the noise in the angular velocity through KF when it puts more confidence in the process model. The reduction of noise improves RLS parameter estimation performance. The entire procedure for KF belongs to the adaptive KF in that the measurement weight is adaptively adjusted to guarantee robustness.

B. DATA PROCESSING ALGORITHM

Data processing is required before RLS estimates the dynamic parameters because the time-optimal trajectory does not always guarantee observability. This algorithm makes RLS aware of observability. Observability determines the condition under which RLS can either estimate or not. By processing the data, RLS prevents the update gain from correcting the dynamic parameters.

Algorithm 3 shows the computation of the column vectors when it receives the index k, angular velocity, and angular acceleration. The function in Algorithm 3 returns the column vector based on (10). Function $\boldsymbol{\phi}_1$ only returns the vector from the first column to the third of the observation matrix $\boldsymbol{H}_1$.

Algorithm 1 Process Model of Kalman Filter

```
Arguments: $\hat{\boldsymbol{\omega}}_i$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}$, $\hat{\boldsymbol{J}}_i$. $\hat{\boldsymbol{r}}_{off}$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{J}}_i}$, $T_i$, $\boldsymbol{M}_i$
$\boldsymbol{T}_i = \begin{bmatrix}0\ 0\ T_i\end{bmatrix}^T$
$\boldsymbol{u}_i = \boldsymbol{M}_i - \hat{\boldsymbol{r}}_{off} \times \boldsymbol{T}_i$
$\overline{\boldsymbol{\omega}}_{i+1} = \hat{\boldsymbol{\omega}}_i + \hat{\boldsymbol{J}}_i^{-1}(\boldsymbol{u}_i - \hat{\boldsymbol{J}}_i \times \hat{\boldsymbol{J}}_i\hat{\boldsymbol{\omega}}_i)dt$
for $k = 1:3$ do
    if $\hat{\boldsymbol{J}}_i(k) \le J_{min}$ then
        $\boldsymbol{\Sigma}_{add}(k,k) = \left(\boldsymbol{\Sigma}_{\hat{J}_i}(k,k)/J_{min}^4\right)\boldsymbol{u}_i(k)^2dt^2$
    else
        $\boldsymbol{\Sigma}_{add}(k,k) = \left(\boldsymbol{\Sigma}_{\hat{J}_i}(k,k)/\hat{\boldsymbol{J}}_i(k)^4\right)\boldsymbol{u}_i(k)^2dt^2$
    end if
end for
$\boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i+1}} = \boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i} + \boldsymbol{\Sigma}_{add}$
Return $\overline{\boldsymbol{\omega}}_{i+1}$, $\boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i\mathcal{C}1}}$
```

Algorithm 2 Measurement Model of Kalman Filter

```
Arguments: $\overline{\boldsymbol{\omega}}_{i+1}$, $\boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i\mathcal{C}1}}$, $\boldsymbol{\omega}_{IMU,i+1}$
initial weight: $\boldsymbol{R} = w_{meas}^2\boldsymbol{I}_{3\times3}$     constraint: c
for $k = 1:3$ do
    $r(k) = |\boldsymbol{\omega}_{IMU,i+1}(k) - \overline{\boldsymbol{\omega}}_{i+1}(k)|$
    if $r(k) \le c$ then
        $\boldsymbol{R}_{meas}(k,k) = \boldsymbol{R}(k,k)$
    else
        $\boldsymbol{R}_{meas}(k,k) = \boldsymbol{R}(k,k)\frac{r(k)}{c}$
    end if
end for
$\boldsymbol{K}_{kalman} = \boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i\mathcal{C}1}}\left(\boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i\mathcal{C}1}} + \boldsymbol{R}_{meas,}^{-1}\right)^{-1}$
$\hat{\boldsymbol{\omega}}_{i+1} = \overline{\boldsymbol{\omega}}_{i+1} + \boldsymbol{K}_{kalman}\left(\boldsymbol{\omega}_{IMU,i+1} - \overline{\boldsymbol{\omega}}_{i+1}\right)$
$\boldsymbol{\Sigma}_{\boldsymbol{\varpi}_{i+1}} = (\boldsymbol{I}_{3\times3} - \boldsymbol{K}_{kalman})\,\boldsymbol{\Sigma}_{\overline{\boldsymbol{\omega}}_{i+1}}$
Return $\hat{\boldsymbol{\omega}}_{i+1}$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_{i+1}}$
```

Algorithm 3 Column Vector Computation $\boldsymbol{\phi}_1$ for $\boldsymbol{H}_1$

```
Arguments: $k$, $\boldsymbol{\omega}$, $\boldsymbol{\alpha}$
$\dot{p} = \alpha(1)$, $\dot{q} = \alpha(2)$, $\dot{r} = \alpha(3)$
$p = \boldsymbol{\omega}(1)$, $q = \boldsymbol{\omega}(2)$, $r = \boldsymbol{\omega}(3)$
if $k = 1$ then
    $\boldsymbol{\phi}_1 = \begin{bmatrix}\dot{p}\ pr\ -pq\end{bmatrix}^T$
else if $k = 2$ then
    $\boldsymbol{\phi}_1 = \begin{bmatrix}-qr\ \dot{q}\ pq\end{bmatrix}^T$
else if $k = 3$ then
    $\boldsymbol{\phi}_1 = \begin{bmatrix}qr\ -pr\ \dot{r}\end{bmatrix}^T$
end if
Return $\boldsymbol{\phi}_1$
```

The standard deviation of the KF can determine the observability. In Algorithm 4, a column vector is inserted when the angular velocity of the k-th component is greater than the corresponding standard deviation. Otherwise, a zero vector is inserted into the column vector.

In Figure 4, the angular rates along the x- and y-axes are greater than the corresponding standard deviations. In other words, sufficient oscillation for RLS to infer MOI parameters

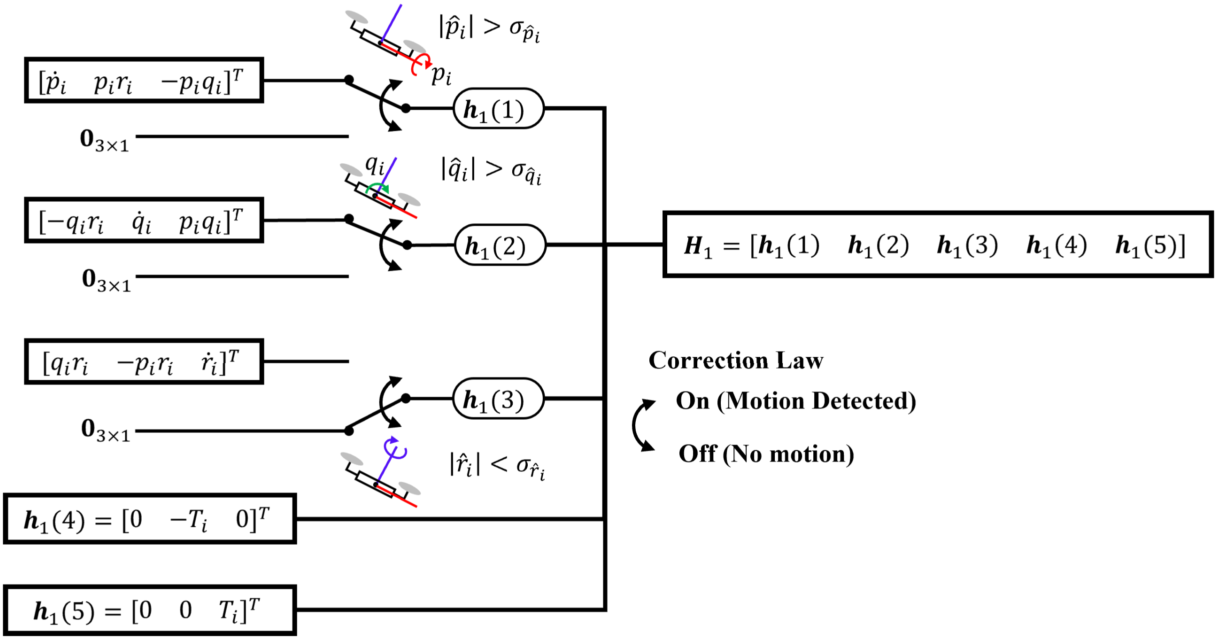
$|\hat{p}_i| > \sigma_{\hat{p}_i}$
p_i
$[\dot{p}_i \quad p_i r_i \quad -p_i q_i]^T$
$\mathbf{0}_{3\times1}$
$\boldsymbol{h}_1(1)$
q_i
$|\hat{q}_i| > \sigma_{\hat{q}_i}$
$[-q_i r_i \quad \dot{q}_i \quad p_i q_i]^T$
$\mathbf{0}_{3\times1}$
$\boldsymbol{h}_1(2)$
$\boldsymbol{H}_1 = [\boldsymbol{h}_1(1) \quad \boldsymbol{h}_1(2) \quad \boldsymbol{h}_1(3) \quad \boldsymbol{h}_1(4) \quad \boldsymbol{h}_1(5)]$
$[q_i r_i \quad -p_i r_i \quad \dot{r}_i]^T$
$\mathbf{0}_{3\times1}$
$\boldsymbol{h}_1(3)$
$|\hat{r}_i| < \sigma_{\hat{r}_i}$
Correction Law
On (Motion Detected)
Off (No motion)
$\boldsymbol{h}_1(4) = [0 \quad -T_i \quad 0]^T$
$\boldsymbol{h}_1(5) = [0 \quad 0 \quad T_i]^T$

FIGURE 4. Data processing algorithm through the use of the correction law.

Algorithm 4 Data Processing Algorithm for $\boldsymbol{H}_1$

```
Arguments: $T_i$, $\hat{\boldsymbol{\omega}}_i$, $\hat{\boldsymbol{\omega}}_{i+1}$, $\boldsymbol{\Sigma}_{\hat{\omega}_i}$,
$\hat{\boldsymbol{\alpha}} = \left(\hat{\boldsymbol{\omega}}_{i+1} - \hat{\boldsymbol{\omega}}_i\right)/dt$
for $k = 1:3$ do
    if $|\hat{\omega}_i(k)| \geq \sqrt{\boldsymbol{\Sigma}_{\hat{\omega}_i}(k,k)}$ then
        $\boldsymbol{h}_1(k) = \boldsymbol{\phi}_1(k, \hat{\boldsymbol{\omega}}_i(k), \hat{\boldsymbol{\alpha}})$.
    else
        $\boldsymbol{h}_1(k) = \mathbf{0}_{3\times1}$ and
    end if
end for
$\boldsymbol{h}_1(4) = \left[0 \; -T_i \; 0\right]^T$
$\boldsymbol{h}_1(5) = \left[T_i \; 0 \; 0\right]^T$
$\boldsymbol{H}_1 = \left[\boldsymbol{h}_1 \; \boldsymbol{h}_2 \; \boldsymbol{h}_3 \; \boldsymbol{h}_4 \; \boldsymbol{h}_5\right]$
Return $\boldsymbol{H}_1$
```

along the x and y axes occurs; therefore, angular acceleration data through differentiation of angular velocity should be inserted into the observation column vector $\boldsymbol{h}_1(1)$ and $\boldsymbol{h}_1(2)$. This forces RLS to correct MOI parameters $\hat{J}_{xx}$ and $\hat{J}_{yy}$. However, the angular velocity along the z-axis is less than the corresponding standard deviation. Therefore, the observation column vector $\boldsymbol{h}_1(3)$ was set as a zero vector $\mathbf{0}_{31}$. This determines when RLS has to correct the dynamic parameters (COM offset and MOI). This is why it is referred to as the correction law.

C. RECURSIVE LEAST SQUARE

The main purpose of RLS is to estimate the COM offset and MOI. Algorithm 5 presents the RLS algorithm in detail. The control input is moment $\boldsymbol{M}_i$ at the current step. $\boldsymbol{M}_i - \boldsymbol{H}_1\hat{\boldsymbol{\theta}}_i$ denotes innovation error. In Algorithm 5, the update gain matrix $\boldsymbol{K}_{update} \in \mathbb{R}^{5\times3}$ is determined by the observation matrix $\boldsymbol{H}_1$ processed in advance using a data-processing algorithm and the previous covariance matrix. This gain matrix determines the extent to which the correction should be achieved for the parameter.

Algorithm 5 RLS Algorithm for COM Offset and MOI

```
Arguments: $\boldsymbol{H}_1$, $\boldsymbol{M}_i$, $\hat{\boldsymbol{\theta}}_i$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_i}$
where $\hat{\boldsymbol{\theta}}_i = \left[\hat{J}_{xx,i} \; \hat{J}_{yy,i} \; \hat{J}_{zz,i} \; \hat{x}_{off,i} \; \hat{y}_{off,i}\right]^T$
RLS weight: $\boldsymbol{R}_{RLS} = w_{RLS}^2 \boldsymbol{I}_{5\times5}$
$\boldsymbol{K}_{update} = \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_i} \boldsymbol{H}_1^T \left(\boldsymbol{H}_1 \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_i} \boldsymbol{H}_1^T + \boldsymbol{R}_{RLS}^{-1},\right)^{-1}$
$\hat{\boldsymbol{\theta}}_{i+1} = \hat{\boldsymbol{\theta}}_i + \boldsymbol{K}_{update}, \left(\boldsymbol{M}_i - \boldsymbol{H}_1 \hat{\boldsymbol{\theta}}_i\right)$
$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{i+1}} = \left(\boldsymbol{I}_{5\times5}, -\boldsymbol{K}_{update} \boldsymbol{H}_1.\right) \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_i}$
Return $\hat{\boldsymbol{\theta}}_{i+1}$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{i+1}}$
```

Data processing through the correction law contributes to RLS. The inertia parameter can be corrected by using an update gain matrix. The updated inertia parameter can be represented as follows:

$$\hat{\boldsymbol{\theta}}_i = \hat{\boldsymbol{\theta}}_{i-1} + \boldsymbol{K}_{update}(\boldsymbol{M}_i - \boldsymbol{H}_1 \hat{\boldsymbol{\theta}}_{i-1}) \tag{30}$$

where $\hat{\boldsymbol{\theta}}_i, = \left[\hat{J}_{xx,i} \; \hat{J}_{yy,i} \; \hat{J}_{zz,i} \; \hat{x}_{off,i} \; \hat{y}_{off,i}\right]^T$.

The update gain matrix can be expressed as follows:

$$\boldsymbol{K}_{update} = \left[\boldsymbol{\Sigma}_1 \boldsymbol{h}_1(1)^T \boldsymbol{P} \; \ldots \; \boldsymbol{\Sigma}_5 \boldsymbol{h}_1(5)^T \boldsymbol{P}\right] \tag{31}$$

where $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_i} = \left[\boldsymbol{\Sigma}_1 \; \boldsymbol{\Sigma}_2 \; \boldsymbol{\Sigma}_3 \; \boldsymbol{\Sigma}_4 \; \boldsymbol{\Sigma}_5\right]$
and $\boldsymbol{P} = (\boldsymbol{H}_1 \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_i} \boldsymbol{H}_1^T + \boldsymbol{R}_{RLS}^{-1})^{-1}$.

In (31), two columns, $\boldsymbol{h}_1(4)$ and $\boldsymbol{h}_1(5)$ are not zero vectors during flight. However, $\boldsymbol{h}_1(1)$, $\boldsymbol{h}_1(2)$, and $\boldsymbol{h}_1(3)$ columns can be zero vectors depending on the correction law. For instance, in Figure 4, only column $\boldsymbol{h}_1(3)$ is set to a zero vector, so there is no correction for the z-axis MOI. Therefore, any MOI parameter other than the z-axis was corrected.

Algorithm 6 Observation Matrix for Mass and Sensor Location Estimation

```
**Arguments**: $\boldsymbol{a}_{IMU}$ $\hat{\boldsymbol{\omega}}_i$, $\hat{\boldsymbol{\omega}}_{i+1}$ $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}$, $x_{offset}$, $y_{offset}$
$x = -\left(x_{offset} + {}^{IMU}x_{COM}\right)$
$y = -\left(y_{offset} + {}^{IMU}y_{COM}\right)$
$\dot{\boldsymbol{\omega}} = \frac{1}{dt}\left(\hat{\boldsymbol{\omega}}_{i+1} - \hat{\boldsymbol{\omega}}_i\right)$
$\boldsymbol{\Omega} = [\hat{\boldsymbol{\omega}}_i]_\times \in \mathbb{R}^{3\times 3}$
$\dot{\boldsymbol{\Omega}} = [\dot{\boldsymbol{\omega}}]_\times \in \mathbb{R}^{3\times 3}$
**if** $|\hat{\boldsymbol{\omega}}_i(3)| < \sqrt{\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}(3,3)}$ or $|\hat{\boldsymbol{\omega}}_i(2)| < \sqrt{\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}(2,2)}$ **then**
    $\boldsymbol{h}_2(:,1) = \boldsymbol{0}_{3\times 1}$
**end if**
**if** $|\hat{\boldsymbol{\omega}}_i(1)| < \sqrt{\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}(1,1)}$ or $|\hat{\boldsymbol{\omega}}_i(3)| < \sqrt{\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}(3,3)}$ **then**
    $\boldsymbol{h}_2(:,2) = \boldsymbol{0}_{3\times 1}$
**end if**
**if** $|\hat{\boldsymbol{\omega}}_i(2)| < \sqrt{\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}(2,2)}$ or $|\hat{\boldsymbol{\omega}}_i(1)| < \sqrt{\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_i}(1,1)}$ **then**
    $\boldsymbol{h}_2(:,3) = \boldsymbol{0}_{3\times 1}$
**end if**
$\boldsymbol{H}_2 = \left[\boldsymbol{a}_{IMU} + x\boldsymbol{h}_2(:,1) + y\boldsymbol{h}_2(:,2)\ \boldsymbol{h}_2(:,3)\right]$
**Return** $\boldsymbol{H}_2$
```

Algorithm 7 RLS2 for Mass and Sensor Location Estimation

```
**Arguments**: $\hat{\boldsymbol{\theta}}_{2,i}$. $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,i}}$, $\boldsymbol{H}_2$, $T_i$
where $\hat{\boldsymbol{\theta}}_{2,i} = \left[\hat{m}_i\ {}^{COM}\widehat{mz}_{IMU,i}\right]^T$
and $\boldsymbol{R}_{RLS2} = w_{RLS2}^2, \boldsymbol{I}_{3\times 3}$
$\boldsymbol{T}_i = \left[0\ 0\ T_i\right]^T$
$\boldsymbol{K}_{RLS2} = \boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_2}\boldsymbol{H}_2^T\left(\boldsymbol{H}_2\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_2}\boldsymbol{H}_2^T + \boldsymbol{R}_{RLS2}^{-1}\right)^{-1}$
$\boldsymbol{\theta}_{2,i+1} = \boldsymbol{\theta}_{2,i} + \boldsymbol{K}_{RLS2}\left(\boldsymbol{T}_i - \boldsymbol{H}_2\hat{\boldsymbol{\theta}}_{2,i}\right)$
$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,i+1}} = \left(\boldsymbol{I}_{2\times 2} - \boldsymbol{K}_{RLS2}\boldsymbol{H}_2^T\right)\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,i}}$
**Return** $\hat{\boldsymbol{\theta}}_{2,i+1}$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,i+1}}$
```

D. MASS AND SENSOR LOCATION ESTIMATION

Algorithms 6 and 7 show how the mass and IMU sensor location estimators operate. In Algorithm 6, x and y represent IMU sensor locations in the COM frame. Algorithm 6 inserts the data by checking the angular velocity motion. When the angular velocity along the z-axis or y-axis exhibits no motion, the column of the temporary observation matrix is filled with zero vectors $\boldsymbol{0}_{3\times 1}$. Otherwise, the column vector is not replaced by a zero vector. Finally, according to (20), observation matrix $\boldsymbol{H}_2 \in \mathbb{R}_{3\times 2}$ is inserted. Then, Algorithm 7 obtains the processed observation matrix $\boldsymbol{H}_2$, collective thrust T_i, estimated values $\boldsymbol{\theta}_{2,i}$, and variances $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,i}}$ at the previous step for mass and sensor location, where $\boldsymbol{\theta}_{2,i} = \left[\hat{m}_i\ {}^{COM}\widehat{mz}_{IMU}\right]^T$. By computing the gain matrix $\boldsymbol{K}_{RLS2}$, the estimated values $\hat{\boldsymbol{\theta}}_{2,i+1}$ and variances $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,i+1}}$. The x- and y-coordinates of the sensor location can be computed using COM offset in (20).

E. RLS AND RLS2 WITH LOW PASS FILTER

The algorithm is the same as Algorithms 5 and 7, except for the fact that the observation matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, collective thrust T_i and moment $\boldsymbol{M}_i$ are processed through a LPF. The domains of (10) and (20) can be converted into the Laplace domain as follows:

$$\boldsymbol{M}(s) = \boldsymbol{H}_1(s)\boldsymbol{\theta}_1(s), \tag{32}$$

$$\begin{bmatrix} 0 \\ 0 \\ T(s) \end{bmatrix} = \boldsymbol{H}_2(s)\boldsymbol{\theta}_2(s). \tag{33}$$

In (32) and (33), the transfer function of the low-pass filter can be multiplied by both sides, and the transfer function of (32) and (33) can be represented as

$$\boldsymbol{H}_{LPF}(s)\boldsymbol{M}(s) = \boldsymbol{H}_{LPF}(s)\boldsymbol{H}_1(s)\boldsymbol{\Theta}_1, \tag{34}$$

$$\boldsymbol{H}_{LPF}(s)\begin{bmatrix} 0 \\ 0 \\ T(s) \end{bmatrix} = \boldsymbol{H}_{LPF}(s)\boldsymbol{H}_2(s)\boldsymbol{\Theta}_2. \tag{35}$$

where $\boldsymbol{H}_{LPF} = \frac{1}{(1/\tau)s+1}\boldsymbol{I}_{3\times 3}$ and τ denotes the time constant. By taking the inverse z-transform to the left-hand side of (34) and (35), the moment M and collective thrust T_i can be expressed as

$$\boldsymbol{M}_{LPF}[i+1] = \left(1 - \frac{T_s}{\tau}\right)\boldsymbol{M}_{LPF}[i] + \frac{T_s}{\tau}\boldsymbol{M}_i \tag{36}$$

$$T_{LPF}[i+1] = \left(1 - \frac{T_s}{\tau}\right)T_{LPF}[i] + \frac{T_s}{\tau}T[i] \tag{37}$$

where T_s is the sampling time and $\alpha_{RLS} = T_s/\tau$ is chosen for convenience. Because each component for the observation matrices cannot be superposed immediately, the whole matrix $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ according to [37] and [42] should be processed through a low-pass filter such as

$$\begin{aligned}\boldsymbol{H}_{1,LPF}[i+1] = (1-\alpha_{LPF})\boldsymbol{H}_{1,LPF}[i] \\ + \alpha_{LPF}\boldsymbol{H}_1[i],\end{aligned} \tag{38}$$

$$\begin{aligned}\boldsymbol{H}_{2,LPF}[i+1] = (1-\alpha_{LPF})\boldsymbol{H}_{2,LPF}[i] \\ + \alpha_{LPF}\boldsymbol{H}_2[i].\end{aligned} \tag{39}$$

F. PD CONTROLLER IMPLEMENTATION

As identification requires data acquisition, it is necessary to implement a controller and trajectory generation. We adopted a cascaded PD control algorithm. The position control law is expressed as follows:

$$\begin{aligned}\boldsymbol{f_p} = m(\ddot{\boldsymbol{r}}_{des} + \boldsymbol{K}_{p,p}(\boldsymbol{r}_{des} - {}^{W}\boldsymbol{r}_{des}) \\ + \boldsymbol{K}_{p,d}(\dot{\boldsymbol{r}}_{des} - {}^{W}\dot{\boldsymbol{r}}_{COM}))\end{aligned} \tag{40}$$

where $\boldsymbol{f}_p = \left[f_x\ f_y\ f_z\right]^T$ is the force vector in the world frame. $\boldsymbol{K}_{p,p} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{K}_{p,d} \in \mathbb{R}^{3\times 3}$ are the proportional and derivative gains for the position control law, respectively, and these matrices are diagonal positive definite matrices. ${}^{W}\boldsymbol{r}_{des}$ denotes the desired position expressed in the world frame.

The attitude control law is as follows:

$$\boldsymbol{M}_o = \boldsymbol{K}_{o,p}\begin{bmatrix}\phi_{des}-\phi_s\\ \theta_{des}-\theta_s\\ \psi_{des}-\psi_s\end{bmatrix} - \boldsymbol{K}_{o,d}\begin{bmatrix}p\\ q\\ r\end{bmatrix} + \boldsymbol{r}_{off}\times\begin{bmatrix}0\\ 0\\ \sqrt{\boldsymbol{f}^T\boldsymbol{f}}\end{bmatrix}, \tag{41}$$

$$\phi_{des} = \arcsin\left(\frac{f_x \sin\psi_{des} - f_y\cos\psi_{des}}{\sqrt{\boldsymbol{f}^T\boldsymbol{f}}}\right), \tag{42}$$

$$\theta_{des} = \arcsin\left(\frac{f_x \cos\psi_{des} + f_y\sin\psi_{des}}{\sqrt{\boldsymbol{f}^T\boldsymbol{f}}}\right), \tag{43}$$

$$\phi_s = \arcsin(2(q_y q_z + q_w q_x)), \tag{44}$$

$$\theta_s = \operatorname{atan2}\left(\frac{2(q_x q_z - q_w q_y)}{\cos\phi_s}, \frac{1-2(q_x^2+q_y^2)}{\cos\phi_s}\right), \tag{45}$$

$$\psi_s = -\operatorname{atan2}\left(\frac{2(q_x q_z - q_w q_z)}{\cos\phi_s}, \frac{1-2(q_x^2+q_z^2)}{\cos\phi_s}\right). \tag{46}$$

$\boldsymbol{K}_{o,p} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{K}_{o,d} \in \mathbb{R}^{3\times3}$ denote the proportional gain and derivative gain matrices, respectively, for the orientation control law. Likewise, the two gain matrices comprise diagonal positive-definite matrices. p, q, and r are the elements of the angular velocity along the x, y, and z-axes, respectively. ϕ_{des}, θ_{des}, and ψ_{des} are desired Euler angles (roll, pitch, and yaw), and the Euler angle about $z{-}x'{-}y''$ is chosen. The desired ψ_{des} was created through trajectory generation. ϕ_s, θ_s, ψ_s are the Euler angles of the quadrotor state (roll, pitch, and yaw, respectively). q_x, q_y, q_z, and q_w are elements of the quaternion that can be acquired by IMU or a camera sensor. Finally, the rotor thrust should be clamped because there is a constraint on rotor speed. The inverse dynamic relationship between the rotor and the tuple of the collective thrust and moment is

$$\begin{bmatrix}F_{R1}\\ F_{R2}\\ F_{R3}\\ F_{R4}\end{bmatrix} = \frac{1}{4}\begin{bmatrix}1 & \frac{\sqrt{2}}{l} & \frac{\sqrt{2}}{l} & \frac{C_T}{C_M}\\ 1 & \frac{\sqrt{2}}{l} & -\frac{\sqrt{2}}{l} & -\frac{C_T}{C_M}\\ 1 & -\frac{\sqrt{2}}{l} & -\frac{\sqrt{2}}{l} & \frac{C_T}{C_M}\\ 1 & -\frac{\sqrt{2}}{l} & \frac{\sqrt{2}}{l} & -\frac{C_T}{C_M}\end{bmatrix}\begin{bmatrix}T\\ M_x\\ M_y\\ M_z\end{bmatrix} \tag{47}$$

where $T = \sqrt{\boldsymbol{f}_p^T\boldsymbol{f}_p}$,

$M_x = M_o(1), M_y = M_o(2),$, and $M_z = M_o(3)$.

The clamping law is

$$F_{Ri} = \min\left(\max\left(F_{Ri}, F_{R,min}\right), F_{R,max}\right). \tag{48}$$

In (48), $F_{R,min}$ and $F_{R,max}$ represent the minimum and maximum rotor thrust, respectively.

G. TRAJECTORY GENERATION

This study adopts the triple integral method for position trajectory generation and the double integral method for yaw trajectory. The triple integral method considers the control input as a jerk. Specifically, to reach the target position, it should be integrated thrice. In contrast, the control input of the double integral is acceleration. To achieve the yaw target, the control input must be integrated twice. The two methods are divided into bang-bang control and bang-off-bang control depending on constraints, target position, and yaw target. This study adopts only the bang-off-bang control to create a trajectory for position and yaw trajectory generation. Because this study mainly focuses on inertia parameter identification, we do not provide a detailed description of trajectory generation.

IV. SIMULATION ENVIRONMENT SETUP

A. SENSOR DATA GENERATION

The software we used was MATLAB 2021 a, and the simulator was based on the customized Runge–Kutta fourth order. The IMU sensor model was implemented using the `imusensor` function. To reflect on a real IMU sensor, we analyzed IMU sensor noise first. The IMU sensor data were acquired using Pixhawk 4. Noise density should be reflected to generate fake IMU data. The definition of gyroscope noise density is

$$ND_{gyro} = \frac{\sigma_\omega}{\sqrt{SR}} \tag{49}$$

where $\quad \sigma_\omega = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\omega_{IMU}[i] - \omega_{gnd}[i],\right)^2}.$

In (49), SR, σ_ω and N denote the sampling rate, root-mean-square error of the angular velocity for each component, and the number of samples, respectively. ND denotes the noise density abbreviation. ω_{gnd} is the ground truth of angular velocity. In the static state, the ground truth of the angular velocity was zero. The noise density of the linear acceleration can be defined as

$$ND_{acc} = \frac{\sigma_a}{\sqrt{SR}} \tag{50}$$

where $\quad \sigma_a = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(a_{IMU}[i] - a_{gnd}[i]\right)^2}.$

In (50), σ_a denotes the root-mean-square error of the linear acceleration. The ground truths of the acceleration element for the static environment were set as $a_x = 0\ m/s^2$, $a_y = 0\ m/s^2$, and $a_z = 9.81\ m/s^2$. Similarly, the root-mean-square error was tested under static conditions. The sampling rate was set to 100 Hz.

Table 1 shows the angular velocity's RMSE and noise density for IMU and the simulated noise density and resolution setting for the angular velocity, which are given.

TABLE 1. RMSE and noise density of gyro data for IMU and simulation setup for gyro data.

	p	q	r
$RMSE_{IMU}$ $(^\circ/s)$	0.0397	0.0359	0.0314
ND_{IMU} $\left(\frac{^\circ}{s\sqrt{Hz}}\right)$	0.00397	0.00359	0.00314
ND_{sim} $\left(\frac{^\circ}{s\sqrt{Hz}}\right)$	0.006		
$resolution$ $(^\circ/s)$	0.001		

TABLE 2. RMSE and noise density of linear acceleration for IMU and simulation setup for linear acceleration.

	a_x	a_y	a_z
RMSE (m/s^2)	0.0116	0.0101	0.0098
ND $\left(\frac{m}{s^3\sqrt{Hz}}\right)$	0.00116	0.00101	0.00098
ND_{sim} $\left(\frac{m}{s^3\sqrt{Hz}}\right)$	0.004		
$resolution$ (m/s^2)	0.001		

TABLE 3. Quadrotor model MOI and COM offset information.

	J_{xx} $(g\cdot m^2)$	J_{yy} $(g\cdot m^2)$	J_{zz} $(g\cdot m^2)$	x_{off} (mm)	y_{off} (mm)
MOI and COM offset	50	50	60	8	-21

TABLE 4. Mass and sensor location information.

	m (g)	$COM_{x_{IMU}}$ (mm)	$COM_{y_{IMU}}$ (mm)	$COM_{z_{IMU}}$ (mm)
mass and sensor location	500	-18	21	-11

We assume that the noise of each rotor thrust obtained through RPM data follows a zero-mean Gaussian normal distribution:

$$w_{F_{Ri}} \sim \mathcal{N}(0, \sigma^2(N^2)) \qquad (51)$$

where $\quad i = 1, 2, 3, 4.$

In (51), σ^2 and i represent the variance of the thrust noise and index of the rotor, respectively. The noise level is approximately ±0.5 g with 150 g rotor thrust when testing thrust with constant pulse width modulation (PWM). By empirically adjusting the noise around that level, the standard deviation σ of each thrust noise was nearly 0.0022 N; therefore, the noise of each thrust in the simulator was set to 0.0022 N.

B. SIMULATION MODEL

The simulation model comprises a dynamic and kinematic model in (6)–(9), computed through the customized Runge–Kutta fourth order. The noise vector of the rotor thrust was added to each rotor thrust after the inverse dynamic in (39) was taken for the tuple of the collective thrust and moment. In the simulation, the ratio of C_T to C_M was 0.7. Table 3 lists the MOI and COM offset values. The mass and sensor locations in the COM frame are presented in Table 4. The COM of quadrotor itself is assumed to be at the CG. The quadrotor itself and the package is assumed to weigh 200 and 300 grams, respectively, and the package fixed to the quadrotor is assumed to cause the CG of the vehicle including the package to move to 8 and -21 millimeters along x and y axis relative to COM frame.

TABLE 5. Control gain and trajectory generator setup.

Position control gain	$\boldsymbol{K}_{p,p} = diag\left(\begin{bmatrix}3 & 3 & 7\end{bmatrix}\right)$ $\boldsymbol{K}_{p,d} = diag\left(\begin{bmatrix}2 & 2 & 4\end{bmatrix}\right)$
Attitude control gain	$\boldsymbol{K}_{o,p} = diag\left(\begin{bmatrix}3 & 3 & 3\end{bmatrix}\right)$ $\boldsymbol{K}_{o,d} = diag\left(\begin{bmatrix}0.5 & 0.5 & 0.5\end{bmatrix}\right)$
Position trajectory constraint	$j_{max} = 0.5\, m/s^3$ $a_{max} = 0.2\, m/s^2$ $v_{max} = 0.2\, m/s$
Yaw trajectory constraint	$\ddot{\psi}_{max} = 15^\circ/s^2$ $\dot{\psi}_{max} = 20^\circ/s$

C. CONTROLLER AND TRAJECTORY GENERATOR SETTING

Table 5 shows the PD gain matrices for the position control and attitude control, as well as the constraints for the position trajectory generator and yaw trajectory generator. First, the position trajectory for hovering was applied. The first target height was 1 m without any horizontal motion. Next, the second position trajectory should be applied to excite sufficiently for the estimator to infer the dynamic parameters. The second target position was applied 1 m to the x and y axes

TABLE 6. Estimator initial values and variance setup.

Minimum MOI of KF	$\boldsymbol{J}_{min} = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^T \; g \cdot m^2$
Initial value of angular velocity	$\hat{\boldsymbol{\omega}}_0 = \mathbf{0}_{3\times1}$
Variance of KF	$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\omega}}_0} = 0.010^2 \, \boldsymbol{I}_{3\times3} \in \mathbb{R}^{3\times3}$
Constraint of KF	$c = 0.010$
Initial values of MOI of RLS	$\hat{J}_{xx,0} = 1 \, g \cdot m^2, \quad \hat{J}_{yy,0} = 1 \, g \cdot m^2, \quad \hat{J}_{zz,0} = 1 \, g \cdot m^2$
Initial values for COM offset of RLS	$\hat{x}_{off} = 0 \, mm \quad \hat{y}_{off} = 0 \, mm$
Initial variance of RLS	$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_0} = diag\left(\begin{bmatrix}1000^2 & 1000^2 & 1000^2 & 100^2 & 100^2\end{bmatrix}\right) \in \mathbb{R}^{5\times5}$
Initial values of mass of RLS2	$\hat{m} = 100 \, g$
Initial values of sensor location of RLS2	${}^{COM}\hat{z}_{IMU} = 0 \, mm$
Initial variance of RLS2	$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\theta}}_{2,0}} = diag\left(\begin{bmatrix}500^2 & 100^2\end{bmatrix}\right) \in \mathbb{R}^{2\times2}$
Low-pass filter setting	$\alpha_{RLS} = 0.2$
Estimator Enable condition	$a_{z,allow} = 0.1 \, m/s^2$

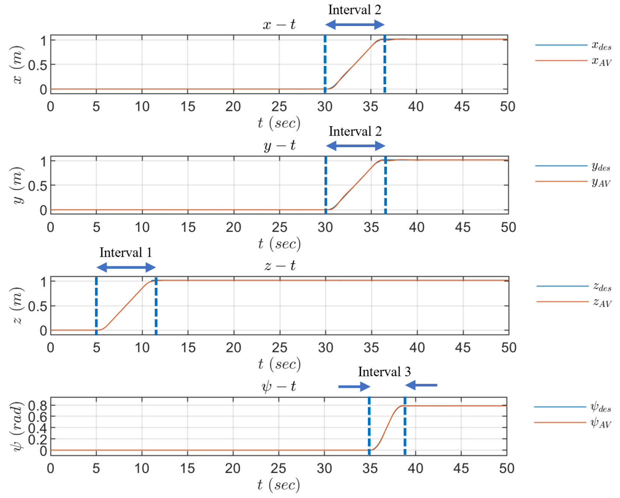

FIGURE 5. Generated trajectory data for inertia parameter estimation.

simultaneously while maintaining the height of the first target position. The final destination of this yaw trajectory was 45° (0.7854 *rad*) from the initial yaw.

D. GENERATED FLIGHT DATA

In Figure 5, the blue and red lines denote the desired position and yaw trajectory generated through the triple integral method and the aerial vehicle's tracked position and yaw data. During Interval 1, the vehicle reached the designated height but did not move in the horizontal direction. In contrast, a horizontal motion was applied during Interval 2. During Interval 3, yaw motion was created.

V. SIMULATION RESULTS

The estimator setup is presented in Table 8 before explaining the simulation analysis. Table 8 shows the initial values and variances for KF, RLS, and RLS2, as well as the low-pass filter coefficients and estimator conditions.

A. ANGULAR ACCELERATION DATA ANALYSIS

The angular acceleration data are shown in Figure 6. The blue, red, and yellow lines represent the differentiation of angular velocity measured by IMU, the one estimated by KF, and the ground truth, respectively. As mentioned in the generated flight data subsection, during Interval 1, only vertical motion exists; therefore, the angular acceleration data show insufficient excitation for the dynamic parameter estimator to infer the parameter. Angular acceleration along the x- and y-axes shows that oscillation occurs during Interval 2. The one along the z-axis oscillates enough for the estimator to correct the z-axis MOI parameter during Interval 3. The differentiation of angular velocity estimated by KF tends to be noisy before Interval 2 or 3. After the beginning of Interval 2, the noise of the x- and y-axis angular accelerations through KF becomes less than that through IMU. In addition, the noise of the z-axis through KF does so after the start of Interval 3. The noise is mitigated when the oscillation starts because the KF puts more confidence in the process model than in the measurement model, and the process model has less noise than the IMU sensor. Table 7 shows each component's root-mean-square error (RMSE) from 0 to 50 s. Angular acceleration through KF is more accurate than that through IMU because, while RMSE of the KF data is approximately 5 to 6 rad/s^2, that of IMU takes about 7.

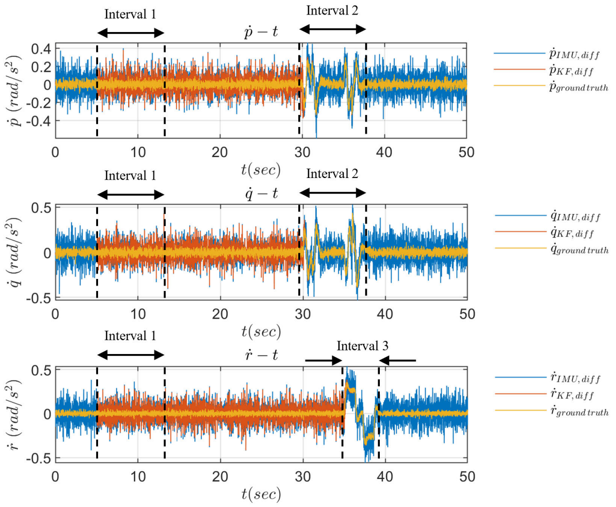

FIGURE 6. Comparison of angular acceleration along x, y, and z axis.

TABLE 7. RMSE of angular acceleration.

RMSE	$\dot{p}\ (rad/s^2)$	$\dot{q}\ (rad/s^2)$	$\dot{r}\ (rad/s^2)$
IMU	7.35	7.46	7.43
KF	5.26	5.44	5.80

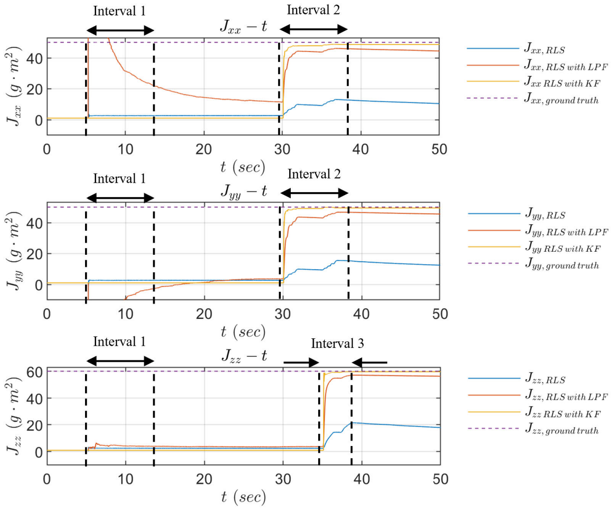

FIGURE 7. MOI value estimation results along x, y, and z axis.

TABLE 8. MOI estimation result.

Relative Error (%)	$J_{xx}\ (g\cdot m^2)$ (Est/True)	$J_{yy}\ (g\cdot m^2)$ (Est/True)	$J_{zz}\ (g\cdot m^2)$ (Est/True)
RLS	-79.2 (%) (10.3/ 50)	-75.1 (%) (12.4/ 50)	-70.1 (%) (17.9/ 60)
RLS with LPF	-11.0 (%) (44.4/ 50)	-8.90 (%) (45.5/ 50)	-6.15 (%) (56.3/60)
Observability-aware RLS with KF	-2.66 (%) (48.6/ 50)	-1.10 (%) (49.4/ 50)	-0.60 (%) (59.6/ 60)

B. MOI ESTIMATION RESULTS

Figure 7 shows that the blue, red, yellow, and dotted lines are the estimated values of MOI through RLS, RLS with LPF, and RLS with KF and ground truth values. Even though Interval 1 does not have sufficient oscillation to correct MOI in Figure 6, RLS and RLS with LPF start to correct MOI parameters during that interval. Eventually, the estimation of the x-axis MOI parameter abruptly increased during Interval 1, and that of the y-axis sharply decreased. However, observability-aware RLS with KF does not correct MOI values at that interval. It corrects the x- and y-axis MOI values during Interval 2 and the z-axis MOI during Interval 3. Therefore, according to Figures 6 and 7, the correction law works. Consequently, the relative error for MOI estimation in Table 8 is much lower for the observability-aware inertia parameter estimation method than for RLS or RLS with LPF. The observability-aware inertia estimation method showed that the absolute value of the relative error was within 3 percent.

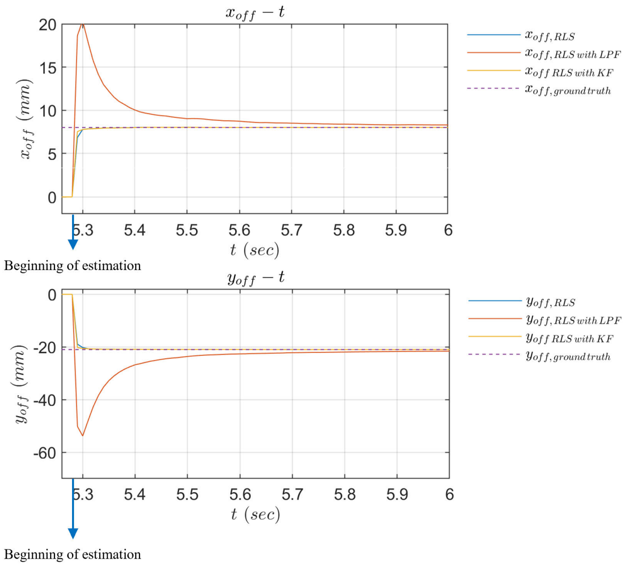

FIGURE 8. COM offset estimation results.

TABLE 9. COM offset estimation result.

Relative Error (%)	$x_{off}\ (mm)$ (Est/True)	$y_{off}\ (mm)$ (Est/True)
RLS	-0.0225 (%) (7.9/ 8)	0.0092 (%) ($-21/\ -21$)
RLS with LPF	0.030 (%) (8/ 8)	0.060 (%) ($-21/\ -21$)
Observability-aware RLS with KF	-0.060 (%) (8/ 8)	0.018 (%) ($-21/\ -21$)

C. COM OFFSET RESULTS

In Figure 9, at 5.28 s, the estimator starts to correct COM offset values. This is because the data processing and dynamic parameter estimator RLS begin to correct these values when the estimator is activated. COM offset estimation using RLS with LPF has surge values, whereas the observability-aware inertia parameter identification method and RLS do not. This incorrect COM offset estimation was attributed to an inaccurate MOI estimation. The final estimation result of COM offset is not different from RLS, RLS with LPF, and the

TABLE 10. COM offset estimation result.

Relative Error (%)	$m\ (g)$ (Est/True)	${}^{COM}x_{IMU}$ (mm) (Est/True)	${}^{COM}y_{IMU}$ (mm)	${}^{COM}z_{IMU}$ (mm) (Est/True)
RLS	-0.003 (%) $(499/500)$	-0.01 (%) $(-17.9/-18)$	0.009 (%) $(21/21)$	-116.8 (%) $(1.8/-11)$
RLS with LPF	-0.003 (%) $(499/500)$	1.35 (%) $(-18/-18)$	0.06 (%) $(21/21)$	-8.60 (%) $(-10/-11)$
Observability-aware RLS with KF	-0.003 (%) $(499/500)$	2.7 (%) $(-17.9/-18)$	1.7 (%) $(21/21)$	-4.7 (%) $(-11.5/-11)$

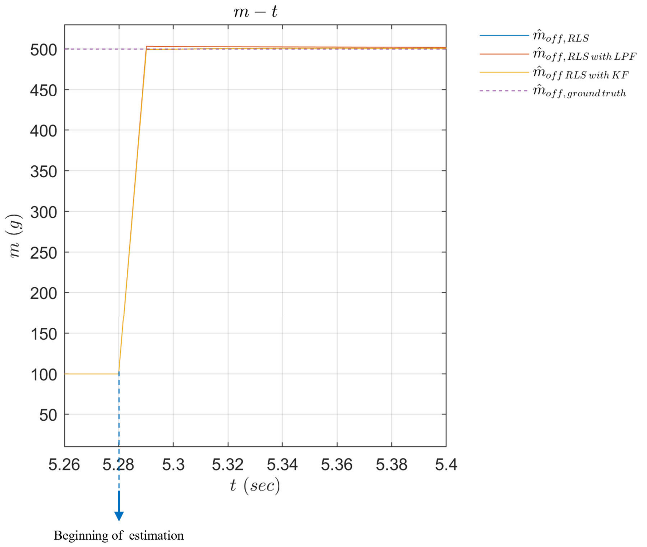

FIGURE 9. Mass estimation results.

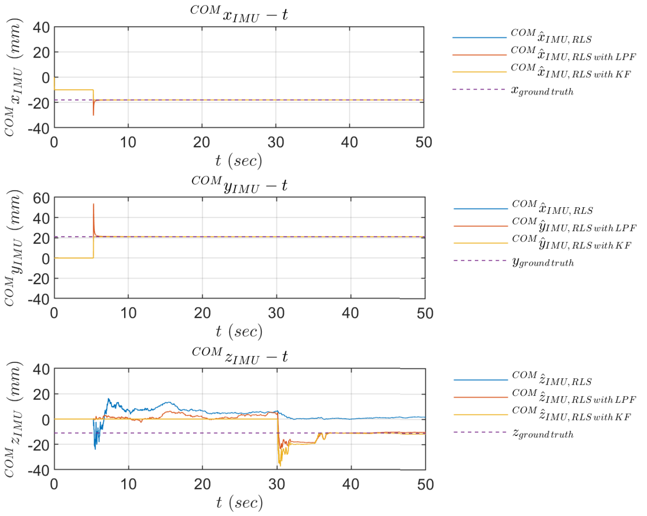

FIGURE 10. Sensor location estimation results.

observability-aware inertia parameter identification method. Table 9 shows that the performances of the three methods are approximately the same because the observability of COM offset is always guaranteed during flight.

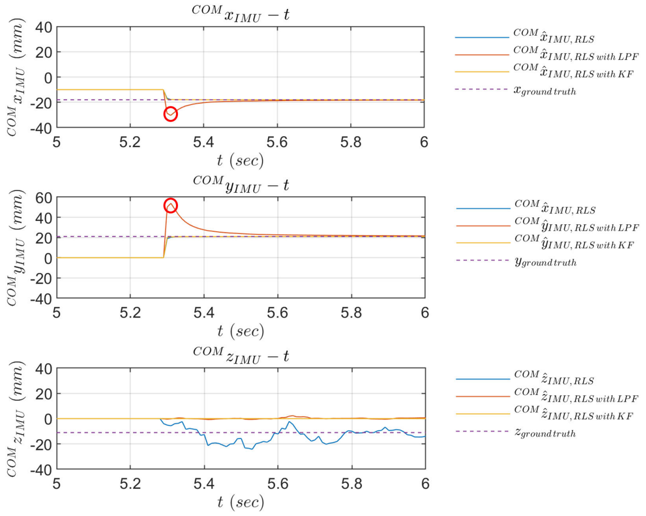

FIGURE 11. The red circle showing that the surge of the estimated sensor location for RLS with LPF method.

D. MASS ESTIMATION RESULTS

The fact that the collective thrust is always greater than zero makes the estimation of mass and COM offset consistently observable. Therefore, there is no significant difference in the performance between RLS, RLS with LPF, and our proposed method, as shown in Figure 9. In Table 10, the relative errors of the mass estimation through the three methods are almost the same.

E. SENSOR LOCATION ESTIMATION RESULTS

With the RLS method, the relative height between COM and IMU is incorrect in Figure 10; however, because COM offset can be easily estimated, the inference on the x and y coordinates of IMU in COM can be performed accurately. In contrast, RLS with a low-pass filter has a surge in the sensor location about x and y, as shown in Figure 11. This is because the COM offset was estimated inaccurately at the beginning of the inference. Our approach does not show surge results for the x and y coordinates, and the observability-aware inertia parameter estimation approach estimates the relative height at the end of all trajectories. In Table 9, the final estimated values for the mass and the x and y components of the sensor location have no significant differences among the three methods. However, RLS tends to be susceptible to the effect of angular acceleration noise; therefore, the relative height is inaccurate. In contrast, our approach and RLS with LPF showed absolute values of relative error within 8 and 5 percent, respectively.

VI. CONCLUSION

This study derived the process model's variance to reflect the MOI's uncertainty. This derived variance makes it possible to adjust the proportion of confidence in the process model to that of the measurement model. Through KF, the noise amplitude significantly decreases when sufficient excitation is applied to the aerial vehicle. In addition, MOI correction law makes RLS aware of the appropriate time to correct the parameter. Our approach has no sharp decrease or increase in MOI, COM offset, or sensor location estimation compared to RLS with LPF and RLS. Therefore, the stable flight would be possible by using the real-time observability-aware inertia parameter identification when the quadrotor transports the non-symmetric package or building block for construction.

In addition, this study shows that the dynamic parameter estimator infers on parameters even during the time-optimal trajectory instead of the extreme one to make the parameter observable. Thus, our overall algorithm pipeline is expected to save energy and guarantee control stability while considering position and attitude control coupling. Hence, applying the coupling between the translational dynamic and orientational models with consideration of the COM offset would be possible through the algorithm of the observability-aware RLS with KF.

In future work, adaptive control and trajectory generation considering the coupling between position control and attitude control will be covered. In addition, a disturbance observer (DOB) will be used to reject the disturbance by combining the observability-aware inertia parameter identification. Moreover, the estimation and compensation of the reaction torque generated by the movement of the manipulator will be dealt with in future work.

ACKNOWLEDGMENT

The authors would like to thank Editage (www.editage.co.kr) for English language editing.

REFERENCES

[1] H. Lee and H. J. Kim, ''Estimation, control, and planning for autonomous aerial transportation,'' *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3369–3379, Apr. 2017.

[2] G. Loianno and V. Kumar, ''Cooperative transportation using small quadrotors using monocular vision and inertial sensing,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 680–687, Apr. 2018.

[3] J. Zhang and S. Singh, ''Laser-visual-inertial odometry and mapping with high robustness and low drift,'' *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, Aug. 2018.

[4] G. Loianno, J. Thomas, and V. Kumar, ''Cooperative localization and mapping of MAVs using RGB-D sensors,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 4021–4028.

[5] L. F. Luque-Vega, B. Castillo-Toledo, A. Loukianov, and L. E. González-Jiménez, ''Power line inspection via an unmanned aerial system based on the quadrotor helicopter,'' in *Proc. IEEE 17th Medit. Electrotech. Conf.*, Beirut, Lebanon, Apr. 2014, pp. 393–397.

[6] K. Máthé, L. Busoniu, L. Barábas, C.-I. Iuga, L. Miclea, and J. Braband, ''Vision-based control of a quadrotor for an object inspection scenario,'' in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Arlington, VA, USA, Jun. 2016, pp. 849–857.

[7] P. E. Pounds, ''Design, construction and control of a large quadrotor micro air vehicle,'' Ph.D. dissertation, Australian Nat. Univ., Canberra, ACT, Australia, 2007.

[8] Q. Lindsey, D. Mellinger, and V. Kumar, ''Construction with quadrotor teams,'' *Auton. Robots*, vol. 33, no. 3, pp. 323–336, Jun. 2012.

[9] H. Lee, H. Kim, W. Kim, and H. J. Kim, ''An integrated framework for cooperative aerial manipulators in unknown environments,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2307–2314, Jul. 2018.

[10] H. Lee, S. Kim, and H. J. Kim, ''Control of an aerial manipulator using online parameter estimator for an unknown payload,'' in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Gothenburg, Sweden, Aug. 2015, pp. 316–321.

[11] E. Fresk, D. Wuthier, and G. Nikolakopoulos, ''Generalized center of gravity compensation for multirotors with application to aerial manipulation,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 4424–4429.

[12] G. Baraban, M. Sheckells, S. Kim, and M. Kobilarov, ''Adaptive parameter estimation for aerial manipulation,'' in *Proc. Amer. Control Conf. (ACC)*, Denver, CO, USA, Jul. 2020, pp. 614–619.

[13] P. Outeiro, C. Cardeira, and P. Oliveira, ''Multiple-model adaptive control architecture for a quadrotor with constant unknown mass and inertia,'' *Aerosp. Sci. Technol.*, vol. 117, Oct. 2021, Art. no. 106899.

[14] G. Antonelli, E. Cataldi, F. Arrichiello, P. R. Giordano, S. Chiaverini, and A. Franchi, ''Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances,'' *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 248–254, Jan. 2018.

[15] S. J. Lee, S. H. Kim, and H. J. Kim, ''Robust translational force control of multi-rotor UAV for precise acceleration tracking,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 562–573, Apr. 2020.

[16] R. Pérez-Alcocer and J. Moreno-Valenzuela, ''Adaptive control for quadrotor trajectory tracking with accurate parametrization,'' *IEEE Access*, vol. 7, pp. 53236–53247, 2019.

[17] W. Alqaisi and C. Z. El-Bayeh, ''Adaptive control based on radial base function neural network approximation for quadrotor,'' in *Proc. 17th Annu. Syst. Syst. Eng. Conf. (SOSE)*, Rochester, NY, USA, Jun. 2022, pp. 214–219.

[18] T. Maki, M. Zhao, F. Shi, K. Okada, and M. Inaba, ''Model reference adaptive control of multirotor for missions with dynamic change of payloads during flight,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, May 2020, pp. 7433–7439.

[19] M. Burri, M. Bloesch, Z. Taylor, R. Siegwart, and J. Nieto, ''A framework for maximum likelihood parameter identification applied on MAVs,'' *J. Field Robot.*, vol. 35, no. 1, pp. 5–22, Jun. 2017.

[20] D.-H. Kim, D.-G. Choi, and H.-S. Oh, ''Inertia estimation of spacecraft based on modified law of conservation of angular momentum,'' *J. Astron. Space Sci.*, vol. 27, no. 4, pp. 353–357, Dec. 2010.

[21] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, ''Design, modeling, estimation and control for aerial grasping and manipulation,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 2668–2673.

[22] M. Vandyke, J. Schwartz, and C. Hall, ''Unscented Kalman Filtering for spacecraft attitude state and parameter estimation,'' *Adv. Astronaut. Sci.*, vol. 119, pp. 217–228, Jan. 2004.

[23] N. Abas, A. Legowo, and R. Akmeliawati, ''Parameter identification of an autonomous quadrotor,'' in *Proc. 4th Int. Conf. Mechatronics (ICOM)*, Kuala Lumpur, Malaysia, May 2011, pp. 1–8.

[24] R. Munguía, S. Urzua, and A. Grau, ''EKF-based parameter identification of multi-rotor unmanned aerial vehicles models,'' *Sensors*, vol. 19, no. 19, p. 4174, Sep. 2019.

[25] N. Abas, A. Legowo, Z. Ibrahim, N. Rahim, and A. M. Kassim, ''Modeling and system identification using extended Kalman filter for a quadrotor system,'' *Appl. Mech. Mater.*, vols. 313–314, pp. 976–981, Mar. 2013.

[26] V. Wuest, V. Kumar, and G. Loianno, ''Online estimation of geometric and inertia parameters for multirotor aerial vehicles,'' in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 1884–1890.

[27] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "IMU-based inertia estimation for a quadrotor using Newton–Euler dynamics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 3861–3867, Jul. 2020.

[28] C. Böhm, M. Scheiber, and S. Weiss, "Filter-based online system-parameter estimation for multicopter UAVs," in *Proc. Robot., Sci. Syst. (RSS)*, Jul. 2021, pp. 1–9.

[29] J. Gośliński, A. Kasiński, W. Giernacki, P. Owczarek, and S. Gardecki, "A study on coaxial quadrotor model parameter estimation: An application of the improved square root unscented Kalman filter," *J. Intell. Robot. Syst.*, vol. 95, no. 2, pp. 491–510, May 2018.

[30] Y. Jiali and Z. Jihong, "An angular acceleration estimation method based on the complementary filter theory," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, Taiwan, May 2016, pp. 1–6.

[31] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Trans. Robot. Autom.*, vol. 13, no. 5, pp. 730–740, Oct. 1997.

[32] K.-J. Park, "Fourier-based optimal excitation trajectories for the dynamic identification of robots," *Robotica*, vol. 24, no. 5, pp. 625–633, Mar. 2006.

[33] J. Vantilt, E. Aertbeliën, F. De Groote, and J. De Schutter, "Optimal excitation and identification of the dynamic model of robotic systems with compliant actuators," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 2117–2124.

[34] J. Jin and N. Gans, "Parameter identification for industrial robots with a fast and robust trajectory design approach," *Robot. Comput.-Integr. Manuf.*, vol. 31, pp. 21–29, Feb. 2015.

[35] J. Swevers, W. Verdonck, and J. D. Schutter, "Dynamic model identification for industrial robots," *IEEE Control Syst.*, vol. 27, no. 5, pp. 58–71, Oct. 2007.

[36] M. Ekal and R. Ventura, "On the accuracy of inertial parameter estimation of a free-flying robot while grasping an object," *J. Intell. Robot. Syst.*, vol. 98, no. 1, pp. 153–163, Apr. 2020.

[37] I. Lopez-Sanchez, J. Montoya-Cháirez, R. Pérez-Alcocer, and J. Moreno-Valenzuela, "Experimental parameter identifications of a quadrotor by using an optimized trajectory," *IEEE Access*, vol. 8, pp. 167355–167370, 2020.

[38] K. Albee, M. Ekal, R. Ventura, and R. Linares, "Combining parameter identification and trajectory optimization: Real-time planning for information gain," in *Proc. Symp. Adv. Space Technol. Robot. Autom. (ASTRA)*, Noordwijk, The Netherlands, Jun. 2019.

[39] C. Böhm, G. Li, G. Loianno, and S. Weiss, "Observability-aware trajectories for geometric and inertial self-calibration," in *Proc. Robot., Sci. Syst. (RSS)*, 2020.

[40] A. J. Krener and K. Ide, "Measures of unobservability," in *Proc. 48th IEEE Conf. Decis. Control (CDC)*, Shanghai, China, Dec. 2009, pp. 6401–6406.

[41] L. Kevin and P. Frank, "Rigid-body motions," in *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2017, ch. 3, pp. 77–78.

[42] F. Reyes and R. Kelly, "Experimental evaluation of identification schemes on a direct drive robot," *Robotica*, vol. 15, no. 5, pp. 1014–1019, Nov. 1997.

GEONWOO KWON received the bachelor's degree in mechanical engineering from Yonsei University, South Korea, in 2018, where he is currently pursuing the Ph.D. degree in robotics and control. His main research interests include adaptive control, trajectory planning, and visual-inertial systems (VINS).

JAEGAK LEE received the bachelor's degree in mechanical engineering from Yonsei University, South Korea, in 2016, where he is currently pursuing the Ph.D. degree in robotics and control. His research interests include adaptive control and bio-mimetic robot application.

HYUNSEOK YANG received the bachelor's degree from the Department of Mechanical Engineering, Yonsei University, Seoul, in 1984, and the M.S. and Ph.D. degrees from the Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), in 1988 and 1993, respectively. He is currently a Professor with the Department of Mechanical Engineering, Yonsei University and coordinates research on motion control and robotics.