Journal of the Korea Convergence Society
Vol. 8. No. 4, pp. 9-18, 2017 https://doi.org/10.15207/JKCS.2017.8.4.009

System Identification of Quadrotor IT Convergence UAV using Batch and RLS Estimation Methods

Sunghun Jung
Department of Drone System, Chodang University

배치추정기법과 RLS추정기법을 사용한 쿼드로터 IT융합 무인항공기 시스템식별

정성훈
초당대학교 항공학부 드론학과

Abstract UAVs began to be actively applied to so-called 3D jobs, including the autonomous exploration, investigation, mapping, search and rescue, etc. since the mid-2000s. With this global trend, having a precise controllability of the UAV will certainly revolutionize the life of the modern human in the aspect of tremendous applications of the UAV. In the first part, a simplified dynamic model of the UAV identified using system identification techniques is compared with the previously built time-discrete linear model. In the second part, the three parameters of the dynamic model are estimated using the batch and RLS methods. Angular acceleration data of the quadrotor UAV at the hovering maneuver are analyzed and shown to be converging at all time. Also, according to the quadrotor flight data from both experiments and MATLAB simulations, the batch estimation method turns out to be more accurate than the RLS estimation method based on the comparison of final parameter values.

• **Key Words** : Batch, Quadrotor, RLS, System Identification, UAV

요 약 무인항공기는 2000년대 중반부터 탐색, 조사, 매핑, 수색 및 구조 등의 3D 작업에 적극적으로 사용되기 시작했다. 이러한 세계적인 추세에 따라, 무인항공기의 정밀한 제어는 엄청난 응용 산업들의 측면에 있어서 혁명을 가져올 것이다. 논문의 첫 번째 파트에서는 시스템식별 기법을 사용하여 간략화 된 무인항공기의 모델을 이전의 이산시간 선형모델과 비교분석 한다. 두 번째 파트에서는 동적 모델의 세 가지 변수가 배치추정기법과 RLS추정기법을 사용하여 추정된다. 쿼드로터 무인항공기 호버링 기동 시의 각가속도 데이터가 항상 수렴한다고 분석되었다. 또한 실험 및 MATLAB 시뮬레이션의 쿼드로터 무인항공기 비행 데이터에 의하면, 배치추정기법이 RLS추정기법보다 더 정확하다고 판명되었다.

• **주제어** : 배치, 쿼드로터, RLS, 시스템식별, 무인항공기

*Corresponding Author : 정성훈(jungx148@gmail.com)
Received February 14, 2017 Revised March 16, 2017
Accepted April 20, 2017 Published April 28, 2017

1. Introduction

Flying objects, such as the unmanned aerial vehicles (UAV), are getting to be a necessary item which a nation, a company, and a person should possess to utilize in various fields [1].

The UAVs, or also called as drones, are attracting attention as one of the major industries that will be popular in the near future. The UAVs, which have limitless application areas, are considered as one of the leading players in the fourth industrial revolution, that is, in the era of physical, digital, and biological technology convergence [2,3,4].

New light on the value of the UAV has acted as a catalyst to further highlight the importance of the UAV. In fact, not only the UAVs, but also other unmanned systems are actively being researched to prepare the fourth industrial revolution [5,6].

With this global trend, having a precise controllability of the UAV will certainly revolutionize the life of the modern human in the aspect of tremendous applications of the UAV.

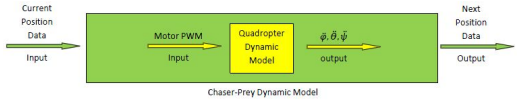
Throughout the paper, a quadrotor UAV dynamic model will be simplified and verified using two methods, including batch and recursive least squares (RLS) methods for the system identification [7,8,9].

The flow of this paper is as follows. Section 2 explains the UAV governing model and Section 3 describes the setup for the data acquisition. Section 4 illustrates parameter estimation and Section 5 shows the convergence of estimates. From Section 6 to Section 8, we show the conditional status of an inverted matrix using singular value and the signal to noise ratio (SNR), the robustness of system identification using the noise probability density function (PDF), and real-time and closed loop state and parameter estimation. Lastly, Section 9 contains the conclusion of this paper.

2. UAV Governing Model

The same governing models of quadrotor UAV shown in [10] are used in this paper. Dynamic models of a quadrotor UAV are very well known and most quadrotor models are appearing to be very similar. A minor difference is to use the z-axis as going upward or downward.

When a path waypoint is applied to the system as an input, the quadrotor dynamic model computes required motor PWM and results in angular accelerations of the roll ($\ddot{\phi}$), pitch ($\ddot{\theta}$), and yaw ($\ddot{\psi}$). The system eventually results in a next position waypoint ([Fig. 1]).



[Fig. 1] Diagram of quadrotor UAV model

The quadrotor dynamic model derived in [5] is,

$$\begin{aligned}
\ddot{\phi} &= \dot{\theta}\dot{\psi}\left(\frac{I_y - I_x}{I_x}\right) - \frac{J}{I_x}\dot{\theta}\Omega + \frac{l}{I_x}U_1, \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) - \frac{J}{I_y}\dot{\phi}\Omega + \frac{l}{I_y}U_2, \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{l}{I_z}U_3,
\end{aligned} \tag{1}$$

in which the system inputs, U_1, U_2, U_3, and Ω, can be rewritten as

$$\begin{aligned}
U_1 &= b\left(\Omega_4^2 - \Omega_2^2\right), \\
U_2 &= b\left(\Omega_3^2 - \Omega_1^2\right), \\
U_3 &= d\left(\Omega_3^2 + \Omega_1^2 - \Omega_4^2 - \Omega_2^2\right), \\
\Omega &= \Omega_4 + \Omega_2 - \Omega_3 - \Omega_1,
\end{aligned} \tag{2}$$

where R is the rotation matrix, ϕ is the roll angle, θ is the pitch angle, ψ is the yaw angle, Ω_i is the ith rotor speed, $I_{x,y,z}$ is the body inertia along x, y, and z-axis, J is the propeller inertia, b is the thrust factor, d is the drag factor, and l is the propeller length.

Though the Eq. 1 contains actuator actions, gyroscopic effects, and propeller rotations, the influence of actuator actions is the only important effects among three influences. So, the Eq. 1 can be simplified as,

$$\ddot{\phi} = \frac{l}{I_x} U_1, \quad (3)$$
$$\ddot{\theta} = \frac{l}{I_y} U_2,$$
$$\ddot{\psi} = \frac{l}{I_z} U_3.$$

By substituting the Eq. 2 into the Eq. 3, we can finally achieve

$$\ddot{\phi} = \frac{lb}{I_x}\left(\Omega_4^2 - \Omega_2^2\right), \quad (4)$$
$$\ddot{\theta} = \frac{lb}{I_y}\left(\Omega_3^2 - \Omega_1^2\right),$$
$$\ddot{\psi} = \frac{ld}{I_z}\left(\Omega_3^2 + \Omega_1^2 - \Omega_4^2 - \Omega_2^2\right).$$

3. Setup for the Data Acquisition

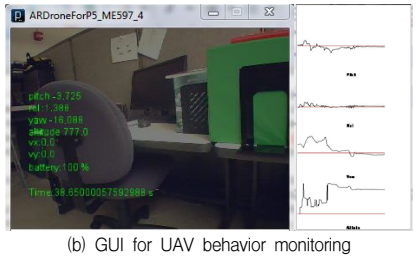
In this paper, the main control objective is to make a quadrotor UAV (AR.Drone [11]) hovers at one position without tilting in any other direction. There is a built-in function for the hovering manipulation of the AR.Drone UAV, but the hovering function does not properly work according to experiments. To achieve hovering behavior, PID controllers for roll, pitch, and yaw angles are designed with controller coefficients of $K_p = 1$, $K_i = 1$, and $K_d = 1$. It is implemented in code as shown in [Fig. 2].

```
// PID control - horizontal positioning of ...
    ARDrone
// Pitch control
pre_error1 =  setpoint1 - pitch;
error1 = setpoint1 - pitch;
integral1 = integral1 + (error1*timeStep);
derivative1 = (error1 - pre_error1)/timeStep;
output1 = int("(Kp*error1) + ...
    (Ki*integral1) +
          (Kd*derivative1)");
if (output1 < -10) {
  output1 = -10;
}
else if (output1 > 10) {
  output1 = 10;
}
if (output1 ≥ 0) {
  ardrone.forward(output1);
}
else if (output1 < 0) {
  ardrone.backward(output1);
}
pre_error1 = error1;
```

[Fig. 2] PID controller for pitch angle of UAV

Using PID controllers, data were collected for 15 s and total hundred data sets were collected. The timestep of the Wifi signal which UAV uses is 0.05 s. Overall experiment setup is shown in [Fig. 3(a)].

(a) Data gathering system set up



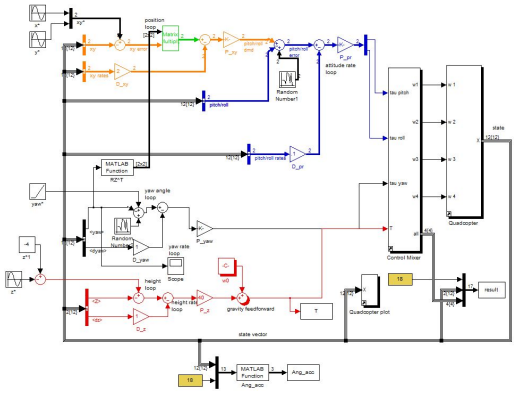
(b) GUI for UAV behavior monitoring

[Fig. 3] Experiment set up

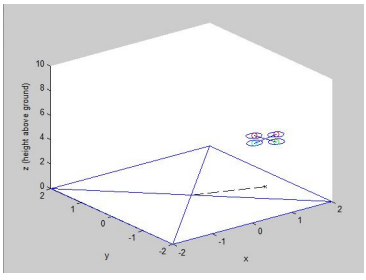
Also, for effective experiments, the GUI shown in [Fig. 3(b)] is developed using Processing open source computer programming language [12] to monitor and save the data of the real time video (from front camera of the UAV), pitch angle, roll angle, yaw angle, and altitude.

The UAV flight is also simulated using the MATLAB/Simulink toolbox developed by [13]. To obtain the desired UAV data, the Simulink model is modified as shown in [Fig. 4]. The modified parts are:

1. Error is introduced to xy data acquisition with 0 mean and 0.01 variance,
2. Error is introduced to z data acquisition with 0 mean and 5 variance,
3. The z data is set to be sinusoidal with an amplitude of 2, the frequency of $2 \cdot \pi \cdot 0125$ *rad/sec*, and phase of $-\pi/2$ *rad*. The visual simulation is shown in [Fig. 5].



[Fig. 4] UAV Simulink model



[Fig. 5] The graphic output of the UAV Simulink model shown in [Fig. 4]

4. Parameter Estimation

In this section, parameter estimation using RLS method is performed. According to the RLS parameter estimation method for the linear static process [14], we can set

$$e(k) = y_p(k) - K_M u(k) \tag{5}$$

where e is the error, y_p is the observation, K_M is the model parameter, u is the system input, $K_M u$ is the model prediction, and k is the given step time.

For the RLS method, the cost function defined as,

$$V = \sum_{k=1}^{N} e^2(k) = \sum_{k=1}^{N} \left[y_p(k) - K_M u(k) \right]^2, \tag{6}$$

has to be minimized to obtain the parameter K_M where N is the total number of data sets. To find the minimum, one first determines the first derivative with regard to K_M as,

$$V' = \frac{dV}{dK_M} = -2 \sum_{k=1}^{N} \left[y_p(k) - K_M u(k) \right] u(k) = 0, \tag{7}$$

so that, the optimal choice of K_M, or $\hat{K}$, becomes

$$\hat{K} = \frac{\sum_{k=1}^{N} y_p(k) u(k)}{\sum_{k=1}^{N} u^2(k)}. \tag{8}$$

where $\hat{K}$ is the estimate of K.

By applying the Eq. 8 to the Eq. 4, we can get

$$\begin{aligned} \hat{\ddot{\phi}} &= \frac{lb}{I_x}\left(\Omega_4^2 - \Omega_2^2\right), \\ \hat{\ddot{\theta}} &= \frac{lb}{I_y}\left(\Omega_3^2 - \Omega_1^2\right), \\ \hat{\ddot{\psi}} &= \frac{ld}{I_z}\left(\Omega_3^2 + \Omega_1^2 - \Omega_4^2 - \Omega_2^2\right). \end{aligned} \tag{9}$$

Then, the Eq. 6 becomes

$$\begin{aligned} V_{N,\phi} &= \frac{1}{N} \sum_{k=1}^{N} \left\{ \ddot{\phi} - \frac{lb}{I_x} \left[\Omega_4^2(k) - \Omega_2^2(k) \right] \right\}^2, \\ V_{N,\theta} &= \frac{1}{N} \sum_{k=1}^{N} \left\{ \ddot{\theta} - \frac{lb}{I_y} \left[\Omega_3^2(k) - \Omega_1^2(k) \right] \right\}^2, \\ V_{N,\psi} &= \frac{1}{N} \sum_{k=1}^{N} \left\{ \ddot{\psi} - \frac{ld}{I_z} \left[\Omega_3^2(k) + \Omega_1^2(k) - \Omega_4^2(k) - \Omega_2^2(k) \right] \right\}^2, \end{aligned} \tag{10}$$

where $V_{N,\phi}$, $V_{N,\theta}$, and $V_{N,\psi}$ are cost functions of roll, pitch, and yaw angles.

Finally, the Eq. 8 becomes

$$\begin{aligned} \left(\widehat{\frac{lb}{I_x}} \right) &= \frac{\sum_{k=1}^{N} \left[\Omega_4^2(k) - \Omega_2^2(k) \right] \ddot{\phi}(k)}{\sum_{k=1}^{N} \left[\Omega_4^2(k) - \Omega_2^2(k) \right]^2}, \\ \left(\widehat{\frac{lb}{I_y}} \right) &= \frac{\sum_{k=1}^{N} \left[\Omega_3^2(k) - \Omega_1^2(k) \right] \ddot{\theta}(k)}{\sum_{k=1}^{N} \left[\Omega_3^2(k) - \Omega_1^2(k) \right]^2}, \\ \left(\widehat{\frac{ld}{I_z}} \right) &= \frac{\sum_{k=1}^{N} \left[\Omega_3^2(k) + \Omega_1^2(k) - \Omega_4^2(k) - \Omega_2^2(k) \right] \ddot{\psi}(k)}{\sum_{k=1}^{N} \left[\Omega_3^2(k) + \Omega_1^2(k) - \Omega_4^2(k) - \Omega_2^2(k) \right]^2}. \end{aligned} \tag{11}$$

The result of the estimated parameter values is shown in <Table 1>.

〈Table 1〉 Comparison of original values with estimated values

Parameter	Estimated Parameter	Original Parameter
$\ddot{\phi}$	0.3968E-4	0.5084E-4
$\ddot{\theta}$	0.4795E-4	0.5084E-4
$\ddot{\psi}$	0.0018E-4	0.2798E-4

5. Convergence of Estimates

In this section, we check if the distribution of parameter estimates converge to a normal distribution using a histogram technique and find asymptotic covariances, that is, the covariance matrices from data.

First of all, the type of distribution is checked by referring [15] as,

$$\frac{1}{N}\sum_{k=1}^{N}\Psi(t,\Theta^*)\epsilon(t,\Theta^*) \in N(0,Q) \tag{12}$$

with

$$Q_\phi = \lim_{N\to\infty} N \cdot E\left[V_{N,\phi}{}'(\Theta^*.Z^N)\right]\left[V_{N,\phi}{}'(\Theta^*.Z^N)\right]^T, \tag{13}$$
$$Q_\theta = \lim_{N\to\infty} N \cdot E\left[V_{N,\theta}{}'(\Theta^*.Z^N)\right]\left[V_{N,\theta}{}'(\Theta^*.Z^N)\right]^T,$$
$$Q_\psi = \lim_{N\to\infty} N \cdot E\left[V_{N,\psi}{}'(\Theta^*.Z^N)\right]\left[V_{N,\psi}{}'(\Theta^*.Z^N)\right]^T,$$

where $\Psi(t,\phi^*)$ is the $(d/d\phi^*)\hat{y}(t|\phi^*)$, ϵ is the prediction error, Θ is the parameter vector, Q_ϕ, Q_θ, and Q_ψ are covariance matrices of the roll, pitch, and yaw angles, E is the expected value, $*$ represents an asymptotic value, and Z^N is the batch of data from the system. The Eq. 12 means that the random variable on the left converges with the normal distribution with zero mean and covariance matrix Q.

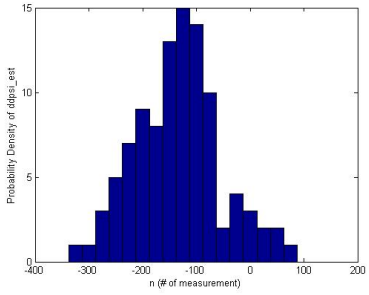
In the Eq. 13, with the assumption that $V_{N,\phi}{}'$, $V_{N,\theta}{}'$, and $V_{N,\psi}{}'$ are independent, it can be stated as

$$\sqrt{N}\left(\hat{\Theta}_N - \Theta^*\right) \in N(0,P_\Theta) \tag{14}$$

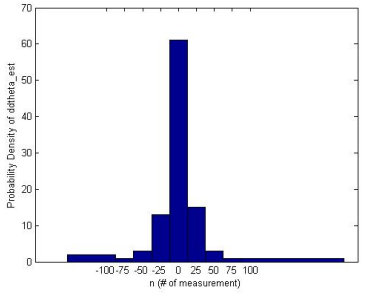
with

$$P_\Theta = \left[\overline{V}''(\Theta)^*\right]^{-1} Q \left[\overline{V}''(\Theta)^*\right]^{-1} \tag{15}$$

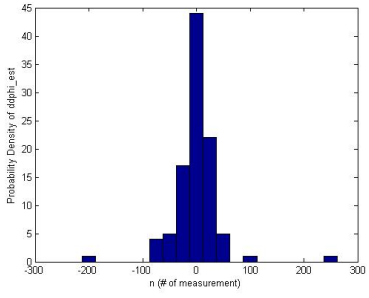
where the covariance matrix Q is given in the Eq. 13. In conclusion, the obtained results are shown in [Fig. 6].



(a) Normal distribution of $\ddot{\psi}$



(b) Normal distribution of $\ddot{\theta}$



(c) Normal distribution of $\ddot{\phi}$

[Fig. 6] Normal distributions of three angular accelerations

Covariance matrices of the asymptotic distribution in the Eq. 15 can be approximated as,

$$Cov\left(\hat{\Theta}_N\right) \sim \frac{1}{N}P_\Theta, \tag{16}$$

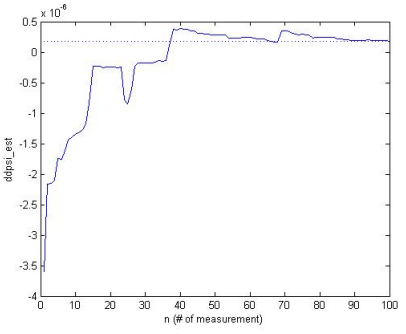
in some special cases when data and noises are sequences of independent random variables with zero value and variance λ_0. In this case, the matrix P in Eq. 15 can be rewritten as,

$$P_\Theta = \lambda_0 \left[\overline{E\Psi}(t,\Theta_0)\Psi^T(t,\Theta_0) \right]^{-1}. \quad (17)$$

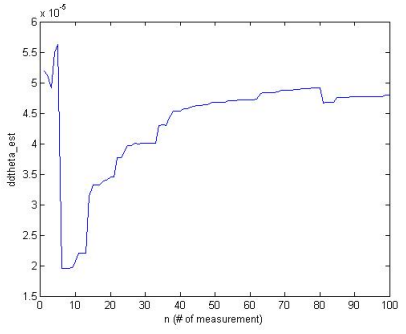
Having processed N data points and determined Θ_N, the Eq. 17 can be rewritten as,

$$\hat{P}_N = \hat{\lambda}_N \left[\frac{1}{N} \sum_{t=1}^{N} \Psi(t,\hat{\Theta}_N)\Psi^T(t,\hat{\Theta}_N) \right]^{-1}, \quad (18)$$
$$\lambda_N = \frac{1}{N} \sum_{t=1}^{N} \epsilon^2(t,\hat{\Theta}_N).$$

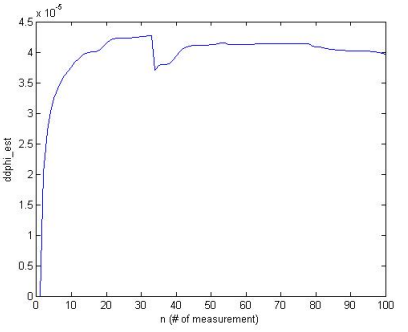
Convergence of the parameters is checked by changing the number of the total experiment from 1 to 100 measurements as shown in [Fig. 7]. Here, each parameter tends to be constant as the number of measurements increase. This illustrates all parameters converge well.



(a) Convergence of $\ddot{\phi}$



(b) Convergence of $\ddot{\theta}$



(c) Convergence of $\ddot{\psi}$

[Fig. 7] Convergence of three angular accelerations

6. Conditional Status of Inverted Matrix using Singular Value and SNR

It is required to check if we have a well-conditioned matrix inversion, ideally all singular values in the same order of magnitude, in the system identification procedure. Also, we need to relate the condition number to the SNR of the data to check whether the persistency of excitation condition has been met.

To check whether a system is well-conditioned or ill-conditioned, we need to calculate a condition number of $R(N)$ as,

$$A = QR = [\Theta, Y] \quad (19)$$

where the matrix R comes from

$$R(N) = \Theta^T\Theta = \begin{bmatrix} R_0 \\ \cdots \\ 0 \end{bmatrix}. \quad (20)$$

In the Eq. 19, the big matrix Q does not need to be calculated since all information is contained in the small matrix R_0 [15]. Also, considerable computations can be saved by using a built-in function in MATLAB, $R = triu(qr(A))$, to compute the matrix A if the matrix A has many more rows than columns ([Fig. 8]).

```
>> A = triu(qr([phi_list,Y_list]));
>> A = A(1:4,1:4);
>> [U,S,V] = svd(A);
>> S

S =

   1.0e+007 *

     2.9945         0         0         0
          0    0.6334         0         0
          0         0    0.0581         0
          0         0         0    0.0092
```

[Fig. 8] Application of the MATLAB built-in function, $R=triu(qr(A))$, to the parameter $\ddot{\hat{\psi}}$

In [Fig. 8], the function *svd* represents the singular value decomposition [16]. Since the matrix S has an almost same order of magnitude, it is a fairly well-conditioned matrix.

The SNR can be defined as the reciprocal of the coefficient of variation, i.e. ratio of the mean to standard deviation of a signal or measurement [17,18],

$$SNR=\frac{\mu}{\sigma} \tag{21}$$

where μ is the signal mean or expected value and σ is the standard deviation of the noise or an estimate thereof. Calculated singular values and SNRs are shown in <Table 2>.

〈Table 2〉 Calculated singular values and SNRs

Parameter	Singular Value	SNR
$\ddot{\phi}$	2.3005, 0.1065	-0.0849
$\ddot{\theta}$	2.3228, 0.1192	0.0087
$\ddot{\psi}$	3.2154, 0.5910, 0.1522, 0.0446	0.0315

Persistency of the excitation condition is met in all three cases. The θ is particularly the most stable among the three parameters against input noise.

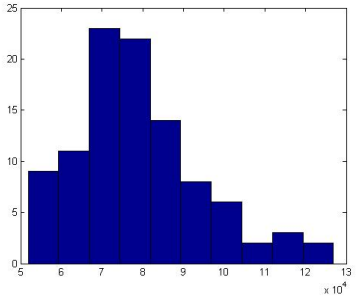
7. Robustness of System Identification using Noise PDF

The robustness of identification procedures to outliers and the optimality criterion producing the minimum parameter variance are dependent upon the PDF since normal noise implies that the quadratic criteria are the best.

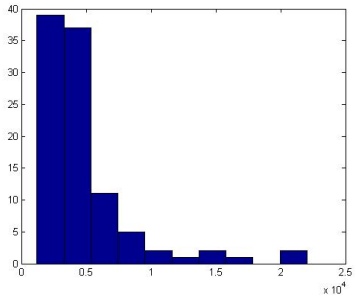
The mean square error is defined as

$$l_{\phi}(\epsilon)=\frac{1}{2}\epsilon_{\phi}^2=\frac{1}{2}\sum_{k=1}^{N}\left(\ddot{\phi}-\frac{lb}{I_x}\left(\Omega_4^2(k)-\Omega_2^2(k)\right)\right)^2,$$
$$l_{\theta}(\epsilon)=\frac{1}{2}\epsilon_{\theta}^2=\frac{1}{2}\sum_{k=1}^{N}\left(\ddot{\theta}-\frac{lb}{I_y}\left(\Omega_3^2(k)-\Omega_1^2(k)\right)\right)^2,$$
$$l_{\psi}(\epsilon)=\frac{1}{2}\epsilon_{\psi}^2=\frac{1}{2}\sum_{k=1}^{N}\left(\ddot{\psi}-\frac{lb}{I_x}\left(\Omega_3^2(k)+\Omega_1^2(k)-\Omega_4^2(k)-\Omega_2^2(k)\right)\right)^2, \tag{22}$$

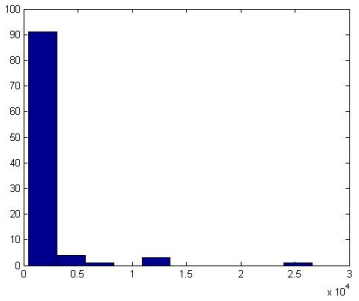
and PDFs of three parameters are shown in [Fig. 9] and <Table 3>.



(a) Noise PDF of $\ddot{\phi}$



(b) Noise PDF of $\ddot{\theta}$



(c) Noise PDF of $\ddot{\psi}$

[Fig. 9] Noise PDF of three angular accelerations

〈Table 3〉 Comparison of original values with estimated values in noise PDF

Parameter	Estimated Value	Original Value
$\ddot{\phi}$	0.3993E-4	0.5084E-4
$\ddot{\theta}$	0.5818E-4	0.5084E-4
$\ddot{\psi}$	0.0019E-4	0.2798E-4

8. Real-time and Closed Loop State and Parameter Estimation

The recursive system parameter estimation is performed to determine if estimates are consistent with parameters previously obtained using the batch estimation method.

There are multiple methods to calculate recursive parameters including RLS, recursive instrumental variable (RIV), recursive prediction error (RPE), recursive pseudolinear regressions (RPLR), etc. When the time-varying linear system is investigated, the RLS method can be used by considering the effect from the time-varying system.

By introducing the white Gaussian noise, w, we can achieve

$$\begin{aligned} &\Theta(t+1)=\Theta(t)+w(t), \\ &Ew(t)w^T(t)=R_1(t), \\ &Ev(t)v^T(t)=R_2(t). \end{aligned} \tag{23}$$

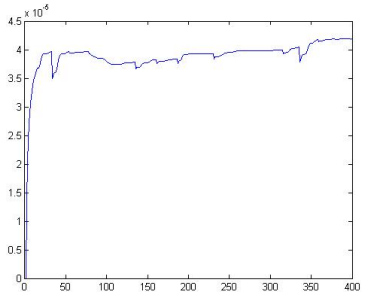
Then, the Kalman filter (KF) interpretation gives

$$\begin{aligned} &\hat{\Theta}(t)=\hat{\Theta}(t-1)+L(t)\left[y(t)-\Phi(t)\Theta^T(t-1)\right], \\ &L(t)=\frac{P(t-1)\Phi(t)}{R_2(t)+\Phi^T(t)P(t-1)\Phi(t)}, \\ &P(t)=P(t-1)-\frac{P(t-1)\Phi(t)\Phi^T(t)P(t-1)}{R_2(t)+\Phi^T(t)P(t-1)\Phi(t)}+R_1(t). \end{aligned} \tag{24}$$

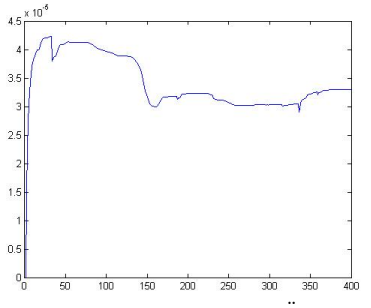
The Eq. 23 and Eq. 24 are implemented in MATLAB as shown in [Fig. 10] and convergence of the parameters is shown in [Fig. 11].

```
% Algorithm for Recursive Estimation (Kalman ...
    Filter Algorithm, Ljung, p.368-369)
% ddpsi_est
P = eye(1,1);
Q = zeros(1,1);
% lamda = 0.98;
lamda = 1;
ddpsi_est = zeros(400,1);
for i=1:400
    phi = data{1}(i,5)^2 + data{1}(i,3)^2 - ...
        data{1}(i,4)^2 - data{1}(i,2)^2;
    sample_out = phi*Q;
    P = 1/lamda*(P - (P*phi*phi*P)/(lamda + ...
        phi*P*phi));
    Q = Q + (P*phi)/(lamda + ...
        phi*P*phi)*(data{1}(i,6) - sample_out);
    ddpsi_est(i,1) = Q;
end
figure(1);plot(1:400,ddpsi_est)
ddpsi_est(end)

% ddtheta_est
P = eye(1,1);
Q = zeros(1,1);
% lamda = 0.98;
lamda = 1;
ddtheta_est = zeros(400,1);
for i=1:400
    phi = data{1}(i,4)^2 - data{1}(i,2)^2;
    sample_out = phi*Q;
    P = 1/lamda*(P - (P*phi*phi*P)/(lamda + ...
        phi*P*phi));
    Q = Q + (P*phi)/(lamda + ...
        phi*P*phi)*(data{1}(i,7) - sample_out);
    ddtheta_est(i,1) = Q;
end
figure(2);plot(1:400,ddtheta_est)
ddtheta_est(end)

% ddphi_est
P = eye(1,1);
Q = zeros(1,1);
% lamda = 0.98;
lamda = 1;
ddphi_est = zeros(400,1);
for i=1:400
    phi = data{1}(i,3)^2 - data{1}(i,5)^2;
    sample_out = phi*Q;
    P = 1/lamda*(P - (P*phi*phi*P)/(lamda + ...
        phi*P*phi));
    Q = Q + (P*phi)/(lamda + ...
        phi*P*phi)*(data{1}(i,8) - sample_out);
    ddphi_est(i,1) = Q;
end
figure(3);plot(1:400,ddphi_est)
ddphi_est(end)
```

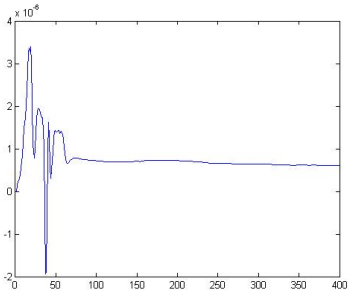
[Fig. 10] Algorithm for the RLS estimation



(a) RLS estimation value of $\ddot{\phi}$



(b) RLS estimation value of $\ddot{\theta}$



(c) RLS estimation value of $\ddot{\psi}$

[Fig. 11] Algorithm for the RLS estimation

According to [Fig. 11], three converged parameter values are shown in <Table 4>.

〈Table 4〉 Comparison of original values with estimated values using RLS estimation method

Parameter	Estimated Value	Original Value
$\ddot{\phi}$	0.4187E-4	0.5084E-4
$\ddot{\theta}$	0.3298E-4	0.5084E-4
$\ddot{\psi}$	6.1211E-7	0.2798E-4

When the RLS method is used, a parameter value of $\ddot{\psi}$ becomes way off compared to the one obtained using the batch method. Except that, the other two parameters nicely converge though those values are more off from the original parameter values than the one obtained using batch method.

9. Conclusion

In this paper, we showed system identification procedures for the quadrotor UAV using batch and RLS estimation methods. The commonly known quadrotor UAV model is adopted and simplified since we only focus on the hovering operation. Angular accelerations of the roll, pitch, and yaw data required for the hovering maneuver are analyzed and shown to be convergent. Also, conditional status using SNR, the robustness of system identification using noise PDF, and real-time parameter estimation using the RLS method are performed for the additional parameter estimations.

REFERENCES

[1] J. P. Lee, J. W. Lee, and K. H. Lee, “A Scheme of Security Drone Convergence Service using Cam-Shift Algorithm,” Journal of the Korea Convergence Society, Vol. 7, No. 5, pp. 29-34, 2016.

[2] J. Bloem, M. V. Doorn, S. Duivestein, D. Excoffier, R. Maas, and E. V. Ommeren, The Fourth Industrial Revolution - Things to Tighten the Link Between IT and OT, Sogeti, 2014.

[3] L. Floridi, The 4th Revolution - How the Infosphere is Reshaping Human Reality, Oxford University Press, 2014.

[4] S. U. Bae, D. G. Kwag, and E. Y. Park, “The Study of the Aviation Industrial Technology Convergence through Patent Analysis,” Journal of the Korea Convergence Society, Vol. 6, No. 5, pp. 219-225, 2015.

[5] Y. S. Byun, J. H. Um, R. G. Jeong, B. H. Kim, and S. W. Kang, “Magnetic Markers-based Autonomous Navigation System for a Personal Rapid Transit (PRT) Vehicle,” Journal of Digital Convergence, Vol. 13, No. 1, pp. 297-304, 2015.

[6] G. W. Lee and J. K. Park, “Geospatial Analysis of Dam Construction Area by Unmanned Vehicles,” Journal of Digital Convergence, Vol. 14, No. 12, pp. 225-230, 2016.

[7] B. Mettler, M. Tischler, and T. Kanade, “System Identification of Small-Size Unmanned Helicopter Dynamics,” 55th American Helicopter Society Annual Forum, pp. 1-12, 1999.

[8] I. Stanculeanu and T Borangiu, “Quadrotor Black-Box System Identification,” International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, Vol. 5, No. 6, pp. 1025-1028, 2011.

[9] A. Hermandez, C. Copot, R. D. Keyser, T. Vlas, and

I. Nascu, “Identification and Path Following Control of an AR.Drone Quadrotor,” 2013 17th International Conference on System Theory, Control and Computing, pp. 1-6, 2013.
[10] S. Bouabdallah and R. Siegwart, “PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor,” Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1-6, 2004.
[11] https://goo.gl/su8QbZ
[12] https://goo.gl/1nknEJ
[13] https://goo.gl/RvWvtc
[14] R. Isermann and M. Munchhof, Identification of Dynamic Systems, Springer-Verlag Berlin Heidelberg, 2011.
[15] L. Ljung, System Identification, Prentice-Hall, 2nd Ed., 1999.
[16] B. Ake, Numerical Methods for Least Squares Problems, Siam, 1996.
[17] D. J. Schroeder, Astronomical Optics, 2nd Ed., Academic Press, 2000.
[18] J. T. Bushberg, The Essential Physics of Medical Imaging, Philadelphia: Lippincott Williams & Wilkins, 2006.

저자소개

정 성 훈(Sunghun Jung) [정회원]

- 2009년 8월 : 미네소타대학교 기계공학과 (공학학사)
- 2010년 12월 : 퍼듀대학교 기계공학과 (공학석사)
- 2013년 12월 : 퍼듀대학교 기계공학과 (공학박사)
- 2014년 1월 ~ 2016년 8월 : 삼성SDI 책임연구원
- 2016년 9월 ~ 현재 : 초당대학교 드론학과 조교수

<관심분야> : 무인항공기 자율기동, 에너지 효율적 경로 최적화, 배터리팩 상태예측 알고리즘 개발