Analysis of Time Delays in Quadrotor Systems and Design of Control

Stephen K. Armah and Sun Yi

Abstract In analyzing and designing control for unmanned aerial vehicles (UAVs), existence of transmission delays caused by wireless communication is one of the critical challenges. Estimation of the delays and analysis of their effects are not straightforward. A delay estimation method is introduced using transient responses of a quadrotor type of UAVs and analytical solutions of delay differential equations (DDEs). Experimental data sets in the time domain are compared to the predicted ones based on the analytical solutions of DDEs. The Lambert W function-based approach for first-order DDEs is used for the analysis. The dominant characteristic roots among an infinite number of roots are obtained in terms of coefficients and the delay. The effects of the time delay on the responses are analyzed via root locations. Based on the estimation result, proportional- plus-velocity controllers are proposed to improve transient altitude responses.

1 Introduction

Time delays exist in autonomous dynamic systems when signals are transmitted wirelessly. Estimation of the delays and understanding of their effects on the performance of dynamic systems is an important topic in many applications [1]. Estimating delays is a challenging problem and has attracted great research interests [2] [3]. Although considerable efforts have been made on parameter estimation, as for time-delay identification there are still many open problems and there is no common approach due to the lack of analytical solutions to delay equations and difficulty in formulation [4–6].

Autonomous control of quadrotor types of unmanned aerial vehicles (UAVs) has been the focus of active research during the past decades. One of the challenges in

S.K. Armah (✉) · S. Yi
Department of Mechanical Engineering, North Carolina A&T State University, Greensboro, NC 27411, USA
e-mail: skarmah@aggies.ncat.edu

S. Yi
e-mail: syi@ncat.edu

T. Insperger et al. (eds.), *Time Delay Systems*,
Advances in Delays and Dynamics 7, DOI 10.1007/978-3-319-53426-8_20

designing effective control systems for UAVs is the existence of signal transmission delay, which has nonlinear effects on the flight performance. A controller designed using a non-delayed system model may result in disappointingly slow and oscillating responses due to the delays. For autonomous aerial robots, typical values of the time delay have been known to be around 0.4 ± 0.2 s [7] or 0.2 s [8]. For large delays (e.g., larger than 0.2 s), the system response might not be stabilized or converged due to the dramatic increase in torque. This poses a significant challenge [9]. Since its effect is not trivial the delay need be estimated and considered in designing controllers.

In this chapter, an estimation method introduced in [6] is studied further and applied to a quadrotor type of UAV, Parrot AR.Drone 2.0. The UAV is controlled with MATLAB/Simulink through WiFi, which introduces a time delay to its dynamics as shown in Sect. 3. The overall time delay is attributed to: (1) the processing capability of the host computer, (2) the electronic devices processing the motion signals, e.g., actuation, (3) the measurement reading devices, e.g., the distance between the ultrasonic sensor, for reading the altitude, and the surface can affect the delay, and (4) the software, on the host computer, being used to implement the controllers. For UAVs wireless communication delays may not be critical when all the controllers are on board. However, delays have significant effects when the control software is run on an external computer and signals are transmitted wirelessly. For example, the experiments on the drone in this chapter were conducted using MATLAB/Simulink on an external computer, and decoding process of navigation data (yaw, pitch, roll, altitude, etc.) contributes to the delay. Also, the numerical solvers in the software introduce additional delay.

Most methods for transfer function identification assume that the delay is already known or just ignore the delays and their effects [10]. Several approaches to estimation of delay have been introduced in the literature. Those include finite dimensional Chebyshev spectral continuous time approximation (CTA) [11]. The finite dimensional CTA was used to approximately solve delay differential equations (DDEs) for the estimation of constant and time-varying delays. In addition, cross-correlation method [12], graphical methods [13, 14], a cost function for a set of time delays in a certain range [15], and frequency-domain maximum likelihood [16] have been used for delay estimation. This chapter presents a method to estimate the time delay in the altitude control system of quadrotor types of UAVs using the approach based on the analytical solutions to DDEs [6]. The altitude dynamics is assumed to be linear time-invariant (LTI) first order and the time delay is incorporated into the model as an explicit parameter. In real applications, drones fly around and the time delay may vary. Here, the delay is not restricted to be a multiple of the sampling interval. Experimental data and analytical solutions of infinite dimensional continuous DDEs are used for estimation. The approach in this chapter is inspired by the well-known time-domain description of step responses of LTI ordinary differential equations (ODEs). Measured transient responses are compared to time-domain descriptions obtained by using the dominant characteristic roots based on the Lambert W function written in terms of system parameters including the delay. Proportional (P) controllers are used to generate the responses for estimation. The effects of the time delay on the transient responses and stability are analyzed. Then, proportional-plus-velocity (PV) control

is designed to obtain better transient responses. This chapter continues with introduction of the approach used for estimating the system's time delay in Sect. 2 with an example. A description of quadrotor's altitude model and the AR.Drone 2.0 control system are provided in Sect. 3. In Sect. 4 the estimation results are summarized. Concluding remarks and future work are presented in Sect. 5.

2 Time-Delay Estimation Using Characteristic Roots

The estimation problem can be formulated using an analytical solution form to DDEs in terms of the scalar Lambert W function [6]. Consider the first-order scalar homogenous DDE:

$$\dot{z}(t) - a_o z(t) - a_1 z(t - T_d) = 0, \tag{1}$$

The characteristic equation of (1) is given by

$$s - a_o - a_1 e^{(-sT_d)} = 0. \tag{2}$$

Then, the characteristic equation in (2) is solved as [17]

$$s = \frac{1}{T_d} W(T_d a_1 e^{(-a_o T_d)}) + a_o. \tag{3}$$

The Lambert W function is defined as $W(x)e^{W(x)} = x$ [18]. As seen in (3), the characteristic root, s, is expressed analytically in terms of parameters, a_o, a_1, and the time delay, T_d. The solution form in (3) enables one to determine how the time delay is involved in the solution and, thus, response. Furthermore, how each parameter affects each characteristic root. Thus, it is possible to formulate estimation of time delays in an analytic way. The Lambert W function is already embedded as *lambertw* in MATLAB [17]. The equation in (2) has an infinite number of roots. Each root is distinguished using branches, $k = -\infty, \ldots, -1, 0, 1, \ldots, \infty$, of the Lambert W function. For first-order scalar DDEs, it has been proved that the rightmost characteristic roots are always obtained by using the principal branch, $k = 0$, and/or $k = -1$ [19]. For the DDE in (1), one has to consider two possible cases for rightmost characteristic roots: characteristic equations of DDEs as in (2) can have one real dominant root or two complex conjugate dominant roots. Thus, when estimating time delays using characteristic roots, it is required to decide whether it is the former or the latter [6].

For ODEs, an estimation technique using the logarithmic decrement provides an effective way to estimate the damping ratio. The technique makes use of the characteristic roots in terms of ζ and ω_n:

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{(1 - \zeta^2)} \tag{4}$$

of second-order ODEs. The variables ζ and ω_n are obtained from the transient response of the system, and different approaches can be applied depending on the nature of the response: oscillatory and non-oscillatory. Here, the transient properties for oscillatory responses ($0 < \zeta < 1$) are used. Property such as the maximum overshoot in percentile, $M_o(\%)$, is related to ζ, as

$$M_o(\%) = 100e^{\left(\frac{-\zeta\pi}{\sqrt{(1-\zeta^2)}}\right)} \tag{5}$$

For a stable system, the dominant roots, s, lie in the left-hand complex plane. Then, ζ is computed as

$$\zeta = \frac{|Re(s)|}{\sqrt{(Re(s))^2 + (Im(s))^2}} \tag{6}$$

Then, the drone control system with the unknown T_d is estimated by the following steps:

Step 1: Measure the maximum overshoot from the experimental transient data,
Step 2: Calculate ζ based on the system altitude response using (5),
Step 3: Solve the nonlinear equation (6) with $s = \frac{1}{T_d} W(T_d a_1 e^{(-a_o T_d)}) + a_o$ for T_d. Equation (6) has only one unknown, which is the time delay, T_d.

The equation in Step 3 can be solved using a nonlinear solver (e.g., *fsolve* in MATLAB). The above method using the dominant characteristic roots assumes that the dominant ones are substantially closer to the imaginary axis than subdominant ones [6]. If not, estimation results may be inaccurate.

2.1 Example: Internet-Based Control

An example for a simple Internet-based feedback control is presented to illustrate how to use the above estimation technique. When tele-operating systems are run through private media variation of the transmission delay value is very small. Thus, the delay can be assumed to be constant and can be well modeled. The Internet, on the other hand, is a public and shared resource in which many end users transmit data through the network simultaneously. The route for transmission between two end points is not fixed and varies dynamically. Also, traffic jams may be caused when too many users use the same route simultaneously. The transmission latency of such public network is difficult to estimate and predict. Time delay is one of the critical obstacles in realizing reliable Internet-based process control systems [20].

The system in Fig. 1 has a plant of an integrator with proportional feedback control. The error, $e(t)$, is calculated in the first PC (PC 1) and sent to the second PC (PC 2). Then, the control input, $u(t) = Ke(t)$, is sent back to the PC1, where K is the

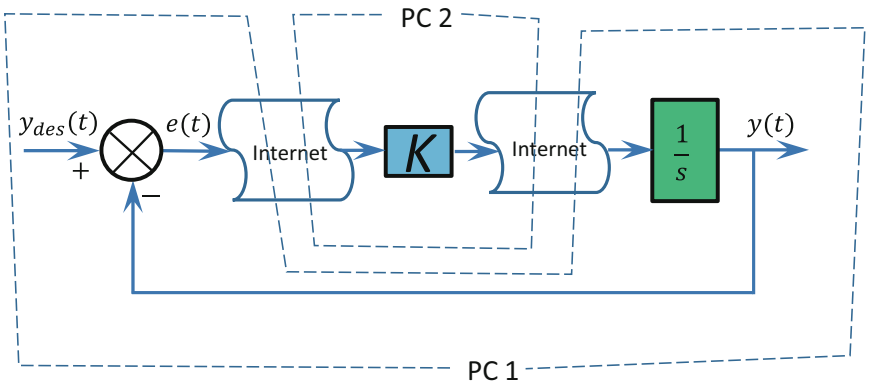

Fig. 1 A plant of an integrator is stabilized with proportional feedback control. The signals are transmitted through Internet using the 'UDP' block of Simulink

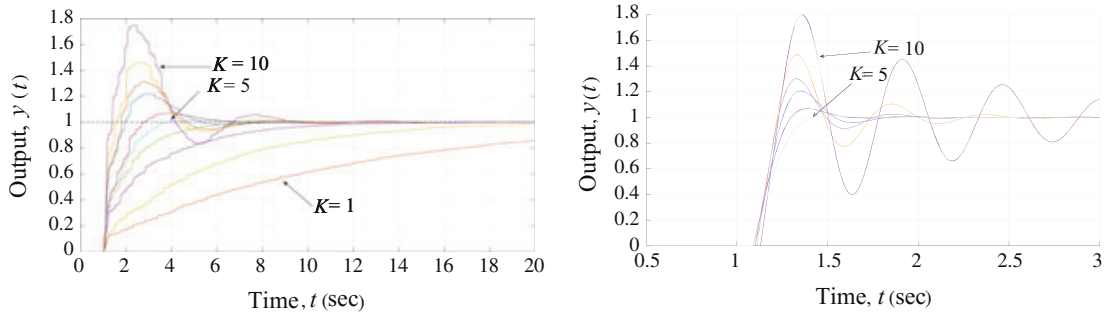

Fig. 2 Outputs for various control gain, K, with Internet delay (*left*) and artificial delay using estimated values in Table 1 (*right*)

control gain. The signals are transmitted through Internet using the 'UDP' blocks of Simulink.

Figure 2 (left) shows response, $y(t)$, of the system. If there is no time delay, the system has one real characteristic root, $-K$. Increases in the values of the control gain, K, do not yield overshoots (the closed-loop system is first order). But when K is greater than 5, overshoots are observed. The observed overshoots are summarized in Table 1.

Table 1 The results in Fig. 2 (left) are summarized

K	1–4	5	6	7	8	9	10
$M_p(\%)$	none	1.7%	7%	22%	31.5%	46%	75%
Estimated T_d (s)	n/a	0.09	0.09	0.104	0.104	0.11	0.13
Rightmost roots	n/a	-9.2	-8.1	-5.2	-4.4	-3.0	-1.0
		$\pm 7.2i$	$\pm 9.6i$	$\pm 10.8i$	$\pm 11.8i$	$\pm 12.1i$	$\pm 11.4i$
Damping ratio (ζ)	n/a	0.7898	0.6454	0.4315	0.3431	0.2373	0.0902

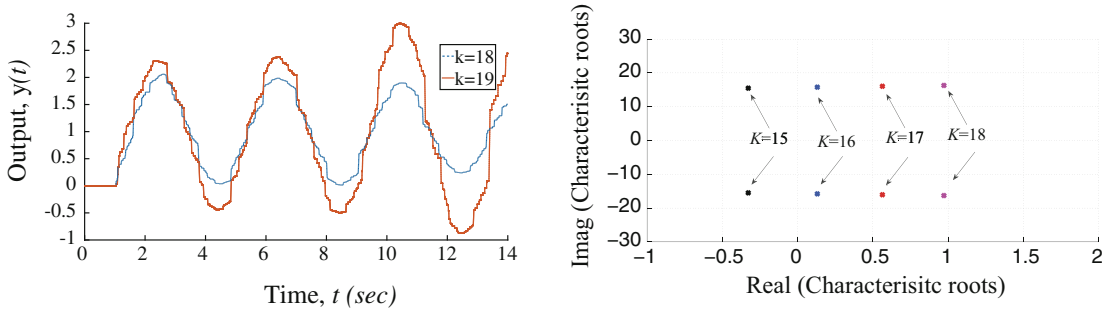

Fig. 3 Internet-based feedback control: when the gain is greater than 19, the output diverge and the controlled system become unstable (*left*). On the other hand, from the position of the rightmost characteristic roots, when the gain is greater than 15 the system is unstable ($T_d = 0.1$ s)

Using the three steps explained above using Eqs (4)–(6), the time delay caused by communication was estimated as summarized in Table 1. All the estimated values, T_d, are not exactly same and they vary from 0.09 to 0.13 s. This variation may happen partly due to Internet delays varying with time and sampling of signals. Also, it may be caused by ignoring other subdominant characteristic roots.

The estimation result can be used for stability analysis. When the value $T_d = 0.1$ s is used for simulation the gain, K, greater than 19 makes the system unstable. Refer to Fig. 3 (left). If the delay is assumed to be zero, no gain value larger than zero makes the system unstable in theory. When the gain is greater than 19 the output diverges over time and the controlled system become unstable. When the rightmost roots are obtained using the Lambert W function the gain greater than 16 makes the system unstable (Fig. 3 (right)). Although the two values (19 and 16) are not exactly same the stability analysis with the estimated time delay provides an approximate stability boundary. Note that using the estimation technique using 'Maximum overhoot' gives relatively good predictions of overshoots (Fig. 2) and stability (Fig. 3). But when the time scales of the simulation and experiment results are compared, there is a difference. Refer to the responses in Fig. 2. This difference is observed in the Internet-based control example above and drone control results to be shown later as well. This needs be studied further.

3 Altitude Model and Control System

This method is applied to control of quadrotors. Quadrotors are typically modeled based on three coordinate systems attached to it; the body-fixed frame, vehicle frame, and global inertial frame. They have six degrees of freedom in terms of position and attitude defined using the Euler angles. The quadrotor has four rotors, labeled 1 to 4, mounted at the end of each cross arm. The rotors are driven by electric motors having electronic speed controllers. The vehicle's total mass is m and total upward thrust, $T(t)$, on the vehicle given by [21]

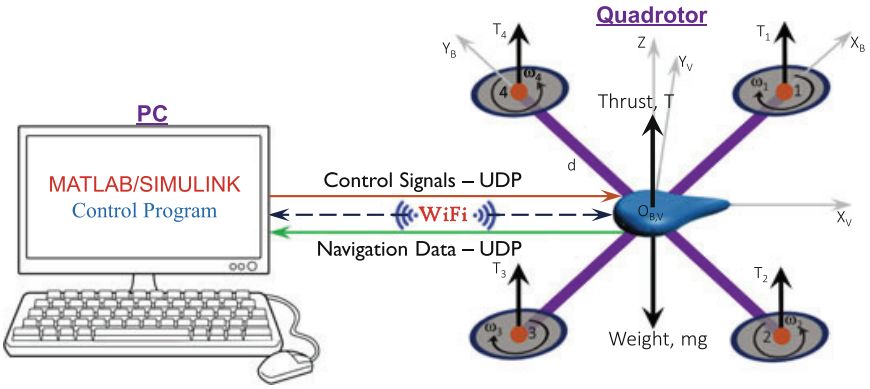

Fig. 4 The drone's navigation data is sent to the PC though WiFi. Then, the altitude error and control signal are calculated and sent back to the drone

$$T(t) = \sum_{i=1}^{i=4} T_i(t) \tag{7}$$

and $T_i(t) = a\omega_i^2(t)$, $i = 1, 2, 3$, and 4, where $\omega_i(t)$ is the rotor angular speed and $a > 0$ is the thrust constant. The equation of motion in the z-direction can be obtained as follows (refer to Fig. 4)

$$\ddot{z}(t) = \frac{4a\omega^2(t)}{m} - g \tag{8}$$

where $\omega(t)$ is the rotor's average angular speed necessary to generate $T(t)$. Thus, only the speed $\omega(t)$ needs be controlled to regulate the altitude, $z(t)$, of the quadrotor, since m, a, and g are constants.

According to the AR.Drone 2.0 SDK documentation, $z(t)$ is controlled by applying a reference vertical speed, $\dot{z}_{ref}(t)$, as control input. The speed, $\dot{z}_{ref}(t)$, has to be constrained to be between -1 and 1 m/s, to prevent damage. The drone's flight management system sampling time, T_s is 0.065 s, which is also the sampling time at which the control law is executed and the navigation data received.

The control block diagram for the drone's altitude motion regulation is shown in Fig. 5. The altitude motion dynamics in (8) is used to determine $\omega(t)$ from $\dot{z}(t)$. The rotors rotate at the equal speed, $\omega(t)$, which will generate $T(t)$ to make $z(t)$ reach the reference ($z_{des} = 1m$). These computations take place in the onboard control system programmed in C. In this chapter, the motor dynamics is assumed to be very fast such that the altitude control system can be represented by a first-order system using an integrator. Under such assumption, the control input, $\dot{z}_{app}(t)$, to the first-order system is approximated to be equal to the actual vertical speed, $\dot{z}(t)$, of the drone. Thus, a first-order model is used for the analytical determination of the time delay and for obtaining the simulation altitude responses. This simplification for control of quadrotor types of UAVs has been used and experimentally verified in the literature [22, 23]. The drone navigation data (from the sensors, cameras, and battery) is received and the control signals are sent using AT commands by UDP protocols.

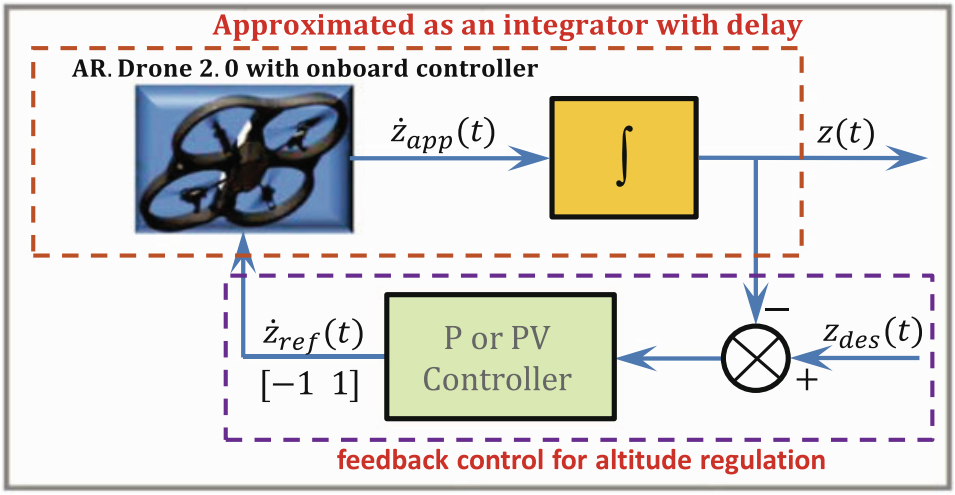

Fig. 5 Diagram for altitude control of the AR. Drone 2.0

AT commands are the combination of short text strings sent to the drone to control its actuation. The drone has ultrasound sensors for ground altitude measurement (at the bottom of the quadrotor). It has 1 GHz 32 bit ARM Cortex A8 processor, 1GB DDR2 RAM at 200 MHz, and USB 2.0 high speed for extensions.

3.1 P and PV Control

The system has an integrator, 1/s, in the closed-loop transfer function as shown in Fig. 5. Thus, only P and PV feedback controllers are used to regulate vertical speed signal. PV control, unlike proportional-plus-derivative (PD) control, does not induce numerator dynamics. The P-feedback controller is used to create transient responses to be used for estimation of the time delay. Based on the estimation, the PV feedback controller is designed to improve the transient response. The transfer function of the time-delay closed-loop system with the P controller is given by

$$\frac{Z(s)}{Z_{des}(s)} = \frac{K_P e^{-sT_d}}{s + K_P e^{-sT_d}} \tag{9}$$

This time-delay system is a retarded type. The characteristic equation is transcendental and the number of the characteristic roots is infinite. The characteristic roots have imaginary parts, which introduce oscillations in outputs, $z(t)$. Comparing the characteristic equation of the closed-loop system in (9) to the first-order system in (2), the coefficients are found as $a_o = 0$ and $a_1 = -K_P$.

The effect of T_d on the drone's altitude response is studied using analytical, simulation, and experimental approaches with the P controller. Then, suitable PV controller gains, K_p and K_v, are obtained to improve on the transient response performance. A high pass filter (HPF) with damping ratio, $\zeta_f = 1.0$ was used for the

derivative controller. A natural frequency value, ω_f, for the filter was selected by tuning and the use of the Bode plot. The transfer function of the time-delay closed-loop system for the PV controller, neutral type, is given by

$$\frac{Z(s)}{Z_{des}(s)} = \frac{K_P e^{-sT_d}}{s + (K_P + K_v s)e^{-sT_d}} \tag{10}$$

The characteristic equations of the neutral types of delay systems cannot be solved by using the Lambert W function [17]. Instead, a numerical method is used to obtain the roots [24].

4 Results and Discussion

4.1 Estimation of Time Delay

First, the drone's altitude responses were collected for several values of K_p, as shown in Fig. 6. Note that if there is no delay in the system ($T_d = 0$), the characteristic root is $-K_p$ (refer to Eq. (9)), which is a real number. Thus, there should be no overshoot in the response. Also, increase in K_p moves the root more to the left in the complex plane and does not destabilize the system. However, as seen in Fig. 6, when the gain, K_p, is higher than 1.0, the output has a nonzero overshoot due to delay in the control loop. Therefore, the delay does have nontrivial effects on response and need to be precisely estimated and considered in designing effective control. Note that for ease of analyzing the responses are shifted to start at (0 s, 0 m).

It was also observed that the saturation applied to the control input has a nonlinear effect on the system's response, especially as K_p increases. After comparing

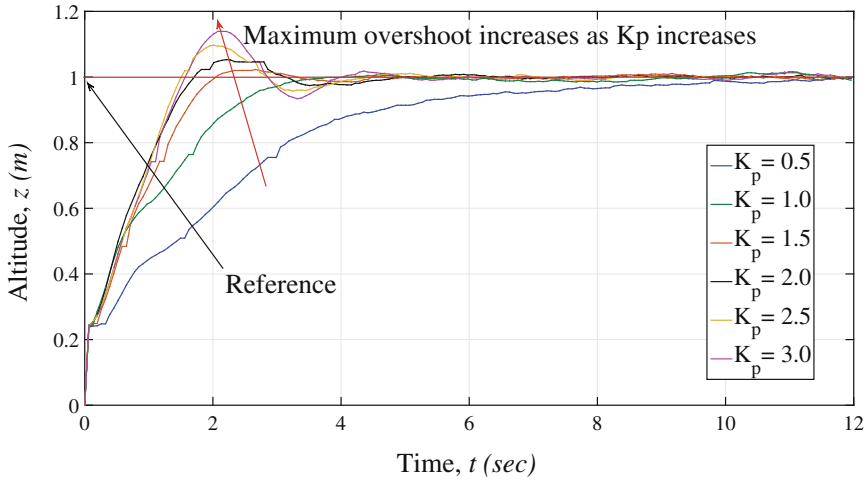

Fig. 6 Measured altitude responses with the gain, K_p, varying from 0.5 to 3

simulation results to experimental data, it was found that the value $K_p = 1.31$ gives a response with a sufficient overshoot for delay estimation with a minimum effect of saturation.

4.2 Numerical Method

Table 2 shows a summary of the simulation altitude responses transient properties by varying T_d at $K_p = 1.31$, where N is a real constant tuning parameter. The drone altitude responses from navigation data of 5 tests (Flights 1–5) with $K_p = 1.31$ are shown in Fig. 7. The measured M_o values are 2.30, 2.29, 2.30, 2.27, and 0.60%. When the values $M_o = 2.30\%$ from the first four flights excluding the last are used to estimate the time delay, as shown in the simulation results in Table 2 the delay T_d is estimated as $5.6646T_s$, which is 0.3682 s.

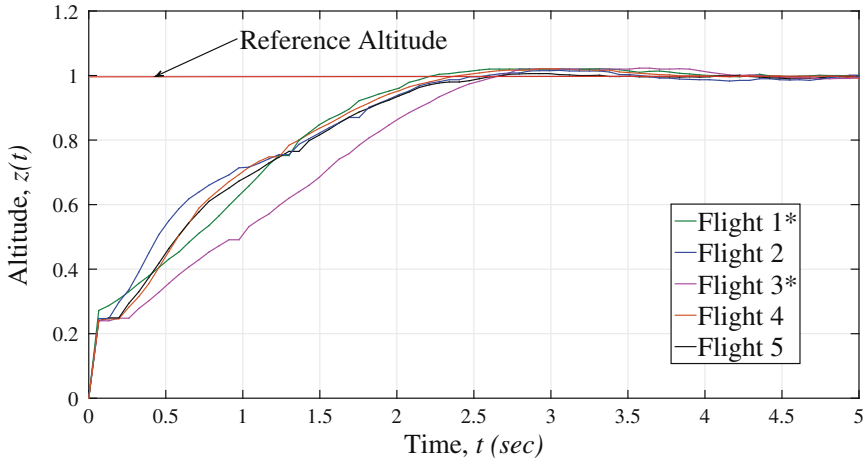

Fig. 7 Experimented altitude responses: $K_p = 1.31$. The measured M_o values are 2.30, 2.29, 2.30, 2.27, and 0.60 %

Table 2 The maximum overshoots of the simulated altitude responses with $K_p = 1.31$

N	$T_d = NT_s$	$M_o(\%)$
4.0000	0.2600	0.000
5.0000	0.3250	0.419
5.6000	0.3640	2.067
5.6640	0.3682	2.298
5.6646	0.3682	2.300
5.6660	0.3683	2.305
5.7000	0.3705	2.429

4.3 Use of Characteristic Roots

The drone altitude responses oscillate and have overshoots as shown in Figs. 6 and 7. Thus, it is assumed that the system has two complex conjugate dominant roots. The maximum overshoot values are used to determine the damping ratio, ζ. When $M_o = 2.30\%$ is used ζ is computed as 0.7684 using Eq. (5). Applying the steps described in Sect. 2, the delay T_d is determined as 0.3598 s. This value is close to the numerical method result (0.3682 s) but the two values are not exactly same. This discrepancy can come from simulation errors and noise in measuring altitudes. Then, the estimated result is used for stability analysis. Again, when stability is predicted assuming no delay any K_p value larger than zero does not destabilize the system. However, as shown in Fig. 8, if K_p is greater than 5, the amplitude of the altitude response grows over time and the system is unstable. The estimated delay value can explain this result. When the value ($T_d = 0.36$ s) is used, the simulated responses obtained by using Simulink agree with the experimental stability result (Fig. 9 (left)). Also, the roots calculated using the Lambert W function-based method shows that when K_p is greater than 5, the rightmost roots are placed in the right half plane (RHP) and, thus, unstable (Fig. 9 (right)).

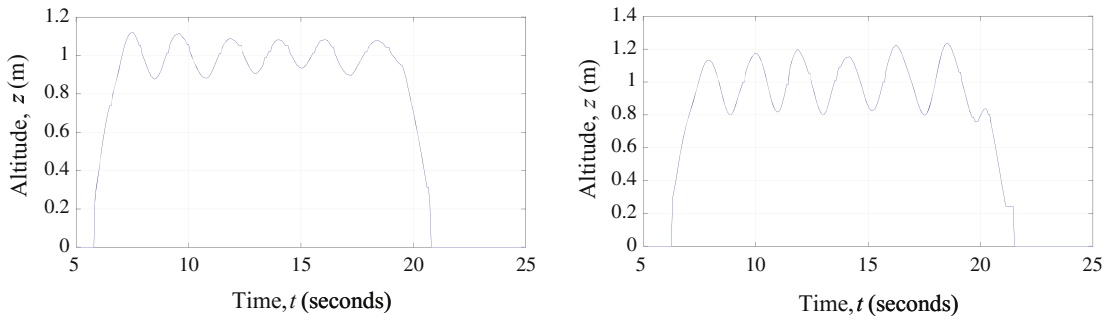

Fig. 8 The altitude responses with K_p is 4 (*left*) and 5 (*right*). If the control gain is greater than 5, the system becomes unstable

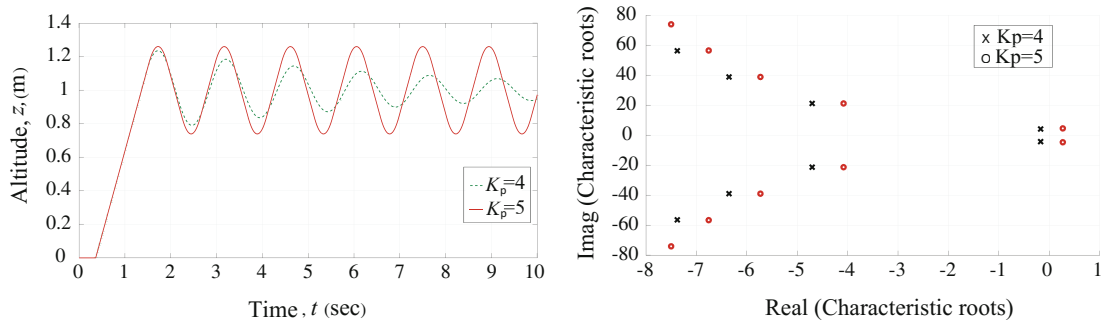

Fig. 9 Simulation also confirms that when K_p is larger than 5, the system becomes unstable (*left*). The rightmost roots crosses the imaginary axis when $K_p > 5$

4.4 PV Control

The PV control used to control altitude of the drone system is designed based on the time delay, 0.36 s, estimated using the analytical method. A MATLAB-based software package [24] was used to study the stability of the neutral type time-delay system, by solving the characteristic equation from the transfer function in Eq. (10). The closed-loop system characteristic roots within a specified region are then plotted for various K_v values. Figure 10 shows the spectrum distribution of the characteristic roots. When the rightmost (i.e., dominant) roots for each case are considered the value $K_v = 0.3$ yields the most stable rightmost roots ($s = -3.32$). For PV control, addition of K_v reduces the maximum overshoot for the same gain value, $K_p = 2$, while maintaining fast response time. But when K_v is greater than 0.97 the roots cross the imaginary axis, and the system becomes unstable. Thus, the estimated time delay value enables stability analysis before implementing.

The corresponding simulation altitude responses for the system were also obtained for the various K_v values, which are not shown in this chapter. It can be seen that as K_v increases at $K_p = 2.0$ and $T_d = 0.36$ s, M_o decreases but the rise time becomes longer. At higher values of K_v, the response oscillates and the system becomes unstable. This is also observed in Fig. 10, that as K_v increases (higher than 0.5) the roots moves back to the right increasing the instability in the system.

Based on these analyses, a controller gain set of $K_p = 2.0$ and $K_v = 0.3$ was selected for the most stable closed-loop system response with transient properties

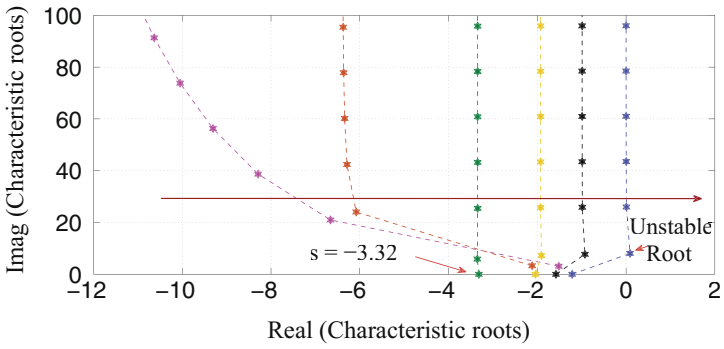

Fig. 10 The characteristic roots of the system with PV control with $K_p = 2.0$ and $T_d = 0.36$ s

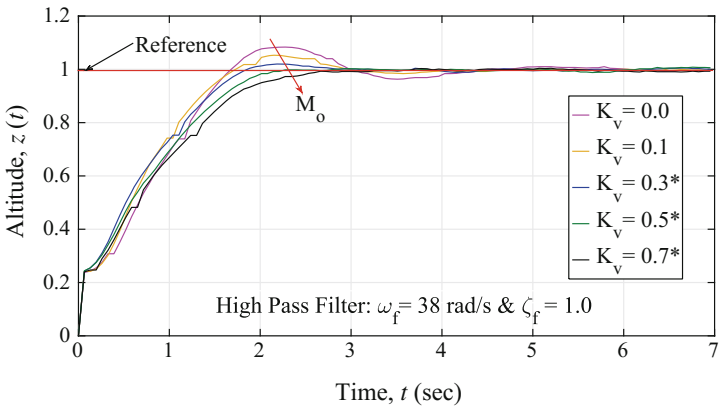

Fig. 11 Experimented PV controller altitude responses with $K_p = 2.0$

of $M_o < 1\,\%$ and $t_s < 2$ s after simulations using Simulink. Using these controller gains, the HPF was included in the simulation control system, and its effects on the altitude transient response, at different ω_f, were studied. It is observed that at lower ω_f values the response oscillates, and at higher values the response distorts [25]. The oscillations and the distortions effects were reduced by using the high-order solver, *ode8* (Dormand-Prince). After trial-and-errors HPF with $\omega_f = 38$ rad/s and $\zeta_f = 1.0$ were selected, with poles of -38 repeated. Now, looking at the poles distribution of the system in Fig. 10, it is observed that the poles of this filter are located much farther to the left than the poles of the PV feedback closed-loop system. Thus, this filter will respond faster, therefore, it has only a minor effect on the drone's altitude transient response. The filter's cutoff frequency was determined as 5.68 rad/s (0.90 Hz).

Figure 11 shows the experimented altitude responses and their corresponding transient properties, with the HPF and $K_p = 2.0$, for different K_v values. It was observed that as the K_v value increases M_o decreases but the responses become slower. The PV controller with $K_v = 0.3$ and $K_p = 2.0$, which has the most stable root ($s = -3.32$) in Fig. 10, gives the best performance in terms of the 2% settling time (1.86 s). As K_v increases, the maximum overshoot decreases but the settling time becomes longer. For example, the PV controller with $K_v = 0.7$ and $K_p = 2.0$, which has the rightmost roots as ($s = -0.93 \pm 7.71i$) and yields 2.47 s for the 2% settling time.

5 Conclusion

This study demonstrated how to use analytical solutions for ODEs and DDEs to estimate the time delay in Internet-based feedback control and quadrotor types of UAVs. Through numerical and analytical approaches, the time delay was estimated as 0.37 s and 0.36 s for the quadrotor, respectively. In the estimation of the time delay, an appropriate P controller was used and the gain that minimizes the effect of the applied control signal saturation on the system's response was selected. The effect of the time delay on the drone's altitude response was analyzed including system stability conditions. The PV controller was designed for improved transient responses based on the rightmost roots calculated using the estimated delay value.

The simulations and experiments were conducted using a high-order solver of MATLAB/Simulink, *ode8* (Dormand-Prince). Investigation through trials revealed that selection of the solver had significant effects on the drone's altitude response. The HPF performance was constrained by the type of solver used and the filter performed better with the high-order solvers. Also, discretization of continuous systems can cause additional errors [26]. These are being investigated further by the authors.

In future, multiaxis dynamics of the drone system considering attitude (pitch, roll, and yaw) and lateral motions (x and y) can be considered for estimating and incorporating the time delay in designing control systems. This problem is significantly more challenging, since the equation of motions are coupled and more complex than that

of the altitude motion. Furthermore, the presented time-delay estimating methods can be extended to general systems of DDEs (higher than first order). Also adaptive control that is updated based on the real-time estimation of delays can be studied.

Acknowledgements This material is based on research sponsored by Air Force Research Laboratory and OSD under agreement number FA8750-15-2-0116. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

1. Armah, S., Yi, S.: Altitude regulation of quadrotor types of UAVs considering communication delays. In: 12th IFAC Workshop on Time Delay Systems, pp. 263–268 (2015)
2. Azadegan, M., Beheshti, M.T., Tavassoli, B.: Using AQM for performance improvement of networked control systems. Int. J. Control Autom. Syst. **13**(3), 764–772 (2015)
3. Kchaou, M., Tadeo, F., Chaabane, M., Toumi, A.: Delay-dependent robust observer-based control for discrete-time uncertain singular systems with interval time-varying state delay. Int. J. Control Autom. Syst. **12**(1), 12–22 (2014)
4. Belkoura, L., Richard, J.P., Fliess, M.: Parameters estimation of systems with delayed and structured entries. Automatica **45**(5), 1117–1125 (2009)
5. Richard, J.P.: Time-delay systems: an overview of some recent advances and open problems. Automatica **39**(10), 1667–1694 (2003)
6. Yi, S., Choi, W., Abu-Lebdeh, T.: Time-delay estimation using the characteristic roots of delay differential equations. Am. J. Appl. Sci. **9**(6), 955–960 (2012)
7. Vásárhelyi, G., Virágh, C., Somorjai, G., Tarcai, N., Szorenyi, T., Nepusz, T., Vicsek, T.: Outdoor flocking and formation flight with autonomous aerial robots. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), pp. 3866–3873 (2014)
8. Sa, I., Corke, P.: System identification, estimation and control for a cost effective open-source quadcopter. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 2202–2209 (2012)
9. Ailon, A., Arogeti, S.: Study on the effects of time-delays on quadrotor-type helicopter dynamics. In: 2014 22nd Mediterranean Conference of Control and Automation (MED), pp. 305–310 (2014)
10. Wang, Q.G., Zhang, Y.: Robust identification of continuous systems with dead-time from step responses. Automatica **37**(3), 377–390 (2001)
11. Torkamani, S., Butcher, E.A.: Delay, state, and parameter estimation in chaotic and hyperchaotic delayed systems with uncertainty and time-varying delay. Int. J. Dyn. Control **1**(2), 135–163 (2013)
12. He, H., Yang, T., Chen, J.: On time delay estimation from a sparse linear prediction perspective. J. Acoust. Soc. Am. **137**(2), 1044–1047 (2015)
13. Ahmed, S., Huang, B., Shah, S.: Parameter and delay estimation of continuous-time models using a linear filter. J. Process Control **16**(4), 323–331 (2006)
14. Mamat, R., Fleming, P.: Method for on-line identification of a first order plus dead-time process model. Electron. Lett. **31**(15), 1297–1298 (1995)
15. Saha, D.C., Rao, G.P.: Identification of continuous dynamical systems: the poisson moment functional (PMF) approach, vol. 56. Springer (1983)
16. Pintelon, R., Van Biesen, L.: Identification of transfer functions with time delay and its application to cable fault location. IEEE Trans. Instrum. Measur. **39**(3), 479–484 (1990)
17. Yi, S., Nelson, P.W., Ulsoy, A.G.: Time-Delay Systems: Analysis and Control Using the Lambert W Function. World Scientific (2010)

18. Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. Adv. Comput. Math. **5**(4), 329–359 (1996)
19. Shinozaki, H., Mori, T.: Robust stability analysis of linear time-delay systems by Lambert W function: some extreme point results. Automatica **42**(10), 1791–1799 (2006)
20. Yang, S.H.: Internet-Based Control Systems Design and Applications. Springer, London (2010)
21. Corke, P.: Robotics, vision and control: fundamental algorithms in MATLAB, vol. 73. Springer Science & Business Media (2011)
22. Hoffmann, G.M., Huang, H., Waslander, S.L., Tomlin, C.J.: Quadrotor helicopter flight dynamics and control: theory and experiment. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, vol. 2 (2007)
23. Hoffmann, G.M., Waslander, S.L., Tomlin, C.J.: Quadrotor helicopter trajectory tracking control. In: AIAA Guidance, Navigation and Control Conference and Exhibit, pp. 1–14 (2008)
24. Vyhlidal, T., Zíítek, P.: Mapping based algorithm for large-scale computation of quasi-polynomial zeros. IEEE Trans. Autom. Control **54**(1), 171–177 (2009)
25. Armah, S., Yi, S., Choi, W.: Design of feedback control for quadrotors considering signal transmission delays. Int. J. Control Autom. Syst. (accepted)
26. Yepes, A.G., Freijedo, F.D., Doval-Gandoy, J., Lopez, O., Malvar, J., Fernandez-Comesa, P.: Effects of discretization methods on the performance of resonant controllers. IEEE Trans. Power Electron. **25**(7), 1692–1712 (2010)