Preprints, 12th IFAC Workshop on Time Delay Systems
June 28-30, 2015. Ann Arbor, MI, USA

Altitude Regulation of Quadrotor Types of UAVs Considering Communication Delays

Stephen Armah* and Sun Yi**

** **Department of Mechanical Engineering, North Carolina A&T State University,*
Greensboro, NC 27411, USA (e-mail: skarmah@aggies.ncat.edu, syi@ncat.edu).

Abstract: Estimation of transmission delays caused by wireless communication and analysis of the delay effects is one of the critical challenges to be considered in designing controllers for quadrotor types of unmanned aerial vehicles (UAVs). This paper presents an estimation method using experimental data and analytical solutions of delay differential equations (DDEs). For the approach, measured transient altitude responses are compared to time-domain descriptions obtained from the analytical solutions. That makes use of the Lambert W function for first-order DDEs. The dominant characteristic roots are obtained in terms of system parameters including the delay. Proportional controllers are used to generate the altitude responses for estimation. The effects of the time delay on the responses are analyzed. Then, proportional plus velocity controllers are designed to obtain better transient altitude responses. MATLAB/Simulink is used for simulations, experiments, and analytical solutions of the DDEs in terms of Lambert W function.

Keywords: UAVs, Flight Control, Delay Differential Equation, Delay Estimation, Lambert W Function

1. INTRODUCTION

Estimating and analyzing time delays in dynamic systems is an important issue in many applications. Estimating delays is a challenging problem and has been an area of great research interest in fields as diverse as radar, sonar, seismology, geophysics, ultrasonic, controls, and communications (Kobra et al., 2013; Ren, 2005). Although considerable efforts have been made on parameter estimation, there are still many open problems in time-delay identification due to difficulty in formulation (Yi et al., 2012; Belkoura et al., 2009; Richard, 2003).

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 designing effective control systems for UAVs is existence of signal transmission delay, which has nonlinear effects on the flight performance of autonomously controlled UAVs. A controller designed using a non-delay system model may result in disappointingly slow and oscillating responses due to the delays. For large delays (e.g., larger than $0.20s$) the system response might not be stabilized or converged due to increased torque, and this poses a significant challenge (Ailon and Arogeti, 2014).

Parrot AR.Drone 2.0 is a UAV controlled through Wi-Fi and, thus, its dynamics contains a time delay. Refer to Section 2 for the control architecture. The time delay is attributed to: (1) the processing capability of the host computer, (2) the electronic devices processing the motion signals, (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, etc. For UAVs wireless communication delays may not be critical when the controllers are on-board. However, delays have significant effects when the control software is run on an external computer and signals are transmitted wireless. For example, the experiments on the drone in this paper were conducted using MATLAB/Simulink on an external computer, and the navigation data (yaw, pitch, roll, altitude, etc.) decoding process contributes to the delay. Also, the different types of numerical solvers introduces delay.

This paper presents how to estimate the constant time delay in AR.Drone 2.0 altitude control system. In real applications, drones fly around and the time delay may vary. 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. Here, the delay is not restricted to be a multiple of the sampling interval. In this brief, experimental data and analytical solutions of infinite-dimensional continuous delay differential equations (DDEs) are used. In Butcher and Torkamani, the finite-dimensional continuous time approximation (CTA) was used to approximately solve DDEs for estimation of constant and time-varying delays. The accuracy is dependent on the size of the Chebyshev spectral differentiation matrix.

For the approach in this paper, measured transient responses are compared to time-domain descriptions obtained by using the Lambert W function. Then, the dominant characteristic roots are obtained 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 responses are analyzed. Then, proportional plus velocity (PV) control is designed to obtain better transient responses.

This paper continues with a description of quadrotor's altitude model and the AR.Drone 2.0 control system in Section 2. Section 3 presents the approaches used for estimating the system's time delay. In Section 4, the P and PV controllers are presented. In Section 5 results are summarized. Concluding remarks and future work is presented in Section 6.

2. ALTITUDE MODEL AND CONTROL SYSTEM

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 the attitude defined using the Euler angles (Corke, 2011). The quadrotor has four rotors, labelled 1 to 4, mounted at the end of each cross arm. The rotors are driven by electric motors powered by electronic speed controllers. The vehicle's total mass is m and d is distance from the motor to the center of mass. The total upward thrust, $T(t)$, on the vehicle is given by

$$T(t) = \sum_{i=1}^{i=4} T_i\,(t) \qquad (1)$$

where $T_i(t) = a\omega_i^2(t),\ \ i = 1, 2,3,4,\ \omega_i(t)$ is the rotor speed, $a > 0$ is the thrust constant, and t is time (Corke, 2011). The equation of motion in the z-direction can be obtain as (Randal, 2008)

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

where $\omega(t)$ is the rotor average angular speed necessary to generate $T(t)$ and g is the gravitational acceleration. Thus, to control the altitude, $z(t)$, of the quadrotor only $\omega(t)$ needs to be varied, 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. $\dot{z}_{ref}(t)$ has to be constrained to $[-1\ \ 1]ms^{-1}$, to prevent damage. The drone's flight management system sampling time, T_s is $0.065s$, which is also the sampling time at which the control law is executed and the navigation data received.

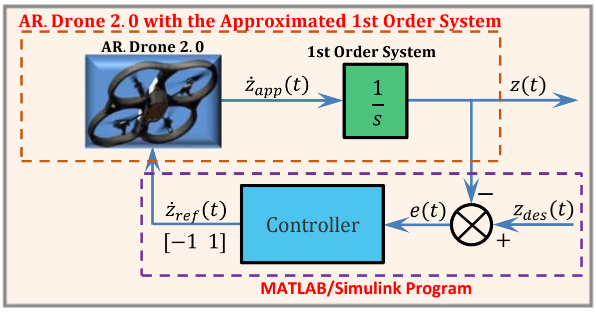

Fig. 1: Diagram for altitude control of the AR.Drone 2.0.

The setup to control the drone's altitude motion using MATLAB/Simulink program is shown in Fig. 1. The error between the desired reference input, $z_{des}(t)$, and the system altitude response, $z(t)$, is denoted as $e(t)$. The altitude motion dynamics in (2) is used to determine $\omega(t)$ from $\dot{z}(t)$, which is obtained from $\dot{z}_{ref}(t)$. The rotors rotate with the same $\omega(t)$, which will generate $T(t)$ to produce $z(t)$. These computations take place on-board the drone control engine program written in C. In this paper, the motor dynamics is assumed to be very fast such that the altitude control system can be represented as a first-order system using an integrator (Fig. 1). 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.

The MATLAB/Simulink program setup developed for the experiments is shown in Fig. 2. The vertical speed control input constraints are applied using the saturation block. For the simulations, the overall constant time delay, T_d, in the system is represented as actuator time delay, and it is implemented using the transport delay block.

The experiments were performed in an office environment, with the AR.Drone 2.0 indoor hull attached. The drone is connected to the host PC using Wi-Fi, and data streaming, sending and receiving, are made possible using UDPs (user datagram protocols). UDP is a communication protocol, an alternative to TCP that offers a limited amount of service when messages are exchange between computers in a network that uses IP.

The drone navigation data (from the sensors, cameras, battery, etc.) are received, and the control signals are sent, using AT commands. AT commands are combination of short text strings sent to the drone to control its actions. The drone has ultrasound sensor for ground altitude measurement (at the bottom). It has 1GHz 32 bit ARM Cortex A8 processor, 1GB DDR2 RAM at 200MHz, and USB 2.0 high speed for extensions.

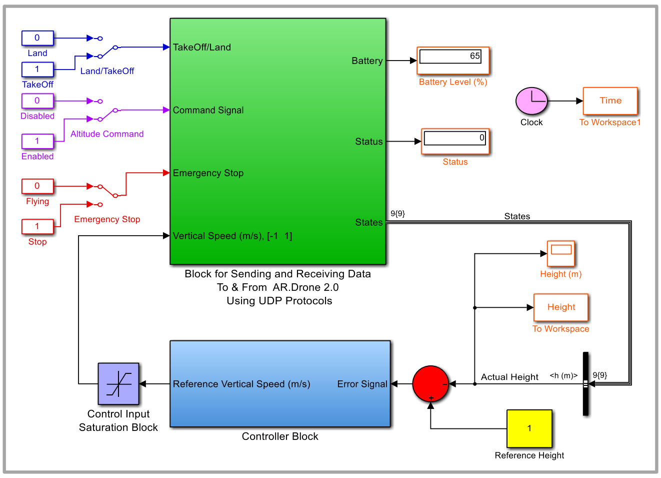

Fig. 2: Simulink diagram for controlling the AR.Drone 2.0.

3. TIME-DELAY ESTIMATION

A continuous control system can be represented for time-delay estimation (TDE) as (Svante, 2003)

$$z(t) = G_p u(t - T_d) + n(t) \qquad (3)$$

where G_p is an LTI dynamic system, single-input-single-output (SISO), $z(t)$ is measured signal, $u(t)$ is the control input signal, and $n(t)$ is measurement noise (here, $n(t) = 0$). The time delay to be estimated is an explicit parameter in the model and it is not restricted to be a multiple of the sampling time. The estimation problem can be formulated using analytical solutions to DDEs. Consider the first-order scalar homogenous DDE shown in (4) below. Unlike ordinary differential equations (ODEs), two initial conditions need to be specified for DDEs: a preshape function, $g(t)$, for $-T_d \leq t < 0$, and initial point, z_o, at $t = 0$.

$$\dot{z}(t) - a_o z(t) - a_1 z(t - T_d) = 0 \tag{4}$$

The characteristic equation of (4) is given by

$$s - a_o - a_1 e^{-sT_d} = 0 \tag{5}$$

Then, the characteristic equation in (5) is solved as (Yi et al., 2012)

$$s = \frac{1}{T_d} W(T_d a_1 e^{-a_o T_d}) + a_o \tag{6}$$

The Lambert W function is defined as $W(x)e^{W(x)} = x$ (Corless et al., 1996). As seen in (6), the characteristic root, s, is obtained analytically in terms of parameters, a_o, a_1, and T_d. The solution in (6) has an analytical form expressed in terms of the parameters of the DDE in (4). One can explicitly determine how the time delay is involved in the solution and, furthermore, how each parameter affects each characteristic root. That enables one to formulate estimation of time delays in an analytic way. Each eigenvalue can be distinguished with the branches of the Lambert W function, which is already embedded in MATLAB (Yi et al., 2012).

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 $k = -1$ (Shinozaki and Mori, 2006). For the DDE in (4), one has to consider two possible cases for rightmost characteristic roots: characteristic equations of DDEs as in (5) 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 (Yi et al., 2012). For ODEs, an estimation technique using the logarithmic decrement provides an effective way to estimate the damping ratio, ξ (Palm, 2010). The technique makes use of the form

$$s = -\xi\omega_n \pm j\omega_n\sqrt{1 - \xi^2} \tag{7}$$

for obtaining s of second-order ODEs. The variables ξ and ω_n are obtained from the response of the system, and different approaches can be applied depending on the nature of the response, oscillatory and non-oscillatory (Yi et al., 2012). Here, the transient properties for oscillatory responses are used. Properties such as the maximum overshoot, M_o, peak time, t_p, and settling time, t_s, are related to ξ and ω_n, as shown below (Palm, 2010)

$$M_o = 100e^{\left(\frac{-\xi\pi}{\sqrt{(1-\xi^2)}}\right)}, t_p = \frac{\pi}{\omega_n\sqrt{(1-\xi^2)}}, \; t_s = \frac{4}{\xi\omega_n} \tag{8}$$

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

Step 1: Calculate ξ and ω_n based on the system altitude response

Step 2: Calculate the 'dominant' roots using $s = -\xi\omega_n \pm j\omega_n\sqrt{(1 - \xi^2)}$

Step 3: Solve the nonlinear equation $s = \frac{1}{T_d} W(T_d a_1 e^{-a_o T_d}) + a_o$ for T_d

The equation in Step 3 can be solved using nonlinear solver such as *fsolve* in MATLAB.

For comparison, numerical approach is also used. In this approach the transient properties, M_o and t_p, of the drone's altitude responses are compared to those of simulation responses for the estimation of T_d.

4. P AND PV CONTROLS

The system has an integral term in the closed-loop transfer function and, thus, only P and PV feedback controllers are used to generate vertical speed signal. PV control, unlike PD control, does not yields numerator dynamics. The P-feedback controller is used in the determination of the time delay, and the PV-feedback controller is used to analyze the effect of the time delay on the AR.Drone 2.0 altitude response. Figs. 3 and 4 show the Simulink setups developed for conducting the simulations, and the controller gains were used in Fig. 2 for the experiments. The transfer function of the time-delay closed-loop system for the P controller is given as

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

This time-delay system is a retarded type. As expected the characteristic equation is transcendental, and therefore the closed-loop poles are infinite; the exponential term in the characteristic equation will introduce oscillations into system. Comparing the characteristic equation of the closed-loop system in (9) to the first-order system in (5), $a_o = 0$ and $a_1 = -K_P$.

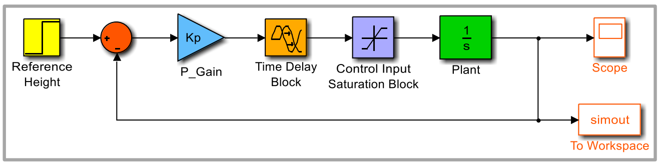

Fig. 3: Simulink block diagram for P-feedback control.

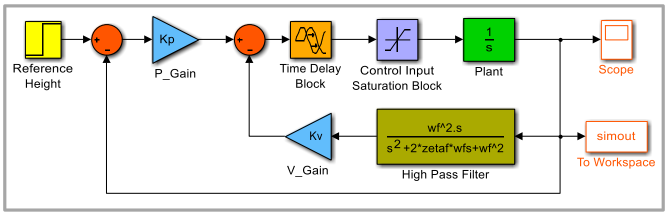

Fig. 4: Simulink block diagram for PV-feedback control.

The effect of T_d on the drone's altitude response was studied using analytical, simulation, and experimental approaches by designing PV controller. Suitable PV controller gains, K_p and K_v, are obtained to improve on the transient response performance. High pass filter (HPF) with damping ratio, $\xi_f = 1.0$ was used for the derivative controller. A suitable natural frequency, ω_f, value was selected, by tuning and the use of Bode plot, for the filter. The transfer function of the time-delay closed-loop system for the PV controller, neutral type, is given as

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

5. RESULTS AND DISCUSSION

5.1 Estimation of the Time Delay

Initially, the drone's altitude responses were obtained for different values of K_p, as shown in Fig. 5. Note that if there is no delay ($T_d = 0$), there should be no overshoot. The characteristic root is $-K_p$ (refer to Eq. (9)), which is a real number. However, as seen in Fig. 5, the delay introduces imaginary parts in the roots and, thus, oscillation in the responses. Therefore, the delay has to be precisely estimated and considered in designing control. For ease of analyzing the responses are shifted to start at $(0s, 0m)$. The gain value, $K_p = 1.0$ seems to be ideal for the controller since the response has no overshoot, however, the response is very slow. As it can also be observed, increasing K_p makes the response faster, the rise time becomes shorter, but introduces higher M_o. This is partly due to the time delay in the system, which introduces nonlinearity on the dynamics.

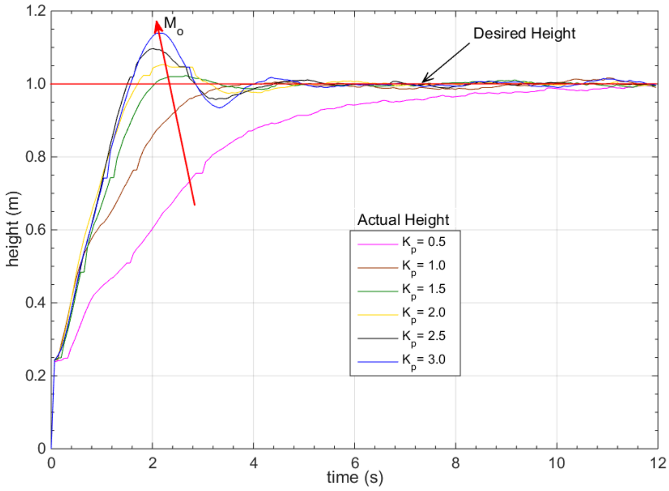

Fig. 5: Experimented altitude responses: varying K_p.

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. Using simulation, an appropriate $K_p = 1.31$ was selected, that gives a response with a sufficient overshoot for estimation and with minimum saturation effect.

5.1.1 Numerical Method

The drone's altitude response oscillates (see Fig 5) and, thus, the system has two complex conjugate dominant, rightmost, roots. Table 1 shows a summary of the simulation altitude responses transient properties, by varying T_d at $K_p = 1.31$, where K is a real constant tuning parameter, a multiplier of T_s. The drone's altitude responses with $K_p = 1.31$ are shown in Fig. 6, with Table 2 displaying their corresponding M_o and t_p values. The value, $t_p = 3.055$s, with the highest $M_o = 2.300\%$ gives the largest T_d. Comparing the $M_o = 2.300\%$ to the results in Table 2, T_d is estimated as $5.6646T_s$, which gives $0.368s$.

Table 1. Simulated altitude responses: K_p = 1.31

K	$T_d = KT_s$ (s)	M_o (%)
4.0000	0.260	0.000
5.0000	0.325	0.419
5.6000	0.364	2.067
5.6640	0.368	2.298
5.6645	0.368	2.301
5.6646*	0.368*	2.300*
5.6660	0.369	2.305

Table 2. Experimented altitude responses: K_p = 1.31

	Flight				
	1	2	3	4	5
M_o (%)	2.300*	2.290	2.300	2.270	2.140
t_p (s)	3.055*	3.084	3.575	3.194	3.096

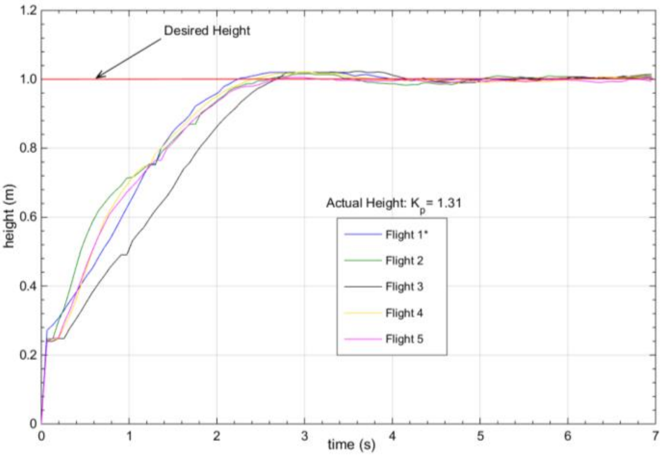

Fig. 6: Experimented altitude responses: K_p = 1.31.

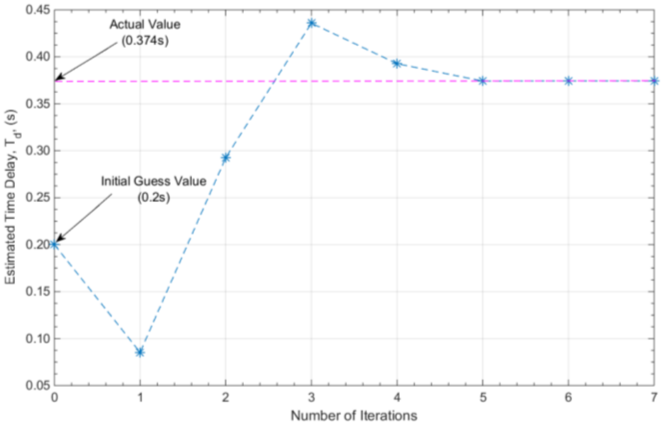

Fig. 7: Iteration of *fsolve* to estimate the time-delay.

5.1.2 Use of Characteristic Roots

From Section 5.1.1, $M_o = 2.300\%$ and $t_p = 3.055$s, thus, ξ and ω_n are computed as 0.7684 and 1.6069 $rads^{-1}$, respectively using (8). Using (7), the dominant characteristic roots, approximated, are calculated as $s = -1.2347 \pm 1.0284j$. Then, from (6), T_d is determined as $0.374s$ using *fsolve* in MATLAB with initial guess value of $0.2s$. See Fig. 7 for the iteration of the *fsolve*.

5.2 PV Control: Design and Implementation

As above the estimated time delay, using both the numerical and the analytical methods, is approximately $0.37s$. A MATLAB-based software package (Vyhlídal, 2013) was used to study the stability of the neutral type time-delay system, by numerically solving the characteristic equation in Eq. (10). The closed-loop system characteristic roots within a specified region are then plotted for various K_v values. Fig. 8 shows the spectrum distribution of the characteristic roots and Table 3 shows a summary of the rightmost (i.e., dominant) roots for each system. The value $K_v = 0.3$ yields the most stable rightmost roots among them.

Table 3. Rightmost characteristic roots of the PV control system with K_p = 2.0 and T_d = 0.37s

K_v	Rightmost Complex Roots
0.0	-1.42 ± 3.07j
0.1	-1.98 ± 3.25j
0.3*	-3.25 ± 24.75j
0.5	-1.84 ± 6.98j
0.7	-0.89 ± 7.47j

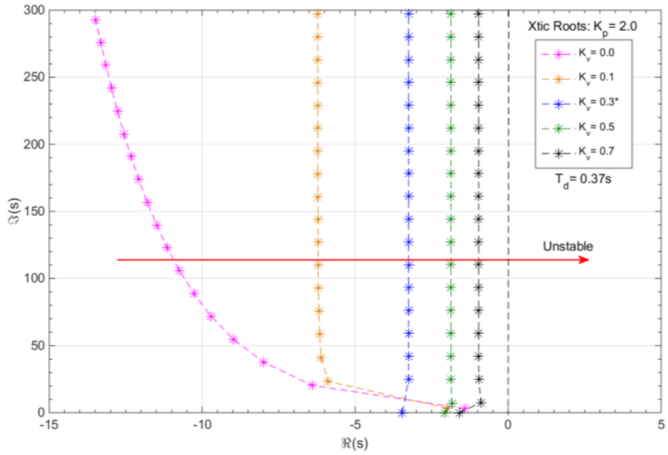

Fig. 8: PV control system characteristic roots spectrum distribution with K_p = 2.0 and T_d = 0.37s.

The corresponding simulation altitude responses for the system were also obtained for the various K_v values, not shown in this paper. It can be seen that as K_v increases at $K_p = 2.0$ and $T_d = 0.37$s, M_o decreases and the rise time becomes longer. At higher values of K_v, the response is oscillatory and the system becomes unstable. This is also observed in Fig. 8, that as K_v increases the roots move to the right, increasing the instability in the system.

Now, based on these analyses, a controller with $K_p = 2.0$ and $K_v = 0.3$ was selected as the most suitable, with closed-loop system response transient properties of $M_o = 0.44\%$, $t_s = 1.52s$, and $t_p = 1.76s$. Using these controller gains, the HPF was included in the simulation control system, and its effects on the altitude transient response was studied for different values of ω_f. It is observed that at smaller ω_f values the response oscillates, and at higher values the response distorts. The oscillations and the distortions effects were reduced by using the high-order solver, *ode8 (Dormand-Prince)*.

The HPF with $\omega_f = 38\,rads^{-1}$ and $\xi_f = 1.0$ was then selected, with poles of -38 repeated. Now, looking at the poles distribution of the system in Fig. 8, it can be observed that the poles of this filter is located to the left than the poles of the PV-feedback closed-loop system, without the filter effect. Thus, this filter will respond faster, therefore, it will have smaller effect on the drone's altitude transient response. The filter's cutoff frequency was determined as $5.68\,rads^{-1}$ (0.90 Hz). Fig. 9 and Table 4 shows the simulation altitude responses and their corresponding transient properties, with the HPF and $K_p = 2.0$, for different K_v values. The results with $K_p = 2.0$ and $K_v = 0.3$ shows an improved transient response performance, which suggests that the estimation of delay and analysis presented help.

Table 4. PV controller altitude response transient properties, with K_p = 2.0 and the high pass filter

	K_v				
	0.0	0.1	0.3*	0.5	0.7
	Simulation (T_d = 0.37s)				
M_o (%)	15.10	10.10	0.32	0.15	0.70
t_p (s)	1.82	1.76	1.69	3.64	4.36
t_s (s)	3.28	2.87	1.52	2.33	3.04
	Experiment				
M_o (%)	8.40	5.30	2.92	0.80	0.40
t_p (s)	2.21	2.15	2.18	4.76	4.07
t_s (s)	4.00	2.67	2.86	1.99	2.48

Fig. 10 and Table 4 also shows the experimented altitude responses and their corresponding transient properties, with the HPF and $K_p = 2.0$, for different K_v values. It can be seen that as the K_v value increases M_o decreases and in general the responses becomes slower. The PV controller performed better for $K_v = 0.3, 0.5,$ and 0.7 at $K_p = 2.0$.

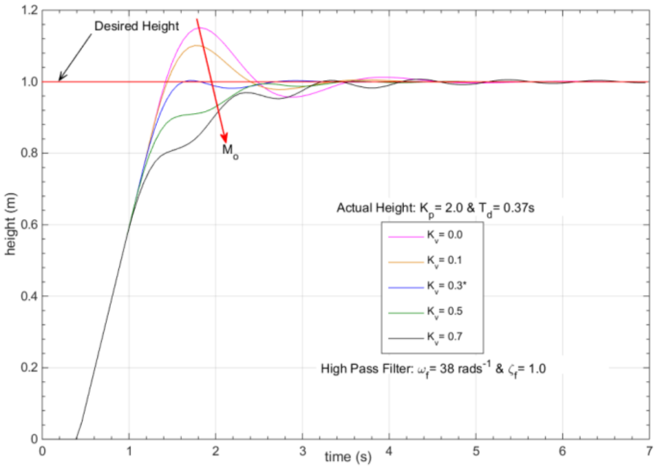

Fig. 9: Simulated PV controller altitude responses with K_p = 2.0 and T_d = 0.37s.

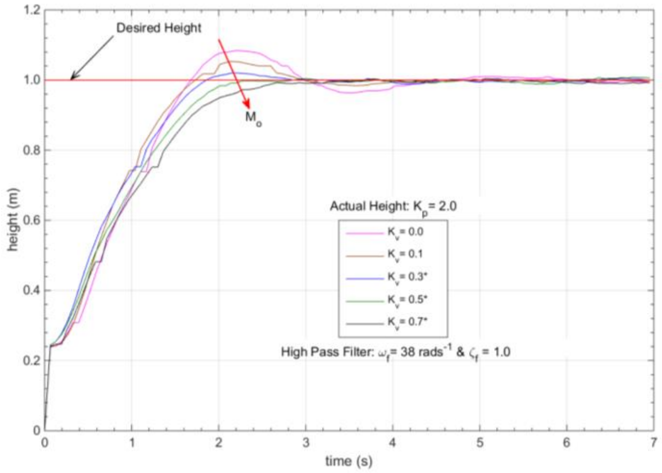

Fig. 10: Experimented PV controller altitude responses with $K_p = 2.0$.

6. CONCLUSIONS

This study has demonstrated how to estimate the time delay in a quadrotor UAV, Parrot AR.Drone 2.0, altitude control system. Through numerical and analytical approaches, the time delay was estimated as $0.37s$. 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, and the designed PV controller performed better than the P controller, especially with gains of $K_v = 0.3, 0.5,$ and 0.7 at $K_p = 2.0$.

The simulations and experiments were conducted using MATLAB/Simulink high-order solver, *ode8 (Dormand-Prince)*. Investigation through trials revealed that selection of the solvers has 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.

In future, robust controllers for the drone's attitude and position (x and y) motions can be developed by estimating and incorporating the time delay in the control systems. This problem is significantly more challenging, since the equation of motions are more complex compared to that of the altitude motion. Furthermore, the presented time-delay estimating methods can be extended to general systems of DDEs (higher than first order), and be applied to delay problems in network systems and fault detection of actuators.

7. ACKNOWLEDGEMENT

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

Ailon, A., and Arogeti, S. (2014). Study on the effects of time-delays on quadrotor-type helicopter dynamics. *Control and Automation (MED)*, 22, 305-310. IEEE.

Asl, F. M., and Ulsoy, A. G. (2003). Analysis of a system of linear delay differential equations. *Journal of Dynamic Systems, Measurement, and Control*, 125(2), 215-223.

Belkoura, L., Richard J.P., and Fliess, M. (2009). Parameters estimation of systems with delayed and structured entries. *Automatica*, 45, 1117-1125.

Corke, P. (2011). *Robotics, vision and control: fundamental algorithms in MATLAB*, vol. 73, pp 78-86. Springer.

Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., and Knuth, D.E. (1996). On the Lambert W function. *Advances in Computational Mathematics*, 5, 329-359.

Gotbolt, B., Vitzilaios, N.I., and Lynch, A.F. (2013). Experimental validation of a helicopter autopilot design using model-based PID control. *Journal of Intelligent & Robotic Systems*, 70, 385-399.

Kobra, M., and Morteza, Z.S. (2013). Implementation of time delay estimation using different weighted generalized cross correlation in room acoustic environments. *Life Science Journal*, 10 (6s), 846-851. Life Sci.

Lim, J. G., and Jung, S. (2013). Design of a Time-Delayed Controller for Attitude Control of a Quadrotor System. *Intelligent Robotics and Applications*, 274-280. Springer, Berlin Heidelberg.

Palm, W.J. (2010). *System dynamics*. 2nd Ed. pp 836. McGraw-Hill, Boston.

Randal, W. B. (2008). Quadrotor dynamics and control. Brigham Young University. Retrieved from http://www.et.byu.edu/groups/ece490quad/control/quadrotor.

Ren, X. (2005). Online identification of continuous-time systems with unknown time delay. *Automatic Control*, 50, 1418-1422. IEEE Trans.

Richard, J.P. (2003). Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39, 1667-1694.

Shinozaki, H., and Mori, T. (2006). Robust stability analysis of linear time-delay systems by Lambert W function: some extreme point results. Automatica, 42.

Svante, B. (2003). A survey and comparison of time-delay estimation methods in linear systems. Retrieved from www.control.isy.liu.se/research/.../.

Torkamani, S., and Butcher, E. A. (2013). Delay, state, and parameter estimation in chaotic and hyperchaotic delayed systems with uncertainty and time-varying delay. *International Journal of Dynamics and Control*, 1(2), 135-163.

Vyhlídal, T. (2013). Quasi-polynomial mapping based rootfinder. CTU in Prague. Retrieved from http://www.cak.fs.cvut.cz/algorithms/qpmr.

Yi, S., Nelson, P.W., and Ulsoy, A.G. (2010). *Time delay systems: analysis and control using the Lambert W Function*. 1st Ed. World Scientific, New Jersey.

Yi, S., Wonchang, C., and Taher, A. (2012). Time-delay estimation using the characteristic roots of delay differential equations. *American Journal of Applied Sciences*, 9 (6), 955-960.