



Research Article

Volume 2 Issue 5 - April 2018

DOI: 10.19080/RAEJ.2018.02.555598

Robot Autom Eng J

Copyright © All rights are reserved by Stephen K Armah

Adaptive Control for Quadrotor UAVs Considering Time Delay: Study with Flight Payload



Stephen K Armah* and Sun Yi

Department of Mechanical Engineering, North Carolina A&T State University, USA

Submission: February 06, 2018; **Published:** April 17, 2018

***Corresponding author:** Stephen K Armah, Department of Mechanical Engineering, North Carolina A&T State University, 1601 E. Market Street, Greensboro, NC 27411, USA, Tel: (518) 961-0906; Email: skarmah@aggies.ncat.edu

Abstract

Design of effective control for unmanned aerial vehicles (UAVs) requires consideration of several sources of uncertainty. These undesired uncertainties affect the flight stability and performance in an unpredictable manner. The existence of transmission delays caused by wireless communication and payload variation are among such critical challenges. Adaptive control (AC) can lead to high performance tracking in the presence of uncertainties. This paper presents the application of model reference adaptive control (MRAC) to quadrotor types of UAVs considering the time delay in the altitude control system. MATLAB system identification tool is applied to obtain the altitude motion model, without time delay, for the quadrotor. Proportional-plus-velocity (PV) and PV-MRAC altitude control systems are designed, by incorporating an estimated constant time delay. The designed controllers are validated using simulations and flight tested in an indoor environment. The robustness of the PV-MRAC controller is tested against the baseline non-adaptive PV controller using the quadrotor's payload capability..

Keywords : Adaptive control; Flight payload; Flight stability; MIT rule; PV control; Quadrotor; Time delay

Abbreviations : UAV: Unmanned Aerial Vehicles; AC: Adaptive Control; MRAC: Model Reference Adaptive Control; PV: Proportional-Plus-Velocity; UDP: User Datagram Protocols; CTA: Continuous Time Approximation; DDE: Delay Differential Equations; LTI : Linear Time-Invariant

Introduction

The application of autonomous control techniques to quadrotor types of unmanned aerial vehicles (UAVs) has been the focus of active research during the past decades. Design of control for aircraft requires several important considerations. There are numerous sources of uncertainty. For example, devices are ageing and wearing (e.g., actuator degradation), external disturbances, control input saturation, payload fluctuations, and potentially uncertain time delays in processing or communication [1]. These undesired nonlinearities affect the flight stability and performance of the controlled systems. Adaptive control (AC) is a candidate to resolve the issues, because of its ability to generate high performance tracking in the presence of uncertainties [1].

Safe, cooperative transportation of possibly large or bulky payloads by UAVs is extremely important in various missions, such as military operations, search and rescue, and personal assistants. Another huge area of application is in the service industry, where commercial quadrotor types of UAVs or drones are being used in the delivery of packages of varying masses. There are several substantial challenges and open research

questions related to this application. For example, this task requires that hovering vehicles remain stable and balanced in flight as payload mass is added to the vehicle. If payload is not loaded centered or the vehicle properly trimmed for offset loads, the drone experiences bias forces that must be rejected [2]. Flying with a suspended load, known as slung load or sling load, can be a very challenging and sometimes hazardous task because the suspended load significantly alters the flight characteristics of the quadrotor [3,4].

Time delays exist in autonomous dynamic systems when signals are transmitted wirelessly. One of the challenges in designing effective control systems for UAVs is existence of signal transmission delay. The delay has nonlinear effects on the flight performance of autonomously controlled UAVs. For autonomous aerial robots, typical values of the time delay have been known to be around 0.4 ± 0.2 s [5] or 0.2s [6]. For large delays (e.g., larger than 0.2s) the system response might not be stabilized or converged due to the dramatic increase in torque. This poses a significant challenge [7]. Since its effect is not trivial the delay need be estimated and considered in designing controllers.

Parrot AR. Drone 2.0, a commercialized drone, is controlled via WiFi network. Data streaming, commanding and reading the vehicle sensors are done using user datagram protocols (UDPs). The drone dynamics contains a time delay due to such communication. Refer to Section 2 for the drone's altitude control architecture. The overall time delay is attributed to:

- I. The processing capability of the host computer,
- II. The electronic devices processing the motion signals,
- III. The measurement reading devices, e.g., the distance between the ultrasonic altitude sensor and the ground, and
- IV. The software on the host computer for implementing the controllers.

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 wirelessly. For example, the experiments presented in this paper were conducted using MATLAB/Simulink on an external computer, and decoding process of the navigation data (yaw, pitch, roll, altitude, etc.) contributes to the delay. Also, the numerical solvers introduce additional delay. 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 [8,9]. A considerable number of system identification methods have been reported in the literature [10,11]. However, most of the existing methods for transfer function identification do not consider the process delay (or dead-time) or just assume knowledge of the delay [12]. Even though considerable efforts have been made on parameter estimation, there are still many open problems and there is no common approach in time-delay identification due to difficulty in formulation [13]. Several approaches for estimation of delay have been introduced in the literature [14-18]. In [14] for example, the finite dimensional Chebyshev spectral continuous time approximation (CTA) was used to solve delay differential equations (DDEs) for estimation of constant and time-varying delays.

In this paper, the constant delay applied in the drone's altitude control system was estimated using an approach based on analytical solutions to DDEs [19,20]. The time delay has been estimated as 0.36s in [21,22]. The altitude dynamics was assumed to be linear time-invariant (LTI) first order and the time delay, assumed to be single constant, was incorporated into the model as an explicit parameter. Experimental data and analytical solutions of infinite dimensional continuous DDEs were used for estimation. Measured transient responses were 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) time-delay control system, retarded type, was used to generate the responses for estimation of the delay.

The contribution of this paper is the identification of altitude motion model, without time delay, for the drone and the extension of incorporating the estimated time delay of 0.36 sin the design of an AC system. Then, validating the controller using the payload capability of the quadrotor. A model reference adaptive control (MRAC) based on the MIT rule is designed. In general, the MIT rule cannot guarantee stability [23], so the MRAC is combined with a tuned proportional-plus-velocity (PV) controller. To validate the robustness of the PV-MRAC controller, the results are compared to a baseline non-adaptive PV-controller.

This paper continues with the description of quadrotor's altitude dynamics and the AR. Drone 2.0 control system in Section 2. The altitude model identification and the design of the PV and PV-MRAC control systems are presented in Section 3. Simulations and experiments setups are presented in Section 4. In Section 5 results are summarized. Concluding remarks and future work are presented in Section 6.

Quadrotor Altitude Dynamics and Control System

The notation for a typical quadrotor model is shown in Figure 1. Quadrotors has three main coordinate systems attached to it; the body-fixed frame, $\{b\}$, the vehicle frame, $\{v\}$, and the global inertial frame, $\{w\}$. There are two other coordinate systems, not shown, that are of interest, the vehicle-1 frame, $\{v^1\}$, and vehicle-2 frame, $\{v^2\}$ [24,25]. The configuration of a quadrotor is represented by six degrees of freedom in terms of position, $(x_w, y_w, z_w)^T$, and the attitude defined using the Euler angles (roll, pitch, and yaw), $(\phi, \theta, \psi)^T$, which gives a 12-state equation of motion [26]. The AR. Drone quadrotor platform adopted for this work has the same architecture design shown in Figure 1.

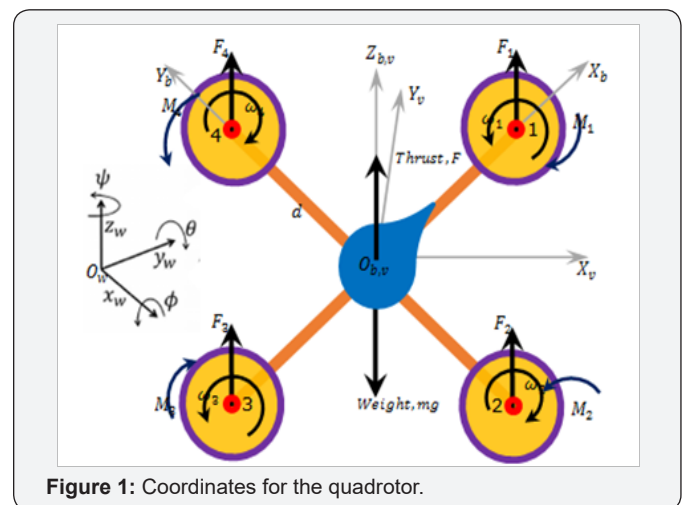


Figure 1: Coordinates for the quadrotor.

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. Each rotor is controlled independently by nested feedback loops as shown in Figure 2. The inner attitude control loop uses on-board accelerometers and gyros to control the roll, pitch, and yaw while the outer position control loop uses estimates of position and velocity of the center of mass to control the trajectory in three dimensions [27]. The components of angular velocity of the robot in the body frame are p , q , and r .

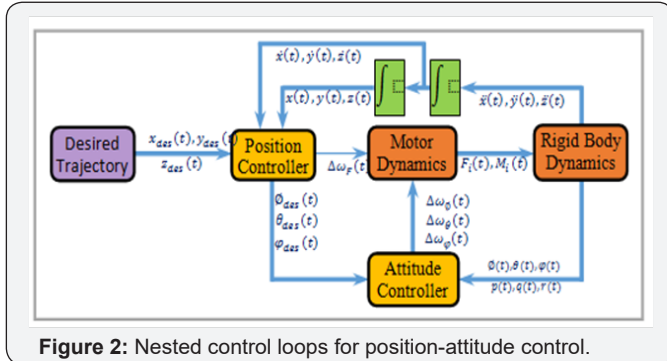


Figure 2: Nested control loops for position-attitude control.

The vehicle's total mass is m and d is distance from the motor to the center of mass. The total upward thrust, $u(t)$, on the vehicle is given by $F(t) = \sum_{i=1}^4 F_i(t)$, and, $F_i(t) = k_F \omega_i^2(t)$, $i=1,2,3,4$ where, $\omega_i(t)$ is the rotor angular speed and $k_F > 0$ is the thrust constant [26]. In addition to forces, each rotor produces a moment perpendicular to the plane of rotation of the blade, $M_i(t)$. The equation of motion in the z -direction can be obtained as [24,27]

$$z(t) = \frac{F(t)}{m} \cos \theta \cos \phi - g \quad (1)$$

Linearizing (1) at an operating point that corresponds to a nominal hover state, z_o , where the roll and pitch angles are small and the nominal thrusts, F_o , from the propellers satisfy. At the hover state the average motor angular speed, $F_o = mg$, is given by $\omega_o \cdot \omega_o = \sqrt{mg/k_F}$. Then, the change in the vertical acceleration can be derived as [27-29]

$$\Delta \ddot{z}(t) = \left(\frac{8k_F \omega_o}{m} \right) \Delta \omega_F(t) \quad (2)$$

Thus, only the rotor's average angular speed, $\omega_F(t)$, necessary to generate $F(t)$ needs to be controlled to regulate the altitude, $z(t)$, of the quadrotor, since m , and k_F are g 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 motion commands for the drone are encoded under a specific protocol, where high-level control signals, including $\dot{z}_{ref}(t)$, are normalized between -1 and 1 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 Figure 3. The desired height and the actual system response are denoted $z_{des}(t)$ and $z(t)$, respectively. The dynamics is implemented using (2), used to determine $\omega_F(t)$ from the actual drone vertical speed, $\dot{z}(t)$. The rotors rotate at equal speed, $\omega_F(t)$, which will generate $F(t)$ to make $z(t)$ reach the reference ($z_{des}(t) = 1m$). These computations take place in the on-board control system programmed in C. The simulation setup used is shown in Figure 4. The overall time delay, T_d , single constant, in the system is represented as actuator time delay, as an explicit plant parameter. $G(s)$ is the plant model for the combined motor and rigid body altitude dynamics without uncertainty effect.

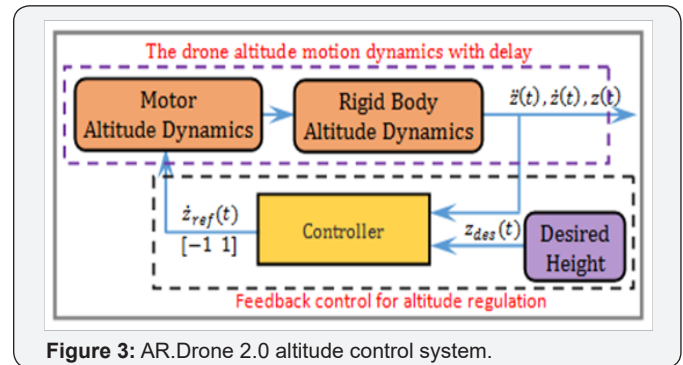


Figure 3: AR.Drone 2.0 altitude control system.

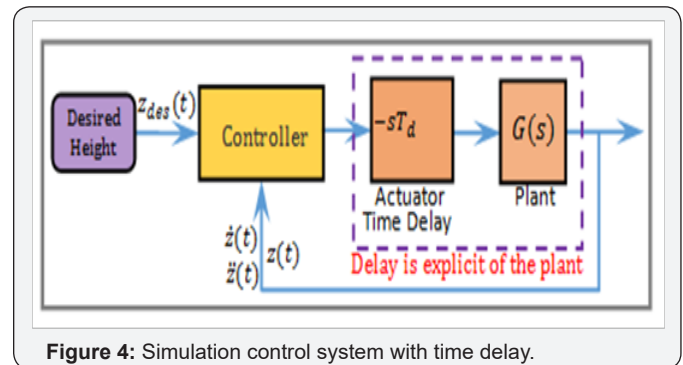


Figure 4: Simulation control system with time delay.

Altitude Model and Design of Controllers

Altitude Model

Quadrotor altitude motion has second-order dynamics (Equation 1 & 2). Black-box approach was used to determine the model for the drone altitude control. MATLAB system identification toolbox App was applied to obtained LTI SISO ARX transfer function, stable system, without time delay, as

$$G(s) = \frac{Z(s)}{\dot{z}_{ref}(s)} = \frac{1.091s + 4.536}{s^2 + 5.809s + 0.01345} \quad (3)$$

Measured data, recorded from the drone's altitude motion response, was imported into the App for the modeling process.

The input data was the reference vertical speed, $\dot{z}_{ref}(t)$, constrained to $[-1 \ 1]$ m/s, and the output data was the vertical height, $z(t)$. A suitable P-feedback controller was used to obtain the data. The model has a fit to estimation (best fit) of 91.04%,

final prediction error (FPE) of 0.000200948, and mean squared error (MSE) of 0.0001864. The system is completely state controllable and completely observable. See Figure 5 for the open-loop unit step response of the plant.

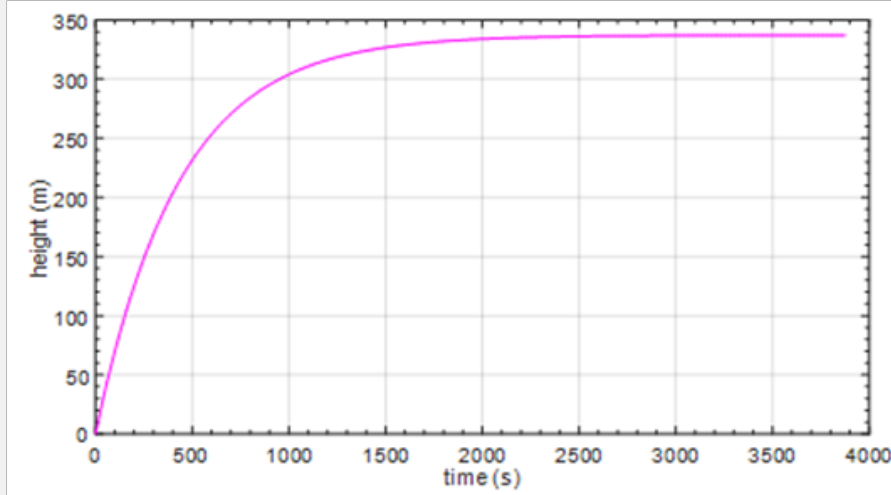


Figure 5: Open-loop unit step response of the plant.

Non-adaptive control: PV

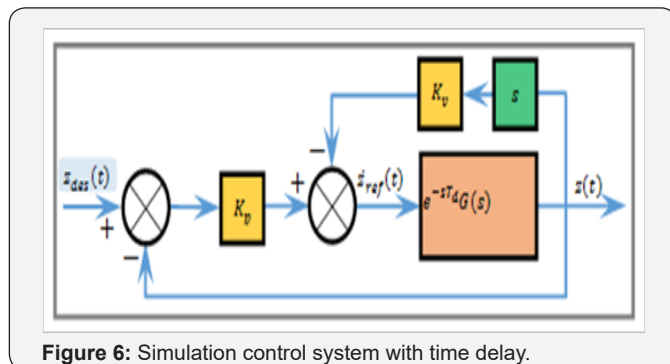


Figure 6: Simulation control system with time delay.

The conceptual application of this control strategy is common in angular position of DC motors, typically the control of robot manipulators [30]. PV-control, unlike proportional-plus-derivative (PD) control, does not induce numerator dynamics. The setup of the PV control system is shown in Figure 6, where K_p and K_v are the proportional (position) and velocity (derivative) gains, respectively. The transfer function of the time-delay closed-loop system is given as

$$\frac{Z(s)}{Z_{des}(s)} = \frac{K_p e^{-sT_d} G(s)}{1 + (K_p + K_v s) e^{-sT_d} G(s)} \quad (4)$$

Note, if $G(s) = \frac{1}{s}$, a first order plant, used in [19,20] for the estimation of the delay, then this time-delay system will be neutral type, see [21,22] for analysis of such control system. Also, if $G(s) = \frac{1}{s}$ and then we have a P-control, retarded type time-delay, system.

Adaptive control: PV-MRAC

The primary function of adaptive controller is to accommodate uncertainties, which may introduce nonlinearities on the system dynamics. There are several broad categories of AC, and in this brief MRAC is applied. The type of MRAC depends on the adjustable mechanism, and one of the modern ones is the MIT rule. The setup of the MIT rule MRAC is shown in Figure 7, where $G_p(s)$ and $G_m(s)$ are transfer functions of the plant (including the uncertainties) and the reference model respectively, and $y_p(t)$ and $y_m(t)$ are their respectively outputs.

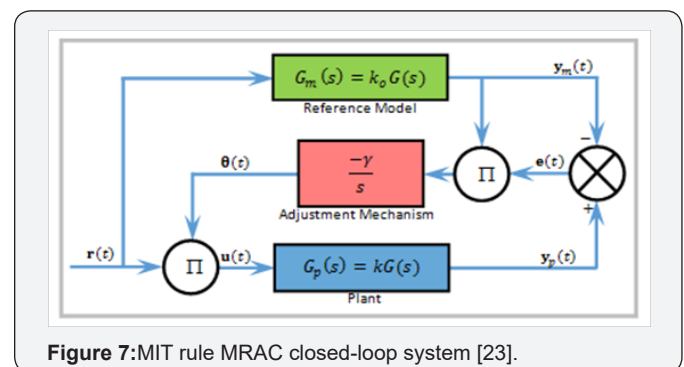


Figure 7: MIT rule MRAC closed-loop system [23].

The reference signal is $r(t)$, $\theta(t) = [\theta_1(t) \ \theta_2(t) \ \dots \ \theta_n(t)]$ is an updating parameter, $\gamma > 0$ is a tuning parameter, and the control input signal, $u(t) = \theta(t)r(t)$, where n is the order of the system. The constant k for the plant is unknown (e.g. is the effect of quadrotor payload fluctuations or time-delay or both). $G_m(s)$ is chosen as multiplied of $G(s)$ by a desired constant, k_o , so that $y_m(t)$ and $y_p(t)$ can be matched. The tracking error, $e(t) = y_p(t) - y_m(t)$, is given by

$$e(t) = G_p(s)u(t) - G_m(s)r(t) \quad (5)$$

Next is to define a cost function, $J[\theta(t)]$, in terms of $e(t)$. The goal here is to adjust $\theta(t)$ to minimize J related to $e(t)$, so that as t increases $e(t) \rightarrow 0$. One simplest way is to expressed $J[\theta(t)] = \frac{1}{2} [e(t)]^2$ [23,31-34]. The rule employs a feed-forward gain adaption to update θ , as shown below [23,32-34].

$$\frac{d\theta(t)}{dt} = -\gamma \frac{\partial J[\theta(t)]}{\partial \theta(t)} = -\gamma e(t) \frac{\partial e(t)}{\partial \theta(t)} \quad (6)$$

Where, $\left(\frac{\partial e(t)}{\partial \theta(t)}\right)$ is sensitivity derivative of the system, and it is evaluated under assumption that θ varies slowly [23,33,34].

Now, $\theta(t) = [\theta_1(t) \theta_2(t)]$ since quadrotor altitude motion dynamics is a second-order system, then. The reference model is selected based on desired transient specifications, given in the form as [23,32]

$$G_m(s) = \frac{a_{om}}{s^2 + a_{1m}s + a_{om}} \quad (7)$$

The control input is given as

$$u(t) = \theta_1(t)z_{des}(t) - \theta_2(t)y_p(t) \quad (8)$$

where $r(t) = z_{des}(t)$. Substituting into (8), and simplifying, we obtain

$$u(t) = \frac{\theta_1(t)z_{des}(t)}{1 + \theta_2(t)G_p(s)} \quad (9)$$

And substituting (9) into (5), and then differentiating with respect $\theta_1(t)$ and $\theta_2(t)$, to obtain

$$\frac{\partial e(t)}{\partial \theta_1(t)} = \left(\frac{G_p(s)}{1 + \theta_2(t)G_p(s)} \right) z_{des}(t) \quad (10a)$$

$$\frac{\partial e(t)}{\partial \theta_2(t)} = \left(\frac{G_p(s)}{1 + \theta_2(t)G_p(s)} \right) y_p(t) \quad (10b)$$

If the reference model is close to the plant, then their characteristic equations can be equally approximated ($1 + \theta_2(t)G_p(s) \cong s^2 + a_{1m}s + a_{om}$), then the MIT update rule, from (6) and (10), can be written as

$$\frac{d\theta_1(t)}{dt} = -\gamma \left(\frac{a_{1m}s + a_{om}}{s^2 + a_{1m}s + a_{om}} z_{des}(t) \right) e(t) \quad (11a)$$

$$\frac{d\theta_2(t)}{dt} = \gamma \left(\frac{a_{1m}s + a_{om}}{s^2 + a_{1m}s + a_{om}} y_p(t) \right) e(t) \quad (11b)$$

It is important to note that the MIT rule by itself does not guarantee error convergence or stability [23,32-34], so the MRAC is combined with a tuned PV controller. An MRAC designed using the MIT rule is very sensitive to the amplitudes of the signals. As a general rule, the value of γ is kept small.

Tuning of γ is crucial to the adaptation rate and stability of the control system [23,32-34].

Simulations and Experiments

The MATLAB/Simulink program was developed for the experiments. The vertical speed control input constraints, $[-1, 1] \text{ ms}^{(-1)}$, are applied using the saturation block. The estimated time delay of 0.36s obtained in [21,22] was implemented using the transport delay block. The experiments were conducted in an office environment with the drone's indoor hull attached. The drone is connected to the host PC through WiFi (Figure 8). Data streaming, commanding and reading, are done using UDPs. UDP is a communication protocol that offers a limited amount of service when messages are exchange between computers in a network that uses IP.

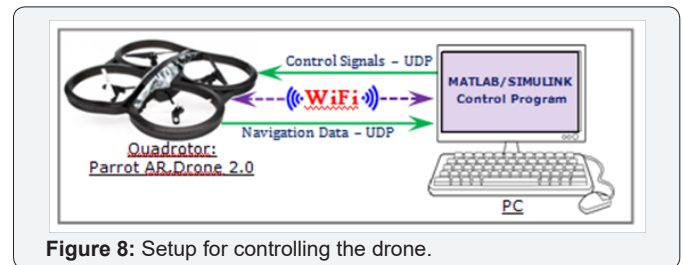


Figure 8: Setup for controlling the drone.

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 (at the bottom) for ground altitude measurement. It has 1GHz 32-bit ARM Cortex A8 processor, 1GB DDR2 RAM at 200MHz, and USB 2.0 high speed for extensions. The drone's total mass (including the indoor hull), m , is 453g.

Figure 9 shows the Simulink diagram developed for conducting the PV control system simulations. For the PV control, a high pass filter (HPF) with damping ratio, $\xi_f = 1.0$ was used for the derivative controller. A natural frequency value, $\omega_f = 38 \text{ rad/s}$, for the filter was selected by tuning and the use of the Bode plot. The filter's cut off frequency was determined as 0.90 Hz. Figure 10 shows the Simulink diagram developed to simulate a unit-step response for the PV-MRAC control system. The same HPF was applied in the control system.

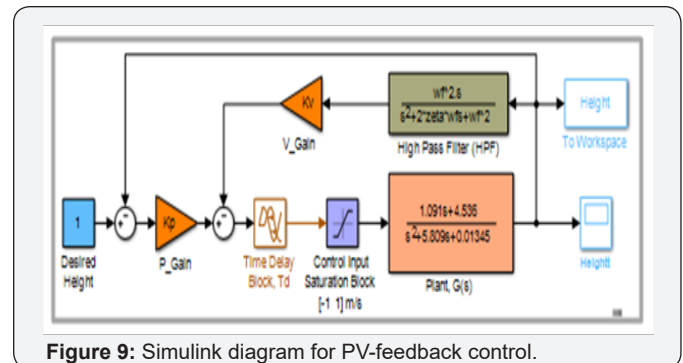


Figure 9: Simulink diagram for PV-feedback control.

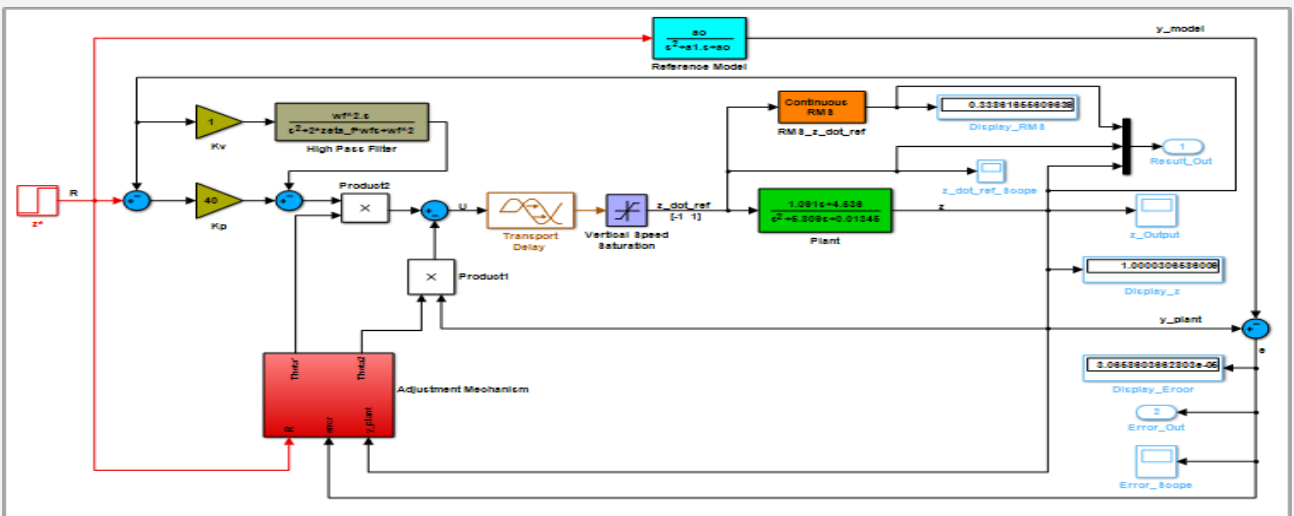


Figure 10: Simulink diagram for PV-MRAC control.

Flight payload: disturbance rejection



Figure 11: Mass attached at the top (Case 1).

To demonstrate the robustness of the PV-MRAC controller, it was tested against the PV controller using the payload capability of the drone. To accomplish this, the three cases below were considered.

Case 1: point-mass load: before flight: Masking tape was used to attach point-mass loads at the top of the drone before

Case 2: cable-suspended point-mass load: Here, a 100g point-mass load was attached to the end of an approximately 45cm length of rope, which was hooked to the drone before flight (Figure 12). Masking tape was used to hold firm the rope across the top and the bottom of the drone. In this second case,

flight (Figure 11). The controllers were tested using different masses. This setup was used to determine the stability bounds within which the changing mass-inertia parameters of the system due to the acquired payload will not destabilize the drone. The mass is assumed to be close to the drone's center of mass.

apart from the drone carrying an extra payload, the suspension system created an oscillating (pendulum) disturbances to the system. The rope is assumed to be massless and is attached to the drone's center of mass.

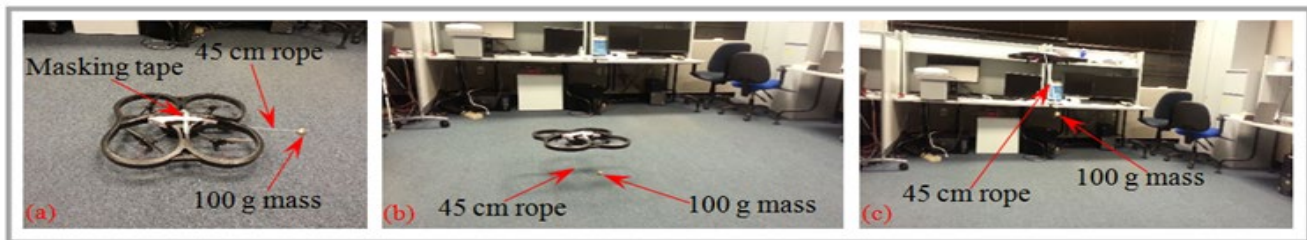


Figure 12: 100g mass hanging by rope (Case 2).

Case 3: point-mass load: during flight: In this setup, the designed controllers were used to demonstrate the stability behavior of the drone undergoing a range of instantaneous step

payload changes. Here, the masses were attached to a sticking masking tape at the top of the drone, after it has stabilized to the desired height, $\approx 1\text{m}$, see Figure13.

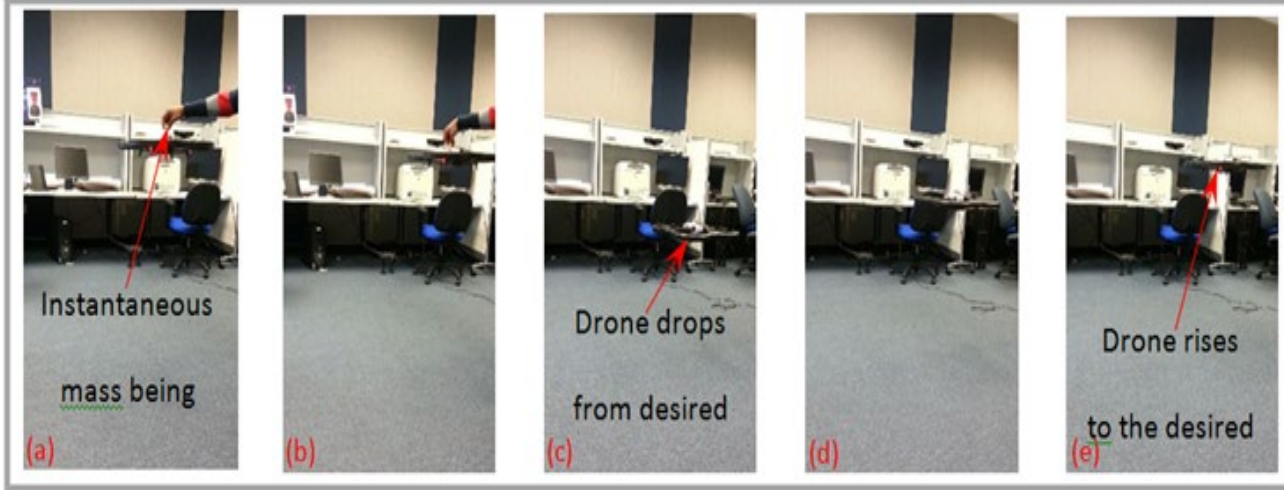


Figure 13: Instantaneous step mass (Case 3).

Results and Discussion

The drone response contains a 'dead-time' due 1) time delay in the control system and 2) the time between when the real-time application from MATLAB/Simulink is connected to the

drone and when the controller is switched on. Consequently, for ease of analyzing the experiments data, some of the drone's responses plots are shifted to start at (0s,0m). For example, see Figure 14 for the drone altitude responses using a P-control system.

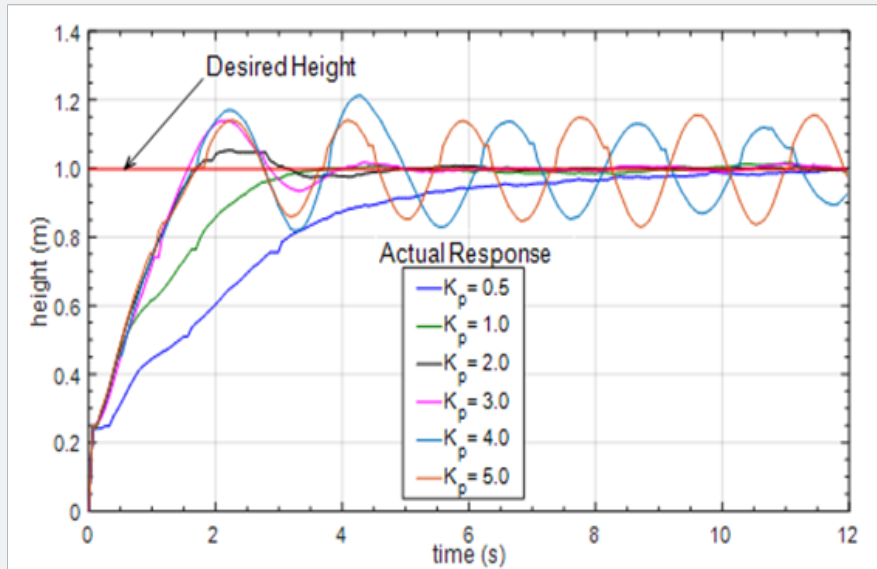


Figure 14: Drone altitude responses using P-control system.

Note from Figure 14 that, for a P-control system, if there is no delay ($\tau_d = 0$) in the system and $K_p > 0$, there should be no overshoot in the response. Also, increase in K_p should not destabilize the system. However, as seen in Figure 14, when $K_p > 1$, the output has a nonzero overshoot. Moreover, when $K_p \geq 5$, the amplitude of the response grows over time and the

system becomes unstable. This is partly due to the time delay in the system, which introduces nonlinearity on the dynamics. Therefore, the delay does have nontrivial effects on response and need be precisely estimated and considered in designing effective control.

Non-adaptive control: pv

The controller gains were obtained by tuning. Figure 14 & 15 show the simulation and experimental responses, respectively. It can be observed that as the K_v value increases M_o decreases, but the responses becomes slower. The PV controller with $K_v = 0.3$ and $K_p = 2.0$ gives the best stable control system performance, with rise time t_r (10%-90% range) of 1.30 s and $M_o = 2.03\%$. The results in Figure 16 and Table 1 compares simulated and experimented with and without the time-delay effects on the control system, with $K_v = 0.3$ and

$K_p = 2.0$. It can be observed from the rise time values the importance of incorporating the time delay into the design of the control system.

Table 1: PV control responses transient properties.

	Experiment	Simulation ($T_d = 0$)	Simulation ($T_d = 0.36s$)
M_o (%)	2.03	0	0
t_r (s)	1.37	1.91	1.34

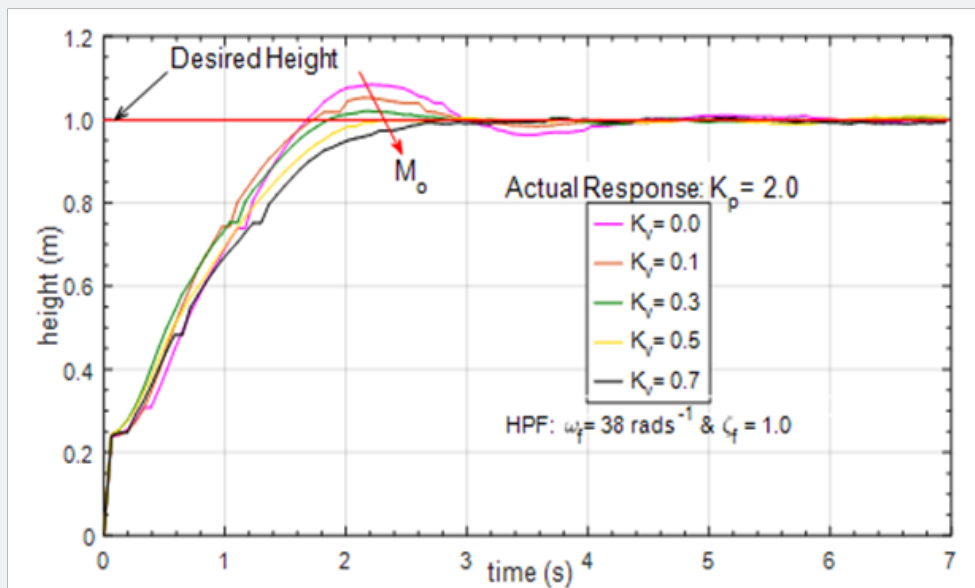


Figure 15: Experimented PV controller responses.

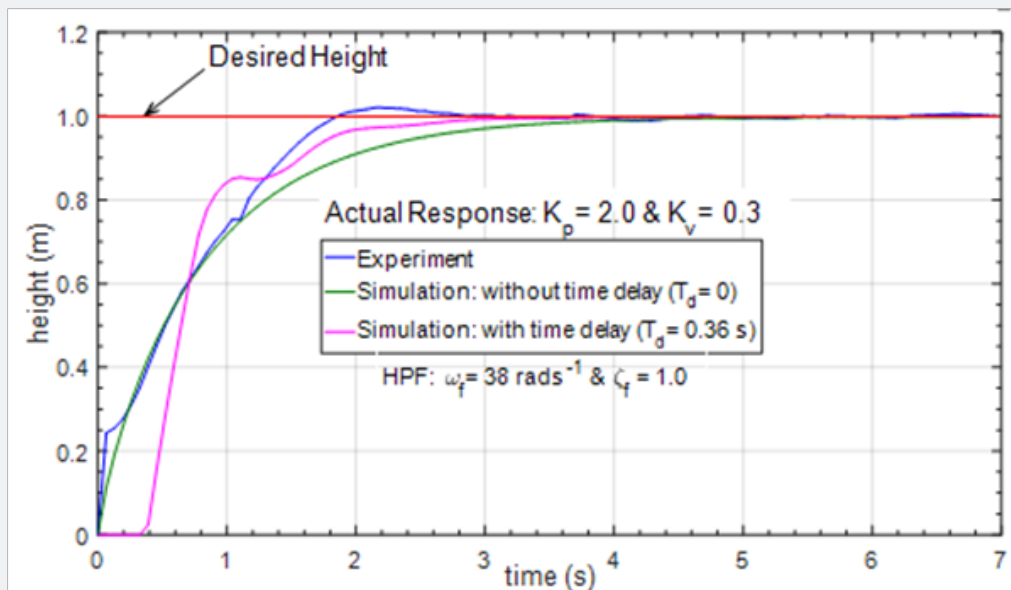


Figure 16: Simulated and experimented PV controller responses: delay effect.

MATLAB-based software package [35] was used to study the stability of the time-delay system, by solving the characteristic

equation, $1 + (K_p + K_v s)e^{-sT_d}G(s) = 0$, from (4). The closed-loop system characteristic roots within a specified region are then plotted

for various κ_v values with $\kappa_p = 2.0$. Figure 17 shows the spectrum distribution of the characteristic roots. When the rightmost (i.e., dominant) roots for each case are considered

$\kappa_v = 0.3$ the value yields the most stable rightmost root ($\Re(s) = -3.10$).

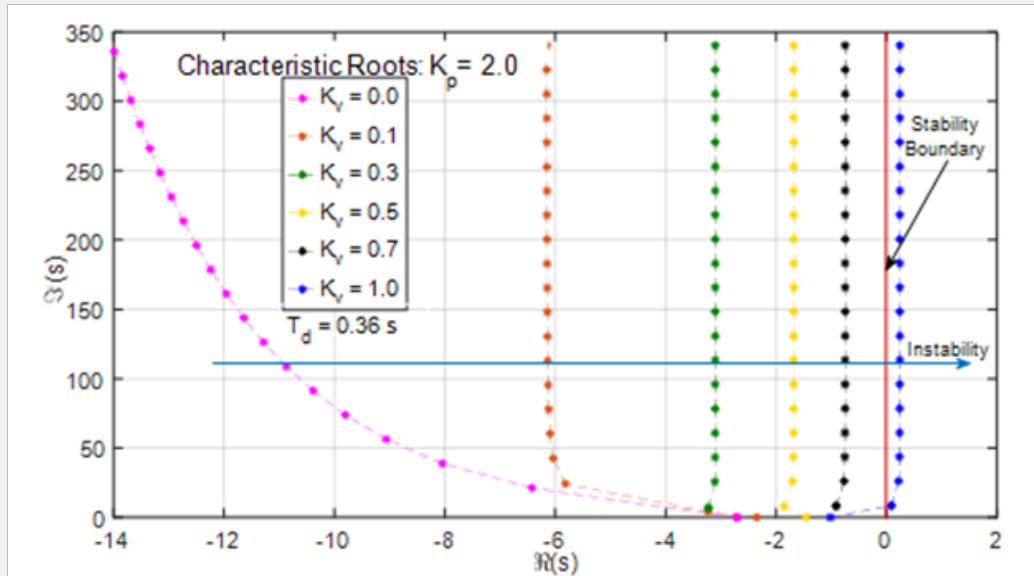


Figure 17: PV control system characteristic roots with $\kappa_p = 2.0$ and ($T_d = 0.36s$) effect of varying κ_v .

Adaptive control: PV-MRAC

Figure 18&19 show simulated and experimented responses, respectively, for different values. It can be seen that, increasing γ makes the response faster but it introduces oscillations, while decreasing γ makes the response slower leading to an unstable system. The results show that there is a range of values of γ to achieve stabilization, assuming the other tuning parameters remains constant. The results in Figure 20 and Table 2 compares simulated and experimented with and without the time-delay effects. Here again, the results show

the effect of including the time delay in the control systems design. Note that, using the PV controller and MRAC controller parameters on their own never stabilized the system.

Table 2: PV-MRAC responses transient properties.

	Experiment	Simulation ($T_d = 0$)	Simulation ($T_d = 0.36s$)
$M_o(\%)$	0.8	0	0
$t_r(s)$	1.39	2.23	1.34

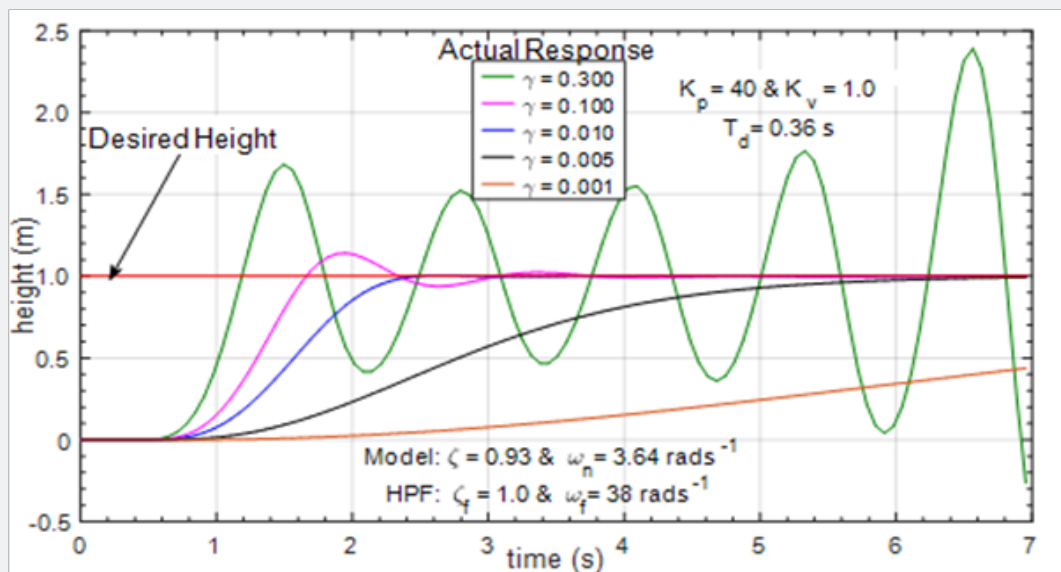


Figure 18: Simulated PV-MRAC responses, varying γ .

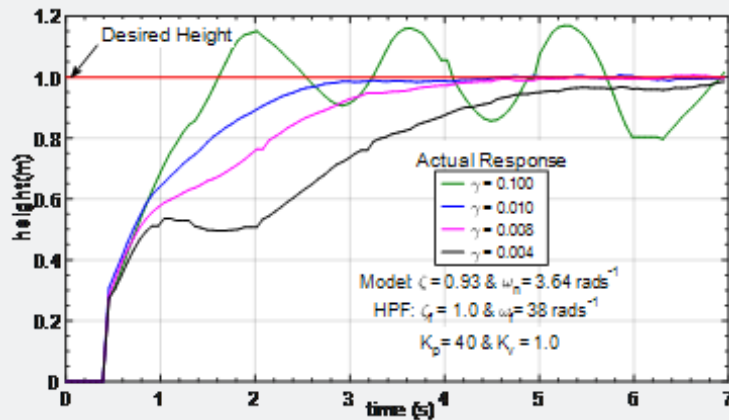


Figure 19: Experimented PV-MRAC responses, varying γ .

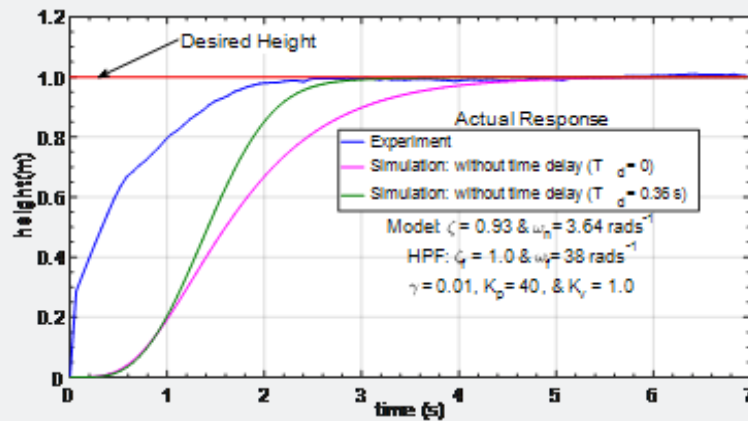


Figure 20: Simulated and experimented PV-MRAC controller responses: delay effect.

Flight payload: disturbance rejection

Figure 21-23 show the responses for the three cases considered in introducing disturbances to the drone, see Section

4.1. In general, the results confirm that the PV-MRAC offers several benefits over the fixed-gain approach, PV controller. The PV-MRAC was found to offer enhanced robustness to the change in mass uncertainty and the oscillating disturbances.

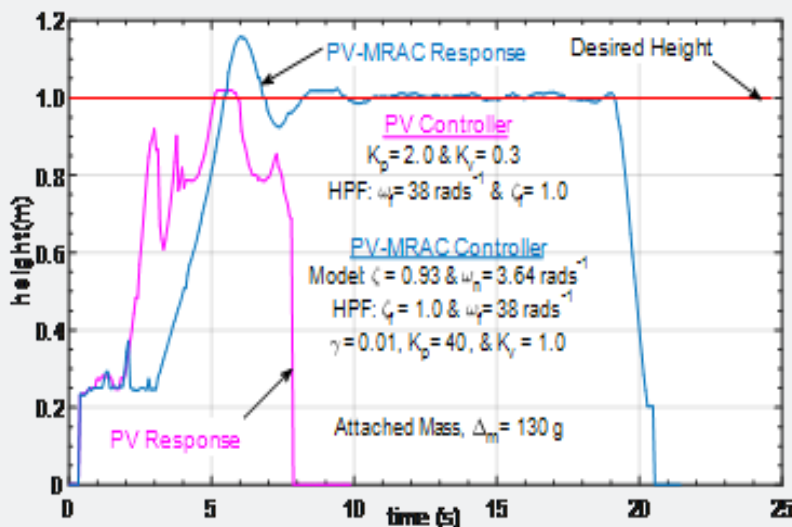


Figure 21: Experimented PV and PV-MRAC responses: 130g attached mass at the top.

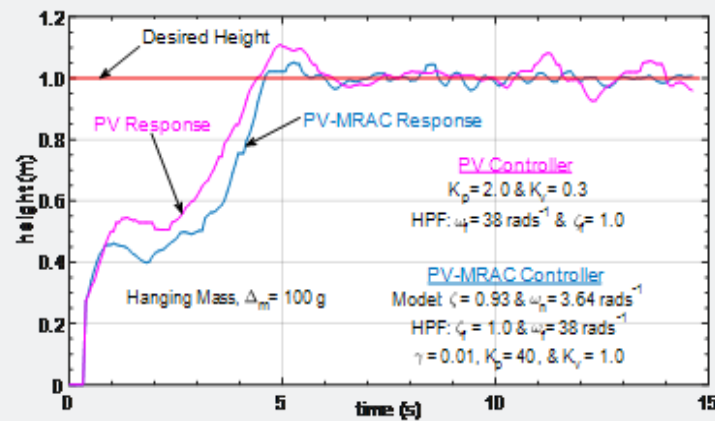


Figure 22: Experimented PV and PV-MRAC responses: 100g hanging mass.

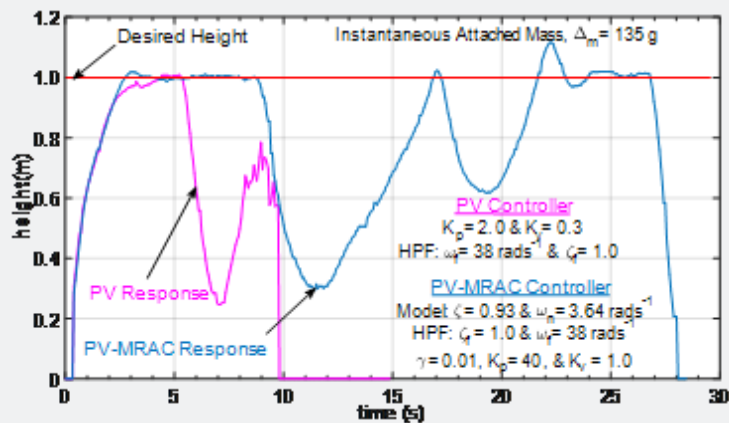


Figure 23: Experimented PV and PV-MRAC responses: 135g instantaneous mass.

Case 1: point-mass load: before flight

Figure 21 shows the altitude responses for the PV and PV-MRAC controllers for the case of attaching 130g point-mass load at the top of the drone. For the PV-MRAC, it was observed that beyond 130g the oscillations were larger, which destabilizes the drone. The drone's stability bounds reached at 130g for the PV-MRAC. For the PV control, it can be seen that with 130g attached, the drone was destabilized. The drone's stability bounds reached at 120g for the PV control. The difference in performance of the controllers was obvious, the adaptive controller allowed safe operation and landing, while the PV controller failed to prevent instability and sometimes resulted in a crash landing.

Case 2: cable-suspended point-mass load

The case of attaching 100g suspended point-mass load, which introduced oscillating disturbances, both controllers achieved the task of rising to the desired height, however, the PV-MRAC controller still performed better, see Figure 22. The PV-MRAC controller response has smaller oscillations about the desired height value compared to the PV controller response.

Case 3: point-mass load: during flight

Figure 23 show the responses for the PV and PV-MRAC controllers, respectively, for the case when instantaneous point-mass load of 135g was attached at the top of the drone. As it can be observed, the PV controller failed to prevent instability and resulted in a crash landing, but the adaptive controller maintained the stability and allowed safe operation and landing. The drone's stability bounds reached at 125g and 135g using the PV and PV-MRAC controllers, respectively.

Conclusion

This paper has presented a second order LTI SISO ARX altitude motion model, without time delay, for a quadrotor types of UAVs. The effect of time delay on the drone's altitude response was analysed including system stability conditions for the tuned PV control system. The non-adaptive PV controller was designed for improved transient responses based on the rightmost roots calculated using the estimated delay value. Also, the MRAC based on the MIT rule was designed by incorporating the time delay. The MRAC is combined with a tuned PV controller, since the MIT rule MRAC on its own cannot guarantee stability.

The robustness of the PV-MRAC controller was validated by comparing its performance to that of the non-adaptive PV controller, using the payload capability of the drone and the introduction of oscillating disturbances to the system. It was shown that the PV-MRAC offers several benefits over the fixed-gain approach of the PV controller. The PV-MRAC was found to offer enhanced robustness to the parametric (uncertainty in mass) and the oscillating disturbances. The difference in performance of the controllers was obvious. The adaptive controller allowed safe operation and landing while the PV controller failed to prevent instability and sometimes resulted in a crash landing. The drone's stability bounds reached at 120g and 130g for added attached mass-inertia using the PV and PV-MRAC controllers, respectively. In the case of the instantaneous introduction of the mass-inertia, the drone's stability bounds reached at 125g and 135g using the PV and PV-MRAC controllers, respectively.

Future work could involve comparing the performance of the MRAC based on the MIT rule to that based on the Lyapunov stability arguments, including composite adaptation. Composite MRAC adapts to both estimation and tracking errors. Also, multi-axis dynamics of the drone system considering attitude and lateral motions (x and y) can be considered. This problem is significantly more challenging, since the equation of motions are coupled and more complex than that of the altitude motion.

References

- Dydek ZT, Annaswamy AM, Lavretsky E (2013) Adaptive control of quadrotor UAVs: A design trade study with flight evaluations. *IEEE Transactions on control systems technology* 21(4): 1400-1406.
- Pounds PE, Bersak DR, Dollar AM (2012) Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control. *Autonomous Robots* 33(1-2): 129-142.
- Palunko I, Fierro R, Cruz P (2012) Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In *Robotics and Automation (ICRA), IEEE International Conference*, pp. 2691-2697.
- Bisgaard M, la Cour-Harbo A, Bendtsen JD (2010) Adaptive control system for autonomous helicopter slung load operations. *Control Engineering Practice* 18(7): 800-811.
- Vásárhelyi G, Virágh C, Somorjai G, Tarcai N, Szörényi T (2014) Outdoor flocking and formation flight with autonomous aerial robots. In *Intelligent Robots and Systems (IROS 2014), IEEE/RSJ International Conference*, pp. 3866-3873.
- Sa I, Corke P (2012) System identification, estimation and control for a cost effective open-source quadcopter. In *Robotics and automation (icra), 2012 IEEE international conference on IEEE*, pp. 2202-2209.
- Ailon A, Arogeti S (2014) Study on the effects of time-delays on quadrotor-type helicopter dynamics. In *Control and Automation (MED), 22nd Mediterranean Conference of*, pp. 305-310. IEEE.
- Azadegan M, Beheshti MT, Tavassoli B (2015) Using AQM for performance improvement of networked control systems. *International Journal of Control, Automation and Systems* 13(3): 764-772.
- Kchaou M, Tadeo F, Chaabane M, Toumi A (2014) Delay-dependent robust observer-based control for discrete-time uncertain singular systems with interval time-varying state delay. *International Journal of Control, Automation and Systems* 12(1): 12-22.
- Sagara S, Zhao ZY (1990) Numerical integration approach to on-line identification of continuous-time systems. *Automatica* 26(1): 63-74.
- Kurz H, Goedecke W (1981) Digital parameter-adaptive control of processes with unknown dead time. *Automatica* 17(1): 245-252.
- Wang QG and Zhang Y (2001) Robust identification of continuous systems with dead-time from step responses. *Automatica* 37(3): 377-390.
- Belkoura L, Richard JP, Fliess M (2009) Parameters estimation of systems with delayed and structured entries. *Automatica* 45(5): 1117-1125.
- Torkamani S, Butcher EA (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.
- He H, Yang T, Chen J (2015) On time delay estimation from a sparse linear prediction perspective. *The Journal of the Acoustical Society of America* 137(2): 1044-1047.
- Ahmed S, Huang B, Shah SL (2006) Parameter and delay estimation of continuous-time models using a linear filter. *Journal of Process Control* 16(4): 323-331.
- Saha DC, Rao GP (1983) Identification of continuous dynamical systems: the Poisson moment functional (PMF) approach Vol. 56.
- Pintelon R, Van Biesen L (1990) Identification of transfer functions with time delay and its application to cable fault location. *IEEE transactions on instrumentation and measurement* 39(3): 479-484.
- Yi S, Choi W, Abu-Lebdeh T (2012) Time-delay estimation using the characteristic roots of delay differential equations. *American Journal of Applied Sciences* 9(6): 955-960.
- Yi S, Nelson PW, Ulsoy AG (2010) Time-delay systems: analysis and control using the Lambert W function. *World Scientific*.
- Armah SK, Yi S, Choi W (2016) Design of feedback control for quadrotors considering signal transmission delays. *International Journal of Control, Automation and Systems* 14(6): 1395-1403.
- Armah SK, Yi S (2017) Analysis of time delays in quadrotor systems and design of control. In *Insperser T, Ersal T, Orosz G (Eds.), Time delay systems*.
- MIT rule MRAC, 2016.
- Beard R (2008) Quadrotor dynamics and control.

25. Armah S, Yi S, Choi W, Shin D (2016) Feedback control of quad-rotors with a Matlab-based simulator. *American Journal of Applied Sciences* 13(6): 779-793.
26. Corke P (2011) Robotics, vision and control: fundamental algorithms in MATLAB. 73: 78-86.
27. Mellinger D and Kumar V (2011) Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on IEEE, pp. 2520-2525.
28. Tanveer MH, Hazry D, Warsi FA, Joyo MK (2013) Stabilized controller design for attitude and altitude controlling of quad-rotor under disturbance and noisy conditions. *American Journal of Applied Sciences* 10(8): 819-831.
29. Armah SK (2015) Adaptive Control for Autonomous Navigation of Mobile Robots Considering Time Delay and Uncertainty (Doctoral dissertation, North Carolina Agricultural and Technical State University), pp. 79-81.
30. Kelly R, Davila VS, Loría A (2005) Proportional control plus velocity feedback and PD control. *Control of Robot Manipulators in Joint Space*, pp. 141-156.
31. Armah S, Yi S (2015) Altitude regulation of quadrotor types of UAVs considering communication delays. *IFAC-Papers OnLine* 48(12): 263-268.
32. Sastry S and Bodson M (2011) Adaptive control: stability, convergence and robustness. Courier Corporation.
33. Adrian C, Corneliu A, Mircea B (2008) The simulation of the adaptive systems using the MIT rule. In *International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering*, pp. 301-305.
34. Chamseddine A, Zhang Y, Rabbath CA, Apkarian J, Fulford C (2011) Model reference adaptive fault tolerant control of a quadrotor UAV. In *Infotech@ Aerospace 2011*, p. 1606.
35. Vyhlídal T (2013) Quasi-polynomial mapping based root finder.



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/RAEJ.2018.02.555598](https://doi.org/10.19080/RAEJ.2018.02.555598)

Your next submission with Juniper Publishers will reach you the below assets

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats
(Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>