

Adaptive Control of a Quadrotor UAV

Alex Paris

Abstract

This paper explains the implementation of a trajectory tracking adaptive controller applied to a quadrotor unmanned aerial vehicle (UAV). The controller is based on model reference adaptive control (MRAC) and augments a baseline, LQR-based controller. The adaptive controller is compared with the baseline in MATLAB simulations and found to be more robust to parametric uncertainties.

Index Terms

Adaptive control, quadrotor, drone, unmanned aerial vehicles.

I. INTRODUCTION

Quadrotor helicopters are becoming increasingly more popular for its variety of uses, such as aerial photography for topology and agriculture [1]–[4], search and rescue operations [5], [6], simultaneous localization and mapping (SLAM) [7], [8] and drone delivery [9], [10]. Nevertheless, these vehicles are inherently unstable, and many sources of uncertainty are present such as actuator degradation, external disturbances, and complex unmodeled dynamics (e.g. blade flapping, and asymmetric angular speed of propellers). These are amplified in the case of an actuator loss of effectiveness (LOE), which often results in a crash. In this context, this paper leverages adaptive control to provide the necessary robustness to quadrotor vehicles that is essential for the successful completion of complex missions. The approach explained is a subset of [11].

II. QUADROTOR DYNAMICS

A. Nonlinear Dynamics

Several authors have derived the dynamics of a quadrotor vehicle [12], [13]. Assuming a rigid body vehicle and low speeds, these dynamics are

$$\begin{aligned}\ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m} \\ \ddot{\phi} &= \dot{\theta} \dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_R}{I_x} \dot{\theta} \Omega_R + \frac{L}{I_x} U_2 \\ \ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{U_1}{m} \\ \ddot{\theta} &= \dot{\phi} \dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \frac{J_R}{I_y} \dot{\phi} \Omega_R + \frac{L}{I_y} U_3 \\ \ddot{z} &= -g + (\cos \phi \cos \theta) \frac{U_1}{m} \\ \ddot{\psi} &= \dot{\phi} \dot{\theta} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} U_4\end{aligned}\tag{1}$$

where x, y, z are the position coordinates of the center of mass in the inertial frame, ϕ, θ, ψ are the Euler angles (roll, pitch, and yaw), U_1 is the collective thrust, U_2 and U_3 are respectively the roll and pitch

A. Paris is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA
E-mail: aleix@mit.edu

forces, U_4 is the yaw torque, I_x, I_y, I_z are the inertia moments of the quadrotor body, m is the vehicle mass, J_R is the propeller inertia, Ω_R is the rotor speed, L is the distance from the center of mass to the quadrotor motors, and g is the gravity. Fig. 1 shows the body-fixed and inertial frames of the quadrotor, as well as the Euler angles, motor numbering and rotor spin direction.

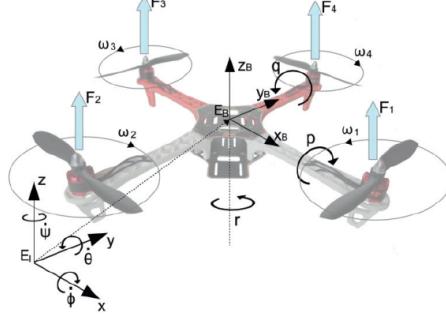


Fig. 1. Diagram of the quadrotor under study, from [14].

B. Linear Dynamics

Since quadrotors usually fly in regimes close to hover, a fair assumption is to consider the Euler angles small. By neglecting higher order terms and performing small-angle approximations, the dynamics in 1 can be simplified to

$$\begin{aligned} \ddot{x} &= g\theta, & \ddot{y} &= -g\phi, & \ddot{z} &= \frac{\Delta U_1}{m} \\ \ddot{\phi} &= \frac{L}{I_x}U_2, & \ddot{\theta} &= \frac{L}{I_y}U_3, & \ddot{\psi} &= \frac{1}{I_z}U_4 \end{aligned} \quad (2)$$

where $\Delta U_1 = U_1 - mg$. As expected, in 2 the x acceleration is affected by the pitch angle, the y acceleration by the roll angle, the collective thrust directly affects z and the roll, pitch and yaw control inputs U_2, U_3, U_4 cause accelerations in the roll, pitch and yaw angles. The equations in 2 are used to design the adaptive controller, but the simulated dynamics used in the experiments are still given by 1.

III. PROBLEM STATEMENT

The linear equations of motion along with uncertainties in the dynamics can be written as:

$$\dot{x}_p = A_p x_p + B_p \Lambda u \quad (3)$$

where $x_p \in \mathbb{R}^{n_p}$ is the vector of $n_p = 12$ states $x, y, z, \phi, \theta, \psi$ and their derivatives, $A_p \in \mathbb{R}^{n_p \times n_p}$ is constant and unknown (because of the uncertainties), $B_p \in \mathbb{R}^{n_p \times m}$ is constant and known, $\Lambda \in \mathbb{R}^{m \times m}$ is an unknown positive-definite matrix, and $u \in \mathbb{R}^m$ is the input vector $u = [\Delta U_1, U_2, U_3, U_4]^T$. The goal is then to track a reference command $r(t) \in \mathbb{R}^m$ in the presence of the unknown matrices A_p and Λ .

Let us define the system output as $y_p = C_p x_p$, $y_p \in \mathbb{R}^m$. As is usual for quadrotors, the output comprises the states x, y, z, ψ , and $m = 4$. The tracking error can then be defined as $e_y = y_p - r$.

The system in 3 can be augmented with the integrated output tracking error

$$e_{y_I} = \int e_y dt \quad \dot{e}_{y_I} = e_y \quad (4)$$

yielding the extended open-loop dynamics

$$\dot{x} = Ax + B\Lambda u + B_c r \quad (5)$$

where $x = [x_p^T \ e_{yI}^T]^T \in \mathbb{R}^n$ is the extended system state vector. The extended open-loop system matrices are given by

$$A = \begin{bmatrix} A_p & 0_{n_p \times m} \\ C_p & 0_{m \times m} \end{bmatrix}, \quad B = \begin{bmatrix} B_p \\ 0_{m \times m} \end{bmatrix}, \quad B_c = \begin{bmatrix} 0_{n_p \times m} \\ -I_{m \times m} \end{bmatrix} \quad (6)$$

and the output is $y = [C_p \ 0_{m \times m}]x = Cx$, where $A \in \mathbb{R}^{n \times n}$, $B, B_c \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{m \times n}$, and $n = n_p + m = 16$. Note that the first n_p rows of the matrices encode the dynamics while the last m rows encode Eq. 4.

IV. CONTROLLER DESIGN

A. Baseline Controller

In order to assess the improvement that an adaptive approach achieves to the problem presented in Section III, a baseline controller must be designed using classical control techniques. The baseline controller assumes there is no uncertainty, that is, $\Lambda = I$ and A is taken at a nominal value \bar{A} , where $A = \bar{A} + A^*$: \bar{A} encodes the linear dynamics in 2 and A^* contains the uncertain terms.

In this paper's case, an infinite-horizon, continuous-time linear-quadratic regulator (LQR) [15] was used, and the control law is

$$u_{bl} = K_{bl}x \quad (7)$$

B. Adaptive Controller

The adaptive controller uses the model reference adaptive controller (MRAC) approach. The model is based on 5 assuming no uncertainty and uses the baseline control input in 7

$$\dot{x}_m = \bar{A}x_m + Bu_{bl} + B_c r \quad (8)$$

where $x_m \in \mathbb{R}^n$ is the reference model state.

The adaptive control input is

$$u_{ad} = \hat{K}_x^T x + \hat{\theta}_r^T r + \hat{\theta}_d^T = \hat{\theta}^T \omega \quad (9)$$

where $\hat{\theta}^T(t) = [\hat{K}_x^T \ \hat{\theta}_r^T \ \hat{\theta}_d^T]$, $\hat{\theta} \in \mathbb{R}^{m \times p}$ is a matrix of adaptive parameters, where $p = m + n + 1 = 21$ and $\omega^T = [x^T \ r^T \ 1]$, $\omega \in \mathbb{R}^p$ is the regressor vector. The control input to the quadrotor plant is then given by

$$u = u_{ad} + u_{bl} = \hat{\theta}^T \omega + K_{bl}x \quad (10)$$

The classic adaptive law [16] is used

$$\dot{\hat{\theta}} = -\Gamma \omega e^T P B \quad (11)$$

where $\Gamma \in \mathbb{R}^{p \times p}$ is the diagonal positive-definite matrix containing the adaptive gains, $e = x - x_m$ is the model tracking error, and $P \in \mathbb{R}^{n \times n}$ is the unique symmetric positive-definite solution of the Lyapunov equation

$$A_m^T P + P A_m = -Q \quad (12)$$

where $A_m = A + B\Lambda K_x$ is the closed-loop reference model Hurwitz matrix, K_x can be obtained via LQR, and Q is any symmetric positive-definite matrix. In this paper, $Q = I$.

Note that in the nominal case, the control strategy is equivalent to the baseline controller, but in case of actuator failures or other types of parametric uncertainty, the adaptive input helps stabilize the system. The Lyapunov function candidate used to prove the stability of the overall system is the classical

$$V = e^T P e + \text{Tr} \left(\tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \right) \quad (13)$$

where $\tilde{\theta} = \hat{\theta} - \theta$ is the parameter estimation error. As shown in [16], this function implies that $\dot{V} = -e^T Q e \leq 0$. Therefore, Barbalat's lemma assures that the system is globally asymptotically stable and $\lim_{t \rightarrow \infty} e(t) = 0$.

Note that this proof applies to the system in 5. Nevertheless, near the hover regime, the nonlinear terms are dominated by the linear terms. These residual nonlinearities result in terms in \dot{V} that are linear in $\|e\|$, and instead of guaranteeing global asymptotic convergence, only boundedness of the error signal is guaranteed.

V. EXPERIMENTS

A. Quadrotor Parameters

The data used to model the quadrotor is based on the Parrot Mambo drone [17]. Most of the parameters were obtained from a script in the Parrot Minidrones Support from Simulink MATLAB package [18], and are shown in Table I.

TABLE I. Quadrotor parameters

Parameter	Symbol	Value	Unit
Mass	m	0.5	kg
Inertia around x-axis	I_x	0.03	kg m^2
Inertia around y-axis	I_y	0.03	kg m^2
Inertia around z-axis	I_z	0.16	kg m^2
Rotor inertia	J_r	2.042e-07	kg m^2
Distance from the center of mass to the rotors	L	0.0624	m
Rotational speed in the hover flight regime	Ω_R	764	rad/s
Thrust coefficient (Thrust = $c_T \cdot \Omega^2$)	c_T	2.1099e-6	$\text{N}/(\text{rad/s})^2$
Torque coefficient (Torque = $c_\tau \cdot \Omega^2$)	c_τ	2.6806e-8	$\text{N m}/(\text{rad/s})^2$
Maximum rotor speed	Ω^{max}	2054.92	rad/s

Experimental tests determined the hover rotor speed Ω_R in Eq. 1, and the thrust coefficient, torque coefficient, and maximum rotor speed (needed to calculate the maximum control inputs U_1, U_2, U_3 , and U_4), in which the RCbenchmark Series 1580/1585 Thrust Stand and Dynamometer [19] was used. Due to the lack of a Parrot Mambo, a DJI Snail Racing Propulsion System motor and propeller was used for this test. Different electronic speed controller (ESC) signals were sent to the motor, and the produced thrust and torque was recorded. Fig. 2 presents the experimental setup.

The results obtained by the thrust stand for different values of rotational speed Ω can be seen in Figs. 3 and 4. The slopes of these curves are, respectively, the thrust and torque coefficients.

B. Simulation Model

The system simulations and the baseline and adaptive controller implementation was carried out in MATLAB Simulink [20]. Fig. 5 shows the diagram of the simulation. The central block ‘compute_dynamics’ applies the nonlinear equations in 1 and Eq. 4 to the states and the input, producing the derivatives of the states. The second input to this block is a signal that comes from a switch to the left that chooses between u_{bl} and $u = u_{ad} + u_{bl}$. The implementation of the reference model is below the ‘compute_dynamics’ block, and its output x_m is subtracted from x to calculate the error and implement the adaptive law in 11 (right-most blocks). Finally, in the bottom-left part, the reference signal r is generated and sent to the ‘compute_dynamics’ block and to the reference model.

Fig. 2. Thrust stand experimental setup.

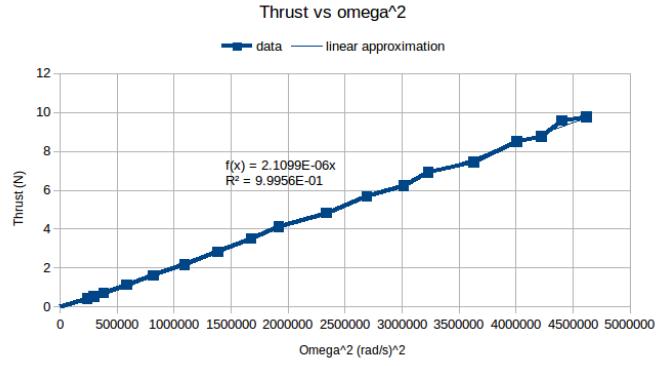
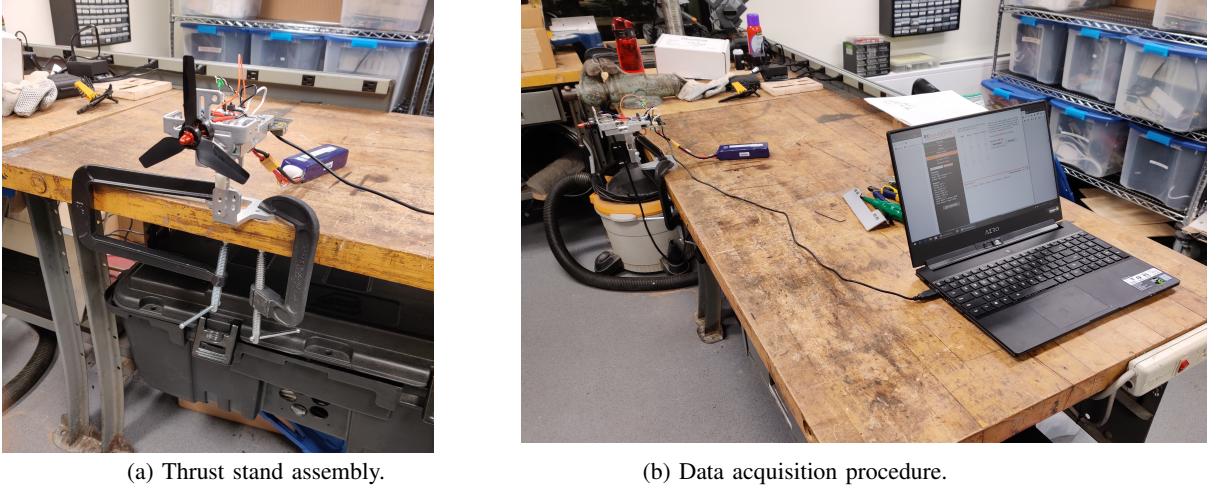


Fig. 3. Thrust produced by a DJI Snail Motor vs the squared rotation speed

Note that this simulation assumes that the states are directly accessible. Nevertheless, in a real quadrotor system, an online estimator would be required to obtain the values of the vehicle's states.

To simulate an actuator failure, a mapping from $\Omega = [\omega_1^2 \omega_2^2 \omega_3^2 \omega_4^2]^T$ to $U = [U_1 U_2 U_3 U_4]^T$ is needed. The matrix H provides this mapping (see Fig. 1):

$$\Omega = HU \quad \rightarrow \quad U = H^{-1}\Omega \quad (14)$$

$$U = c_T \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -L & 0 & L \\ -L & 0 & L & 0 \\ -\frac{c_T}{c_T} & \frac{c_T}{c_T} & -\frac{c_T}{c_T} & \frac{c_T}{c_T} \end{bmatrix} \Omega \quad (15)$$

Therefore, to encode a 60% LOE failure of motor number 1, the control input to be used for the simulation U_{fail} is given by

$$U_{fail} = H \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H^{-1}U \quad (16)$$

C. Results

To test the adaptive controller, the quadrotor was commanded to climb to an altitude of 5 m and to move 1 m in the x-direction at $t = 1$ s. Then, at $t = 12$ s, one of the motors suffered a 60% LOE failure. The videos of both experiments can be found in <https://bit.ly/2t6PMWT>.

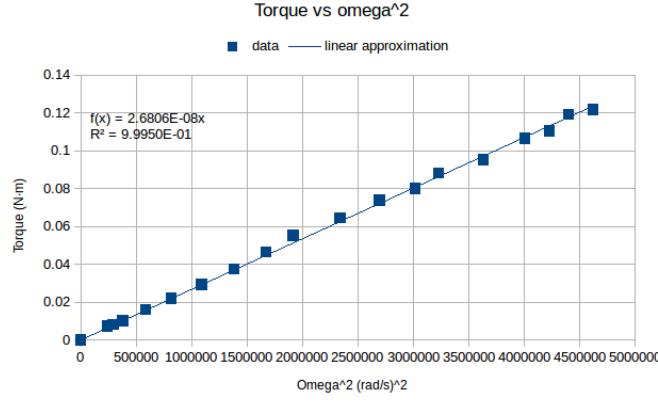


Fig. 4. Torque produced by a DJI Snail Motor vs the squared rotation speed

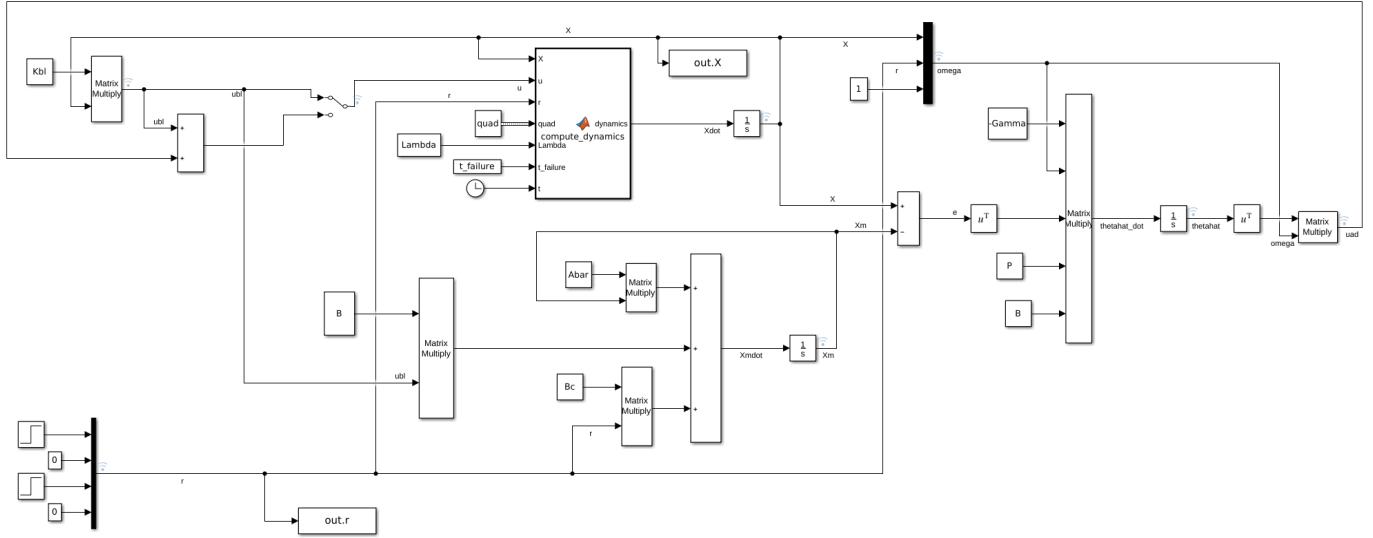


Fig. 5. Simulink model of the system under study.

1) *Baseline Controller:* The results obtained when using the baseline controller outlined in Eq. 7 are shown in Fig. 6. It can be observed that the system converges to the equilibrium point around $t = 8$ s with a slight overshoot, but after the failure it diverges from the setpoint, becoming unstable and crashing. That is, if the controller is not adaptive, the quadrotor is *not robust to parametric uncertainties*.

2) *Adaptive Controller:* In Fig. 7, the results for the same test but with the overall control input defined in 10 is shown. Similarly as in the previous case, the system converges at $t = 8$ s with a small overshoot. Nevertheless, after $t = 12$ s, in this case the system is almost not affected at all: only a very small oscillation in the pitch angle can be seen. Therefore, the adaptive controller is *robust to parametric uncertainties*.

VI. CONCLUSION

This paper has developed an adaptive controller that is capable of stabilizing a quadrotor even in the case of a 60% LOE failure, as shown in MATLAB simulations. This approach was compared with a baseline controller, which on the other hand was not capable of stabilizing the system. The simulation data is based on the Parrot Mambo minidrone and additional hardware experiments with a DJI Snail motor were conducted to determine parameters such as the thrust and torque coefficient.

Future work could implement this adaptive controller in hardware, which might require more advanced techniques such as the use of the projection operator.

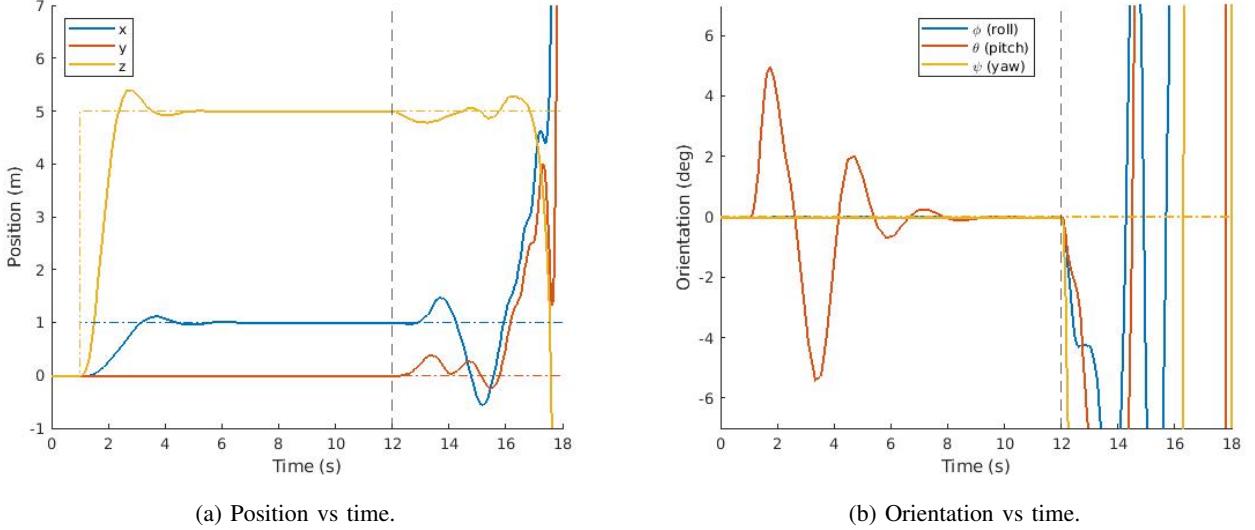


Fig. 6. Results for the experiment using the baseline controller. A 60% LOE failure occurs at $t = 12$ s and the system becomes unstable.

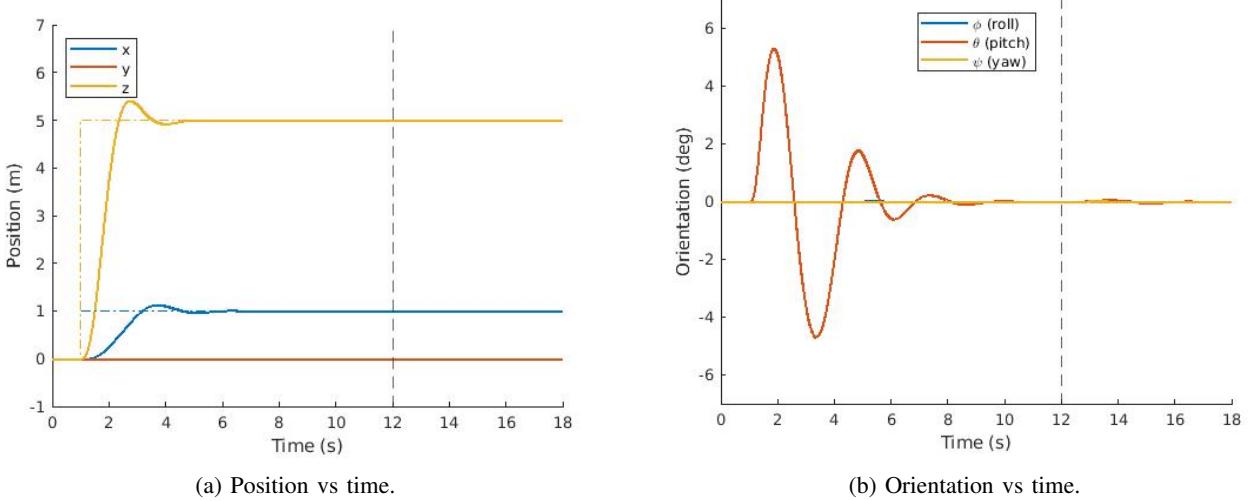


Fig. 7. Results for the experiment using the adaptive controller. A 60% LOE failure occurs at $t = 12$ s, and the system hardly notices it.

ACKNOWLEDGMENT

The author would like to thank Zachary T. Dydek for clarifying some aspects of his paper [11], which serves as the base for this work.

REFERENCES

- [1] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli, "Using unmanned aerial vehicles (UAV) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments," *Remote Sensing*, vol. 5, no. 12, pp. 6880–6898, 2013.
- [2] K. Anderson and K. J. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, no. 3, pp. 138–146, 2013.
- [3] E. Honkavaara, H. Saari, J. Kaivosoja, I. Pöölönen, T. Hakala, P. Litkey, J. Mäkinen, and L. Pesonen, "Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture," *Remote Sensing*, vol. 5, no. 10, pp. 5006–5039, 2013.
- [4] P. Tokek, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.
- [5] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kasseecker, E. Mair, I. L. Grix, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

- [6] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-UAV system for search and rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. ACM, 2015, pp. 33–38.
- [7] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and SLAM for medium and high altitude flying UAVs," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 137–161, 2009.
- [8] P. Schmuck and M. Chli, "Multi-UAV collaborative monocular SLAM," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3863–3870.
- [9] J. Scott and C. Scott, "Drone delivery models for healthcare," in *Proceedings of the 50th Hawaii international conference on system sciences*, 2017.
- [10] S. M. Shavarani, M. G. Nejad, F. Rismanchian, and G. Izbirak, "Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of amazon prime air in the city of san francisco," *The International Journal of Advanced Manufacturing Technology*, vol. 95, no. 9-12, pp. 3141–3153, 2018.
- [11] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Transactions on control systems technology*, vol. 21, no. 4, pp. 1400–1406, 2012.
- [12] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4393–4398.
- [13] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *AIAA guidance, navigation and control conference and exhibit*, 2007, p. 6461.
- [14] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, "Mathematical modelling and parameter identification of quadrotor (a survey)," *Procedia Engineering*, vol. 96, pp. 172–181, 2014.
- [15] P. O. Scokaert and J. B. Rawlings, "Infinite horizon linear quadratic control with constraints," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 5905–5910, 1996.
- [16] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*. Courier Corporation, 2012.
- [17] Parrot, "Parrot mambo drone," <https://www.parrot.com/us/drones/parrot-mambo-fly> [2019-11-10], 2019.
- [18] MATLAB, "Parrot minidrones support from simulink," <https://www.mathworks.com/hardware-support/parrot-minidrones.html> [2019-11-10], 2019.
- [19] RCbenchmark, "Rcbenchmark series 1580/1585 thrust stand and dynamometer," <https://www.rcbenchmark.com/pages/series-1580-thrust-stand-dynamometer> [2019-11-16], 2019.
- [20] The MathWorks, Inc., "Matlab simulink r2019b," <https://www.mathworks.com/help/simulink/index.html> [2019-11-08], 2019.