

# Experimental Implementation of an Adaptive Digital Autopilot

Ankit Goel, Juan Augusto Paredes, Harshil Dadhaniya, Syed Aseem Ul Islam, Abdulazeez Mohammed Salim, Sai Ravela, Dennis Bernstein

**Abstract**—This paper develops an adaptive digital autopilot for quadcopters and presents experimental results. The adaptive digital autopilot is constructed by augmenting the PX4 autopilot control system architecture with adaptive digital control laws based on retrospective cost adaptive control (RCAC). In order to investigate the performance of the adaptive digital autopilot, the default gains of the fixed-gain autopilot are scaled by a small factor, which severely degrades its performance. This scenario thus provides a venue for determining the ability of the adaptive digital autopilot to compensate for the detuned fixed-gain autopilot. The adaptive digital autopilot is tested in simulation and physical flight tests, and the resulting performance improvements are examined.

## I. INTRODUCTION

Multicopters are ubiquitous and are increasingly being used in a wide range of applications, such as sports broadcasting, wind-turbine inspection and agricultural monitoring [1]–[6]. In the most common quadcopter (multicopters with four propellers) configuration, the rotation directions and spin rates of the four motors provides thrust for translational motion as well as moments for attitude control. However, the nonlinear and unstable dynamics makes autonomous operation of quadcopters a very challenging problem. The control system of a multicopter is thus finely tuned and tailored to the geometry and mass properties of the vehicle. In fact, the open-source autopilots PX4 and ArduPilot contain finely-tuned controller gains for many commercially available unmanned aerial vehicle (UAV) configurations [7], [8]. In some applications, however, vehicle properties are subject to changes due to hardware alterations, such as airframe, payload, sensors, and actuators, and environmental conditions, such as wind speed and air density, which occur especially in experimental and field operations. In these cases, there is no guarantee that stock autopilot gains, tuned to perform well in a particular setting, will perform in an acceptable manner in off-nominal environments. Along the same lines, unanticipated and unknown changes that occur during flight due to failure or damage may significantly degrade the performance of the stock autopilot.

This research was supported in part by the Office of Naval Research under grant N00014-19-1-2273.

Ankit Goel, Juan Augusto Paredes, Harshil Dadhaniya, Syed Aseem Ul Islam, and Dennis Bernstein are with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109. [ankngoel](mailto:ankngoel@umich.edu), [jparedes](mailto:jparedes@umich.edu), [hdadhani](mailto:hdadhani@umich.edu), [aseemisl](mailto:aseemisl@umich.edu), [dsbaero@umich.edu](mailto:dsbaero@umich.edu)

Abdulazeez Mohammed Salim is with the Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139. [azez@mit.edu](mailto:azez@mit.edu)

Sai Ravela is with the Department of Earth, Atmospheric, and Planetary Sciences, MIT, Cambridge, MA 02139. [ravela@mit.edu](mailto:ravela@mit.edu)

With this motivation in mind, the present paper develops an adaptive digital autopilot for quadcopters with unknown dynamics. To do this, the PX4 autopilot architecture is modified so that the feedback and feedforward P and PID controllers are augmented with adaptive control laws based on retrospective cost adaptive control (RCAC) [9]. In particular, each controller in PX4 is augmented by an adaptive digital controller as described in [10]. The adaptive digital controller is based on recursive least squares (RLS), and thus involves the update of a matrix of size up to  $4 \times 4$  (3 PID gains + feedforward gain) at each time step, which makes it suitable for meeting the time constraints imposed by the real-time embedded systems used to implement the PX4 autopilot.

Learning algorithms have been previously implemented in UAV autopilots. In [11], the performance of model reference adaptive control (MRAC) was tested in a quadcopter under flight failure conditions, where a propeller was cut mid-flight. Previous knowledge of quadcopter dynamics was used to design the algorithm based on MRAC. In contrast, the adaptive algorithm proposed in this paper does not require any previous knowledge regarding the system dynamics. Fuzzy neural network based sliding mode control was used in [12] to control a fixed-wing aircraft flying under wind disturbances and, simultaneously, learn the inverse dynamics of the plant model. However, the autopilot required the gains from the proportional controllers to be appropriately initialized to allow sufficient time for learning. In this paper, in contrast, the adaptive digital autopilot controller coefficients are all initialized at zero and, thus, don't require any conditioning. Retrospective-cost based PID controllers were used in the attitude controller in [13], and were applied with fixed hyperparameter tuning to a quadcopter, a fixed-wing aircraft, and a VTOL aircraft in a simulation environment. In [14], retrospective-cost adaptive control law was used to compensate for the payload mass uncertainty. The present paper extends this work by augmenting both the attitude and position controllers in the PX4 autopilot with adaptive controllers and testing it in both simulation and experimental settings.

The contribution of the paper is the development of an adaptive digital autopilot and its experimental demonstration. The finely tuned stock autopilot is degraded by scaling down all fixed-gain controllers and the adaptive digital autopilot is then used to recover the stock autopilot's performance. The improvements are demonstrated through simulation and flight tests results.

The paper is organized as follows: Section II presents the

control system architecture implemented in the PX4 autopilot in detail. Section III describes the augmentation of the PX4 autopilot. Section IV presents simulation tests and flight test results of the stock PX4 and the adaptive PX4 autopilot. Finally, section V concludes the paper with a summary and future research directions.

## II. PX4 AUTOPILOT

In this section, the control system implemented in the stock PX4 autopilot is described. The control system consists of a mission planner and two nested loops as shown in Figure 1. The mission planner generates the position, velocity, azimuth, and azimuth rate setpoints from the user-defined waypoints. The notation used in this paper for quadcopter dynamics is described in [15]. The outer loop consists of the *position controller* whose inputs are the position setpoint  $\vec{r}_{c/w}|_E^{sp}$  and velocity setpoints  $\vec{r}_{c/w}|_E^{sp,ff}$  as well as the measured position  $\vec{r}_{c/w}|_E^{meas}$  and measured velocity  $\vec{r}_{c/w}|_E^{meas}$  of the quadcopter. The output of the position controller is the thrust vector setpoint  $\vec{f}_c|_E^{sp}$ . The inner loop consists of the *attitude controller* whose inputs are the thrust vector setpoint, the azimuth setpoint  $\Psi_{sp}$ , and azimuth rate setpoints  $\dot{\Psi}_{sp,ff}$ , as well as the measured attitude  $q_{Q/E}^{meas}$  and the angular velocity measured in the body-fixed frame  $\vec{\omega}_{Q/E}|_Q^{meas}$ . The output of the attitude controller is the moment setpoint  $\vec{M}_{Q/c}|_Q^{sp}$ . The magnitude of the thrust vector and the moment vector uniquely determine the rotation rate of the four propellers.

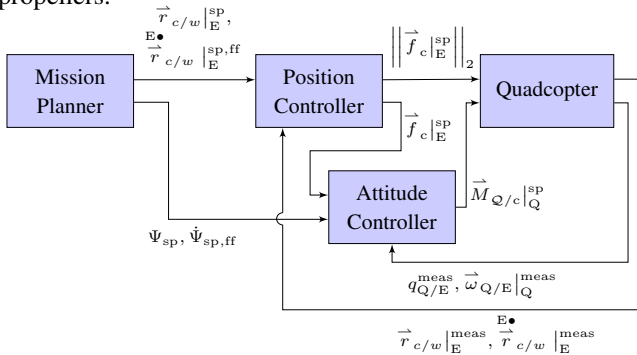


Fig. 1. PX4 autopilot architecture.

The position controller consists of two cascaded linear controllers as shown in Figure 2. The first controller  $G_r$  consists of a proportional controller and a feedforward controller. The velocity setpoint is thus given by

$$\vec{r}_{c/w}|_E^{sp} = K_r z_r + \vec{r}_{c/w}|_E^{sp,ff}, \quad (1)$$

where  $z_r \triangleq \vec{r}_{c/w}|_E^{sp} - \vec{r}_{c/w}|_E^{meas}$ ,  $K_r$  is a  $3 \times 3$  diagonal matrix, and  $\vec{r}_{c/w}|_E^{sp,ff}$  is the feedforward velocity setpoint specified by the mission planner. Note that diagonal entries of  $K_r$  are the tuning gains.

The second controller  $G_v$  consists of three decoupled PID controllers. The force setpoint  $\vec{f}_c|_E^{sp}$  is thus given by

$$\vec{f}_c|_E^{sp} = G_v z_v, \quad (2)$$

where  $z_v \triangleq \vec{r}_{c/w}|_E^{sp} - \vec{r}_{c/w}|_E^{meas}$ ,

$$G_v = K_{v,P} + \frac{K_{v,I}}{\mathbf{q}-1} + K_{v,D} \left(1 - \frac{1}{\mathbf{q}}\right), \quad (3)$$

and  $K_{v,P}$ ,  $K_{v,I}$ , and  $K_{v,D}$  are  $3 \times 3$  diagonal matrices; and  $\mathbf{q}$  is the forward-shift operator. Note that diagonal entries of  $K_{v,P}$ ,  $K_{v,I}$ , and  $K_{v,D}$  are the tuning gains.

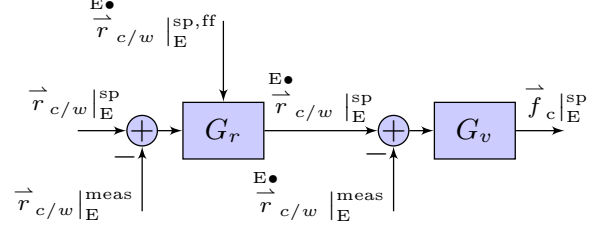


Fig. 2. PX4 autopilot position controller.

The Attitude controller consists of a static map  $f2q$  and two cascaded controllers  $G_q$  and  $G_\omega$  as shown in Figure 3. The static map  $f2q$  converts the force setpoint into the quaternion setpoint as described below.

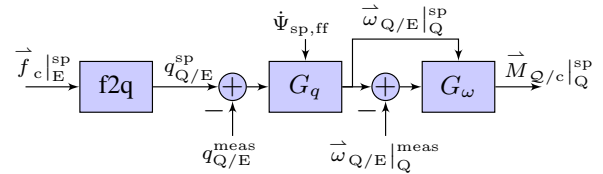


Fig. 3. PX4 autopilot attitude controller.

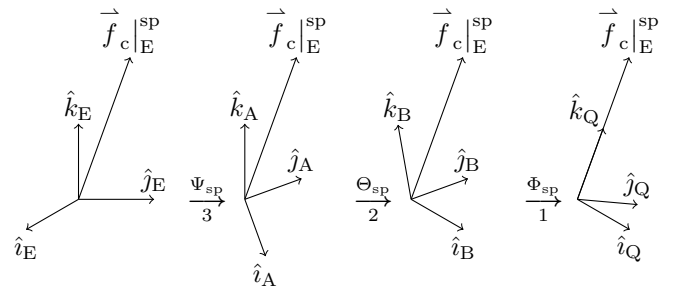


Fig. 4. 3-2-1 Euler angles that uniquely specify the attitude setpoint given the force setpoint and the azimuth setpoint.

The force setpoint  $\vec{f}_c|_E^{sp}$  and the azimuth setpoint  $\Psi_{sp}$  uniquely determine the attitude setpoint. Using the 3-2-1 Euler angle sequence shown in Figure 4 and the force setpoint  $\vec{f}_c|_E^{sp}$ , it follows that  $\hat{k}_Q|_E^{sp}$  satisfies

$$\hat{k}_Q|_E^{sp} = \frac{\vec{f}_c|_E^{sp}}{\|\vec{f}_c|_E^{sp}\|_2}. \quad (4)$$

Using the azimuth setpoint  $\Psi_{sp}$  specified by the mission planner and the rotation about  $k_E$ , it follows that

$$\hat{k}_Q|_A^{sp} = O_{A/E} \hat{k}_Q|_E^{sp} = O_3(\Psi_{sp}) \hat{k}_Q|_E^{sp}. \quad (5)$$

Since the frame  $F_B$  is obtained by rotating it about the  $\hat{j}_A$  axis so that  $\vec{f}_c|_E^{sp}$  lies in the  $\hat{j}_B - \hat{k}_B$  plane, it follows that

$$\tan \Theta_{sp} = \frac{e_1^T \hat{k}_Q|_A^{sp}}{e_3^T \hat{k}_Q|_A^{sp}}, \quad (6)$$

and thus  $\hat{k}_Q|_B^{sp} = \mathcal{O}_{B/A} \hat{k}_Q|_A^{sp} = \mathcal{O}_2(\Theta_{sp}) \hat{k}_Q|_E^{sp}$ . Finally, frame  $F_Q$  is obtained by rotating it about the  $\hat{i}_B$  axis so that  $\hat{k}_Q|_E^{sp}$  is along  $\vec{f}_c|_E^{sp}$ . It thus follows that

$$\tan \Phi_{sp} = \frac{e_2^T \hat{k}_Q|_B^{sp}}{e_3^T \hat{k}_Q|_B^{sp}}. \quad (7)$$

Next, the attitude setpoint given by the 3-2-1 Euler angles  $\Psi_{sp}$ ,  $\Theta_{sp}$ , and  $\Phi_{sp}$  is converted to the quaternion form. Using the measured attitude  $q_{Q/E}^{meas}$  and the setpoint attitude  $q_{Q/E}^{sp}$ , the attitude error  $\tilde{q}$  is given by

$$\tilde{q} \triangleq (q_{Q/E}^{meas})^{-1} q_{Q/E}^{sp} \quad (8)$$

Finally, writing the quaternion  $q = [\eta \ \varepsilon^T]$ , where  $\eta \in [-1, 1]$  and  $\varepsilon$  is a  $3 \times 1$  unit vector, the body-fixed angular velocity setpoint is given by

$$\vec{\omega}_{Q/E}|_Q^{sp} = \frac{2}{\tau} \text{sgn}(\hat{\eta}) \hat{\varepsilon}, \quad (9)$$

where  $\tau > 0$  is a tuning parameter. Note that the attitude controller (9) is an almost globally stabilizing controller [16]. However, the controller (9) may not provide good tracking since the azimuth response is usually slower than the elevation and bank responses due to the larger moment of inertia about the azimuth axis. Alternatively, a mixed attitude controller consisting of a reduced attitude error and a feedforward azimuth-rate controller can be used to generate the body-fixed angular velocity setpoint as described below.

The *reduced attitude error*  $\tilde{q}_{red}$  is defined as the rotation that aligns  $\hat{k}_Q$  with  $\hat{k}_Q^{sp}$  using the smallest angle of rotation, which is shown in Figure 5. The reduced attitude error is thus given by

$$\tilde{q}_{red} = \begin{bmatrix} \cos \frac{\alpha}{2} \\ \sin \frac{\alpha}{2} (\hat{k}_Q^{meas} \times \hat{k}_Q^{sp}) \end{bmatrix} \quad (10)$$

where  $\alpha \triangleq \arccos(\hat{k}_Q^{meas} \cdot \hat{k}_Q^{sp})$  and  $\hat{k}_Q^{meas} \times \hat{k}_Q^{sp}$  is the axis of rotation.

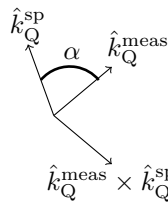


Fig. 5. Reduced attitude error. The reduced attitude is the smallest rotation that aligns  $\hat{k}_Q$  with  $\hat{k}_Q^{sp}$ .

Using the reduced attitude error (10), the body-fixed angular-velocity setpoint is given by

$$\vec{\omega}_{Q/E}|_Q^{sp} = K_q z_q + \dot{\Psi}_{sp,ff} \hat{k}_E|_Q = K_q z_q + \dot{\Psi}_{sp,ff} \mathcal{O}_{Q/E} e_3, \quad (11)$$

where  $z_q \triangleq \text{sgn}(\hat{\eta}_{red}) \hat{\varepsilon}_{red}$ ,  $K_q$  is a  $3 \times 3$  diagonal matrix and  $\dot{\Psi}_{sp,ff}$  is the feedforward azimuth rate setpoint specified by the mission planner. Note that the diagonal entries of  $K_q$  are the tuning gains.

Finally, the moment setpoint is given by

$$\vec{M}_{Q/c}|_Q^{sp} = G_\omega \begin{bmatrix} z_\omega \\ \vec{\omega}_{Q/E}|_Q^{sp} \end{bmatrix}, \quad (12)$$

where  $z_\omega \triangleq \vec{\omega}_{Q/E}|_Q^{sp} - \vec{\omega}_{Q/E}|_Q^{meas}$ ,

$$G_\omega = \begin{bmatrix} K_{\omega,P} + \frac{K_{\omega,I}}{\mathbf{q}-1} + K_{\omega,D} \frac{\mathbf{q}-1}{\mathbf{q}} & K_{\omega,ff} \end{bmatrix}, \quad (13)$$

and  $K_{\omega,P}$ ,  $K_{\omega,I}$ ,  $K_{\omega,D}$ , and  $K_{\omega,ff}$  are  $3 \times 3$  diagonal matrices. Note that diagonal entries of  $K_{\omega,P}$ ,  $K_{\omega,I}$ ,  $K_{\omega,D}$ , and  $K_{\omega,ff}$  are the tuning gains.

The control system implemented in the PX4 autopilot thus consists of 27 gains. In particular, the position controller includes three gains in  $G_r$  and nine gains in  $G_v$ ; and the attitude controller includes three gains in  $G_q$  and 12 gains in  $G_\omega$ . In practice, these 27 gains are manually tuned and require considerable expertise.

### III. ADAPTIVE PX4 AUTOPILOT

This section describes the augmentation of the PX4 autopilot with adaptive controllers. The adaptive digital autopilot is constructed by modifying the PX4 autopilot. The adaptive controllers are updated by the retrospective cost adaptive control algorithm described in [10], which is further specialized for the PX4 control architecture as shown in [15]. The controllers  $G_r$  and  $G_v$  in the position controller are augmented with adaptive components as shown in Figure 6. The velocity setpoint is given by

$$\vec{r}_{c/w}|_E^{sp} = K_r z_r + \vec{r}_{c/w}|_E^{sp,ff} + u_r, \quad (14)$$

where  $u_r = \phi_r \theta_r$  is the output of the adaptive  $G_r$ ,  $\phi_r \triangleq \text{diag}(z_r)$ , and  $\theta_r \in \mathbb{R}^3$  is updated by RCAC. Note that the structure of the adaptive  $G_r$  is same as that of  $G_r$ . The timestep  $k$  in the notation for  $u$ ,  $\phi$ , and  $\theta$  is omitted to improve readability.

Next, the force setpoint  $\vec{f}_c|_E^{sp}$  is given by

$$\vec{f}_c|_E^{sp} = G_v z_v + u_v, \quad (15)$$

where  $u_v = \phi_v \theta_v$  is the output of the adaptive  $G_v$ ,

$$\phi_v = \begin{bmatrix} \phi_{1,v} & 0 & 0 \\ 0 & \phi_{2,v} & 0 \\ 0 & 0 & \phi_{3,v} \end{bmatrix}, \quad (16)$$

$\theta_v \in \mathbb{R}^9$  is updated by RCAC, and, for  $i \in \{1, 2, 3\}$ ,

$$\phi_{i,v} \triangleq \begin{bmatrix} z_{i,v,k-1} & \gamma_{i,v,k-1} & z_{i,v,k-1} - z_{i,v,k-2} \end{bmatrix}, \quad (17)$$

where  $z_{i,v,k}$  is the  $i$ th component of the vector  $z_v$  at  $k$ .

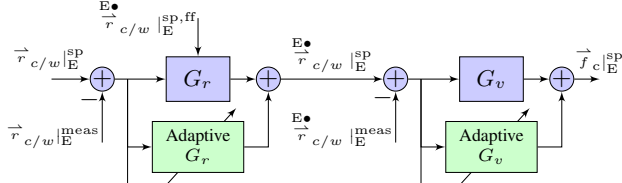


Fig. 6. Adaptive PX4 autopilot position controller.

The controllers  $G_q$  and  $G_\omega$  in the attitude controller are augmented with adaptive components as shown in Figure 7. The body-fixed angular velocity setpoint is given by

$$\vec{\omega}_{Q/E}|_Q^{\text{sp}} = K_q z_q + \dot{\Psi}_{\text{sp,ff}} \mathcal{O}_{Q/E} e_3 + u_q, \quad (18)$$

where  $u_q = \phi_q \theta_q$  is the output of the adaptive  $G_q$ ,  $\phi_q \triangleq \text{diag}(z_q)$ , and  $\theta_q \in \mathbb{R}^3$  is updated by RCAC.

Finally, the moment setpoint is given by

$$\vec{M}_{Q/c}|_Q^{\text{sp}} = G_\omega \begin{bmatrix} z_\omega \\ \vec{\omega}_{Q/E}|_Q^{\text{sp}} \end{bmatrix} + u_\omega, \quad (19)$$

where  $u_\omega = \phi_\omega \theta_\omega$  is the output of the adaptive  $G_\omega$ ,

$$\phi_\omega = \begin{bmatrix} \phi_{1,\omega} & 0 & 0 \\ 0 & \phi_{2,\omega} & 0 \\ 0 & 0 & \phi_{3,\omega} \end{bmatrix}, \quad (20)$$

$\theta_\omega \in \mathbb{R}^{12}$  is updated by RCAC, and, for  $i \in \{1, 2, 3\}$ ,

$$\phi_{i,\omega} \triangleq \begin{bmatrix} z_{i,\omega,k-1} \\ \gamma_{i,\omega,k-1} \\ z_{i,\omega,k-1} - z_{i,\omega,k-2} \\ e_i^T \vec{\omega}_{Q/E}|_Q^{\text{sp}} \end{bmatrix}^T. \quad (21)$$

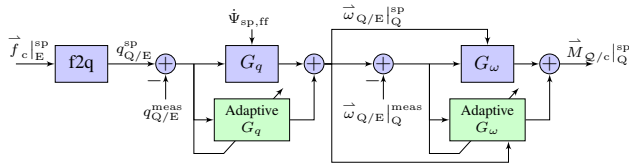


Fig. 7. Adaptive PX4 autopilot attitude controller.

Variable	Controller coefficient	$\sigma$	Simulation $P_0$	Flight test $P_0$
$u_r$	$\theta_r$	$I_3$	0.01	0.01
$u_v$	$\theta_v$	$I_9$	0.001	0.001
$u_q$	$\theta_q$	$I_3$	0.01	0.01
$u_\omega$	$\theta_\omega$	$I_{12}$	0.001	0.0001

TABLE I  
HYPERPARAMETERS USED BY RCAC IN THE ADAPTIVE PX4  
AUTOPILOT.

## IV. FLIGHT TESTS

This section describes the results of the flight tests conducted with the adaptive PX4 autopilot. First, the adaptive digital autopilot is tested with the SITL simulator. In this work, jMAVSim<sup>1</sup> is used to simulate the quadcopter dynamics. The default controller gains and the actuator constraints in PX4 are specified in the `mc_pos_control_param.c` and `mc_att_control_param.c`<sup>2</sup>. Note that the results in this paper are based on PX4 version V1.10.0dev. Table I shows the hyperparameters used by RCAC in the simulation environment and the flight tests.

The default controller gains in PX4 are well-tuned for the jMAVSim simulator. To investigate the potential improvements in the performance, the default controller gains in PX4 are multiplied by a scalar  $\alpha_p$  to degrade the performance of the autopilot. This is equivalent to the case of poor initial choice of controller gains. The baseline performance is obtained by setting  $\alpha_p = 1$  and the degraded autopilot performance by setting  $\alpha_p < 1$ . The solid blue trace in Figure 8 shows the trajectory-following response of the jMAVSim model in the baseline case. The solid red trace in Figure 8 shows the trajectory-following response in the case where  $\alpha_p = 0.3$ . Note the large overshoots. Next, the adaptive digital autopilot is used to fly the jMAVSim model. The dashed blue trace in Figure 8 shows the trajectory-following response in the case where  $\alpha_p = 1$ , and the dashed red trace in Figure 8 shows the trajectory-following response in the case where  $\alpha_p = 0.3$ . Note that with  $\alpha_p = 1$ , the performance is similar to the baseline case since the default controller is tuned well for the jMAVSim model. However, in the case of  $\alpha_p = 0.3$ , the adaptive digital autopilot recovers the baseline performance. Furthermore, as shown by the dashed red trace in Figure 8, the trajectory-following response after the second waypoint is similar to the baseline case.

The corresponding azimuth errors for the four cases are shown in Figure 9. Note that the mission takes about 65 s to complete compared to about 40 s in the baseline case. Note that, in the case where  $\alpha_p = 1$ , the adaptive digital autopilot marginally improves the performance, and in the case where  $\alpha_p = 0.3$ , the adaptive digital autopilot recovers the baseline performance. Finally, Figure 10 shows the controller gains optimized by RCAC in the adaptive digital autopilot. Note that the magnitude of the adaptive gains increase as  $\alpha_p$  is reduced. This suggests that RCAC compensates for the poor choice of gains in the fixed-gain controllers.

Next, the Holybro X500 quadcopter is flown in the M air facility at The University of Michigan, Ann Arbor. The default controller gains and the actuator constraints for the Holybro X500 quadcopter are specified in the QGroundControl mission planner. Table I shows the hyperparameters used by RCAC in the flight tests. It was observed that adaptive controller gain multiplying  $z_{1,\omega}$  diverged and destabilized the flight. Thus, in all subsequent flight test, it was set to

<sup>1</sup><https://github.com/PX4/jMAVSim>

<sup>2</sup><https://github.com/ankgoe18188/Firmware>

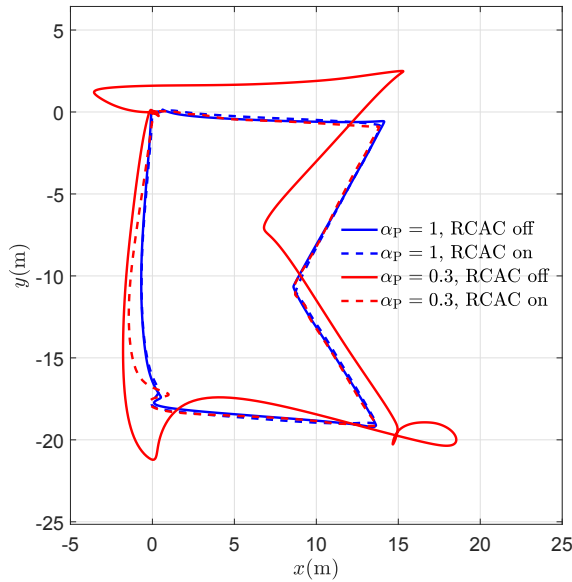


Fig. 8. Closed-loop trajectory-following response of the jMAVSIM model. The four traces show the trajectory-following response with  $\alpha_p = 1$  and  $\alpha_p = 0.3$  with the stock PX4 and the adaptive PX4 autopilot.

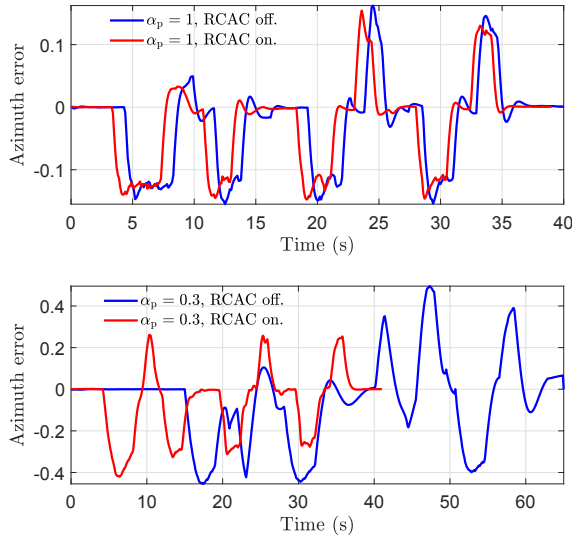


Fig. 9. Closed-loop azimuth response of the jMAVSIM model. The blue trace shows the azimuth error with the stock PX4 autopilot and the red trace shows the azimuth error with adaptive PX4 autopilot for two values of  $\alpha_p$ . Note that, in the case of  $\alpha_p = 1$ , the azimuth error is marginally better with RCAC, whereas in the case of  $\alpha_p = 0.3$ , note that RCAC optimizes the controller to improve the response and the mission is completed in the same time taken in the baseline case.

zero. The solid blue trace in Figure 11 shows the trajectory-following response of the Holybro X500 quadcopter in the baseline case. Next, the autopilot performance is degraded by setting  $\alpha_p = 0.5$  and  $\alpha_p = 0.3$ . The solid red trace in Figure 11 shows the trajectory-following response with  $\alpha_p = 0.5$ . However, with the stock PX4 autopilot, the quadcopter does not takeoff in the case where  $\alpha_p = 0.3$ . Next, the adaptive digital autopilot is used to fly the Holybro X500 quadcopter. The dashed blue, red, and the green traces in Figure 8 show the trajectory-following response with  $\alpha_p = 1$ ,  $\alpha_p = 0.5$ , and  $\alpha_p = 0.3$ , respectively.

The corresponding azimuth errors for the five cases are

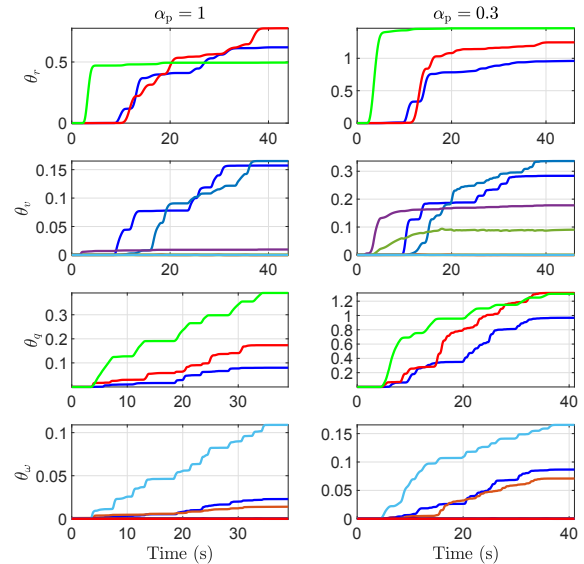


Fig. 10. Adaptive gains optimized by RCAC in the adaptive digital autopilot. Note that the magnitude of the adaptive gains increase as  $\alpha_p$  is reduced.

shown in Figure 12. Note that the mission takes about 70 s to complete with the degraded autopilot compared to about 55 s in the baseline case. Finally, Figure 13 shows the controller gains optimized by RCAC in the adaptive digital autopilot. Note that, in the case where  $\alpha_p = 1$ , the adaptive digital autopilot marginally improves the performance, and, in the case where  $\alpha_p < 1$ , the adaptive digital autopilot recovers the baseline performance.

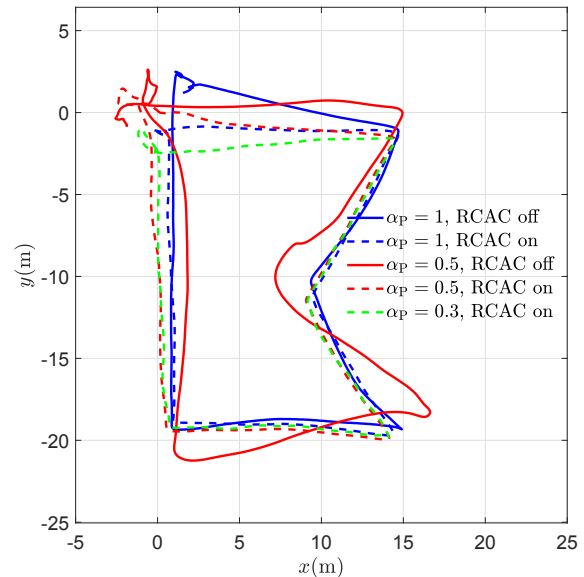


Fig. 11. Closed-loop trajectory-following response of the Holybro X500 quadcopter. The five traces show the baseline response, the degraded controller response, and the corresponding responses obtained with the adaptive PX4 autopilot.

## V. CONCLUSIONS AND FUTURE WORK

This paper presented an adaptive digital autopilot that can improve an initial poor choice of controller gains. The adaptive autopilot is constructed by augmenting the fixed-gain

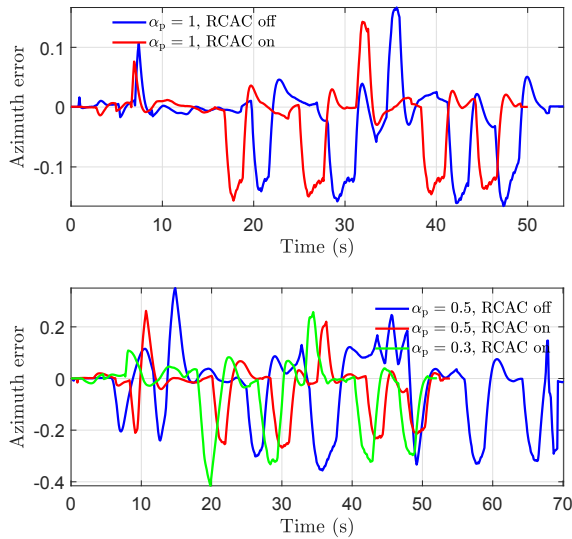


Fig. 12. Closed-loop azimuth response of the Holybro X500 quadcopter. The blue trace shows the azimuth error without RCAC; and the red and the green traces show the azimuth error with RCAC for two values of  $\alpha_p$ . Note that, in the case of  $\alpha_p = 1$ , the azimuth error is marginally better with RCAC. However, in the case of  $\alpha_p < 1$ , RCAC optimizes the controller to improve the trajectory-following and the mission is completed in the same time taken in the baseline case.

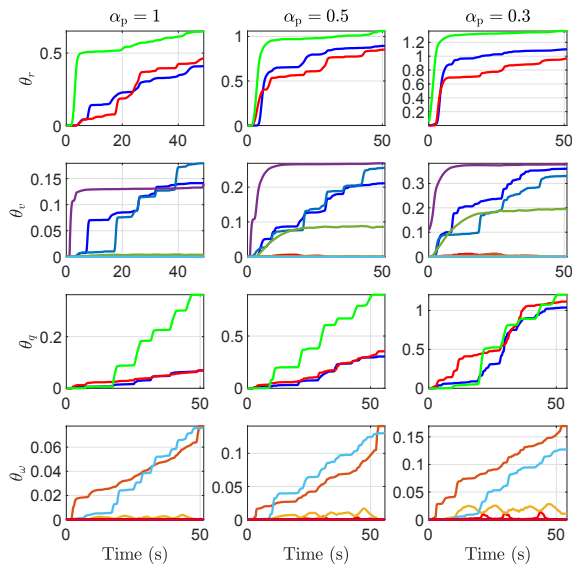


Fig. 13. Adaptive gains optimized by RCAC in the adaptive digital autopilot. Note that the magnitude of the gains increase as  $\alpha_p$  is reduced.

controllers in the stock autopilot with adaptive controllers. The adaptive autopilot was used to fly a quadcopter model in jMAVSIM simulator and the X500 Holybro quadcopter. The adaptive autopilot recovered the performance in the case where the default controllers were degraded both in simulations and physical flight tests. Future work will focus on systematically assessing the sensitivity of the quadcopter performance to the four controllers in the PX4 autopilot, targeting the augmentation to the most sensitive controller, and investigating the performance improvements using variable rate forgetting, integrator anti-windup, and IIR position and attitude controllers. Furthermore, the performance of the adaptive autopilot under unknown suspended payload and

chipped propellers, will be assessed.

## REFERENCES

- [1] C.-C. Chang, J.-L. Wang, C.-Y. Chang, M.-C. Liang, and M.-R. Lin, "Development of a multicopter-carried whole air sampling apparatus and its applications in environmental studies," *Chemosphere*, vol. 144, pp. 484–492, 2016.
- [2] S. Anweiler and D. Piwowarski, "Multicopter platform prototype for environmental monitoring," *Journal of Cleaner Production*, vol. 155, pp. 204–211, 2017.
- [3] V. H. Andaluz, E. López, D. Manobanda, F. Guamushig, F. Chicaiza, J. S. Sánchez, D. Rivas, F. Pérez, C. Sánchez, and V. Morales, "Nonlinear controller of quadcopters for agricultural monitoring," in *International Symposium on Visual Computing*, Springer, 2015, pp. 476–487.
- [4] B. E. Schäfer, D. Picchi, T. Engelhardt, and D. Abel, "Multicopter unmanned aerial vehicle for automated inspection of wind turbines," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 244–249.
- [5] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 998–1007.
- [6] J. A. Paredes, J. González, C. Saito, and A. Flores, "Multi-spectral imaging system with uav integration capabilities for crop analysis," in *2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE)*, 2017, pp. 1–4.
- [7] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE international conference on robotics and automation (ICRA)*, 2015, pp. 6235–6240.
- [8] ArduPilot Dev Team, *ArduPilot*, <https://ardupilot.org/ardupilot/>.
- [9] Y. Rahman, A. Xie, and D. S. Bernstein, "Retrospective Cost Adaptive Control: Pole Placement, Frequency Response, and Connections with LQG Control," *IEEE Control System Magazine*, vol. 37, pp. 28–69, Oct. 2017.
- [10] M. Kamaldar, S. A. U. Islam, S. Sanjeevini, A. Goel, J. B. Hoagg, and D. S. Bernstein, "Adaptive digital PID control of first-order-lag-plus-dead-time dynamics with sensor, actuator, and feedback nonlinearities," *Advanced Control for Applications*, vol. 1, no. 1, e20, 2019.
- [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, 2013.
- [12] E. Kayacan, M. A. Khamis, J. Rubio-Hervas, and M. Reyhanoglu, "Learning control of fixed-wing unmanned aerial vehicles using fuzzy neural networks," *International Journal of Aerospace Engineering*, vol. 2017, 2017.
- [13] A. A. Ansari, N. Zhang, and D. Bernstein, "Retrospective cost adaptive pid control of quadcopter/fixed-wing mode transition in a vtol aircraft," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018.
- [14] S. Dai, T. Lee, and D. S. Bernstein, "Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 6149–6154.
- [15] A. Goel, A. M. Salim, A. Ansari, S. Ravela, and D. Bernstein, "Adaptive digital pid control of a quadcopter with unknown dynamics," *arXiv preprint arXiv:2006.00416*, 2020.
- [16] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-Body Attitude Control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 30–51, 2011.