NA583: Adaptive Control of hexacopter rotor degradation

Prashin Sharma * and Arpita Petkar†
University of Michigan

This paper describes the application of model reference adaptive control to a hexacopter unmanned aerial vehicle platform. A baseline LQR with integrator trajectory tracking controller is assisted by an adaptive controller. The approach is implemented and validated using simulation.The adaptive controller is found to offer increased robustness to parametric uncertainties. In particular, it is found to be effective in mitigating the effects of a loss-of-thrust anomaly, which may occur due to rotor failure or physical damage to the propeller. The design of the adaptive controller is presented, followed by a comparison of simulation results using the baseline and adaptive controller.

Nomenclature

(Nomenclature entries should have the units identified)

C_T	=	Thrust Coefficient
C_Q	=	Torque Coefficient
C_x, C_y, C_z	=	Drag Coefficients
T	=	Thrust
τ_x, τ_y, τ_z	=	Torques along the roll, pitch, yaw axis
ϕ	=	Roll Angle
θ	=	Pitch Angle
ψ	=	yaw angle
m	=	mass of the hexacopter
g	=	acceleration due to gravity
p, q, r	=	angular velocities along x,y and z axis
γ	=	ration of moment coefficient to thrust coefficient

*PhD Candidate, Robotics
†MS, Electrical and Computer Engineering

I. Introduction and Problem Statement

Hexacopters have been an increasingly popular research platform in recent years. In designing a controller for these aircraft, there are several important considerations. There are numerous sources of uncertainty in the system some of which are actuator degradation, battery degradation, external disturbances,and potentially uncertain time delays in processing or communication. These problems are amplified in the case of actuator failures, where the aircraft has lost some of its control effectiveness. Adaptive control is an attractive candidate for this type of scenarios because of its ability to generate high performance tracking in the presence of parametric uncertainties.

In designing a controller for these aircraft's, there are several important vehicle-specific considerations. The dynamics of hexacopters are nonlinear and multivariate. There are also several effects to which a potential controller must be robust:

- aerodynamics of the rotor blades (propeller and blade flapping)
- inertial anti-torques (asymmetric angular speed of propellers)
- gyroscopic effects (change in orientation of the hexacopter and the plane of the propeller)

Most of the previously used approaches acknowledge the significant uncertainties present in the system and the potential for additional uncertainty due to failures, but none of the approaches explicitly accounts for these uncertainties in the controller design.

In this paper, we describe the application of a direct model reference adaptive approach that is based on Lyapunov stability arguments to improve tracking performance. We describe an adaptive controller which would deal specifically with rotor degradation failure scenario.

This paper is organized as follows. Section II describes the dynamics of the hexacopter. Using this model, the design of baseline controller (LQR with Integrator state) and direct model reference adaptive controller is described in Section III. The adaptive controller is then simulated for nominal i.e. no rotor degradation and off-nominal i.e. single rotor degradation scenarios. The tuning of parameters and plots of system response for both the scenarios are presented in Section IV. And finally the conclusions are presented in Section V.

II. Hexacopter Model

This section presents multicopter dynamics model relevant to a hexacopter. Each propulsion unit includes a motor, an electronic speed controller (ESC) and a propeller. All multicopter motors are connected to a single battery.

The aerodynamic forces generated by multicopter blades can be understood from lumped parameter model derived from momentum theory yielding simple relationships between thrust (T), torque (τ), power (P), and rotor speed (ω), for further details readers can refer to [1],[2]. These relationships are typically based on a hover assumption yielding a static thrust model.

The static free air model for thrust and torque are as follows :

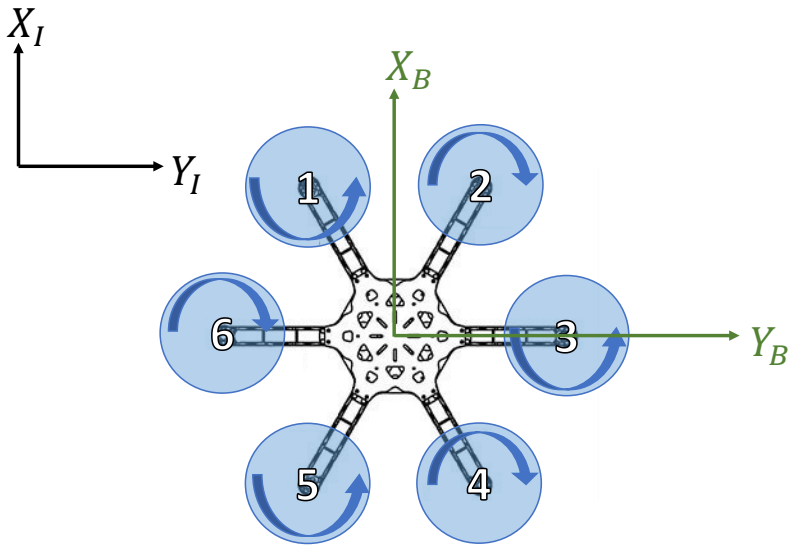


Fig. 1 Illustration of body and world hexacopter frame as seen from the top view

$$T = C_T \omega^2 \tag{1}$$

$$\tau = C_Q \omega^2 \tag{2}$$

where, C_T = thrust coefficient and C_Q = torque coefficient

To specify the dynamics of the system, two coordinate frames are defined: inertial frame denoted by $\{ X_I, Y_I, Z_I \}$ and body frame denoted by $\{ X_B, Y_B, Z_B \}$ as shown in Figure 1. We define roll angle ϕ about X_B, pitch angle θ about Y_B and yaw angle ψ about Z_B.

The rotors are numbered from 1-6 and are located at a distance L from the hexacopter center of gravity. The hexacopter's total motor thrust magnitude T and torque magnitude τ_i about each body axis are related to six motor commands as described by the following equations [3]:

$$\begin{pmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} C_T & C_T & C_T & C_T & C_T & C_T \\ \frac{LC_T}{2} & \frac{-LC_T}{2} & -LC_T & \frac{-LC_T}{2} & \frac{LC_T}{2} & LC_T \\ \frac{\sqrt{3}lC_T}{2} & \frac{\sqrt{3}lC_T}{2} & 0 & \frac{-\sqrt{3}LC_T}{2} & \frac{-\sqrt{3}lC_T}{2} & 0 \\ C_Q & -C_Q & C_Q & -C_Q & C_Q & -C_Q \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \\ \omega_5^2 \\ \omega_6^2 \end{pmatrix} \tag{3}$$

Let C_x,C_y and C_z represent hexacopter drag coefficients, m the mass of the hexacopter, and g acceleration due to gravity. Then, Newton's equations of motion for the hexacopter can be defined as

$$m\ddot{r} = R_B^W \begin{pmatrix} 0 \\ 0 \\ -T \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} + \begin{pmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \\ 0 & 0 & C_z \end{pmatrix} \dot{r} \tag{4}$$

where rotation matrix R_B^W is defined by Z-X-Y Euler angle rotation and is given as follows,

$$R_B^W = \begin{pmatrix} C\psi C\theta - S\phi S\psi S\theta & -C\phi S\psi & C\psi S\theta + C\theta S\phi S\psi \\ C\theta S\psi + C\psi S\phi S\theta & C\phi C\psi & S\psi S\theta - C\psi C\theta S\phi \\ -C\phi S\theta & S\phi & C\phi C\theta \end{pmatrix} \tag{5}$$

The attitude equations of motion for the hexacopter are defined as follows, where p is angular velocity about the x-axis, q is angular velocity about the y axis, and r is angular velocity about the z-axis,

$$I \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{L}{2} & \frac{-L}{2} & -L & \frac{-L}{2} & \frac{L}{2} & L \\ \frac{\sqrt{3}L}{2} & \frac{\sqrt{3}L}{2} & 0 & \frac{-\sqrt{3}L}{2} & \frac{-\sqrt{3}L}{2} & 0 \\ \gamma & -\gamma & \gamma & -\gamma & \gamma & -\gamma \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \end{pmatrix} - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times I \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{6}$$

where $\gamma = \frac{C_Q}{C_T}$, γ is the ratio of moment coefficient to thrust coefficient.

The relation between body and world frame angular velocity is given by,

amsmath

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) \\ sin(\theta)tan(\phi) & 1 & -cos(\theta)tan(\phi) \\ -\frac{sin(\theta)}{cos(\phi}) & 0 & -\frac{cos(\theta)}{cos(\phi)} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{7}$$

Since the hexacopter typically operates very near the hover position, we can make small angle approximations,

neglecting higher order terms and let $T = mg + \delta T$, with the resulting linear dynamics as follows :

$$\begin{aligned} \ddot{x} = g\theta, \quad \ddot{y} = -g\phi, \quad \ddot{z} = \frac{\Delta T}{m} \\ \ddot{\phi} = \frac{L}{I_x}\tau_x, \quad \ddot{\theta} = \frac{L}{I_y}\tau_y, \quad \ddot{\psi} = \frac{1}{I_z}\tau_z \end{aligned} \tag{8}$$

While simple, this model captures the dominant dynamics of the hexacopter, and is accurate near the hover position. As expected, the roll, pitch, and yaw inputs command moments about their respective axes and the thrutst input commands acceleration in the positive z-direction. Accelerations in the x- and y-directions are achieved primarily through vectoring the collective thrust. It can be seen that the dynamics of z, ϕ, θ, and ψ are double integrators, while the dynamics of x and y are quadruple integrators. The former group can be thought of as "fast" states, or as the vehicle dynamics while the latter group can be thought of as "slow" states of the vehicle kinematics.

III. Adaptive Controller Design

A. Plant Model

The equations of motion in (8) , along with the uncertainties can be written as

$$\dot{x_p} = A_p x_p + B_p \Lambda u \tag{9}$$

where $B_p \epsilon \mathbb{R}^{n_p \times m}$ is constant and *known*, $A_p \epsilon \mathbb{R}^{n_p \times n_p}$ is constant and *unknown*, $x_p \epsilon \mathbb{R}^{n_p}$, u $\epsilon \mathbb{R}^m$ and $\Lambda \epsilon \mathbb{R}^{m \times m}$ is an unknown positive definite matrix [4]. The state x_p thus consists of x, y, z, θ, ϕ, ψ, and their derivatives. The matrix A_p is the linearized state matrix representation of the non-linear dynamics. The goal is to track a reference command r(t) $\epsilon \mathbb{R}^m$ in the presence of the unknown A_p and Λ.

We define the system output $y_p \epsilon R^m$ as $y_p = C_p x_p$. In the case of the hexacopter, the output states are x, y, z, and ψ. The output tracking error is then given by e_y = y_p - r . Augmenting (9) with the integrated output tracking error $\dot{e_{yI}} = e_y$, leads to the extended open-loop dynamics

$$\dot{x} = Ax + B\Lambda u + B_c r \tag{10}$$

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

$$A = \begin{bmatrix} A_p & 0_{n_p \times m} \\ C_p & 0_{m \times m} \end{bmatrix}, B = \begin{bmatrix} B_p \\ 0_{m \times m} \end{bmatrix}, B_c = \begin{bmatrix} 0_{n_p \times m} \\ -I_{m \times m} \end{bmatrix} \tag{11}$$

and the extended system output $y = [C_p 0_{m \times m}]x = Cx$. Note that the dimension of the extended system state vector is

$n = n_p + m$, therefore $A \ \epsilon \ \mathbb{R}^{n\times n}, B, Bc \ \epsilon \ \mathbb{R}^{n\times m}, and \ \ C \ \epsilon \ \mathbb{R}^{m\times n}$.

B. Baseline Controller Design

A baseline controller can be designed for the system in (10) as :

$$u_{bl} = K_{bl}x \tag{12}$$

assuming there is no uncertainty, that is $\Lambda = I_{m\times m}$.The feedback gains are selected using LQR-I technique.To account for the uncertainties in the model we use the Model Reference Adaptive Control (MRAC) technique.The refernce model used by MRAC is the closed-loop system given by (10), in case of no uncertainty and along with the control input as defined in (12) the reference model can be written as :

$$\dot{x}_m = Ax_m + Bu_{bl} + B_c r \tag{13}$$

where x_m is the reference model state.

C. Adaptive Law

An adaptive input is added to the baseline controller as

$$u_{ad} = \hat{K}_x^T x + \hat{\theta}_r^T \tag{14}$$

where $\hat{\theta}^T = [\hat{K}_x^T \ \hat{\theta}_r \ ]$ is a matrix of time-varying adaptive parameters with dimensions m×p with p = m+n+1 and $\omega^T = [x^T \ r^T]$ is a regressor vector of dimension p. The adaptive parameters will be adjusted in the adaptive law given in (16) below. The overall control input is thus

$$u = u_{ad} + u_{bl} = \hat{\theta}^T\omega + \hat{K}_x^T x \tag{15}$$

The adaptive law is given as

$$\dot{\hat{\theta}} = -\Gamma\omega e^T PB \tag{16}$$

where $\Gamma \ \epsilon \ \mathbb{R}^{p\times p}$ is a diagonal, positive definite matrix of adaptive gains, $e = x - x_m$ is the model tracking error, and P ϵ $\mathbb{R}^{n\times n}$ is the unique symmetric positive definite solution of the Lyapunov equation, ${A_m}^T\mathrm{P} + \mathrm{P}A_m = -\mathrm{Q}$, where Q is also symmetric positive definite. This adaptive controller is based on nonlinear stability theory. The augmented structure of the adaptive controller implies that in the nominal case, that is the case with no parameter uncertainty, the overall system is equivalent to the baseline control. However, when failures or other uncertainties arise, the adaptive controller works to assist the baseline controller in maintaining stability and performance.

The proof of stability for the adaptive system uses a Lyapunov approach with the Lyapunov function candidate given by

$$V = e^T Pe + Tr(\tilde{\theta}^T \Gamma^{-1} \tilde{\theta}) \tag{17}$$

where $\tilde{\theta} = \hat{\theta} - \theta$ is the parameter estimation error. The value of $\tilde{\theta}$ will in general be unknown; however, it is not required by the control law, it is used only in the proof. It can be shown that the derivative of the Lyapunov function candidate is given by equation (18) .

$$\dot{V} = -e^T Qe \leq 0 \tag{18}$$

The system is globally asymptotically stable by Barbalat's lemma and the tracking error asymptotically converges to 0, that is $\lim_{e(t)\to\infty} = 0$

IV. Results

A. Simulation Model

Simulation model was developed in MATLAB R2019b Version Simulink 9.7.0 as shown in Fig. 2.The MATLAB code used for the generation of parameters and plant model is added in the appendix. The parameters used for simulation are detailed in Table 1. $Q_{lqr}, R_{lqr}, Q_{lyap}, \Gamma$ are the tuning parameters in our simulation. The matrix Λ as defined in (15) is used as the degradation matrix. The values of this matrix are chosen by setting thrust from one of the motors to zero.

Table 1 Parameter Values used in the simulation

Sr.No	Parameters	Values
1	Q_{lqr}	diag([30,30,30,10,10,10,10,10,10,10,10,10,10,1000,1000,1000,1000])
2	R_{lqr}	diag([100,100,100,100])
3	Q_{lyap}	diag([10,10,10,1,1,1,10,10,10,10,10,10,10,1000,1000,1000,1000])
4	Γ	diag([0.5,0.5,0.5,1,1,0.1,1,1,1,1,1,1,1,1,1,1,1,8000,8000,10000,10000)])
5	Λ	diag([0.85,0.80,0.80,0.90])

B. Simulation Results

For the simulations, x, y, z step reference signals were provided to the model. The response of the plant was analyzed with and without degradation of thrust and torque, with the adaptation ON and OFF in each of the scenarios. The results obtained from the simulations are presented in Figures 3-5.

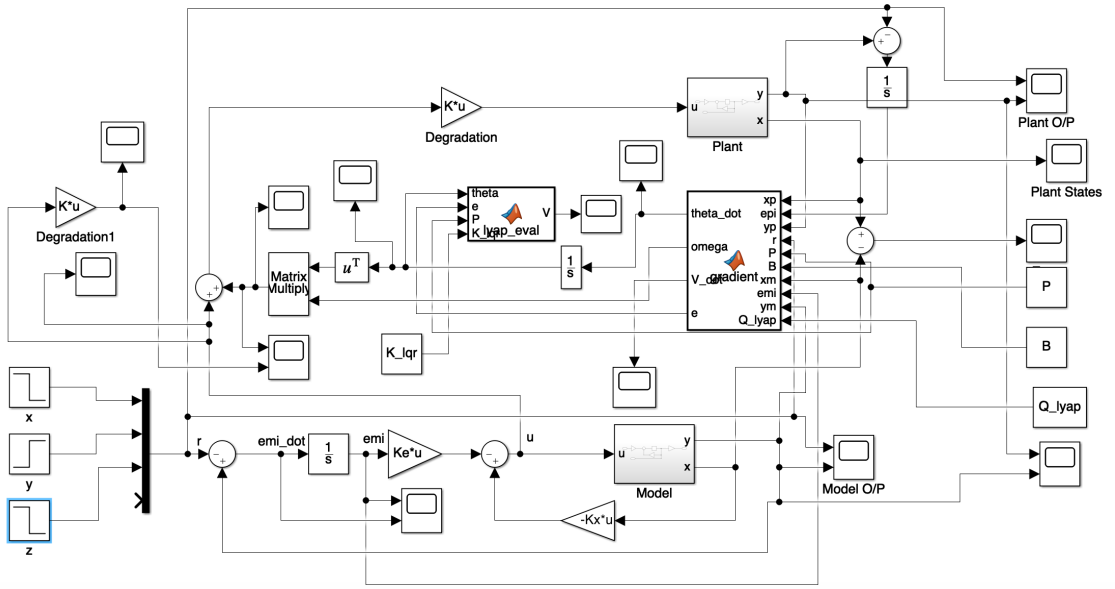
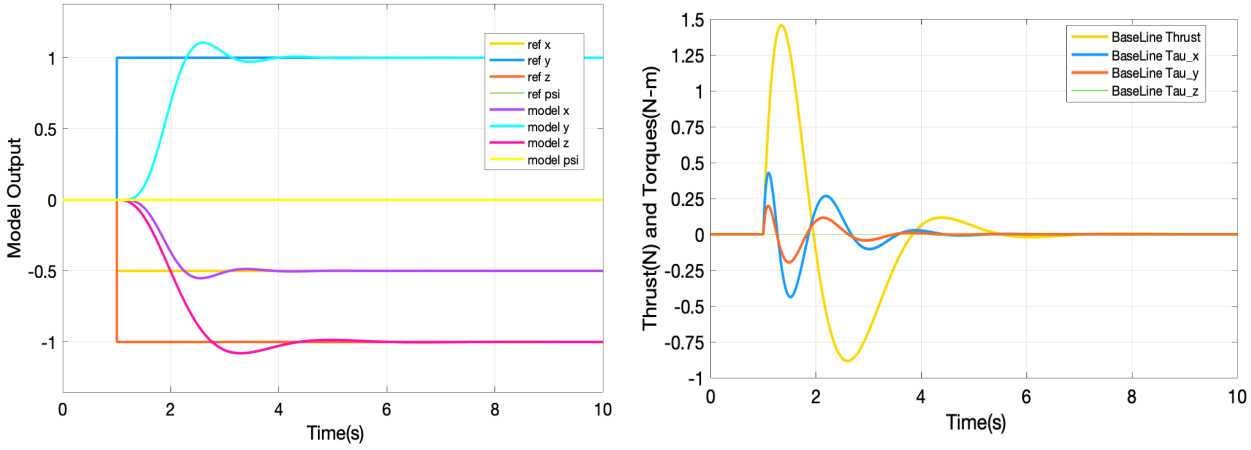


Fig. 2 Simulink model developed for simulations

1. Without rotor degradation

The tracking behaviour of the reference model of hexacopter is shown in Figure 3. From Figure 3a it can be seen that the hexacopter can track the reference signals with no steady state error. The control input required to achieve this tracking is shown in Figure 3b. The plant also exhibits similar behaviour for tracking reference signals, hence for brevity the plots are omitted.



(a) Model output

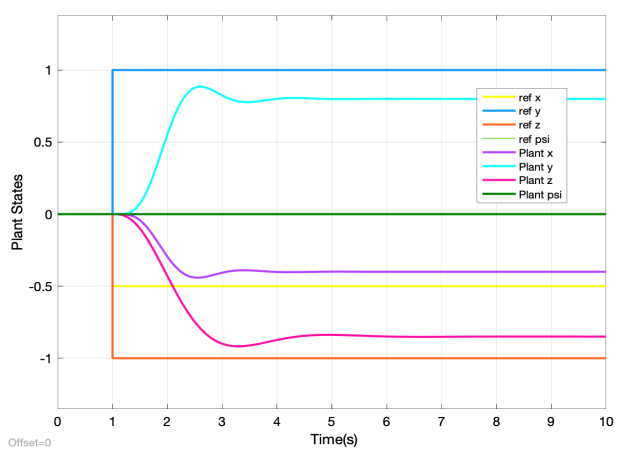
(b) Baseline thrust and torques

Fig. 3 Reference Hexacopter Model response to track reference command

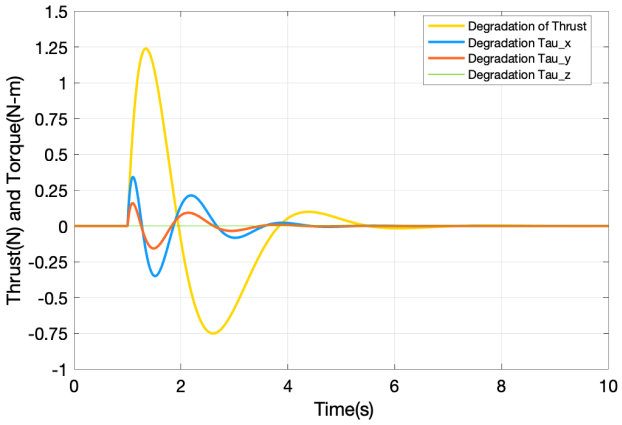
2. With rotor degradation

For simulating degradation of rotors, the control inputs obtained from baseline controller were multiplied with matrix Λ and applied to the plant.

The results when no adaptation takes place are plotted in Figure 4. It can be seen in Figure 4a that due to the degradation in the thrust and torque, there is a steady state error in tracking reference signals. Also the control command applied to plant reduces as a result of degradation, as can be seen by comparing Figure 4b and 3b.



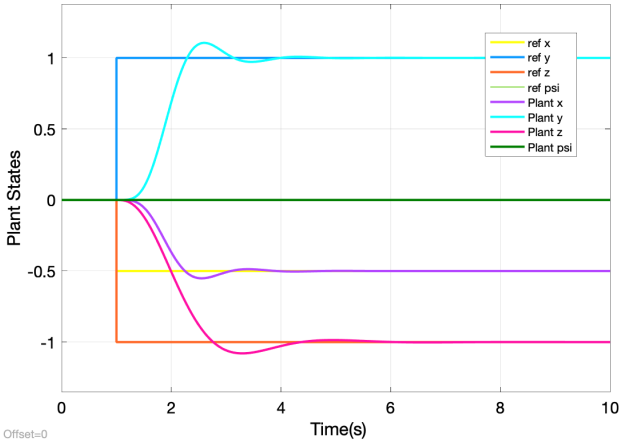
(a) Plant output with adaptive controller OFF



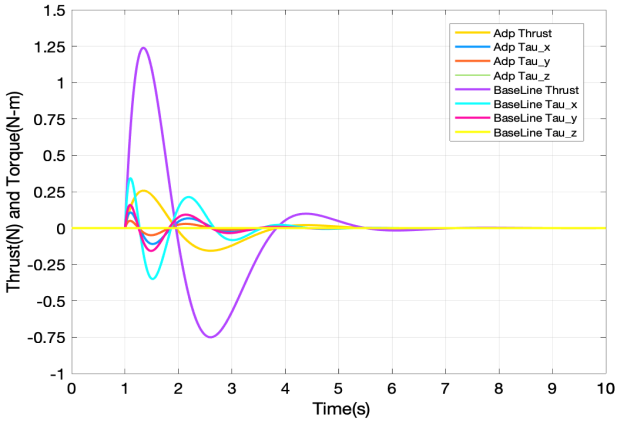
(b) Baseline thrust and torques degraded and applied to the plant

Fig. 4 Hexacopter Plant response when the forces are degraded and with no adaptation

The results of simulations when adaptation occurs in plotted in Figure 5. It can be seen from Figure 5a, there is no steady state tracking error. From Figure 5b its can be seen that in order to compensate the degraded control input, additional forces and torques are generated from adaptation and applied to baseline controller.



(a) Plant output with adaptive controller ON



(b) Baseline + adaptive thrust and torques applied to the plant

Fig. 5 Hexacopter Plant response when the forces are degraded and with adaptation

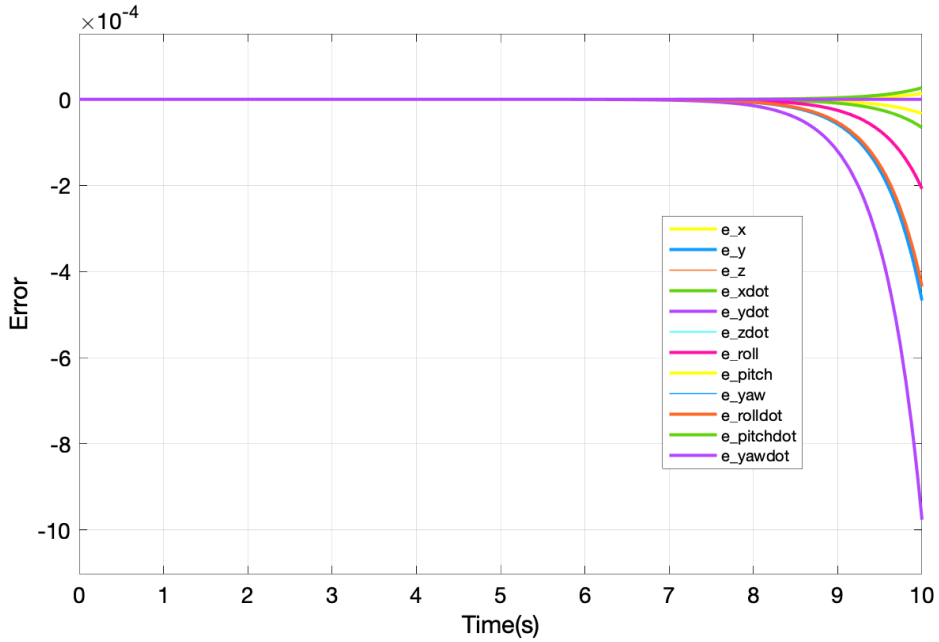
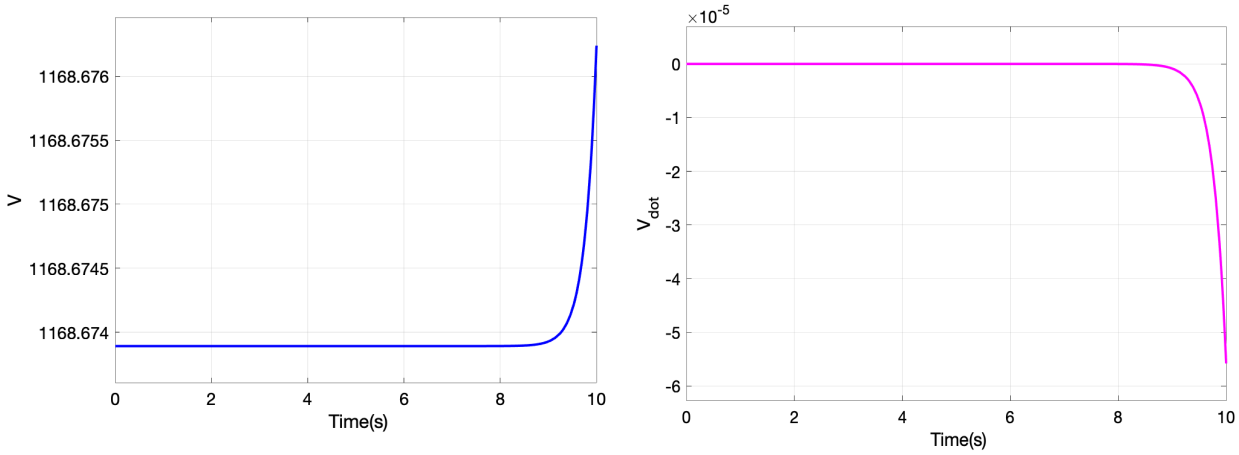


Fig. 6 Error of plant and mode states.

To further analyse the results from simulations $e, V, \dot{V}$ were also plotted as shown in Figures 6 and 7. Although from the plots as shown in Figure 5, the plant seems to converge to reference signal in presence of degradation, however, the error blows up as shown in Figure 6 towards the end of simulation. To further analyse this behaviour the lyapunov function from equation (17) and its derivative (18) are also plotted in Figure 7. At about nine seconds into the simulation $\dot{V}$ suddenly drops and V increases. This is not an ideal behaviour for the system and results in an anomoly which needs further investigation of the implementation.



(a) Progression of V with time

(b) Progression of $\dot{V}$ with time

Fig. 7 Analysis of Lyapunov function and its derivative

V. Conclusions

In this paper, a description of an adaptive controller based on Lyapunov stability and its application to a hexacopter simulation model was presented. From results of the simulation model it was shown that the adaptive controller was able to adapt to control input degradation and achieve zero steady state tracking error.

While analysing the adaptive controller it was also noticed that it destabilizes towards the end of the simulation stop time. Preliminary analyses of the anomaly was presented, however requires further in depth investigation.

For future work saturation of the control inputs and gradient method using projection could be used to bound the adaptive gains. Also, in the current simulations it is assumed that there is no noise in output measurements. It would be helpful to understand the system response in presence of measurement noise. Lastly, application of dead-zone for the error could be used in presence of noise so that the controller robustly adapts to any changes in parameters.

Acknowledgments

We would like to thank Prof. Jing Sun for her support and guidance throughout the project.

References

[1] Quan, Q., *Introduction to Multicopter Design and Control*, Springer Singapore, 2017, Chap. 4, pp. 73–95. https://doi.org/10.1007/978-981-10-3382-7.

[2] Mahony, R., Kumar, V., and Corke, P., "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics Automation Magazine*, Vol. 19, No. 3, 2012, pp. 20–32. https://doi.org/10.1109/MRA.2012.2206474.

[3] Lindblom, S., "Modelling and control of a hexarotor UAV," PhD dissertation, Linkoping University,Sweden, 2015.

[4] Dydek, Z. T., Annaswamy, A. M., and Lavretsky, E., "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Transactions on control systems technology*, Vol. 21, No. 4, 2012, pp. 1400–1406.

[5] Dydek, Z., "Adaptive control of unmanned aerial system," PhD dissertation, Massachusetts Institute of Technology, 2010.

[6] Petros A. Ioannou, J. S., *Robust Adaptive Control*, Prentice Hall, Englewood, Cliffs, NJ, 1996. https://doi.org/0-13-439100-4.

[7] Narendra, K. S., and Annaswamy, A. M., *Stable Adaptive Systems*, Prentice Hall, Englewood, Cliffs, NJ, 1989. https://doi.org/0-13-839994-8.

Appendix

Initialization of Parameter

```
%{
  Code replicating results from :Dydek, Zachary T.,Anuradha M. Annaswamy,
```

```
and Eugene Lavretsky. "Adaptive control of quadrotor UAVs: A design trade study
with flight evaluations."
IEEE Transactions on control systems technology 21.4 (2012): 1400-1406.

LQR- with Integrator state has been implemented
u = -[Kx Ke][x ;eyi]
%}

close all
clear all

g = 9.81;
m = 1;

Ixx = 0.0169;
Iyy = 0.0132;
Izz = 0.0327;

% State vectors are of the form
% X = [x y z xdot ydot zdot phi theta psi phidot thetadot psidot ex ey ex epsi]

Ap = [0 0 0 1  0 0 0 0 0 0 0 0 ;...
      0 0 0 0  1 0 0 0 0 0 0 0 ; ...
      0 0 0 0  0 1 0 0 0 0 0 0 ;...
      0 0 0 0  0 0 0 -9.81 0 0 0 0;...
      0 0 0 0  0 0 9.81 0 0 0 0 0; ...
      zeros(1,12) ; ...
      0 0 0 0  0 0 0 0 0 0 1 0 0;...
      0 0 0 0  0 0 0 0 0 0 0 1 0;...
      0 0 0 0  0 0 0 0 0 0 0 0 1;...
      zeros(3,12) ] ;
Bp = [zeros(5,4); [-1 0 0 0] ; zeros(3,4); ...
```

```
      [0 L/Ixx 0 0];[0 0 L/Iyy 0];[0 0 0 L/Izz] ];

% Output y= [x; y; z ;psi]
Cp = [[eye(3,3); zeros(1,3)] zeros(4,5) [0;0;0;1] zeros(4,3)] ;

A  = [ Ap  zeros(12,4) ;...
       Cp  zeros(4,4) ];

B  = [Bp; zeros(4,4) ] ;

Bc = [zeros(12,4) ; -eye(4)];

C  = [Cp zeros(4,4)];

Q = diag([30 30 30 10*ones(1,9) 10000*ones(1,4)]);
rho =100 ;
R = rho*eye(4);

[K_lqr,S,P] = lqr(A,B,Q,R);

Ke = K_lqr(:,13:16);
Kx = K_lqr(:,1:12);

% Closed loop matrix is Hurwitz
A_m = [Ap-Bp*Kx   -Bp*Ke ;...
         Cp         zeros(4,4) ];

Q_lyap = diag([10 10 10 1 1 1  10*ones(1,6) 1000*ones(1,4)]);

P =lyap(A_m, Q_lyap);

mix_matrix =[   -1          -1       -1       -1      -1       -1        ;...
```

```
    -0.225    -0.225    -0.45     0.225     0.225     0.45      ;...
     0.3897         0   -0.3897   -0.3897         0    0.3897    ;...
    -1             1        -1         1        -1         1    ];
cntrl_alloc = pinv(mix_matrix);

open_system('MRACpaper')
```

Gradient Descent Algorithm

```
function [theta_dot,omega,V_dot,e] = gradient(xp,epi,yp,r,P,B,xm, emi,ym,Q_lyap)

% x_ = [x y z xdot ydot zdot phi theta psi phidot thetadot psidot ex ey ex epsi]
x_ = [xp;(yp-r)];

% omega = [x y z xdot ydot zdot phi theta psi phidot thetadot psidot ex ey ex epsi rx ry rz
omega = [x_;r];
Tau = diag([ 0.5*ones(1,3) 1*ones(1,2) 0.1,...
            1*ones(1,3) 1*ones(1,3) 1*ones(1,4) 8000 8000 10000*ones(1,2) ]);
V_dot = 0;
% e = [ dx dy dz dxdot dydot dzdot dphi dtheta dpsi dphidot dthetadot dpsidot dex dey dez
e = [(xp-xm); (epi-emi)];

V_dot = -e'*Q_lyap*e ;
edz = 10^6*[0.0005  0.0005  0.0005  0  0  0  0.00009  0.00009  0.00009  0.00001  ...
             0.00001  0.00001  0.0001  0.0001  0.0001  0.00009 ]';

theta_dot = -(Tau*omega*e'*P*B)/(1+omega'*omega) ; % normalized value
```

Lyapunov Function Evaluation

```
function V = lyap_eval(theta , e,P,K_lqr)

V = 0 ;
Tau = diag([ 0.5*ones(1,3) 1*ones(1,2) 0.1,...
            1*ones(1,3) 1*ones(1,3) 1*ones(1,4) 8000 8000 10000*ones(1,2) ]);
```

```
theta_t = zeros(4,20);
theta_t = [K_lqr , zeros(4,4)];
V = e'*P*e + trace(theta_t*inv(Tau)*theta_t');
```