Adaptive Control of Quadrotor Susceptible to Disturbance and Rotor Failure

Brian Grimaldi Anthony Nguyen Pratik Ponnarassery

***Abstract*—In this document, we construct and test the robustness of a H_{inf} controller and PD controller with adaptive terms against rotor failure and wind disturbance respectively. We find that the H_{inf} controller met our expectations in the case of easily stabilizing our quadrotor in the case of a single rotor failure. For the PD controller with adaptive terms, we found that while it better stabilized the quadrotor in the presence of wind disturbance, the method of determining the adaption constant, γ, was inefficient, creating a long process that generally resulted in a failed stabilization. These results highlight the strengths and weaknesses of these controllers, allowing for further research in creating a robust controller to stabilize a quadrotor in presence of rotor failure and wind disturbance while following a flight trajectory.**

I. Introduction

Quadrotor research is an exceedingly popular field of research that has become a interdisciplinary topic in the process. Most notably, autonomous drone applications such as search rescue [1], mapping [2], surveying and inspection [3] are becoming more relevant with modern needs. However, these drones are generally light weight and fragile, making them susceptible to disturbances and damage such as rotor failure and wind. Researchers Razinkova, Gaponov, and Cho develop an adaptive control function to work in tandem with a PD controller to attenuate the effects of wind disturbance [4] while Lanzon, Freddi and Longhi utilize H_{inf} control theory to stabilize a quadrotor in the case of a rotor failure [5]. These robust controllers effectively maintain stability in the presence of disturbances, making their contribution to the field of quadrotor research very valuable.

Our project aims to create a robust controller that allows a quadrotor to fly on a desired trajectory while being disturbed by both wind disturbance and rotor failure. In order to achieve this, we rebuilt the PD controller with adaptive terms and H_{inf} controller using the dynamics and controller theory provided in the two respective papers [4] [5]. With these controllers, we are able to simulate the disturbances a quadrotor might experience, determining what the strengths and weaknesses of these implementations are. This information is necessary for future work to implement the two controllers together and determine if a robust controller for a micro aerial vehicle (MAV), modeled after the crazyflie 2.1 released by Bitcraze, could be developed to handle these simultaneous disturbances.

II. Theory

A. Quadrotor Dynamic Model

In this section we use the same model and methods described by D. Mellinger [6] and will briefly describe it here and in the next section describing the motor model. Using the coordinate system and free body diagram shown in Fig. 1 where W is defined as the world frame, and B is defined as the body fixed frame.

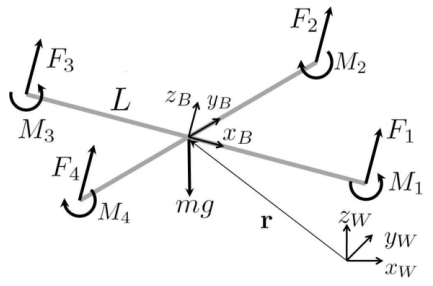

Fig. 1. Coordinate system and FBD of quadrotor

We use Z-Y-X Euler angles to model the rotation of the quadrotor frame to the world frame, defining ψ, as the rotation about z_W, or the yaw angle, ϕ as the rotation about x_W, or the roll angle, and θ as the rotation about y_W, or the pitch angle. The rotation matrix for transforming coordinates from B to W is

$$R = \begin{bmatrix} c\psi c\theta - s\theta s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

where $c\psi$ and $s\psi$ denote $cos(\psi)$ and $sin(\psi)$ respectively, and similarly for ϕ and θ. The forces acting on the system are gravity, acting in the z_W direction, and the thrust force from each motor acting in the z_B direction. If we denote the position of the quadrotor as r, then the equations of motion governing the acceleration of the COM is

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \sum F_i \end{bmatrix} \tag{1}$$

Angular velocity of the robot in the body-fixed frame, denoted as $p, q,$ and r, are a function of the rate of change in the

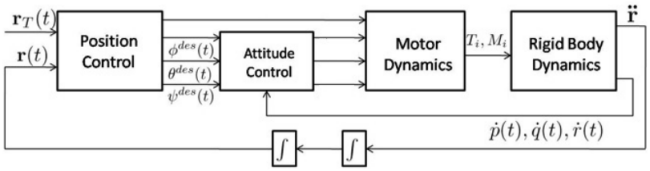

Fig. 2. Nested control loops for position and attitude

orientation of the quadrotor.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2}$$

Each rotor also produces a moment M_i where rotors 1,3 act in the z_B direction and rotors 2,4 act in the $-z_B$ direction. If we denote L as the length from the center of the quadrotor to the axis of rotation of the rotors, and I as the inertia matrix, then

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ \sum M_i \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3}$$

B. Motor Model

Each of the rotors has its own angular speed ω_i that produces a vertical force F_i

$$F_i = k_F \omega_i \tag{4}$$

and also produces a moment, M_i, about the axis of rotation

$$M_i = k_m \omega_i^2 \tag{5}$$

where constants k_F and k_M were found by the students and staff in the Kumar Lab.

C. PD Control with Adaptive Control

The quadrotor is controlled by a nested feedback loop, seen in Fig. 2, where the inner loop controls the attitude of the robot and the outer loop controls the position.

We initially assume that the quadrotor is at near hover points, meaning that the roll,pitch, and yaw angles are small. This allows us to both linearize the rotation matrix described in the previous section and decouple the position and attitude loops, acknowledging that the attitude can be controlled at a much smaller time step.

The decoupling of the position and attitude control is essential in this project, as you can imagine if there is an occurrence of actuator failure, then we lose control of the yaw angle about the z_B axis due to the unbalanced moments about the axis of rotation for the rotors.

If we assume that a state estimation system gives us r and $\dot{r}$, we find that the desired acceleration can be calculated by

$$\ddot{r}^{des} = \ddot{r}_T - K_d(\dot{r} - \dot{r}_T) - K_p(r - r_T) \tag{6}$$

where K_d and K_p are the diagonal, positive definite gain matrices.

By further exploiting the linear approximation of the rotation matrix, we can also state that the angular velocity of the body fixed frame is equal to the Euler angle velocities,

$$\begin{bmatrix} \dot{\theta} & \dot{\phi} & \dot{\psi} \end{bmatrix}^T = \begin{bmatrix} p & q & r \end{bmatrix}^T \tag{7}$$

allowing us to directly command the acceleration of the Euler angles. By using equations 6 and 7, we can find our system inputs

$$T = m(\ddot{r}^{des} + g) \tag{8}$$

$$\tau = I \begin{bmatrix} -k_{p,\phi}(\phi - \phi^{des}) - k_{d,\phi}(p - p^{des}) \\ -k_{p,\theta}(\theta - \theta^{des}) - k_{d,\theta}(q - q^{des}) \\ -k_{p,\psi}(\psi - \psi^{des}) - k_{d,\psi}(r - r^{des}) \end{bmatrix} \tag{9}$$

To model the wind disturbance, we assume that there is a known, constant or slow changing force F_v which acts in the X-Y plane of the world frame. Similarly to Razinkova et al's methods, the linearized dynamics is now

$$m\ddot{r} = m \begin{bmatrix} T\phi \\ T\theta \\ g - T \end{bmatrix} - \begin{bmatrix} F_{v,x} \\ F_{v,y} \\ 0 \end{bmatrix} \tag{10}$$

According to Razinkova et al, a conventional PD controller, like the one mentioned above, would fail to compensate for the effect of un-modeled disturbances due to rotor saturation limits and sensor noise amplifications. To deal with wind disturbance, Razinkova et al proposes to augment the controller by adding adaptive terms to the control inputs

$$\tau_1' = \tau_1 + A_x \tag{11}$$

$$\tau_2' = \tau_2 + A_y \tag{12}$$

where the A_x and A_y terms are the adaptation terms

$$A_x = \int \dot{A}_x dt = \gamma(x^{des} - x) \tag{13}$$

$$A_y = \int \dot{A}_y dt = \gamma(y^{des} - y) \tag{14}$$

The adaptation should converge with a positive adaptation rate, γ.

D. H Infinity Controller

Rotor failure is a serious event where the dynamics of the quadrotor is drastically modified. Most papers argue that loss of a rotor removes the ability to control and stabilize the quadrotor. Lanzon et al propose a strategy where by, one of the degree of freedom is sacrificed to ensure stable hovering state [5].

Most quadrotor controllers involve a cascading architecture with an outer loop that controls the 'trajectory following' aspects of the control. These aspects include the 3D translational coordinates of the quadrotor which are considered, in literature, the slow dynamics of the quadrotor. The required coordinates are inputs to the outer loop. The outer loop outputs the required roll, pitch and yaw angles required to reach the translational coordinates and are passed as inputs to the inner loop of the cascading controller.

Lanzon's strategy involves focusing on the roll, pitch, yaw and altitude of the quadrotor [5]. He opines that these parameters are necessary for obtaining a stable hover state. The loss of one of the rotor means that one of the four degree of freedom needs to be sacrificed. Non-zero roll and pitch angles immediately destabilize the quadrotor. A time varying altitude is also not a feasible option. The only option left is to allow a non-zero yaw angle. This means that the quadrotor will constantly rotate only along one axis. This still is not enough as the dynamics is very complex as the roll and pitch axes is constantly rotating about the yaw axis. This problem can be mitigated by driving the yaw rate to zero along with the roll and pitch angles, i.e. the quadrotor rotates with a constant velocity in the yaw axis.

$$\begin{bmatrix} u_f \\ \tau_p \\ \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{15}$$

$$x = [\hat{x_1} \quad \hat{x_2} \quad \hat{x_3} \quad \hat{x_4} \quad \hat{x_5}]^T = [\phi \quad \theta \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} - \tfrac{mgd}{k_r}] \tag{16a}$$

$$u = [\hat{u_1} \quad \hat{u_2} \quad \hat{u_3}] = [u_f - mg \quad \tau_q \quad \tau_r - mgd]^T \tag{16b}$$

Equation (15) maps the transformation from the individual rotor thrusts to the net thrust (u_f) and the torques across the roll, pitch and yaw axes, respectively. The plant for the inner loop is defined using five variables (eqn 16a). They are roll, pitch, and angular velocities in the three axes [5]. As stated before, yaw angle control is sacrificed. The objective of the controller will be to regulate $\hat{x_1}$, $\hat{x_2}$ and $\hat{x_5}$ to zero.

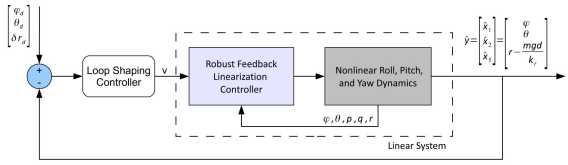

Fig. 3. Architecture of the Inner Loop [5]

The inner loop architecture is visualized in Figure 3. The H_{inf} controller in the paper [5] is applied to the linearized quadrotor dynamics. The linearized dynamics is summarised in the equation (17). It is derived from the jacobian of the dynamics at the point of stability. The detailed derivation of the linearization procedure is available in Lanzon *et al*'s paper [5] . For our example, we will switch off rotor 2 in Fig. 1. We lose control of a rotor across the roll axis. We need to remove τ_p and F_2 from equation (15) because τ_p can not be used to stabilize the rotor. The non linear dynamics of equation 2 and 3 can be written as equation 17a. The dynamics is linearized by finding a non-linear function in $\hat{x}$ for u that is linear in input v (see equation 17b). The $\alpha(\hat{x})$ and $\beta(\hat{x})$ terms are derived in the appendix of Lanzon's paper [5]. The linearized dynamics is obtained in equation 17c.

$$\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{G}(\hat{x})\hat{u} \tag{17a}$$

$$u(\hat{x}, v) = \alpha(\hat{x}) + \beta(\hat{x})v \tag{17b}$$

$$\dot{x} = A_r x_r + B_r v \tag{17c}$$

$$A_r = \begin{bmatrix} 0 & \frac{dgm}{k_r} & 1 & 0 & 0 \\ -\frac{dgm}{k_r} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{k_r}{I_{xx}} & \frac{dg(I_{xx}-I_{zz})m}{I_{xx}k_r} & 0 \\ 0 & 0 & -\frac{dg(I_{xx}-I_{zz})m}{I_{xx}k_r} & -\frac{k_r}{I_{xx}} & 0 \\ 0 & 0 & 0 & 0 & -\frac{k_r}{I_{zz}} \end{bmatrix} \tag{18a}$$

$$B_r = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{L}{2I_{xx}} & 0 & -\frac{L}{2dI_{xx}} \\ 0 & \frac{1}{I_{xx}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \tag{18b}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18c}$$

$$D = 0 \tag{18d}$$

Loop shaping algorithms attempt to shape the plant dynamics in frequency domain (visualized in a Bode Plot) into a target loop shape that have favourable properties. We define the A_r, B_r, C and D matrix of equation 18 as the (A, B, C, D) matrix for the state-space model of the plant. The target loop shape is defined based on the following parameters [5]:

1) High gain (+60 dB) for low frequencies for good reference tracking and high disturbance rejection.
2) Low gain at high frequencies for noise attenuation.
3) Crossover point is set at 100 rad/s. This factor is dependent on constituent frequencies of disturbance (low f) and noise (high f). We need to choose a balancing frequency that allows both disturbance rejection and noise attenuation.
4) Low gain roll-off (-20 dB/decade) at the crossover point for robustness.

The target loop shape, plant gains and the gains of the shaped plant is plotted in Figure 4. The MATLAB function *loopsyn* [9] is used to get the shaped plant. The inner loop controls the fast dynamics; so model reduction was considered for quicker response. This is done by finding the Hankel singular value of the states of the controller. It effectively tells you how much energy (quantity that measures the relative contribution

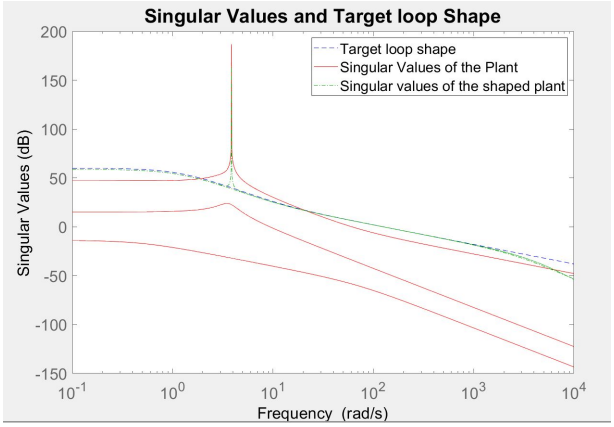

Fig. 4. Gains of the plant, shaped plant and target loop shape

of states to the outputs of the plant) each state utilizes. The lower energy states can be removed using MATLAB function *reduce*.

III. Results

A. PD Controller Adapting to Wind Disturbance

For the PD controller with the adaptive terms defined by Razinkova, Gaponov, and Cho [4], we simulated a matplotlib environment that updated the position, velocity, orientation, and angular velocity of a quadrotor within a given time frame of 30 seconds.

We simulate two different types of trajectories: a linear bang-bang trajectory with a goal position of [5,5,0] and a circular, minimum jerk trajectory where the quadrotor stops at points along the circle to simulate a more complex mission for a real-life scenario.

TABLE I
Experiment Parameters for y=x Trajectory

Case #	Parameters		
	Type of Force	***Forces (x,y,z)***	γ
1	No Force	$(0,0,0)m/s^2$	0
2	Constant F_x	$(0.00015,0,0)N$	0.004
3	Constant F_x and F_y	$(0.00015,0,0)N$	0.005
4	Sinusoidal F_x	$(\sin(t)*0.002,0,0)N$	0.004

In Table 1, we present the following experiments for the linear bang-bang trajectory, where we simulate different scenarios of wind disturbance acting on the quadrotor. It should be noted that if the error in position between the quadrotor's position and goal position is small enough, the quadrotor is expected to hover at the goal position. If not, the quadrotor will overshoot the goal and continue its trajectory.

For Case 1, we simulate no disturbances on the quadrotor. As seen in Figure (5), our PD controller successfully works, with minimal error due to the tuned gain parameters.

To allow for comparison, we provide the graphs for position and velocity with a constant disturbance and no adaptation terms. This can be seen in Figure (6).

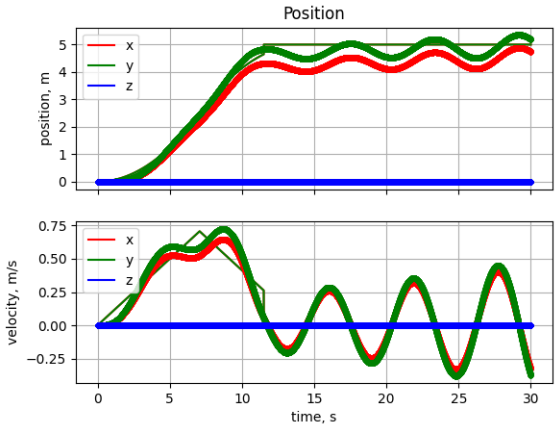

Fig. 5. PD Controller with no adaptive control due to absence of constant linear wind disturbance. The more thin lines in the graphs above represent the desired position and velocity

For Case 2, the adaptive terms are added to the controller to accurately account for the disturbance with an experimentally determined adaptive rate term, γ. It should be noted that in this case the quadrotor overshot the goal position, seen in Fig. 7

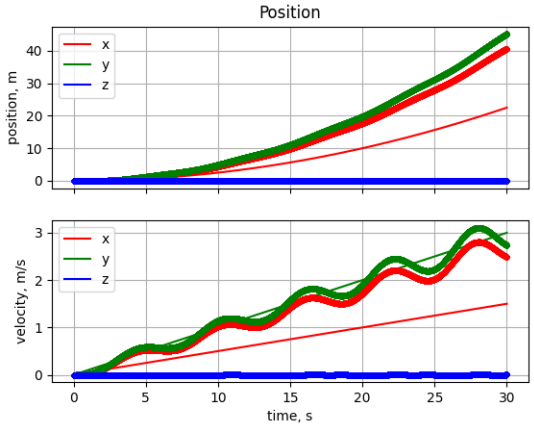

Fig. 6. PD Controller with no adaptive control in the presence of constant linear wind disturbance

Case 3 can be seen in Figure (8). Once again is important to note that the quadrotor does not reach the goal position, flying past it.

The last case can be seen in Figure (12). We find that we initially were able to travel near the desired position, but flew out of control soon after. Testing for a "perfect" γ term has proven inconclusive.

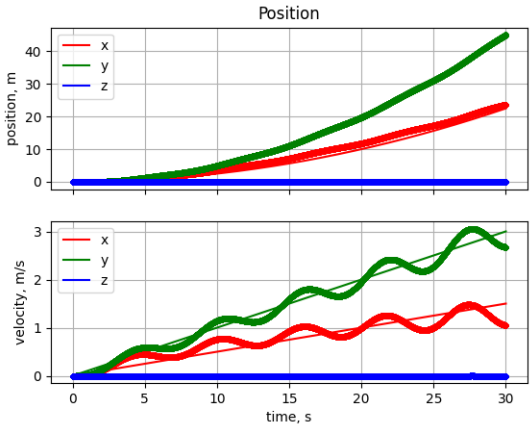

Fig. 7. PD Controller with adaptive control due to the presence of a constant linear wind disturbance

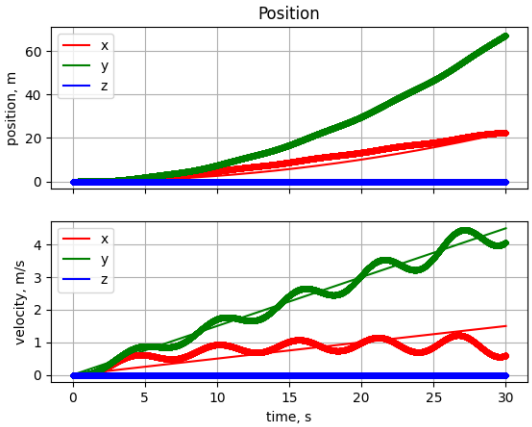

Fig. 8. PD Controller with adaptive control in the presence of linear wind disturbance in X-Y plane

TABLE II
EXPERIMENT PARAMETERS FOR CIRCULAR TRAJECTORY

Case #	Parameters		
	Type of Force	***Forces (x,y,z)***	γ
1	No Force	$(0, 0, 0)N$	0
2	Constant F_x	$(0.08, 0, 0)N$	0.0157
3	Constant F_x and F_y	$(0.08, 0.08, 0)N$	0.018
4	Sinusoidal F_x	$(\sin(t)*0.5, 0, 0)N$	0.5

Again, we present various cases to test wind disturbance for the circular trajectory [7] [8] in Table II. We reached very similar results for the more complex circular trajectory but for all cases we observe that the quadrotor reaches the waypoints with very little error due to the boundary constraints placed in the optimization problem but the error in position occurs more often along the spline traveling from waypoint to waypoint. It can be seen how the disturbance effects the flight path from Fig 9 to Fig 10 and how the adaptation term changes the flight path in Fig 11. The results of the various simulation is tabulated in Table III.

TABLE III
EXPERIMENT RESULTS FOR CIRCULAR TRAJECTORY

Case #	*Largest Error in Flight*	Largest Error at Waypoint
1	1.91m	0.02m
2	2.21m	0.35m
3	3.12m	0.68m
4	3.27m	0.22m

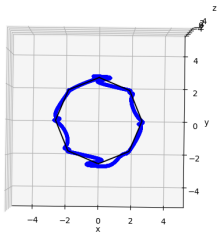

Fig. 9. Circular trajectory flight with no disturbance

B. H_{inf} Controller Adapting to a Single Rotor Failure

The simulation of the H_{inf} controller is to be done in the following steps:

1) The output of the plant ($\hat{x_1}$, $\hat{x_2}$, $\hat{x_5}$) is fed to the controller and it outputs *v*.
2) *v* is input to equation 17b to get the linearized input *u*.
3) Convert force and force rate limits to input limits on *u* and clip *u* based on the limiting values.
4) Iterate the quadrotor plant with the input obtained from the above steps.

Figure 10 shows us the roll, pitch and yaw rate for a closed loop system initialized with a random $x_{5\times5}$ matrix without the input saturation limits. Randomness is introduced to check the robustness of the controller. Multiple iterations were run on different initial states and stabilization was achieved for all the cases. The yaw rate is stabilized the quickest while the roll takes the most time to stabilize. This may be because the loss of quadrotor across the roll axis makes that degree of freedom difficult to control. The rotation of the quadrotor about yaw is dependent on the rotor blade drag among other factors. In Lanzon's paper, the results of simulations show that the working rotor across roll axis is switched on only

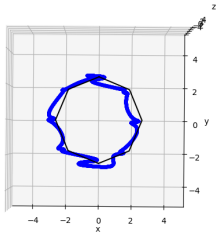

Fig. 10. Circular trajectory flight disturbed by a 0.08 N force

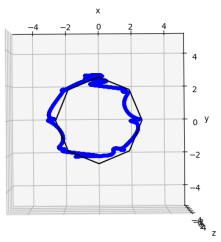

Fig. 11. Circular trajectory flight with adaptation terms

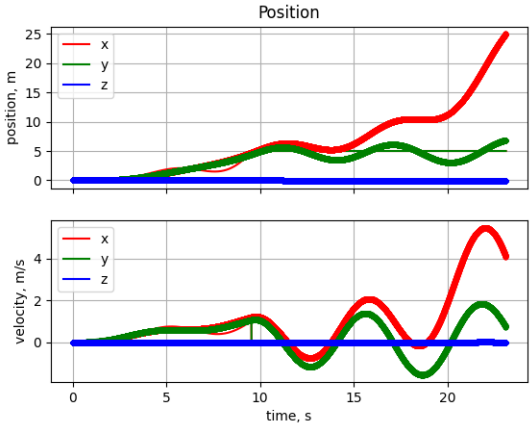

Fig. 12. PD Controller with adaptive control in the presence of a sinusoidal wind disturbance in X direction.

intermittently, providing pulses of thrusts because continous operation will destabilize the quadrotor along the roll axis.

We also tried to reduce the order of the controller. The controller output by *loopsyn* [9] has 22 states. This value is dependent on the complexity of the target loop and the *GAMMA* (performance metric of loop shaping) value we input into the function. We have plotted the Hankel Singular Values in Figure 14. Based on the results, model reduction was done on a trial and error basis using a bilinear search iteration between 6 and 13 states. The least states that could visibly follow the dynamics of the original controller was 8. The step response of the open loop model (Plant x Controller) of both the controllers (original and reduced) was compared in Figure 15. The reference tracking was similar for the both the controllers.

IV. Discussion

A. Adaptive Controller Analysis

In our experimentation of the adaptive controller, we found it to be an unreliably and inefficient method of attenuating the force that the quadrotor experiences. While we are simulating a much smaller quadrotor compared to that of the researchers (.03 kg being our quadrotor mass while Razinkova et al quadrotor is unspecified, but traditionally heavier), it is still unreliable to determine an adaption constant for every different type of force. The experimentation necessary to determine a

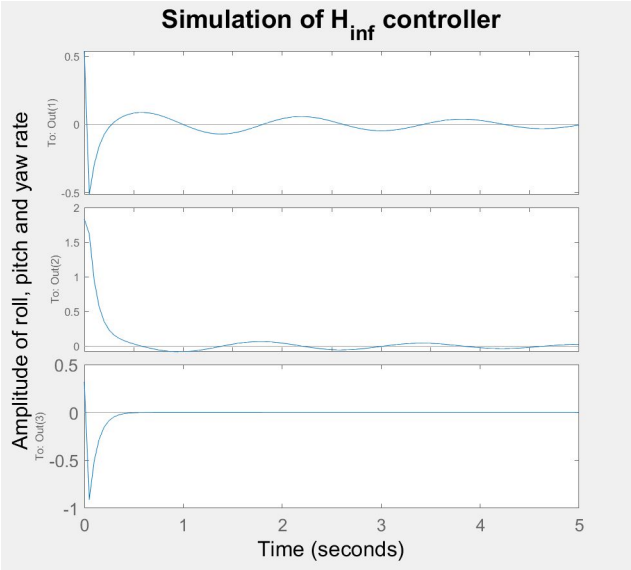

Fig. 13. Simulation of the closed loop with the controller obtained through k

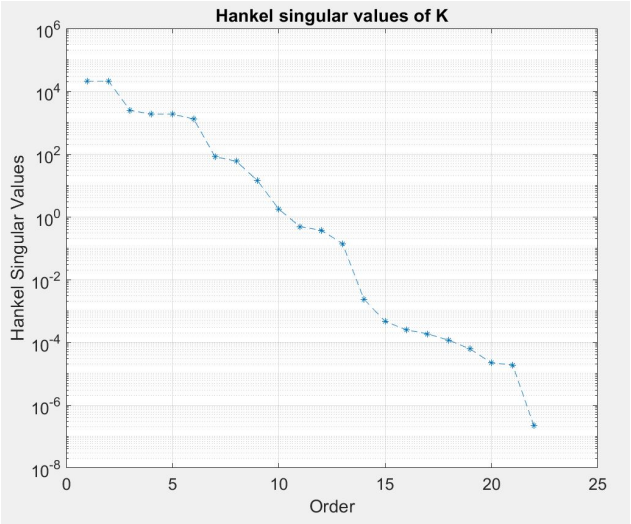

Fig. 14. Hankel Singular Values of the controller K. There are 22 states in the controller plant

γ constant that stabilizes a trajectory towards the destination is lengthy, leading to coefficients that get the quadrotor very close to the desired position, but overshooting it and flying out of control in another direction. We see this very clearly in figures 7 and 8. In both cases, the quadrotor was tuned to follow a desired trajectory, but due to the very precise values that the adaption term demands, there is still some error that does not bring the quadrotor exactly to the desired position. Due to the strictness of our simulation, we do not stop the motion of the quadrotor until it has reached the point leading to this overshoot. When given more time in the simulation, the quadrotor would begin to correct this overshoot, flying out of control and causing the simulation to stop once it is 20 meters away from the the goal. This tells us that maybe our

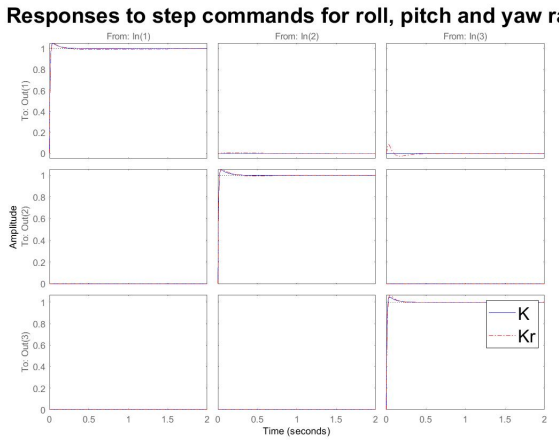

Fig. 15. Difference in tracking of a step signal for models with the two cases of shaped plant controller and reduced order controller

gain matrices could be better tuned, but given our time-frame we did the best we could manage.

When looking at the sinusoidal force in the x direction (Figure (12)), we see that the quadrotor does get to the desired position as shown by the desired position x and y lines staying constant at 5 meters, however the quadrotor soon flies out of control. While the results of cases 1 and 2 might have been reflective of a problem in the implementation of the adaptive controller, case 3 confirms that it is indeed functional, but requires the adaptive constant to be finely tuned to the scenario that is being simulated.

In the simulation run by Razinkova, Gaponov, and Cho, they were able to stabilize their quadrotor along the y=x trajectory in response to a force of 1N in the x direction [4]. We hoped to potentially adapt to larger forces, but found that it is not possible to due to the low weight of our MAV and our controller built around low roll, pitch, yaw angles. It may be better to build a controller that allows for more aggressive maneuvers. Razinkova et al also did not specify the method that was used to determine the adaption term γ. For this reason, we made the assumption that the γ term was a constant, determined through trial and error as there were no methods or functions to determine this value. We would like to research more methods to determining this adaption constant in future work, determining a function or algorithm to expedite the search for a γ value. This would allow for a much more reliable adaptive term that could be very useful in driving a quadrotor along the desired trajectory with minimal error.

B. Rotor Failure Control Analysis

The simulation in Figure 13 shows that the controller creates a closed loop system that is stable and satisfies the objectives put forward at the start of the project. The order reduction of the controller also shows that removing lower energy states does not drastically affect the efficacy of the controller (see Figure 15). The input saturation limits were not incorporated into the simulations even though they were earmarked at the start of the project. They may affect the stability of the quadrotor as input saturation may increase the time taken for stabilizing the quadrotor.

The roll and pitch dynamics of the closed loop system shows overshoot with it being prominent in the roll dynamics. Overshoot may be improved by adding an auxillary differential element to the roll and pitch section of the controller but further testing needs to the done to verify this claim.

We need to test the H_{inf} controller with the adaptive controller. Since the dynamics were linearized for H_{inf}, the optimal input should be a sum of the adaptive and H_{inf} inputs. However, the adaptive controller will not consider failure of a rotor. Its control action could destabilize the quadrotor. One solution is to run one iteration of adaptive control, then stabilize, and repeat. Loop shaping techniques like H_{inf} allow easy disturbance rejection and noise attenuation as an intrinsic feature. Testing just the wind disturbance part for hovering without trajectory following part for H_{inf} controller could also be a possible research direction

V. Conclusion

While we were unable to integrate both the adaptive terms and H_{inf} controller together in the quadrotor simulation, we were able to test the robustness of the individual controllers and determine their strengths and weaknesses. The H_{inf} controller was successful at stabilizing the quadrotor in the case of a single rotor failure, as we were successful in stabilizing the roll and pitch angles to 0 and stabilize the yaw rate to a constant speed at a given altitude. Our PD controller with adaptive terms was able to mitigate the wind disturbance cases that we simulated, however the method of determining gamma was inefficient and generally ineffective. This led to many trajectories getting very close to reaching the destination, but never truly getting there, causing the quadrotor to fly out of control as a result.

In future work, we plan on continuing to develop our H_{inf} controller, adding input saturation limits. We also will integrate the H_{inf} controller into the simulation with the adaptive term, determining if our combined controller would be robust enough to stabilize a quadrotor in a hover, linear, or complex trajectory in the presence of rotor failure and wind disturbance.

Acknowledgment

We would like to sincerely thank Professor Michael Posa, William Yang, and Yu-Ming Chen for all the help they have provided during the development and implementation of this research project.

References

[1] A. Ryan and J.K. Hedrick, “A mode-switching path planner for UAV-assisted search and rescue,” Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 , pp.1471- 1476, 2005.

[2] B. Renardi, E. Khosasi, Y. Y. Nazaruddin and E. Juliastuti, ”Using Multi-Quadrotor System for Effective Road Mapping,” 2019 6th International Conference on Electric Vehicular Technology (ICEVT), Bali, Indonesia, 2019, pp. 246-252, doi: 10.1109/ICEVT48285.2019.8994003.

[3] K. Alexis, G. Nikolakopoulos, A. Tzes, and L. Dritsas, “Coordination of Helicopter UAVs for Aerial Forest- Fire Surveillance”, Applications of Intelligent Control to Engineering Systems, Springer, vol.39, pp.169-193, 2009.

[4] Razinkova, A., Gaponov, I., amp; Cho, H. (2014). Adaptive control over quadcopter UAV under disturbances. Retrieved December 4, 2020, from https://www.researchgate.net/publication/288189410

[5] Lanzon, A., Freddi, A., amp; Longhi, S. (2014). Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure. Journal of Guidance, Control, and Dynamics, 37(2), 580-591. doi:10.2514/1.59869

[6] D. Mellinger and V. Kumar, ”Minimum snap trajectory generation and control for quadrotors,” 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 2520-2525, doi: 10.1109/ICRA.2011.5980409.

[7] Spedicato, S., amp; Notarstefano, G. (2018). Minimum-Time Trajectory Generation for Quadrotors in Constrained Environments. IEEE Transactions on Control Systems Technology, 26(4), 1335-1344. doi:10.1109/tcst.2017.2709268

[8] Richter, C., Bry, A., amp; Roy, N. (2016). Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. Springer Tracts in Advanced Robotics Robotics Research, 649-666. doi:10.1007/978-3-319-28872-7_37

[9] loopsyn. (n.d.). Retrieved December 09, 2020, from https://www.mathworks.com/help/robust/ref/loopsyn.html