Post-Rotor-Failure Performance of a Feedback Controller for a Hexacopter

Michael McKay
PhD Student

Robert Niemiec
PhD Candidate

Farhan Gandhi
Redfern Chair in Aerospace Engineering

Center for Mobility with Vertical Lift (MOVE)
Department of Mechanical, Aerospace and Nuclear Engineering
Rensselaer Polytechnic Institute
Troy, NY, United States

ABSTRACT

A feedback controller is designed and implemented for a regular hexacopter based on the AeroQuad Cyclone ARF kit. This controller is designed with an inner loop control law as a set of parallel PID controllers for aircraft altitude, pitch, roll, and yaw attitudes, as well as an outer loop for control over aircraft body velocities. Rotor failure is modeled in the dynamic simulation by setting the rotor force and moment output to be zero regardless of the commanded control input to that rotor, the feedback controller utilizes no knowledge of this fault during simulation. Various trajectories are commanded to examine the performance of the baseline feedback controller in the event of forward rotor failure, including hover, forward flight, and more complex maneuvers. The controller is demonstrated to recover the aircraft states after the transient effects of the rotor failure, as well as complete the defined state trajectory, demonstrating tolerance to single rotor failure.

NOTATION

Ω	Rotor rotational velocity
R	Rotor radius
λ	Rotor Inflow Ratio
A	State Evolution Matrix
B	Control Sensitivity Matrix
C	Output Matrix
X	Aircraft x-position (Inertial Frame)
Y	Aircraft y-position (Inertial Frame)
Z	Aircraft z-position (Inertial Frame)
ϕ	Aircraft roll attitude (Inertial Frame)
θ	Aircraft pitch attitude (Inertial Frame)
ψ	Aircraft yaw attitude (Inertial Frame)
u	Aircraft x-velocity (Body Frame)
v	Aircraft y-velocity (Body Frame)
w	Aircraft z-velocity (Body Frame)
p	Aircraft roll rate(Body Frame)
q	Aircraft pitch rate (Body Frame)
r	Aircraft yaw rate (Body Frame)

INTRODUCTION

Multicopters are gaining popularity in a number of applications ranging from hobbyist and recreational use to professional arenas including law enforcement, military, and commercial uses. This is mainly due to the mechanical simplicity of such aircraft, utilizing distributed electric propulsion to independently vary rotor speed on multiple fixed pitch rotors in lieu of traditional collective and cyclic rotor pitch controls. Multicopters that have more than four rotors possess control redundancy, a feature that may be exploited in order to provide tolerance to the failure of one or more rotors.

Fault tolerant control schemes are well-researched for small-scale multicopters, with many methods published in the literature recently. Marks et al. (Ref. 1) use knowledge of actuator effectiveness to allocate a set of controls to the functioning effectors via a pseudoinverse-type approach. Falconi et al. (Ref. 2) show that a Model Reference Adaptive Control (MRAC) scheme can be augmented to retain system stability in the event of a control failure on a hexacopter aircraft. Provided that the rotor failure is known, the control laws can be adapted to reallocate controls and this is shown to stabilize the aircraft. Falconi and Holzapfel (Ref. 3) later published a fault-tolerant control strategy for a hexacopter that required no exact knowledge of the fault type. This control scheme utilized current state knowledge as well as a reference model for the aircraft to adapt the control laws, based on a linear static actuator model and a backstepping baseline controller with adaptive augmentation, and stabilize the aircraft post-failure.

Other groups have demonstrated the loss of controllability of a hexacopter when a rotor fails (Refs. 4,5), indicating that not all aircraft states can be maintained at a desired value. This conclusion is local to the hover flight condition, and can be handled in different ways. Du et al. (Ref. 4) present their controllability analysis of a hexacopter with single rotor failure, then suggest that the directional dynamics tracking (heading, yaw rate) be ignored in order to recover the aircraft. As such,

Presented at the AHS International 74th Annual Forum & Technology Display, Phoenix, Arizona, USA, May 14–17, 2018.

the study overwrites the measured yaw states to be identical to the desired such that the controller applies no compensatory effort, then demonstrate that the aircraft can be brought in to land post-rotor-failure. It is worth noting that this aircraft will be spinning rapidly throughout the maneuver. Achtelik et al. (Ref. 5) go through a similar derivation for the controllability of a hexacopter post-rotor-failure, showing that the control sensitivity matrix is not full rank and therefore the aircraft is not controllable. Instead of ignoring the directional dynamics, the study suggests that the rotor diametrically opposite to the failed rotor be operated in forward and reverse in order to compensate for external disturbance in yaw. However, this assumes that the diametrically opposed rotor produces only torque without any thrust (an impossibility for fixed-pitch rotors) and also ignores the complex aerodynamics associated with fully reversed flow.

Other control schemes for fault tolerance include work by Du et al. (Ref. 6), where the tracking error of the aircraft is decomposed into contributions from external disturbance and those from the rotor failure itself. This state additive decomposition allows for the separation of error leading to separate compensatory efforts generated from the controller for the rotor failure disturbance as well as external disturbances, and then the recombination of both control efforts to stabilize the aircraft. The authors demonstrate the effectiveness of this approach without knowledge of the rotor failure itself. Schneider et al. (Ref. 7) use parametric programming to determine the appropriate control allocation given a certain desired virtual control input. This allocation is determined with knowledge of the failure type and consequently an attainable control set for the aircraft. Given a desired virtual control input and the known attainable set, a constrained optimization routine determines the appropriate set of rotor controls. Lastly, Sadegzadeh et al. (Ref. 8) solve the fault-tolerant control problem by finding PID gains to stabilize the aircraft at different airspeeds and fault scenarios, scheduling them appropriately in their controller. This method is then compared to MRAC (using MIT rule), demonstrating that this simpler control structure performs well.

All of the cited studies are required to solve a control allocation problem, a necessary step in controlling systems with control redundancy. Oppenheimer et al. (Ref. 9) provide a survey of many control allocation methods, including explicit ganging, pseudoinverse, and various optimal control methods. Johansen et al. (Ref. 10) provide another detailed survey of control allocation methods, Tjønnås and Johansen (Ref. 11) have also published on adaptive control allocation.

Overall, the literature surveyed present various fault-tolerant control methods for multicopter aircraft. Qi et al. (Ref. 12) provide a survey of fault tolerant control methods (as well as fault identification methods) for manned and unmanned helicopters, Ducard (Ref. 13) provides another brief survey of fault tolerant control methods for small UAVs.

Many of the given methods require knowledge of the fault type to adapt the control laws, a problem that is challenging on its own. Another potential issue with the work cited here is the lack of sophistication in the dynamic model of the aircraft. Most prior work utilizes simple dynamic models to approximate aircraft dynamics, assuming rotor thrust and torque are proportional to the square of the rotor speed. This model is a fair approximation local to hover, but will fail to capture additional hub forces and moments in edgewise flow that may be predicted by a more sophisticated model. This is demonstrated by Niemiec and Gandhi (Ref. 14), where the authors show that a Blade Element Theory based model coupled with a Peters-He (Ref. 15) dynamic wake is capable of predicting rotor forces and moments both in hover and varying edgewise flow conditions. An accurate prediction of rotor dynamics can prove to be useful in the modeling and simulation of a multirotor aircraft with a rotor failure.

The focus of the present study is to simulate the performance of a hexacopter using the dynamic model set forth in (Ref. 14) and to implement a baseline controller to stabilize the aircraft dynamics. The performance of this controller after single rotor failure with no change to the baseline feedback control laws will then be examined.

APPROACH

Aircraft Model

The aircraft considered in the present study is a hexacopter derivative of the AeroQuad Cyclone ARF kit. The nominal gross weight of the aircraft is 2kg, with other aircraft and rotor properties given by Table 1. Note that the blade is assumed to have linear taper and chord distribution along the span. A schematic of the aircraft is depicted in Fig. 1.

Table 1. Aircraft Parameters

Parameter	Value
Gross Weight	2 kg
Boom Length	0.3048 m
No. of Rotors	6
Rotor Radius	0.1245 m
Tip Airfoil	Clark Y
Tip Chord	0.0098 m
Tip Pitch	11.5°
Root Airfoil	NACA 4412
Root Chord	0.0253 m
Root Pitch	21.5°

To simulate aircraft performance, a dynamic simulation is implemented using summation of forces and moments to determine aircraft accelerations. The force and moment contributions considered include gravity, fuselage drag (modeled as a cylinder), as well as rotor forces and moments. Blade Element Theory coupled with a 3×4 (10 state) Peters-He Dynamic wake model is to calculate rotor induced velocities. To determine rotor loads, sectional contributions are first integrated along the span and then averaged about one rotor revolution. These steady rotor loads are then used in conjunction with loads on the airframe and due to gravity to determine the

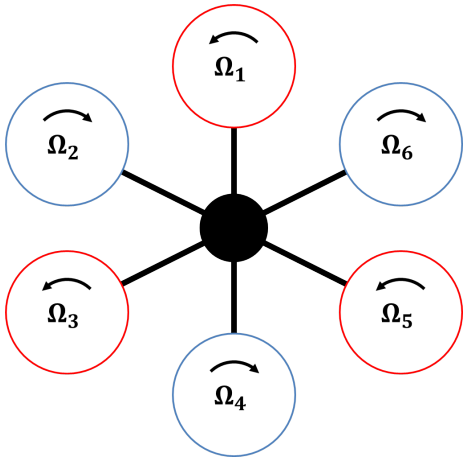

Fig. 1. Hexacopter Schematic

overall aircraft forces and moments given a specific operating condition.

The dynamic model of the aircraft allows for a trim routine to be performed to estimate control inputs to maintain steady level flight in a desired condition. Trim is said to be achieved when the aircraft accelerations are zero, the solution can be found with a Newton-Rhapson method to find the root of the nonlinear set of dynamic equations representing the 6 equilibrium equations for the aircraft. This method is outlined as follows, given the set of equations represented by:

$$\dot{x} = f(x,u) \tag{1}$$

Trim is satisfied when $\dot{x} = 0$, in order to solve this, a Jacobian matrix is formed by use of centered finite differences to approximate the sensitivities of aircraft accelerations to control input (as well as roll and pitch attitude).

$$J_{ij} = \frac{\partial f_i}{\partial u_j} \tag{2}$$

This Jacobian matrix is then used to update the control vector, u, by the following:

$$u_{k+1} = u_k - J^{-1} f(x,u_k) \tag{3}$$

We note that because the number of control inputs (8, 6 rotor speeds + roll attitude + pitch attitude) exceeds the number of equilibrium equations (6), the Jacobian matrix that is formed is non-square ($J \in \mathbb{R}^{6\times 8}$), and is not invertible. To circumvent this issue, the Moore-Penrose pseudoinverse (J^+) is used in place of the traditional inverse in Eq. 3.

$$J^+ = (J^T J)^{-1} J^T \tag{4}$$

Once trim is solved, the system is numerically linearized to obtain a state-space model for the aircraft (Eq. 5). This allows for linear control techniques to be applied to the simplified model which can be used on the nonlinear plant local to the trim point. Centered finite differences are used to approximate the state evolution matrix (A) and the control sensitivity matrix (B).

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{5}$$

The matrices in Eq 5 take the following form:

$$\begin{aligned} A_{ij} &= \frac{\partial f_i}{\partial x_j} \quad B_{ij} = \frac{\partial f_i}{\partial u_j} \\ C &= I_n \qquad D = \emptyset \end{aligned} \tag{6}$$

With $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, where n is the number of states and m the number of control inputs.

For the aircraft considered in the present study, the linearized model contains 72 dynamic states comprised of the 12 rigid body states along with 60 inflow states (6 rotors, 10 inflow states per rotor). The system has 6 controls given by the 6 control inputs (speed of each rotor, or multirotor controls). That is, the matrix dimensions previously defined are $n = 72$ and $m = 6$.

If the autonomous system is analyzed via an eigenvalue decomposition, it can be seen that the poles corresponding to the rigid body states exist relatively close to the origin when compared to those for the inflow dynamics, which exist very far in the left-half plane. Because the inflow dynamics evolve at such a high frequency compared to the other modes, they can be treated as static and the method of static condensation can be used to reduce the model back down to the 12 rigid body states.

Static condensation is performed by partitioning the state-space model as follows, where $x \in \mathbb{R}^{12\times 1}$ and $\lambda \in \mathbb{R}^{60\times 1}$:

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A_{xx} & A_{x\lambda} \\ A_{\lambda x} & A_{\lambda\lambda} \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} + \begin{bmatrix} B_x \\ B_\lambda \end{bmatrix} u \tag{7}$$

The model reduction comes from setting $\dot{\lambda} = 0$ and solving for the system in terms of x and u.

$$\begin{aligned} \dot{\lambda} = 0 &= A_{\lambda x} x + A_{\lambda\lambda}\lambda + B_\lambda u \\ \lambda &= -A_{\lambda\lambda}^{-1}(A_{\lambda x} x + B_\lambda u) \end{aligned}$$

Substituting into the equation for $\dot{x}$:

$$\begin{aligned} \dot{x} &= A_{xx}x + A_{x\lambda}(-A_{\lambda\lambda}^{-1}A_{\lambda x}x - A_{\lambda\lambda}^{-1}B_\lambda u) + B_x u \\ \dot{x} &= \underbrace{[A_{xx} - A_{x\lambda}A_{\lambda\lambda}^{-1}A_{\lambda x}]}_{A_{RO}} x + \underbrace{[B_x - A_{x\lambda}A_{\lambda\lambda}^{-1}A_{\lambda x}B_\lambda]}_{B_{RO}} u \end{aligned}$$

Which results in the reduced system:

$$\begin{aligned} \dot{x} &= A_{RO}x + B_{RO}u \\ y &= x \end{aligned} \tag{8}$$

In this system, the state vector $x \in \mathbb{R}^{12\times 1}$ is comprised of the 12 rigid body states, so the matrices in Eq. 8 have dimension $A_{RO} \in \mathbb{R}^{12\times 12}$ and $B_{RO} \in \mathbb{R}^{12\times 6}$. The control vector is

comprised of the 6 multirotor controls defined for a regular hexacopter:

$$x = \{X \quad Y \quad Z \quad \phi \quad \theta \quad \psi \quad u \quad v \quad w \quad p \quad q \quad r\}$$
$$u = \{\Omega_0 \quad \Omega_{1S} \quad \Omega_{1C} \quad \Omega_D \quad \Omega_{2S} \quad \Omega_{2C}\} \tag{9}$$

Multirotor controls are outlined in (Ref. 16) and portrayed in Fig. 2. For a regular multicopter, multirotor controls are a transformation from individual rotor controls, yielding a set of primary controls that generate force and moment along only one axis, and additional reaction-less controls. Classical multirotors aircraft have four primary controls regardless of the number of rotors, corresponding to the control of thrust, pitching moment, rolling moment, and yaw. These controls are written as a combination of individual rotor speeds, determined by the location of an individual rotor on the aircraft.

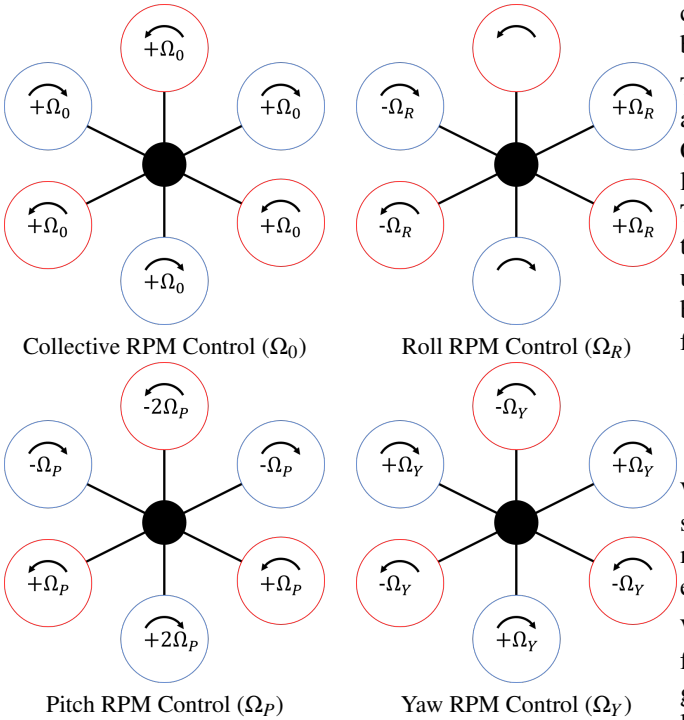

Fig. 2. Primary Multirotor Controls (Ref. 16)

Controller Design

The state space system defined by the linearized system was first examined for autonomous stability. This was done via an eigenanalysis of the state evolution matrix A. The results for hover are given in Fig. 3.

The figure plots the location of the eigenvalues of the A matrix, or the location of the open-loop poles in the complex plane for the hexacopter in hover. Farthest in the left-half plane are the (2) poles corresponding to the roll and pitch subsidence modes, closer to the origin is the pole corresponding to the heave mode, closer still to the origin is the yaw rate pole. At the origin, there are 4 integrators corresponding to aircraft position and heading. Finally, there are two sets of

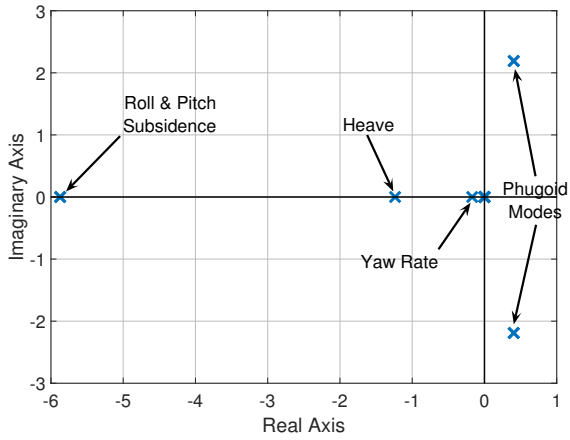

Fig. 3. Open-Loop Poles for Hexacopter in Hover

conjugate poles in the right-half plane representing the unstable longitudinal and lateral phugoid modes.

The existence of poles in the right-half plane indicates that the aircraft plant is unstable at the current flight condition (hover). Consequently, a PID controller was implemented for the baseline aircraft (no failure) using the linear models generated. This is made easier by use of multirotor controls, which tend to impart changes in only one axis, enabling the appropriate use of SISO control design. First, a transfer function is created between a desired output and control input, in the following form:

$$Y = G(s)U$$
$$G(s) = C_i(sI - A)^{-1}B_j + D \tag{10}$$

Where C_i is the i^{th} row of the output matrix to select the desired output and B_j is the j^{th} column of the control sensitivity matrix to select the desired control input. Note that D is an empty matrix, but it is included here for completeness.

With the input-output relationship characterized by $G(s)$, feedback gains can be determined using any number of tuning guides. For the purposes of the present study, an inner control loop was first constructed that compensated for altitude tracking error with Ω_0 and error in tracking attitudes (ϕ, θ, ψ) with Ω_R, Ω_P, and Ω_Y, respectively.

With altitude and attitude tracking stabilized, a closed-loop system could now be defined that takes reference altitude and attitude as an input. This closed-loop is stable and was wrapped by an outer control loop to allow for velocity tracking. This was done in a similar manner to the inner loop design, instead choosing attitude as an input to the closed loop system to control the velocity output. To summarize, the aircraft plant model takes 6 inputs, the primary and reaction-less multirotor controls, and outputs the aircraft state. The inner loop stabilizes the plant, taking input of altitude and attitudes and outputs a compensation in terms of the primary multirotor controls (reaction-less controls Ω_{2s} and Ω_{2c} set to 0). This in turn is closed by the outer loop, which takes input of velocity and outputsa reference attitude. Figure 4 illustrates the architecture of the control design as it has been described.

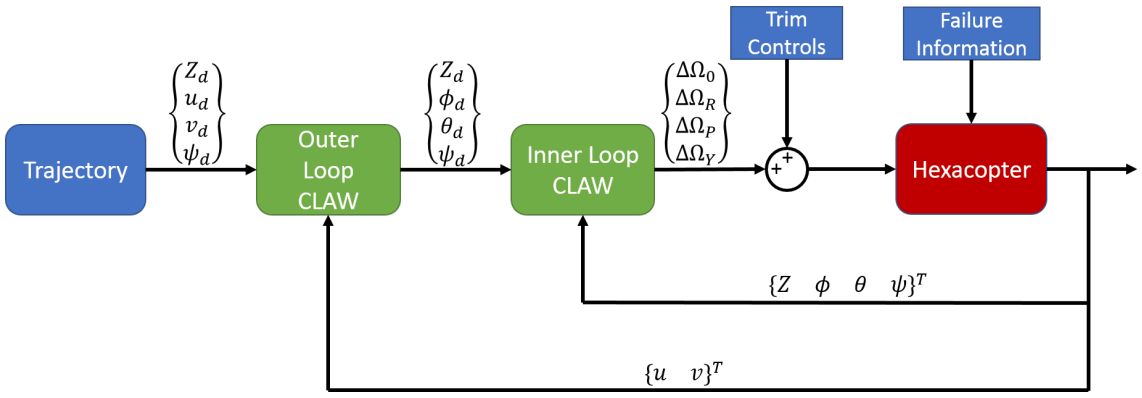

Fig. 4. Controller Block Diagram

The control architecture for the present work utilizes control laws that are tuned for a given flight condition. As such, the compensatory control inputs are determined relative to the reference trim condition which are provided to the controller just upstream of the plant (and calculated by an off line trim routine prior to simulation).

In addition,rotor failure is modeled by setting the force and moment output of the failed rotor to be zero regardless of the commanded rotor speed input to the plant. Dynamically, this effectively removes the rotor and its inflow states from the aircraft model. Previously, the aircraft has been modeled in steady state with single rotor failure and shown to be able to successfully trim (Refs. 17, 18).

RESULTS

The controller as discussed is simulated for different flight conditions using different components of the controller to examine performance in recovery post-rotor-failure, as well as the ability of the control architecture to re-capture the desired trajectory.

Inner Loop Simulation

Hover First, instead of fully specifying the aircraft trajectory (full state trajectory), only the desired altitude and attitudes were specified to demonstrate the performance of the inner loop control laws. The first case simulated the aircraft in hover, defined by an altitude hold and zero attitude. The front rotor (rotor 1) is failed at 5 seconds, indicated by the vertical red dashed line on Fig. 5.

Prior to rotor failure, all rotors operate at the same speed (5400 RPM), an intuitive result given the geometry and number of rotors on the aircraft. Once a rotor fails, it has been demonstrated (Refs. 5, 17, 18) that in order to trim in hover, the rotor diametrically opposite to the failed rotor should be turned off and the remaining rotors be sped up, in this case from 5400-6600 RPM. The remaining four rotors still operate at identical speeds to maintain moment balance at the aircraft level.

Dynamically, rotor four cannot be instantaneously turned off and the failure of the front rotor causes the aircraft to pitch

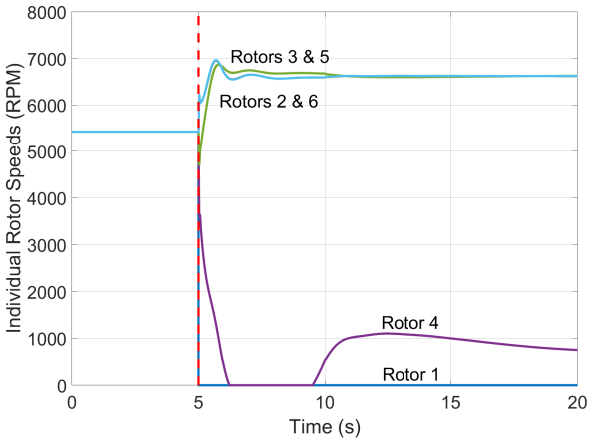

Fig. 5. Rotor Speed Time History for Rotor 1 Failure in Hover (Inner Loop)

nose-down, yaw nose-left (Fig. 6, North-East-Down coordinate system), and descend (Fig. 7). These transients create errors in the state tracking, which results in the feedback controller generating compensatory inputs, whereas prior to rotor failure the feedback controller was generating no control inputs (Fig. 8).

If we refer to classical multirotor controls as defined for a fully operational hexacopter in (Ref. 16), compensation should therefore occur with collective RPM (Ω_0), pitch RPM (Ω_R), and yaw RPM (Ω_Y). Figure 8 shows the commanded multirotor control input from the feedback controller, which reflects these compensations.

A trim collective input is summed with these feedback commands to give the total multirotor control input for the hexacopter. If these total controls are transformed into the individual rotor speeds, they take the values given in Fig. 9.

The speeds in Fig. 9 reflect the rotor speeds that the inner loop control laws determine will compensate the error in reference state tracking. However, saturation limits (minimum 0 RPM, maximum 10,000 RPM) and rotor failure ($\Omega_1 \equiv 0$) alter the actual operational speeds of the individual rotors on the hexacopter, yielding the rotor speed time history seen in Fig. 5.

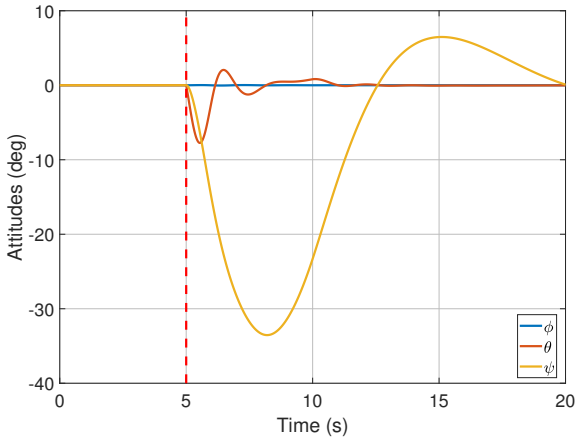

Fig. 6. Attitude Time History for Rotor 1 Failure in Hover (Inner Loop)

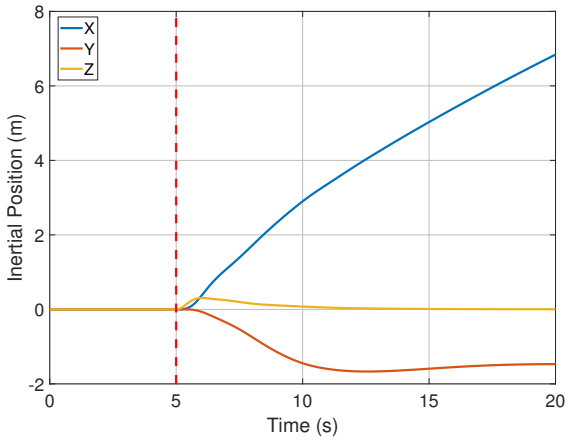

Fig. 7. Position Time History for Rotor 1 Failure in Hover (Inner Loop)

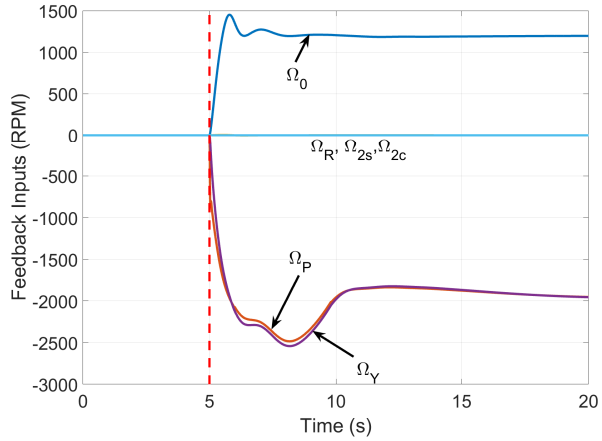

Fig. 8. Commanded Changes in Multirotor Controls from the Feedback Controller for Rotor 1 Failure in Hover (Inner Loop)

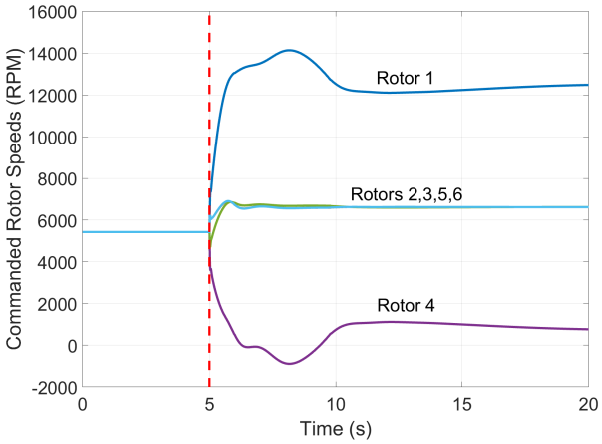

Fig. 9. Total Commanded Rotor Speeds for Rotor 1 Failure in Hover (Inner Loop) prior to Saturation Limits

If all 6 rotors were fully operational, the use of the primary multirotor controls would restore the aircraft to the desired state easily. However, the front rotor is no longer functional, therefore regardless of the feedback controller input to the plant the front rotor will output no forces or moments. As a consequence, the primary multirotor controls for a hexacopter no longer function as originally intended. In other words, the controls as determined by the feedback controller no longer impart only single axis responses.

Given that the controller as it is designed utilizes no knowledge of the failure at hand, no feedback laws are changed nor is the reference input for the desired flight condition. As such, the feedback laws still attempt to rectify state tracking errors with the primary multirotor controls and the reference inputs to the system still correspond to aircraft trim with six operational rotors. However, because the aircraft still possesses five functional rotors, the primary multirotor controls largely impart forces and moment in the desired axis. This leads to a steady set of rotor speeds (from the feedback controller) that can be expressed as a linear combination of the primary multirotor controls. These compensatory control inputs, when added to the reference inputs, give a set of rotor speeds that drive the aircraft back to a steady hover defined by no vertical translation and zero attitude. It should be noted that these rotor speeds match those determined in Refs. 17, 18.

If the compensation of the transients generated by the failure of the front rotor is considered outside of the definition of the multirotor controls, it would be expected that in order to recover balance in pitching moment, the rear rotor (diametrically opposite the failed rotor) should be slowed significantly to alleviate the net nose-down moment on the aircraft. Simultaneously, the functioning rotors should increase their speeds to retain the aircraft altitude and there should also exist a difference between the clockwise and counter clockwise spinning rotors to balance the net yawing moment on the aircraft. Figure 5 shows all of these compensations occurring in the few seconds immediately after rotor failure, with some slight variation to account for the off-axis effect of the multirotor controls. In large part, the control history behaves as previously discussed, with the exception of the speed of rotor 4. This speed is governed by the pitch RPM (Ω_P), collec-

tive RPM (Ω_0), and yaw RPM (Ω_Y) inputs, so it decreases due to pitch and yaw attitude error (red and yellow curves in Fig. 6), which outweigh the collective RPM input (see Figs. 8 and 9). The remaining rotor speeds collectively increase in order to climb back up to the desired altitude, with a noticeable difference in rotor speed between the front and rear rotor to regain the pitch attitude. An examination of the attitude history reveals that pitch is regained quickly (with respect to heading). The aircraft heading takes longer to recover mainly because of the lack of damping (reduced autonomous decay in the yaw-rate mode relative to heave and pitch) in that axis and the coupling between yaw control and pitching motion as explained in 17, 18. This coupling can be explained by a rank deficiency in the control sensitivity matrix B (in the linearized system local to hover) when rotor 4 is completely deactivated. Altitude is recovered during the simulation (Fig. 7), however the X and Y positions of the aircraft drift from the original location due a lack of control over these aircraft states.

Forward Flight To simulate a forward flight condition at approximately 5 m/s, a nose-down pitch attitude determined by the off-line trim routine is commanded, while altitude and other attitudes are held at zero as in hover. Again, rotor 1 is failed at 5 seconds. From Refs. 17, 18, the steady rotor speeds that trim the aircraft post failure involve rotor 4 slowing significantly but not shutting off completely as in hover, while the remaining four rotors speed up to maintain thrust, with a slight differential between aft rotor speeds and front rotor speeds to produce the commanded nose-down pitch attitude in forward flight. The actual rotor speeds in simulation are shown in Fig. 10.

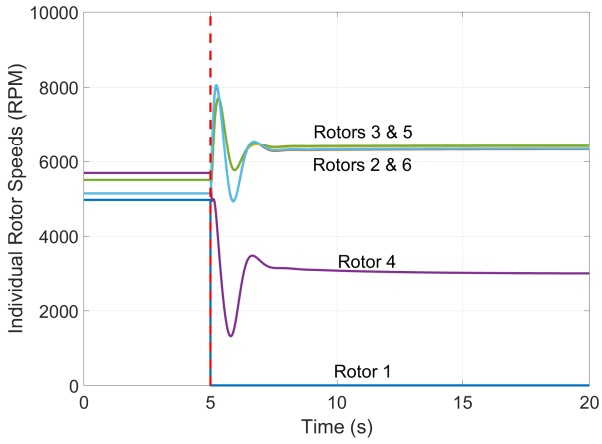

Fig. 10. Rotor Speed Time History for Rotor 1 Failure at 5 m/s (Inner Loop)

Prior to rotor failure, the steady rotor speeds can be described succinctly as a linear combination of collective and pitch RPM alone, producing only thrust and nose-down pitching moment. Post-failure, the aircraft again experiences a transient in the form of descent, pitching nose-down, and yaw nose-left (Fig. 11). Intuitively, the aircraft should respond in a similar manner as in hover, which it does except for the pitch attitude which remains non-zero as commanded.

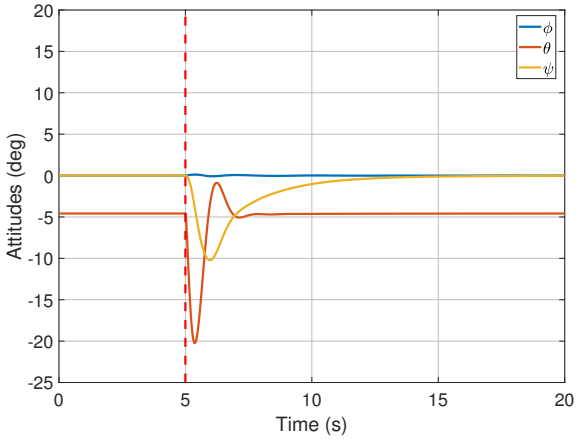

Fig. 11. Attitude Time History for Rotor 1 Failure at 5 m/s (Inner Loop)

Rotor 4 is slowed significantly to balance pitching moments, but does not turn off completely as in hover. The operation of rotor 4 allows the aircraft more authority in yaw than it had previously (the rotor thrust is small compared to rotors 2, 3, 5, and 6 but can produce torque and side-force at a large moment arm), which results in the aircraft being capable of recovering its heading faster than it did in hover. Overall, the aircraft is able to return to its desired trajectory in less time than it did for hover. Again, the steady rotor speeds (Fig. 10) at the end of the simulation approach those previously published in Refs 17, 18.

Outer Loop

Hover Once the outer loop control laws are implemented, state reference can be defined in terms of aircraft velocities rather than attitudes. The first case considered with outer loop control was hover, where the body velocities are zero. Simultaneously, the desired altitude and heading are commanded to be zero. Again, rotor 1 failure occurs at 5 seconds.

Similar to the inner loop simulation, all of the rotors operate at the same speed up until the failure of rotor 1, at which point the speed of rotor 4 is decreased until it is at 0 RPM, while the remaining four rotors speed up to maintain thrust, settling out at approximately 6500 RPM (Fig. 12). The compensation is nearly identical to what was observed for hover using only inner loop control, with the exception that rotor 4 remains deactivated for the remainder of the simulation post-rotor-failure. From the previous discussion, this should allow for a pitching moment balance (recovery of 0° pitch attitude) as well as a recovery in the aircraft altitude, at the expense of aircraft heading (Fig. 13).

Clearly, the aircraft recovers its pitch and roll attitudes in a similar amount of time as the inner loop simulation, but the persistent zero speed of rotor 4 precludes a similar recovery

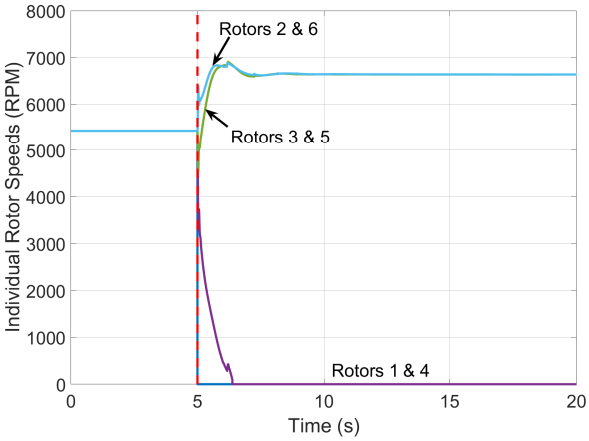

Fig. 12. Rotor Speed Time History for Rotor 1 Failure in Hover (Outer Loop)

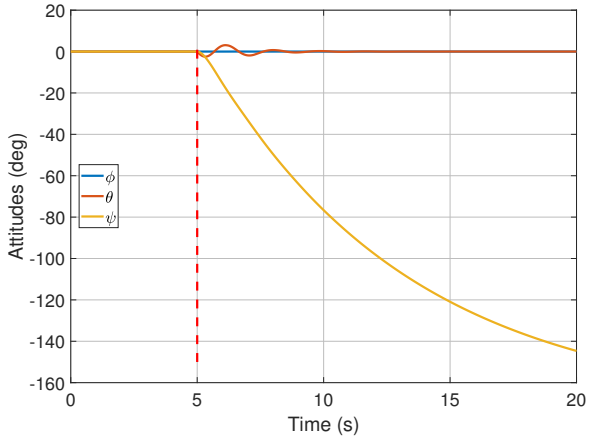

Fig. 13. Attitude Time History for Rotor 1 Failure in Hover (Outer Loop)

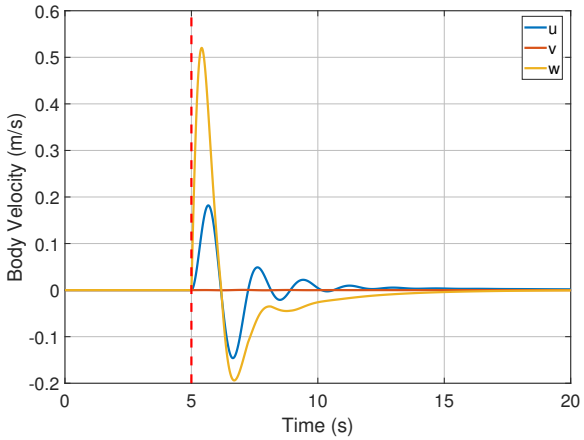

Fig. 14. Body Velocity Time History for Rotor 1 Failure in Hover (Outer Loop)

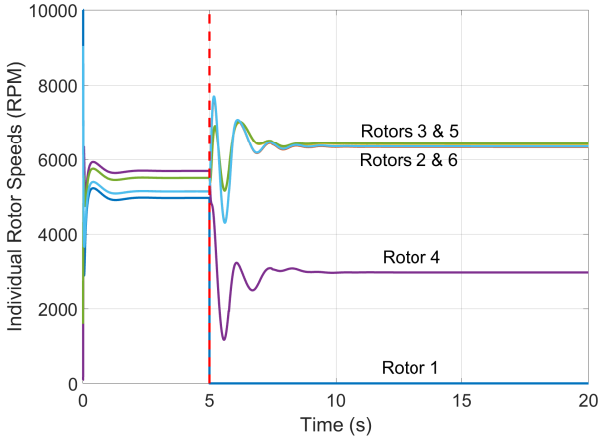

Fig. 15. Rotor Speed Time History for Rotor 1 Failure at 5 m/s (Outer Loop)

of the aircraft heading. It is noted that the aircraft does recapture a zero body velocity condition again, seen in Fig. 14, but the aircraft is continuing to yaw 15 seconds post-rotor-failure. Interestingly, relaxing control of aircraft heading is a suggested strategy to retain control of the aircraft in Refs. 4 and 19.

Forward Flight Next, forward flight was simulated at 5 m/s with the outer loop control laws. A fully operational hexacopter trims in this flight condition using a combination of collective and pitch RPM (Ω_0 and Ω_P), which results in aft rotors spinning faster than front rotors to produce a nose-down pitching moment required to overcome drag and rotor hub moments in forward flight. This is demonstrated in Fig. 15 by the rotor speeds prior to the vertical dashed line.

Post-rotor-failure, the rotor speeds approach the values that were observed from the inner loop controller previously in hover. The rear rotor (rotor 4) slows down, but not to zero speed, allowing the aircraft to retain 5 independent controls and consequently retain the ability to reconcile the aircraft heading back to the desired value (Fig. 16). The remaining rotor speeds are again increased to maintain thrust, and are used in the transient recovery period along with rotor 4 to recover the nose-down pitch attitude required in forward flight, which is now determined by the outer loop control laws rather than being commanded directly.

The commanded signals in this simulation are given in terms of the body velocities, altitude, and heading for the hexacopter. From the inner loop simulations previously discussed, it is clear that the aircraft is able to recover altitude and heading trajectories post-rotor-failure, Fig. 17 demonstrates the controller's ability to recover and track commanded velocity values as well as attitude commands. The simulation was designed for a 5 m/s inertial X-velocity (V_x), which translates into body velocities defined by $u = V_x \cos\theta$ and $w = V_x \sin\theta$ where θ is the pitch attitude of the aircraft, nose-down in forward flight.

Loiter A possible idea to complete a mission task is a low speed circling of a target area, or a loitering maneuver. From previous results, it should follow that the aircraft could maintain a slow circling maneuver post-rotor failure and be able to

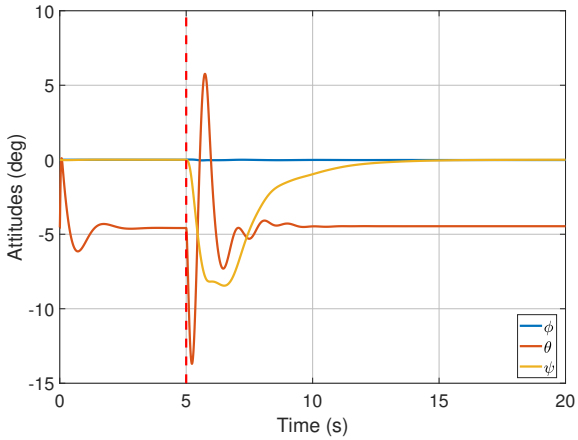

Fig. 16. Attitude Time History for Rotor 1 Failure at 5 m/s (Outer Loop)

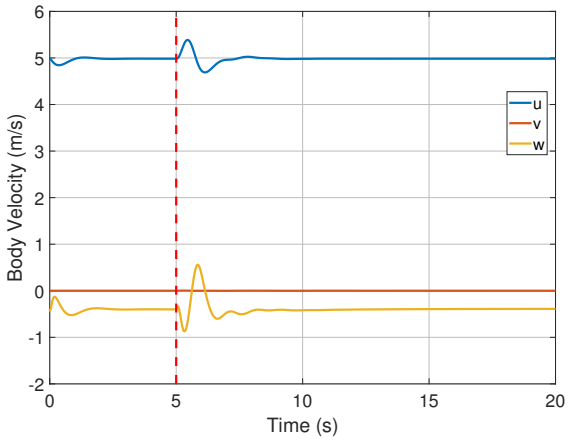

Fig. 17. Body Velocity Time History for Rotor 1 Failure at 5 m/s (Outer Loop)

complete a mission. A trajectory is drawn to circle the aircraft around the origin at a radius of 2 meters and a forward flight speed of 2 m/s, rotor 1 is failed at 5 seconds.

Loitering involves a low speed forward flight as well as a constant change in heading to follow a circular trajectory. This can be achieved through a combination of collective, pitch, and yaw RPM (Ω_0, Ω_P, and Ω_Y). Prior to rotor failure, the individual rotor speeds are similar to previous cases of fully operational forward flight, except that the strict pattern of rotor speed increasing from front to aft of the aircraft is no longer present. In general, aft rotors still operate at a higher speed than front rotors, but the ordering among the 3 aft and 3 front rotors differs from cruise. This is a consequence of commanded constant change in yaw as the trajectory evolves in time, which requires the yaw RPM (Ω_Y) input, speeding up clockwise-spinning rotors and slowing down counter-clockwise-spinning rotors. Post-failure, the rotor speeds again approach those seen for other forward flight conditions using both the inner and outer loop control laws (Fig. 18). Again, the rear rotor (rotor 4) is slowed dramatically compared to the remaining four rotors to compensate for the loss of rotor

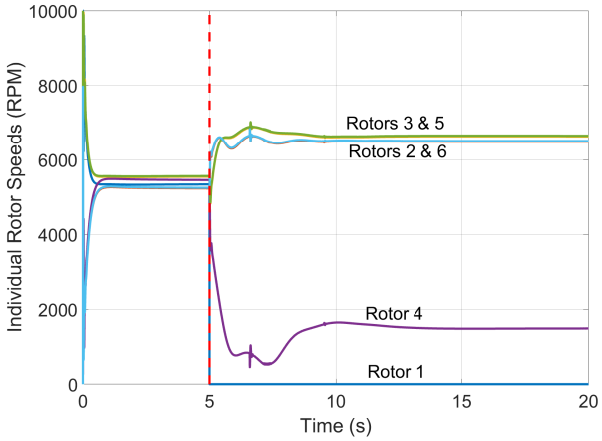

Fig. 18. Rotor Speed Time History for Rotor 1 Failure during Loiter (Outer Loop)

1 thrust, but also provides a nose-down pitching moment to maintain forward flight. The remaining rotors again speed up to maintain thrust, but there is a small differential between the rotors that is different from previous simulation in order to maintain the yaw rate that defines the circular trajectory.

Figure 19 depicts the time history of the body velocities for the loiter simulation.

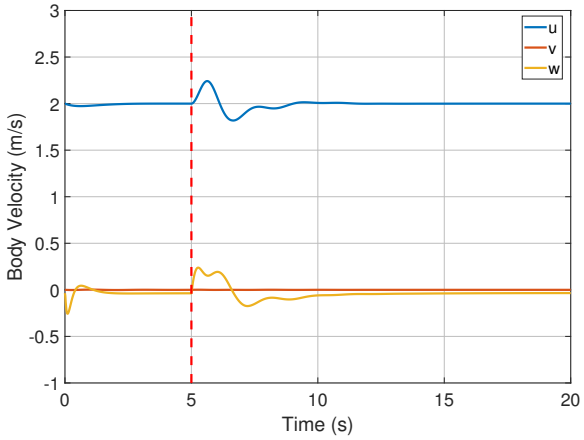

Fig. 19. Body Velocity Time History for Rotor 1 Failure during Loiter (Outer Loop)

Because the circular trajectory is tracked using only yaw, the aircraft experiences no sideward body velocity ($v \equiv 0$). As a result, the time history of the body velocity for the aircraft appears similar to that for the aircraft holding 5 m/s forward flight with outer loop control, shown previously in Fig. 17.

The path of the hexacopter through space can be plotted to garner additional understanding of the simulation results. This is given in Fig. 20, where the red "X" denotes the position at which the front rotor fails.

After completing the majority of one revolution, the front rotor fails, denoted by the red "X". After this point, the aircraft descends rapidly due to loss of thrust and a sudden increase

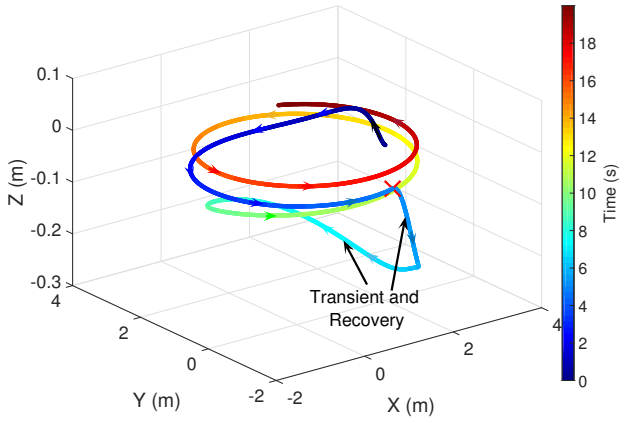

Fig. 20. Position Time History for Rotor 1 Failure during Loiter (Outer Loop)

in the nose-down pitch attitude. As seen in Fig. 18, rotor 4 is slowed at this point, while the remaining rotors speed up. This leads to the trajectory leveling out and eventually climbing back up to the desired altitude coincident with the X-Y plane. It should be noted that the final circles drawn by the aircraft position do not lie coincident with the original (orange-red circle vs blue circle in Fig. 20), although they exist at the same altitude. This is because the controller tracks a commanded velocity signal rather than position. Had the outer loop been implemented using a desired position input, the trajectory would have returned to the original circle. However, the new center of rotation is shifted away from (0,0) where it was originally to approximately (1,1) in the X-Y plane (Fig. 21). Note that the transient portion of the trajectory is excluded from this figure as it occurs below the X-Y plane, again the red "X" denotes the location at which rotor one fails in simulation.

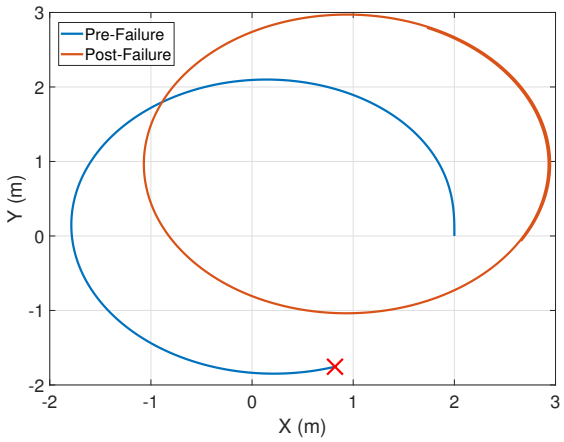

Fig. 21. X-Y Trajectories for Rotor 1 Failure during Loiter

Landing Options To this point, simulation cases have explored the ability of the baseline feedback controller to continue a basic command input, be it hover, forward flight, or a slightly more complex maneuver. An important feature of a controller that allows for rotor failure would also be the ability to recover the aircraft by landing in some manner that does not result in severe damage or catastrophic loss. Of a number of possible landing maneuvers, in the present study two options are considered: one in which the hexacopter decelerates from a forward flight condition and descends in such a way that the aircraft arrives at the ground with zero velocity (commanded), and another where the aircraft slows from forward flight to a momentary hover above a landing area, then descends straight down to the ground.

In both cases, the aircraft is commanded to track a velocity trajectory that involves an acceleration to a forward speed of 6 m/s with a climb to an altitude of 5 meters, cruise, and deceleration with descent. The front rotor is failed at 5 seconds as in the previous cases. Successful tracking of the trajectory as it is drawn would demonstrate a possible landing maneuver for the aircraft post-rotor failure. The general shape of the trajectory is depicted in Fig. 22.

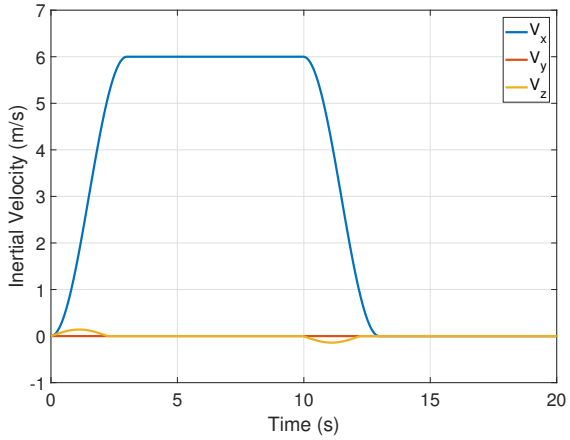

Fig. 22. Reference Velocity Trajectory for Landing Methods

Note that this velocity trajectory results in the aircraft accelerating to 6 m/s ($\approx V_{BE}$), cruising, then decelerating to stop 60 meters ahead of its starting position (in the $+X$ direction). Simultaneously, the aircraft is commanded to climb to 5 meters for the cruise portion of the trajectory, then descend to the ground coincident with the point where the aircraft reaches zero forward speed. Landing occurs at 12 seconds.

When the aircraft tracks this trajectory, the command signal follows Fig. 23. Again, the trends for rotor speeds of a hexacopter in forward flight are exhibited, where aft rotors spin faster than forward rotors. For the first portion of the trajectory (from 0 to 2 seconds), the aircraft is climbing, which is accomplished by speeding up all of the rotors simultaneously, followed by a decrease in collective RPM back to a steady value once the aircraft has achieved its cruising altitude. Figure 24 shows that the aircraft achieves a steady level flight condition between 3 to 4 seconds into the simulation.

Because rotor 1 fails in steady level flight in a forward flight condition, the rotor speed history (Fig. 23) is nearly identi-

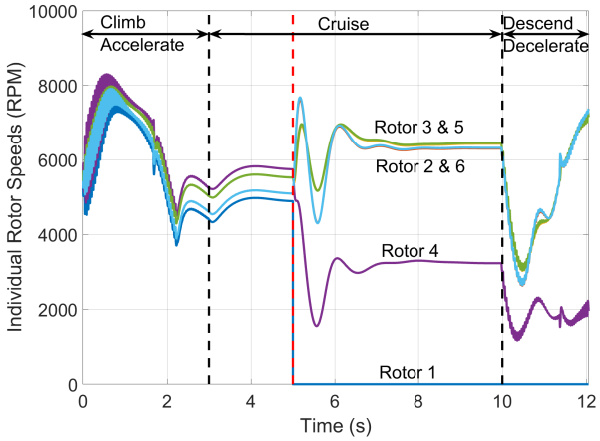

Fig. 23. Rotor Speed Time History for Rotor 1 Failure in Landing Simulation (Outer Loop)

cal to the rotor speeds given in Fig. 15. There are slight differences in the actual value of the rotor speeds because the present simulation occurs at 6 m/s rather than 5 m/s.

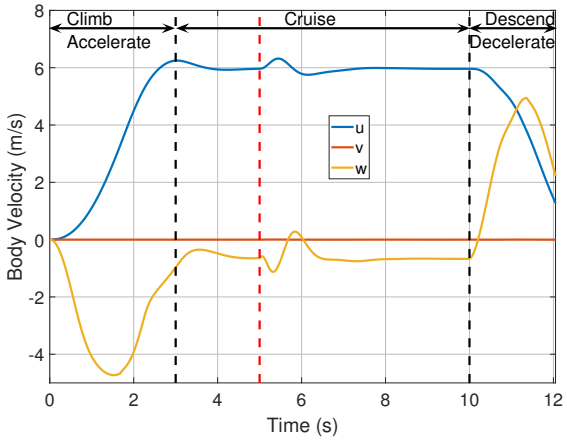

Fig. 24. Body Velocity Time History for Rotor 1 Failure in Landing Simulation (Outer Loop)

After the aircraft recovers the velocity profile, it must decelerate and descend. In the first portion of the deceleration and descent, the rotor speed history is effectively the opposite of the acceleration and climb phase. All rotor speeds decrease to allows the aircraft to descend from its cruise altitude, while rotors on the aft portion of the hexacopter slow and forward rotors speed up to pitch the aircraft nose-up and halt its forward speed (Fig. 25 gives the attitude time history).

An interesting observation from Fig. 25 is that the aircraft begins to yaw as the forward flight speed decreases as it did for the zero velocity hold case shown previously. This can again be attributed to rotor 4 being slowed down all the way to zero speed, at which point the aircraft loses independent control of pitch and yaw attitude (Refs. 17, 18). Once rotor 4 has lost its effectiveness, the only device being used to slow the aircraft yaw rate is the aerodynamic damping coming from the

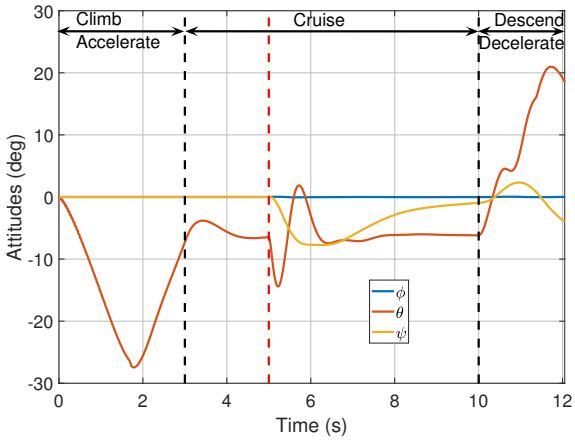

Fig. 25. Attitude Time History for Rotor 1 Failure in Landing Simulation (Outer Loop)

rotor and airframe drag. Because this damping is small (compared to damping in roll and pitch), the heading of the aircraft begins to rotate once the hexacopter has slowed its forward speed. It should be noted that the yaw rate is decreasing as the simulation continues and would eventually come to rest, this is indicated by the positive curvature of the yellow curve in Fig. 25.

If the position of the aircraft is plotted in 3-dimensional space as it was for the loitering case, the time history is given by Fig. 26.

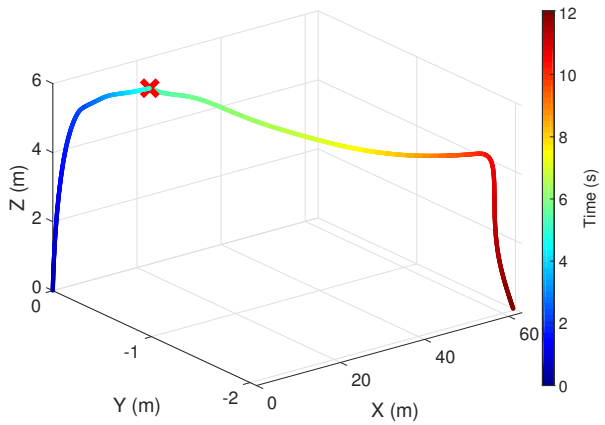

Fig. 26. Position Time History for Rotor 1 Failure in Landing Simulation (Outer Loop)

Again, the location at which rotor one fails is denoted by the red "X" and the aircraft moves from its initial position at the origin to its landing position at (63 m, -2 m) in the X-Y plane. It is clear from the figure that the aircraft follows a smooth path from start to finish, with a slight deviation from course due to the rotor failure causing the Y-position of the landing to be non-zero. Following this trajectory, the aircraft lands with a vertical speed of approximately 1.7 m/s and a nose-up pitch attitude of 18.5°. This vertical speed could be reduced

by changing the commanded altitude trajectory accordingly, the present trajectory is more aggressive than is necessary.

The second landing option involves shifting the descent period of the trajectory after the end of the aircraft deceleration from cruise. The rotor speed history for this case is given in Fig. 27. From 0 to 10 seconds, the rotor speeds are identical to the previous case. At 10 seconds, the aircraft is commanded to slow down to a hover, which is achieved by inputting a pitch RPM command and pitching the aircraft nose-up to slow down. Once the aircraft reaches zero velocity, there is an artificial spike in rotor speeds that arises from gain scheduling in the controller itself. This spike has no real impact in the state trajectories, depicted for the aircraft velocities in Fig. 28 and attitudes in Fig. 29

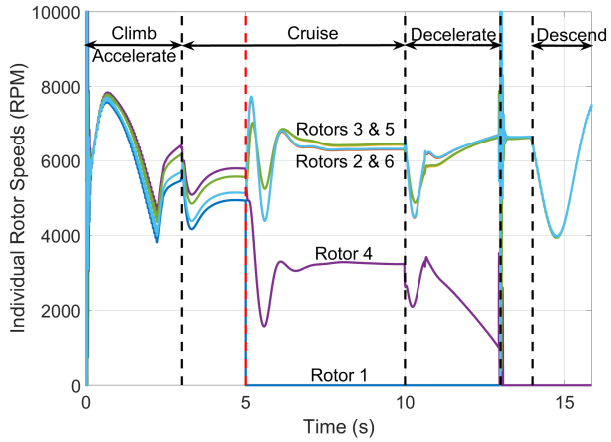

Fig. 27. Rotor Speed Time History for Rotor 1 Failure (Outer Loop)

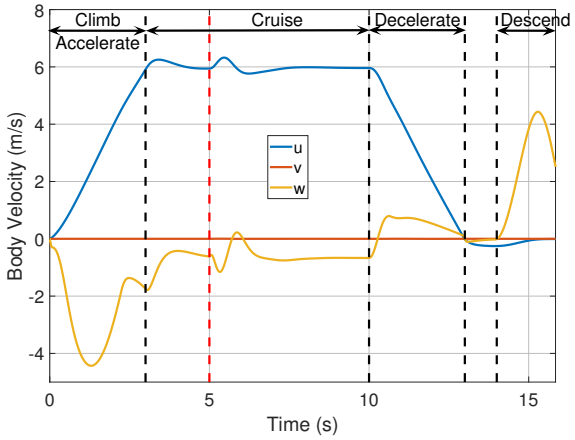

Fig. 28. Body Velocity Time History for Rotor 1 Failure (Outer Loop)

Figure 28 highlights the shift in the aircraft trajectory from a simulataneous deceleration and landing to two separate phases in the simulation. Again, through 10 seconds, the simulation appears identical to that which was previously shown. Between 10 and 14 seconds, the aircraft slows to a stationary hover. Then, from 14 seconds to approximately 16 seconds, the aircraft descends to land (touching down at 15.8 seconds). In this case, the aircraft lands with a vertical speed of approximately 2.5 m/s, which again could be reduced with the appropriate changes in the commanded trajectory.

Again, the aircraft begins to spin (Fig. 29) once its forward speed is reduced because of the need to reduce the speed of rotor 4 (Fig. 27) to maintain pitching moment balance. In the present simulation, the hexacopter would be yawing at more or less a constant rate as it descends from its cruise altitude down to landing.

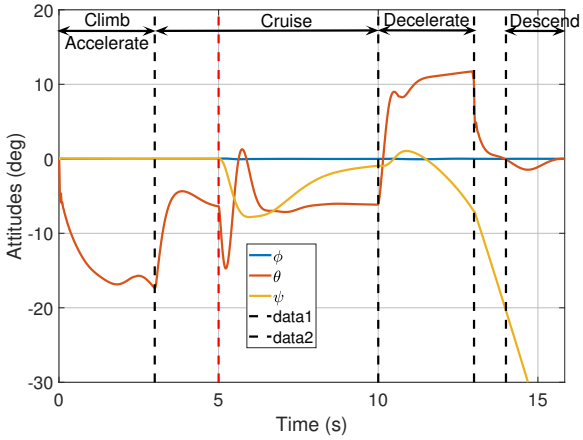

Fig. 29. Attitude Time History for Rotor 1 Failure (Outer Loop)

Finally, the position of the aircraft in space is presented in Fig. 30.

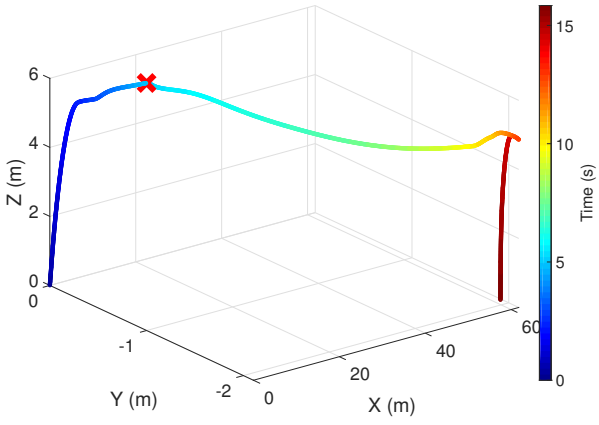

Fig. 30. Position Time History for Rotor 1 Failure at 5 m/s (Outer Loop)

Similar to Fig. 26, there is a slight deviation of the Y position of the aircraft due to the change in heading caused by the failure of rotor one. The aircraft in this simulation lands at (61.6 m, -1.9 m). A clear distinction can be made between

these two simulations in the descent portion of the flight, in Fig. 26 the aircraft is translating in the X direction as it descends and lands with a significant nose-up attitude, whereas in Fig. 30 there is virtually no motion in the X-Y plane while the aircraft descends.

Both landing cases demonstrate the ability of the aircraft to perform a controlled descent after the front rotor fails. Although the aircraft may be rotating away from the commanded heading, it is able to approach the ground and land near a desired location for recovery.

CONCLUSIONS

A feedback controller is designed and implemented for hexacopter in a fully operational condition. The controller is designed by approximating the system as a set of parallel SISO systems in altitude, roll, pitch, and yaw, with feedback input defined by the set of primary multirotor controls $(\Omega_0, \Omega_R, \Omega_P, \Omega_Y)$. An outer control loop is also implemented for control over aircraft body velocities.

Controller performance is examined for various trajectories in the event of rotor failure. These trajectories include holding hover, forward flight, a loitering maneuver, as well as options for landing and aircraft recovery. When the rotor fails, the feedback controller remains identical to the baseline case, only the plant dynamics are impacted by the rotor failure itself. In the dynamic simulation, a rotor failure is simulated by setting the force and moment output of the affected rotor to zero regardless of the commanded control input to that rotor.

The controller is demonstrated to recover desired state values post-rotor-failure following the trajectories as previously outlined. When a rotor fails, the multirotor controls as defined for the fully operational aircraft no longer produce only single axis thrust and moments, that is the controls are now coupled. However, the multirotor controls as they are defined still mostly produce single-axis force and moment, and the off-axis response can be mitigated by use of the other multirotor controls, allowing the feedback controller to perform and track reference trajectories post-rotor-failure. Successful tracking of these trajectories demonstrates tolerance to rotor failure for a baseline feedback controller designed using multirotor controls as defined for a fully operational hexacopter.

Author contact:

Michael McKay	mckaym2@rpi.edu
Robert Niemiec	niemir@rpi.edu
Farhan Gandhi	fgandhi@rpi.edu

ACKNOWLEDGMENTS

The authors would like to acknowledge the Department of Defense and the American Society of Engineering Education, for funding Michael McKay and Robert Niemiec through the National Defense Science and Engineering Graduate Fellowship.

REFERENCES

[1]A. Marks, J. Whidborne, and I. Yamamoto, "Control Allocation for Fault Tolerant Control of a VTOL Octrotor," in *UKACC International Conference on Control, Cardiff, UK*, UKACC, Sept. 2012.

[2]G. Falconi, C. Heise, and F. Holzapfel, "Novel Stability Analysis of Direct MRAC with Redundant Inputs," in *24th Mediterranean Conference on Control and Automation, Athens, Greece*, IEEE, June 2016.

[3]G. Falconi and F. Holzapfel, "Adaptive Fault Tolerant Control Allocation for a Hexacopter system," in *American Control Conference, Boston, MA*, AACC, July 2016.

[4]G.-X. Du, Q. Quan, and K.-Y. Cai, "Controllability Analysis and Degraded Control for a Class of Hexacopters Subject to Rotor Failures," *Journal of Intelligent and Robotic Systems*, vol. 78, pp. 143–157, Sept. 2015.

[5]M. Achtelik, K.-M. Doth, D. Gurdan, and J. Stumpf, "Design of a Multi Rotor MAV with regard to Efficiency, Dynamics, and Redundancy," in *AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN*, AIAA, Aug. 2012.

[6]G.-X. Du, Q. Quan, and K.-Y. Cai, "Additive-State-Decomposition-Based Dynamic Inversion Stabilized Control of a Hexacopter Subject to Unknown Propeller Damages," in *32nd Chinese Control Conference, Xi'an, China*, July 2013.

[7]T. Schneider, G. Ducard, K. Rudin, and P. Strupler, "Fault-Tolerant Control Allocation for Multirotor Helicopters using Parametric Programming," in *Int. Micro Air Vehicle Conference and Flight Competition, Braunschweig, Germany*, July 2012.

[8]I. Sadeghzadeh, A. Mehta, Y. Zhang, and C.-A. Rabbath, "Fault-Tolerant Trajectory Tracking Control of a Quadrotor Helicopter Using Gain Scheduled PID and Model Reference Adaptive Control," in *Annual Conference of the Prognostics and Health Management Society, Montreal, Quebec, Canada*, Sept. 2011.

[9]M. Oppenheimer, D. Doman, and M. Bolender, "Control Allocation for Over-Actuated Systems," in *14th Mediterranean Conference on Control and Automation, Ancona, Italy*, IEEE, June 2006.

[10]T. Johansen and T. Fossen, "Control Allocation - A Survey," *Automatica*, vol. 49, pp. 1087–1103, Mar. 2013.

[11]J. Tjonnas and T. Johansen, "Adaptive Control Allocation," *Automatica*, vol. 44, pp. 2754–2765, Mar. 2008.

[12]X. Qi, D. Theilliol, J. Qi, Y. Zhang, J. Han, D. Song, L. Wang, and Y. Xia, "Fault Diagnosis and Fault Tolerant Control Methods for Manned and Unmanned Helicopters: A Literature Review," in *Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France*, Oct. 2013.

[13]G. Ducard, *Fault-Tolerant Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. phdthesis, ETH Zurich, 2007. No. 17505.

[14]R. Niemiec and F. Gandhi, "Effects of Inflow Model on Simulated Aeromechanics of a Quadrotor Helicopter," in *American Helicopter Society 72nd Annual Forum, West Palm Beach, FL*, AHS, May 2016.

[15]D. Peters and C. He, "A Finite-State Induced Flow Model for Rotors in Hover and Forward Flight," in *American Helicopter Society 43rd Annual Forum, St. Louis, MO*, AHS, May 1987.

[16]R. Niemiec and F. Gandhi, "Multi-Rotor Coordinate Transforms for Orthogonal Primary and Redundant Control Modes for Regular Hexacopters and Octocopters," in *42nd Annual European Rotorcraft Forum, Lille, France*, AHS, Sept. 2016.

[17]M. McKay, R. Niemiec, and F. Gandhi, "Control Reconfiguration for a Hexacopter Experiencing Single Rotor Failure," in *27th International Conference on Adaptive Structures and Technologies, Lake George, NY*, ICAST, Oct. 2016.

[18]M. McKay, R. Niemiec, and F. Gandhi, "An Analysis of Classical and Alternate Hexacopter Configurations with Single Rotor Failure," in *American Helicopter Society 73rd Annual Form, Fort Worth, TX*, AHS, May 2017.

[19]M. Mueller and R. D'Andrea, "Relaxed Hover Solutions for Multicopters: Application to Algorithmic Redundancy and Novel Vehicles," *International Journal of Robotics Research*, vol. 35, no. 8, pp. 873–889, 2016.