Feedback motion replanning during high-stakes scenario

Mohamed Khalid M Jaffar, Michael Otte
Department of Aerospace Engineering
University of Maryland, College Park

***Abstract*— This paper proposes a novel algorithm for a quadrotor to replan its motion in the event of one, two or three rotor loss. Further, during the course of its replanned trajectory, the MAV avoids collision with static obstacles including the ground.**

***Index Terms*— Quadrotor, Optimal control, Dynamic trajectory planning, Obstacle avoidance, Sampling-based motion planning**

I. INTRODUCTION

Autonomous aerial robots are increasingly used for risky outdoor applications such as firefighting, search and rescue, surveillance in uncharted territories, etc. An event of structural damage or addition of slung load is highly likely in such scenarios. This would result in an unanticipated change in the dynamics of the system and hence, require the quadcopter to revise its motion planning strategies in order to complete the mission statement.

This project addresses replanning in the specific case of single, two opposing, or three rotor loss. Given that the micro aerial vehicle (MAV) identifies the type of failure correctly, it has to replan its motion *'on-the-fly'*. We refer this problem as 'high-stakes motion replanning'. The overall goal of this research is to develop and characterize general techniques that can be applied to any aerial vehicle that must relearn how to move in a high-stakes scenario. In this research, we plan to extend ideas from motion replanning by incorporating ideas from system identification, numerical optimal control and Monte Carlo tree search.

Specifically, the MAV's mission statement is to fly along a predefined trajectory while avoiding static or dynamic obstacles, which is perceived only when the aerial robot is sufficiently close enough to the obstacle. These maneuvers have to be performed even in the event of rotor loss. We plan to use a quadrotor as the test platform. Quadrotors are an ideal platform for this research because they have nontrivial dynamics and control, VTOL capabilities, and also are easier to assemble and widely popular.

Such a mission statement might find relevance in warehouse management and delivery logistics. In warehouses, an aerial robot taking stock of the inventory in a pre-defined path has to avoid obstacles in the form of manual forklifts or racks being moved by ground robots. In the event of an unprecedented damage, it has to fly back to the control station while still avoiding the dynamic obstacles. In the outdoor scenario of aerial package deliveries, robot's dynamics might be altered due to the change in the distribution of the payload, and its major obstacles are buildings, trees and poles.

Corresponding author's email: khalid26@umd.edu

Paper Organisation: Section II states the related work and the methodology is outlined in section III. The remainder of the paper is organised as follows, Section V describes the modelling of the dynamics and open loop response of the quadrotor. Section VI states the control law during normal operation and in case of rotor loss. Section VII proposes the motion planning strategy including collision avoidance, and an optimal control policy to achieve it. Finally, results are discussed in Section VIII and Section IX states our conclusions and scope for future work.

II. RELATED WORK

In the relevant control literature, strategies in the event of propeller loss have been proposed and validated [1][2]. Such controllers are designed to stabilize about a stable equilibrium and have a cascaded structure comprising of separate position and attitude controllers, which might not be the best solution in the case of executing aggressive maneuvers. We would like to incorporate optimal strategies for full-state control at all instants. Also, in the event of rotor loss, the system dynamics abruptly changes due to the increase in non-linearities and dominance of higher order dynamics as discussed in [3].

[4] proposes an elegant way of constructing random search trees based on region of attraction of local trajectory controllers. The constructed tree in the form of "funnels" ultimately causes the system to go from start to goal. An LQR controller is synthesised for tracking a time-optimal trajectory. Further, the performance analysis and region of attraction are constructed using Lyapunov functions. This project aims to extend the results to higher order and fast dynamic systems.

Extensive research in sampling-based motion planning for dynamic environments [5][6] has been carried out recently. We intend to extend notions from such dynamic planning algorithms to the case of internal change in vehicle dynamics in addition to those in the environment. Related work in the intersection of control and planning [4][7] often implement a non-linear feedback controller synthesis and optimisation of a cost/value function.

With increasing on-board computational capabilities, numerical approaches to optimal control of quadrotors have become popular [8]. Unlike analytical controllers, these adopt a direct multiple shooting approach [9] and solve the complex problem by using an off-the-shelf optimization solver. Such

controllers use recursive quadratic programming technique to significantly improve the convergence behaviour and reduce the computing time.

This work will incorporate ideas from sampling-based dynamic motion planning to construct feasible time-optimal trajectories and implement numerical optimal control algorithms to track these trajectories. The control strategies will change according to the vehicle state i.e. one, two, three or four operational rotors. The sampling would be based on the feedback provided by the controller about its own capabilities and performance. Lastly, an appropriate collision avoidance framework [10] would be implemented to avoid obstacles in the workspace.

III. METHODOLOGY

This work is aptly divided into controller synthesis and motion planner design. The algorithm generates a random exploring tree based on the possible values of the available rotor thrust inputs. The dynamics are modelled considering the quadrotor system and the actuator system separately, the former modelled using Newton-Euler approach [11][12]. The actuator dynamics are modelled based on test-bench studies. In the event of rotor loss, some of the branches cease to exist and hence, the search tree has to be repaired and the motion needs to be replanned. The same is true for dynamic obstacles as discussed in [5].

The developed algorithms will be first tested and analysed on a suitable simulation platform. Initial open-loop simulations will be run in MATLAB/Simulink owing to the ease of implementation. Later, we intend to use DRAKE [13] library in C++ for simulating the complex non-linear quadcopter dynamics. MCTS and motion planning will be implemented through efficient C++ codes. Using C++ throughout will help to prototype and test ideas in a safe sand-box, yet provide quick migration from simulation to the real vehicle.

The quadrotor test platform comprises of an off-the-shelf frame and motors, NAVIO autopilot by Emlid® and Raspberry Pi as the computational resource. Software will be implemented using ROS and C++ which will enable us to use open source code for software system components that we will not be modifying, e.g., sensor interfaces, localization filters, etc. Vehicle state (position and orientation) feedback would be obtained from the VICON® motion capture system installed at Dr. Otte's Laboratory and broadcast to the quadrotor using a wireless router.

IV. PROBLEM FORMULATION

The problem is divided into path-planning and controls sub-problems, formulated as follows,

A. Path-Planning Problem

The robot exists in the workpsace, W and motion is planned in the configuration space, C. Additionally, we have a collision checker, $f_{cc} : C \rightarrow [true, false]$. C-free space is defined as $X_{free} = \{x \in C : f_{cc}(x) = false\}$. Given X_{start}, $X_{goal} \in X_{free}$, determine a feasible path, $\sigma : [0, 1] \rightarrow X_{free}$. The solution path would be concatenation of edges,

$$\sigma = \sigma_1 \oplus \sigma_2 \oplus .... \oplus \sigma_n$$

where, edges, σ_i are characterised by nodes (v_{i-1}, v_i)

B. Control Problem

Given the state space, S and the control space, U, dynamics model, $\dot{s}(t) = f(s(t), u(t))$, along with the system parameters such as inertia, mass and thrust coefficients, determine a control policy, $\pi : S \rightarrow U$ such that, $s(0) = v_{i-1}$ and $s(T) = v'_i$, Where, $v'_i \in B_\delta(v_i)$ and T is the constrained finite horizon. The δ-ball is defined as, $B_\delta(v) = \{x \in S : ||x - v|| < \delta\}$

V. DYNAMICS MODELLING

The degrees of freedom of a quadrotor along with body frame and earth frame are illustrated in Fig. 1.

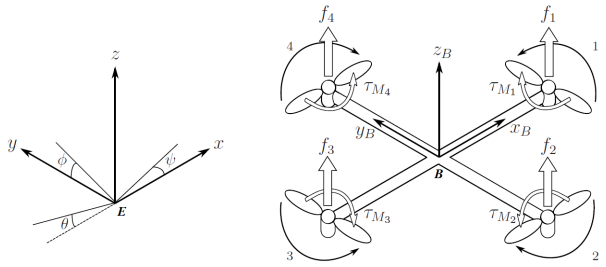

Fig. 1. Quadcopter Model [14] and reference frames

A. Equations of motion

The dynamic model of the quadrotor are derived using Newton-Euler formulation [11], [12]. Dynamics of any rigid body under the influence of external forces, $\boldsymbol{F}$ and moments, $\boldsymbol{\tau}$ expressed in a rotating frame of reference (body axes) is,

$$\begin{bmatrix} mI_{3\times3} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega} \times m\boldsymbol{v} \\ \boldsymbol{\omega} \times J\boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{\tau} \end{bmatrix} \tag{1}$$

Where, m is the mass, J the inertia matrix, $\boldsymbol{v}$ the body linear velocity and $\boldsymbol{\omega}$ is body angular rates. In the case of quadrotors, (1) can be reformulated as,

$$\begin{aligned} &\dot{\boldsymbol{\xi}} = \boldsymbol{v} \\ &m\dot{\boldsymbol{v}} = -mg\boldsymbol{e_3} + \boldsymbol{F_d} + \mathbf{R}\boldsymbol{e_3}T \\ &\dot{\boldsymbol{\eta}} = W_\eta \boldsymbol{\omega} \\ &J\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times J\boldsymbol{\omega} - J_r(\boldsymbol{\omega} \times \boldsymbol{e_3})\Omega + \boldsymbol{\tau_d} + \boldsymbol{M} \end{aligned} \tag{2}$$

Where, $\boldsymbol{F_d}$ and $\boldsymbol{\tau_d}$ are the drag forces and moments, respectively. $\mathbf{R} \in SO(3)$ is the rotation matrix from the body frame to earth frame, and W_η is the transformation matrix for angular velocities [11]. $\boldsymbol{e_3} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and g is acceleration due to gravity. Thrust, T and Moment, $\boldsymbol{M}$ are defined as,

$$\begin{aligned} T &= k \sum_{i=1}^{4} \Omega_i^2 \\ \boldsymbol{M} &= \begin{bmatrix} M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} = \begin{bmatrix} kl(\Omega_4^2 - \Omega_2^2) \\ kl(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{bmatrix} \end{aligned} \tag{3}$$

where, J_r is rotor inertia, Ω_i the rotor speed, and Ω is the net rotor speed. k is the thrust co-efficient, d the counter-moment drag co-efficient and l is the arm length. Simplifying (2) and further assuming that the aerodynamic drag is not predominant in rolling and pitching motion, the rotational dynamics can be expressed as,

$$\begin{aligned}\dot{p} &= \left(\frac{J_{yy}-J_{zz}}{J_{xx}}\right)qr - \frac{J_r}{J_{xx}}q\Omega + \frac{1}{J_{xx}}M_\phi \\ \dot{q} &= \left(\frac{J_{zz}-J_{xx}}{J_{yy}}\right)rp + \frac{J_r}{J_{yy}}p\Omega + \frac{1}{J_{yy}}M_\theta \\ \dot{r} &= \left(\frac{J_{xx}-J_{yy}}{J_{zz}}\right)pq - \gamma r + \frac{1}{J_{zz}}M_\psi\end{aligned} \tag{4}$$

$\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is body angular rates and $\gamma > 0$ is the aerodynamic drag opposing the yawing motion of the quadrotor. Consequently, dynamics of the position in the inertial frame, including the velocity drag term, b are,

$$\begin{aligned}\ddot{x_E} &= -b_x\dot{x_E} + \frac{1}{m}(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)T \\ \ddot{y_E} &= -b_y\dot{y_E} + \frac{1}{m}(\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)T \\ \ddot{z_E} &= -b_z\dot{z_E} - g + \frac{1}{m}(\cos\theta\cos\phi)T\end{aligned} \tag{5}$$

B. Open Loop Simulation

The mathematical model with the system parameters is simulated in Simulink and the open-loop behavior was studied. It is observed that quadrotors are under-actuated systems, with 6 degrees of freedom and only 4 inputs. Therefore, the rotor speed inputs are designed so as to activate each of the 4 controllable degree of freedom - z, ϕ, θ and ψ.

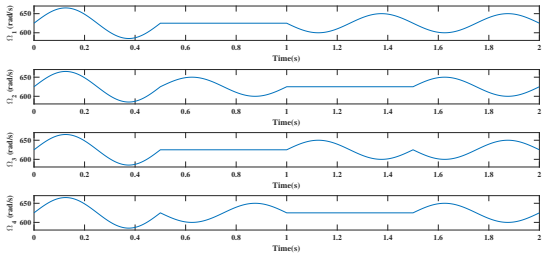

Fig. 2. Simulation: Speed input to the 4 rotors, Ω_i

The simulation progresses at steps of $0.005s$ until 2 seconds. The entire simulation time is divided into 4 intervals for each of the controllable DOF. In the first $0.5s$, all the rotor speeds are increased from the hover speed, decreased and increased back. This increases and decreases the thrust and hence, the quadrotor ascends to a higher altitude. In the next $0.5s$, a sinusoidal rolling moment is created by changing the speeds of rotors 2 and 4. Similarly, pitching and yawing moments are created by changing the corresponding rotor speeds as shown in Fig. 2.

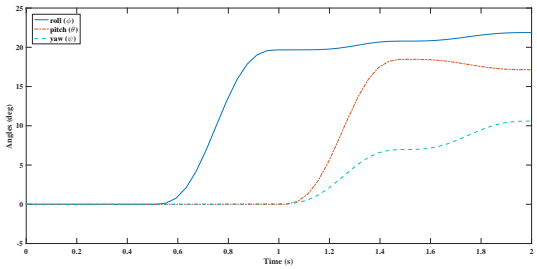

Fig. 3. Simulation: Rotational angles

The complex coupling between the rotational dynamics is observed from Fig. 3. As expected, rotations were not observed in the first $0.5s$. In the next $0.5s$, only rolling motion is observed - increasing and settling at around $20°$. Due to the pitching moment input in the next interval, the pitch angle shows similar behavior as that of roll angle. It is worthwhile to note that, due to the coupling, even rolling and yawing motions are observed during this interval. And in the final interval, even though only yawing moment was applied, the quadrotor rotates about all the three axes. Therefore, such a simple choice of inputs delineates the dynamic coupling between the rotational degrees of freedom. It is noted that the coupling is more evident when the roll and pitch angles are high due to the kinematic equations described in (4).

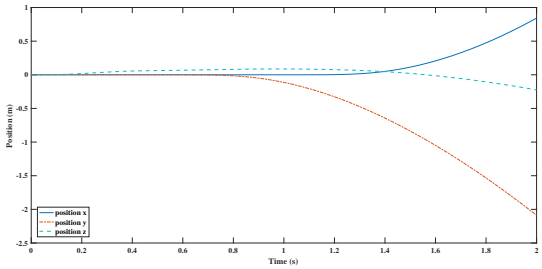

Fig. 4. Simulation: Translational position

From Fig. 4, a positive roll angle drives the quadrotor in the negative y_E axis, whereas a positive pitch drives it in positive x_E direction. After the initial $0.5s$, the total thrust of the rotors had remained close to that during hover. Thus, the deviations of the roll and pitch angles from the zero values decrease the value of the thrust in the direction of the z_E axis. Consequently, the quadcopter accelerates in the direction of the negative z_E axis and descends after the first $0.5s$.

A few simulations were run by switching off 1,2 and 3 rotors for $0.25s$ during hover. The open loop response were studied and the results have been elucidated through the following plots.

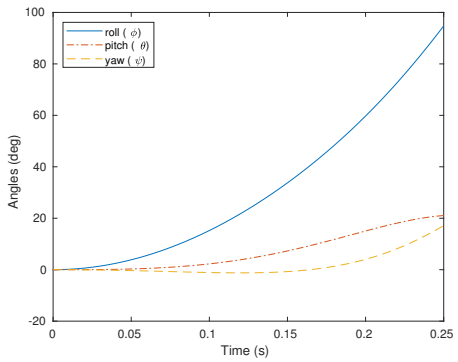

Fig. 5. Simulation: One rotor loss, $\Omega_4 = 0$

From Fig. 5, it is observed that the roll angle increases unboundedly due to the unbalanced roll moment created due to loss of rotor-4 loss. Due to the gyroscopic coupling, the pitch angle also starts to increase, but at a lower rate.

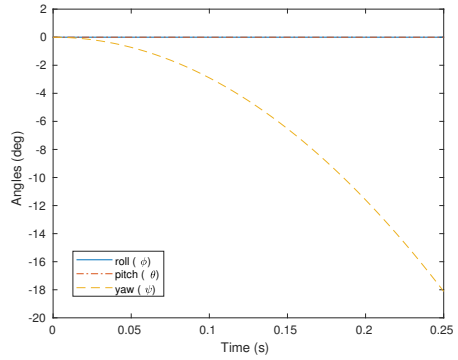

Fig. 6. Simulation: Two opposing rotor loss, $\Omega_2 = \Omega_4 = 0$

Fig. 6 and 7 illustrate the angle dynamics in case of 2 rotor loss. Due to the system dynamics, 2 adjacent and 2 opposite rotor loss will have different characteristics as corroborated in the open loop responses. In the case of 2 opposing rotor loss (rotor-2 and rotor-4), only unbalanced yawing moment exists and hence the yaw angle starts to change. However, the rate is much slower than in the first case, alluding to the fact that the yaw dynamics are inherently slower than roll and pitch.

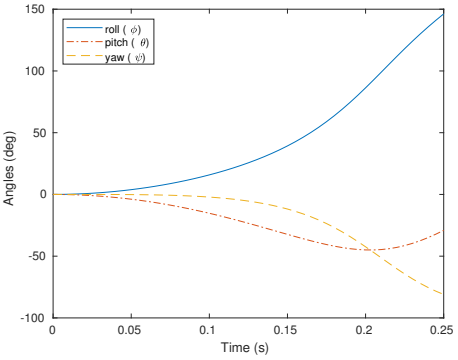

Fig. 7. Simulation: Two adjacent rotor loss, $\Omega_3 = \Omega_4 = 0$

In the scenarios of loss of 2 adjacent rotors (Fig. 7) and 3 rotors (Fig. 8), it is observed that there will be unbalanced moments along all the 3 directions and hence, roll, pitch and yaw angles deviate from the equilibrium hover case and that too at a faster rate in the 4^{th} case.

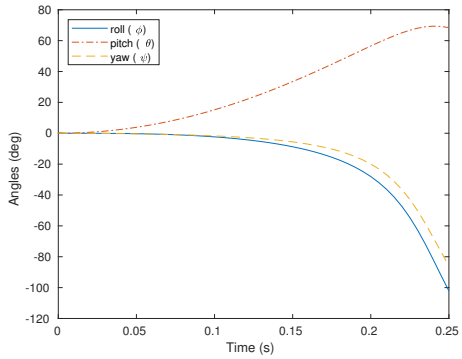

Fig. 8. Simulation: Three rotor loss, $\Omega_2 = \Omega_3 = \Omega_4 = 0$

Therefore, for this set of system parameters, it seems that an appropriate autopilot should recognize the failure and switch to an appropriate controller within 0.1-0.15s to minimize the chances of a crash.

VI. CONTROL LAW

During normal operation of 4 rotors, it is possible to control the full attitude (roll, pitch, yaw) of the quadrotor by varying the 4 rotor speeds appropriately [15][14]. The position is then controlled by varying the attitude, thus tilting the net thrust vector towards the desired position. However, when the vehicles loses one or more propellers, the rotational sub-system loses its controllability and hence, one can no longer control the full attitude of the vehicle. Therefore, the strategy adopted in this paper is to control the attitude to only one degree of freedom, a unit stationary in the inertial frame and represented by $\boldsymbol{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T$ in the body frame. The dynamics of this unit vector is given by the Euler equation,

$$\dot{\boldsymbol{n}} = -\boldsymbol{\omega}_b \times \boldsymbol{n} \tag{6}$$

The goal of the attitude controller is to maintain this unit vector stationary in the body frame, mathematically, $\dot{\boldsymbol{n}}_{des} = 0$. This unit attitude vector could be then oriented along the desired position, thus guaranteeing position control.

A. Reduced Attitude kinematics and Equilibrium solution

In the event of rotor loss, the net moments acting on the quadrotor frame will non-zero and hence, a static equilibrium does not exist. The equations of motion are solved to derive the dynamic equilibria (represented by an overbar) in the hover case.

$$\begin{aligned} \dot{\bar{\boldsymbol{\omega}}}_b &= \mathbf{0} \\ \dot{\bar{\boldsymbol{n}}} &= \mathbf{0} \end{aligned} \tag{7}$$

From (6), $\bar{\boldsymbol{n}} = \bar{\epsilon}\, \bar{\boldsymbol{\omega}}$, $\bar{\epsilon}$ defined such that,

$$||\bar{\boldsymbol{n}}|| = 1 \tag{8}$$

Enforcing the force balance equation,

$$\bar{T}\bar{n}_z = mg \tag{9}$$

In order to solve for the equilibrium condition, we have to solve 8 equations (7)-(9) to determine 11 unknowns, ($\bar{n}_x$, $\bar{n}_y$, $\bar{n}_z$), ($\bar{p}$, $\bar{q}$, $\bar{r}$), $\bar{\epsilon}$ and ($\bar{\Omega}_1$, $\bar{\Omega}_2$, $\bar{\Omega}_3$, $\bar{\Omega}_4$). For each rotor loss, there will be an addition of a constraint equation, $\Omega_i = 0$. In the case of 3 rotors loss, there will be 11 equations and 11 unknowns and hence, the equilibrium solution would be unique, if it exists. For two and one rotor loss scenarios, additionally we require 1 and 2 constraint relations, respectively.

1 rotor loss: Without any loss of generality, we assume that the 4^{th} rotor has failed. The equilibrium state is a 2-dimensional hyper-plane in the 10-D space. Hence, we have to come up with 2 constraint equations to get a equilibrium point. Intuitively, from (4), we set $\Omega_1 = \Omega_3$, and $\Omega_2 = \lambda\Omega_1$. The nonlinear equations (7)-(9) are solved using the `fsolve` routine in MATLAB. The equilibrium states as a function of the parameter, λ are studied through the following plots.

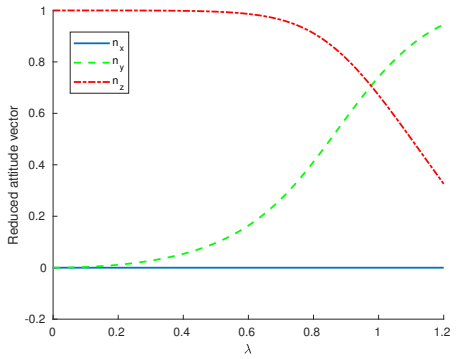

Fig. 9. Reduced attitude vector

From Fig. 10, it is observed that the angular velocity of the quadrotor is minimum at $18.60 rad/s$ for $\lambda = 0.82$. Therefore, this is chosen to be the parameter value and the corresponding equilibrium state is,

$$\begin{aligned} \bar{\mathbf{n}} &= (0, 0.45, 0.89) \\ \bar{\boldsymbol{\omega}}_b &= (0, 8.27, 16.67) \\ \bar{\Omega}_i &= (565.37, 463.61, 565.37, 0) \end{aligned} \tag{10}$$

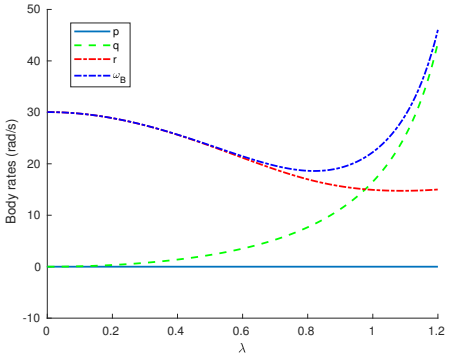

Fig. 10. Body rates

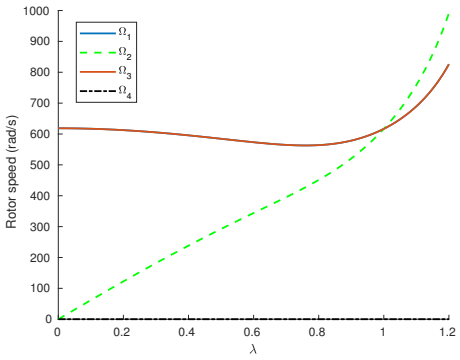

Fig. 11. Rotor speeds

The inputs were given to the system starting from the state as described in (10) and the system response was studied. From Fig. 12, we observe that the we have periodic roll and pitch motion and continuous yawing motion, indicating that we do not have control over yaw. The fact that roll and pitch are bounded gives us some hope that we could possibly control the reduced attitude of the quadrotor.

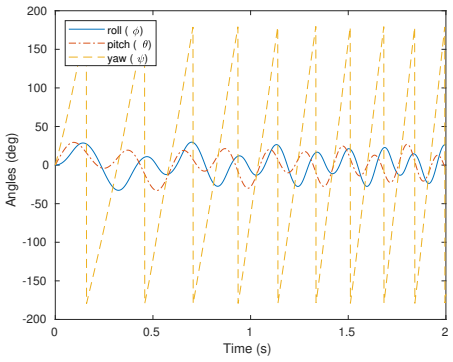

Fig. 12. Simulation: rotational angles with inputs as defined in (10)

2 rotor loss: The case of 2 opposite rotors failing is addressed first, assuming rotors 2 and 4 have failed. From (4) and (7)-(9), it is intuitive to have rotors 1 and 3 to rotate at the same speed to balance out the pitching moment. Then, the equilibrium conditions are derived to be,

$$\begin{aligned}\bar{\mathbf{n}} &= (0, 0, 1)\\ \bar{\boldsymbol{\omega}}_b &= (0, 0, 30.05)\\ \bar{\Omega}_i &= (618.55, 0, 618.55, 0)\end{aligned} \tag{11}$$

The system response to these inputs is similar to the one in Fig. 6. The case of 2 adjacent rotors failing will be considered in future work.

3 rotor loss: In this particular scenario, the number of constraint equations equal the system unknowns and hence, we have only one equilibrium solution. Assuming that only rotor-1 is functional,

$$\begin{aligned}\bar{\mathbf{n}} &= (0.41, 0, 0.91)\\ \bar{\boldsymbol{\omega}}_b &= (14.74, 0, 32.93)\\ \bar{\Omega}_i &= (915.32, 0, 0, 0)\end{aligned} \tag{12}$$

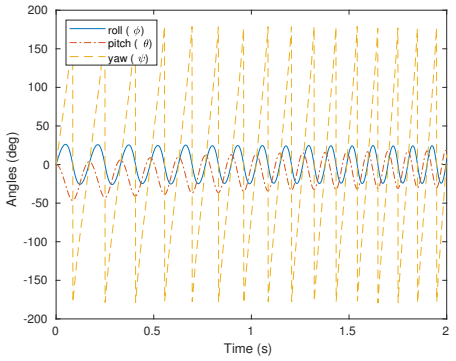

Fig. 13. Simulation: Rotational angles $\Omega_1 = 915.32$ rad/s, $\Omega_2 = \Omega_3 = \Omega_4 = 0$

Again similar trends for rotational angles are observed in Fig. 13 as in 1 rotor out case (Fig. 12). This subtly hints at controllability of the reduced attitude (roll, pitch) of the system.

All the above mentioned equilibrium are stable, verified by analysing the sign of $\dot{\mathbf{n}}$ and $\dot{\boldsymbol{\omega}}_b$ in the neighborhood of the equilibrium state. Having stated that, the next step would be to design a controller to drive the system to the desired attitude and hence position.

B. Controllability

The controllability of the system needs to be investigated before controller synthesis. This section proves that the reduced attitude vector is controllable near the equilibrium states described in the previous section. The state vector, $\boldsymbol{X}$ corresponding to the reduced attitude is $[n_x\ n_y\ n_z\ p\ q\ r]$. The state dynamics are described by the control-affine form,

$$\dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{g}(\boldsymbol{U}) \tag{13}$$

Where, $\boldsymbol{f}(\boldsymbol{X})$ and $\boldsymbol{g}(\boldsymbol{U})$ are from (4) and (6). The deviation from the equilibrium point is denoted by, $\tilde{\boldsymbol{X}} = \boldsymbol{X} - \bar{\boldsymbol{X}}$. The system is linearised about the equilibrium point using first-order Taylor expansion and represented in the state space form as,

$$\dot{\tilde{\boldsymbol{X}}} = A\tilde{\boldsymbol{X}} + B\boldsymbol{U} \tag{14}$$

Where, A is defined as follows. B depends on the scenario and will be discussed in the next section.

$$A = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{X}}\right|_{\boldsymbol{X}=\bar{\boldsymbol{X}}} \tag{15}$$

$$A = \begin{bmatrix} 0 & \bar{r} & -\bar{q} & 0 & -\bar{n}_z & \bar{n}_y \\ -\bar{r} & 0 & \bar{p} & \bar{n}_z & 0 & -\bar{n}_x \\ \bar{q} & -\bar{p} & 0 & -\bar{n}_y & \bar{n}_x & 0 \\ 0 & 0 & 0 & 0 & a_1\bar{r} + a_4 & a_1\bar{q} \\ 0 & 0 & 0 & a_2\bar{r} + a_5 & 0 & a_2\bar{p} \\ 0 & 0 & 0 & a_3\bar{q} & a_3\bar{p} & a_6 \end{bmatrix} \tag{16}$$

Where, the constants are defined in (27). Since, the reduced attitude vector,$\boldsymbol{n}$ is of unit magnitude, it suffices to control n_x and n_y. That is, n_z is controlled through the constraint relation. Therefore, the state vector is effectively $[n_x\ n_y\ p\ q\ r]$.

1 rotor loss: With three rotors available, there are only 2 control inputs possible (other input is derived from the desired thrust). An intuitive choice of inputs are,

$$\begin{aligned}u_1 &= f_3 - f_1\\ u_2 &= f_3 + f_1\end{aligned} \tag{17}$$

f_1 and f_3 are solved using the above equation. f_2 is determined from the equation,

$$f_1 + f_2 + f_3 = T_{des} \tag{18}$$

The state matrices, A and B then are,

$$A_{(3)} = \begin{bmatrix} 0 & \bar{r} & 0 & -\bar{n}_z & \bar{n}_y \\ -\bar{r} & 0 & \bar{n}_z & 0 & -\bar{n}_x \\ 0 & 0 & 0 & a_1\bar{r} + a_4 & a_1\bar{q} \\ 0 & 0 & a_2\bar{r} + a_5 & 0 & a_2\bar{p} \\ 0 & 0 & a_3\bar{q} & a_3\bar{p} & -a_6 \end{bmatrix} \tag{19}$$

$$B_{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -b_1 \\ b_2 & 0 \\ 0 & b_3 \end{bmatrix} \tag{20}$$

The controllability of the system is investigated by observing the row rank of the controllability matrix, $\mathcal{C}_{(3)} = [B_{(3)} \quad A_{(3)}B_{(3)} \quad A^2_{(3)}B_{(3)} \quad A^3_{(3)}B_{(3)}]$. For the system matrices (19)-(20), the rank is 5 implying full row rank. Therefore, the linearized system is controllable near the equilibrium state.

2 rotor loss: In this case, there will be effectively only one input to the reduced attitude controller given by,

$$u = f_3 - f_1 \tag{21}$$

f_3 and f_1 are calculated by including the desired Thrust equation,

$$f_3 + f_1 = T_{des} \tag{22}$$

Similar analysis was carried out with the same choice of state vector as in the previous section. However, for the choice of inputs, the row rank turned out to be 4 implying non-controllability. Therefore, the control over r was compromised, further justified by its sluggish rate of dynamics.

Therefore, the corresponding state matrices are

$$A_{(2)} = \begin{bmatrix} 0 & \bar{r} & 0 & -\bar{n}_z \\ -\bar{r} & 0 & \bar{n}_z & 0 \\ 0 & 0 & 0 & a_1\bar{r} + a_4 \\ 0 & 0 & a_2\bar{r} + a_5 & 0 \end{bmatrix} \tag{23}$$

$$B_{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_2 \end{bmatrix} \tag{24}$$

The controllability matrix, $\mathcal{C}_{(2)} = [B_{(2)} \quad A_{(2)}B_{(2)} \quad A^2_{(2)}B_{(2)} \quad A^3_{(2)}B_{(2)}]$ has full row rank and hence, the linearized system is controllable, at least near the equilibrium state.

3 rotor loss: In this case, the only input to the system is f_1 with which both the position and the attitude needs to be controlled. Here, we utilize f_1 to control attitude and place just an inequality constraint that,

$$u = f_1 \geq mg \tag{25}$$

For the same choice of state vector, the linearized system was found to be controllable. Hence, $A_{(1)} = A_{(2)}$ given by (23). The control matrix is,

$$B_{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -b_2 \end{bmatrix} \tag{26}$$

The constants appearing in the matrices are defined as follows,

$$\begin{array}{l|l} a_1 = (J_{yy} - J_{zz})/J_{xx} & \\ a_2 = (J_{zz} - J_{xx})/J_{yy} & \\ a_3 = (J_{xx} - J_{yy})/J_{zz} & b_1 = l/J_{xx} \\ a_4 = -J_r\Omega/J_{xx} & b_2 = l/J_{yy} \\ a_5 = J_r\Omega/J_{yy} & b_3 = (d/k)/J_{zz} \\ a_6 = -\gamma/J_{zz} & \end{array} \tag{27}$$

C. LQR Controller Synthesis

The role of attitude controller is to track $\boldsymbol{n}_{des}$, output by the position controller. An appropriate mixer model then calculates the rotor speeds from the inputs given by the attitude controller. A *finite-horizon* LQR controller was designed with the defined state matrices, A and B. The choice of LQR controller is largely because of its optimality and smoothness of actuator inputs. For a linear system as in (14), an LQR controller gives the linear feedback control law that minimizes the cost functional (28),

$$J = \int_0^T (\boldsymbol{X}^T Q \boldsymbol{X} + \boldsymbol{U}^T R \boldsymbol{U})dt + \boldsymbol{X}^T(T) P_1 \boldsymbol{X}(T) \tag{28}$$

$$\boldsymbol{U} = -K\boldsymbol{X} \tag{29}$$

Where, K is given by,

$$K = R^{-1}B^T P(t) \tag{30}$$

P is determined by solving the Riccati ODE with the terminal constraint $P(T) = P_1$,

$$A^T P + PA - PBR^{-1}B^T P + Q = -\dot{P} \tag{31}$$

1 rotor loss: The state error cost matrix, Q is diagonal with cost values of 20 for attitude deviations and $1s^2$ on the angular rates. The input cost matrix, R is diagonal with a value of $1N^{-2}$. The K matrix is determined using the `lqr` routine in MATLAB,

$$K_{(3)} = \begin{bmatrix} -4.04 & -1.85 & -0.01 & 1.06 & -0.30 \\ 1.95 & -4.04 & -1.06 & -0.01 & 0.13 \end{bmatrix} \tag{32}$$

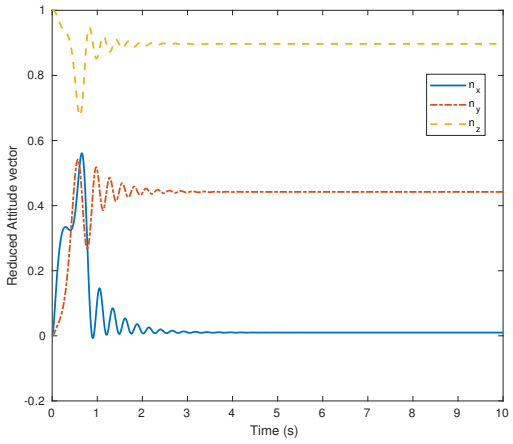

Fig. 14. Simulation: Convergence of reduced attitude with LQR controller $\boldsymbol{n}_{des} = [0, 0.45, 0.89]$

From Fig. 14, it is observed that the reduced attitude vector converges to the desired set-point after a period of oscillatory transient response for $2s$.

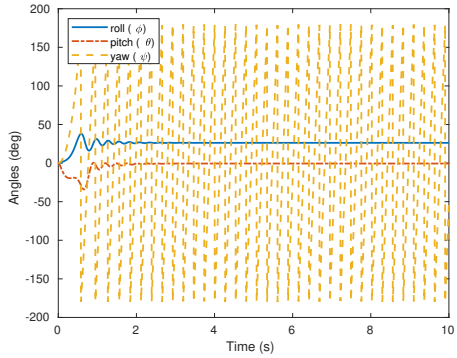

Fig. 15. Simulation: Euler angles with 3 functional rotors controlled using the proposed LQR controller

Fig. 15 depicts the settling of Roll (ϕ) and Pitch (θ) angles albeit with steady-state error, and the unconstrained spinning along the yaw axis.

2 rotor loss: The state error cost matrix, Q is diagonal with cost values of 100 for attitude deviations and $0s^2$ on the angular rates. The input cost matrix, R is diagonal with a value of $0.75N^{-2}$. The K matrix is determined using the `lqr` routine in MATLAB,

$$K_{(2)} = \begin{bmatrix} -25.89 & -25.79 & -0.09 & 1.02 \end{bmatrix} \tag{33}$$

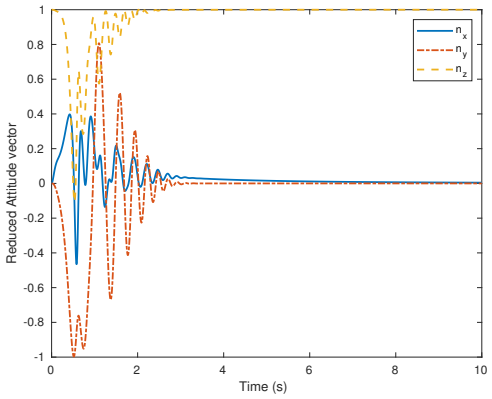

Fig. 16. Simulation: Convergence of reduced attitude with LQR controller $\boldsymbol{n}_{des} = [0, 0, 1]$

From Fig. 16, the transient response lasts longer and has more oscillatory behaviour than the one-rotor out scenario. An additional integral control had to be added to remove the steady-state error associated with this controller.

3 rotor loss: The state error cost matrix, Q is diagonal with cost values of 100 for attitude deviations and $1s^2$ on the angular rates. The input cost matrix, R is diagonal with a value of $4N^{-2}$. The K matrix is determined using the `lqr` routine in MATLAB,

$$K_{(1)} = \begin{bmatrix} 6.56 & 2.61 & 0.08 & -0.73 \end{bmatrix} \tag{34}$$

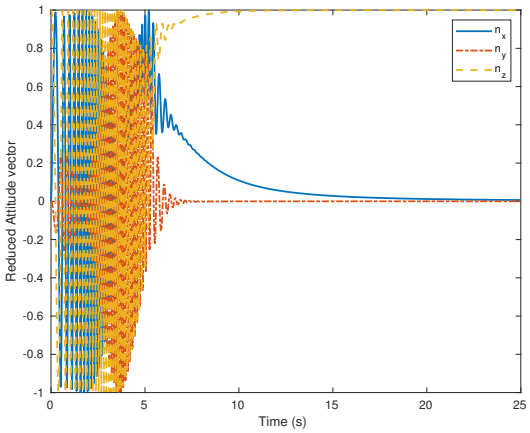

Fig. 17. Simulation: Convergence of reduced attitude with LQR controller $\boldsymbol{n}_{des} = [0, 0, 1]$

Compared to the previous 2 scenarios, the oscillations are far more prominent and have a really high frequency. The reduced attitude system takes a lot more time (approx. $15s$) to settle to the desired setpoint as seen in Fig. 17. This is because there is only one actuator to control the entire system.

The performance of the attitude controller largely depends on the choice of our cost matrices, Q and R. Therefore, more attention needs to be given to tuning these matrices according to our requirements. More in-depth theoretical analysis is required to characterize the system's behaviour in response to the cost matrices.

VII. MOTION PLANNING

Given that we have an attitude controller, we then design the high-level position controller and motion planner. The architecture of the feedback system is such that position controller will output the desired attitude trajectory which would be tracked by the designed controller as mentioned in the previous section. A cascaded approach is adopted because of the notable difference in the pace of the dynamics of the position and the attitude dynamics.

A. Position Control

The translational position of the quadcopter is controlled by varying the total thrust and the direction of attitude vector, hence varying the direction of the acceleration. Let the translational deviation of the vehicle from the desired position be denoted by $\boldsymbol{d}$, in the inertial frame. The desired acceleration is derived so as to force the position dynamics resemble a second order system, described below,

$$\ddot{\boldsymbol{d}}_{des} = -k_p \boldsymbol{d} - k_d \dot{\boldsymbol{d}} \tag{35}$$

In order to achieve this desired acceleration, the Thrust vector should be,

$$\begin{aligned} \boldsymbol{n}_{des} \bar{n}_z T_{des} &= m\boldsymbol{R}^{-1}(\ddot{\boldsymbol{d}}_{des} - \boldsymbol{g}) \\ T &= f_1 + f_2 + f_3 + f_4 \end{aligned} \tag{36}$$

Where, $\boldsymbol{R}$ is the transformation matrix from body to inertial frame and f_i are individual rotor thrusts. For the one rotor case, it is noted that one doesn't have enough control inputs to control the full system and hence, condition on the desired thrust magnitude is relaxed. That is f_1 is determined solely by the attitude controller.

(35) is analogous to a PD controller and the gains are tuned such that we have fast convergence and sufficient difference in the time-scales of the dynamics of the 2 sub-systems - attitude and position. Further, different gains were chosen for the horizontal DOF (x, y) and vertical DOF (z). This is justified for the nature of the mission profile considered, discussed in the next section.

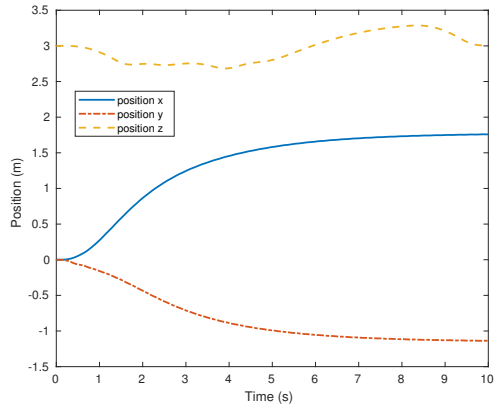

Fig. 18. 1-rotor loss simulation: Convergence of translational position to $(1.5, -1)$ with altitude held at 3 m

As observed in Fig. 18, the translational position converge to the desired setpoint after 7-8 seconds. A steady-state error of 0.25-0.3 m was observed, which has to be taken into account when designing the motion planner.

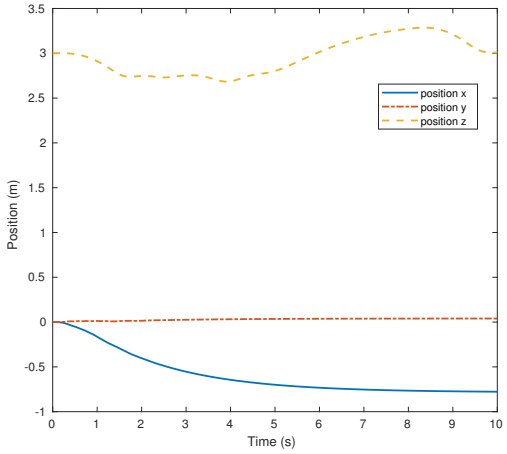

Fig. 19. 2-rotor loss simulation: Convergence of translational position to $(0, -1)$ with altitude held at 3 m

A higher steady-state error and a longer settling time was observed than the previous case. This implies that the parameters associated with the motion planner have to account for this. For the one rotor scenario, the desired thrust was out of bounds of the thrust limits and hence, only hover control was possible.

B. RRT

Quadrotors are highly dynamic systems requiring fast solutions for path-planning problems. Therefore, exact methods such as combinatorial, based on geometric path-planning are not so relevant. Also, it doesn't make sense to snap on an artificial grid on our configuration-space and use grid-based methods such as A* or D*-lite. Discretizing the space (configuration or action) limits the capabilities of the continuous system which might be critical especially in the scenario of rotor failures. Therefore to address all these shortcomings, a sampling based motion planner, RRT [16] was chosen to solve the path-planning problem. The algorithm has been outlined as follows,

Algorithm 1 RRT_FLY(q_{start},q_{goal})

$V \leftarrow q_{start}$
$E \leftarrow \phi$
while $V \cap q_{goal} = \phi$ **do**
  $u \leftarrow$ `randomSample`
  $v \leftarrow$ `closestNeighbor`
  $w \leftarrow$ `extend`(v, u, ϵ)
  $w' \leftarrow$ `FLY`$(v,\ w)$ ▷ 2-point BVP solver
  if `collisionFree`(v, w') **then**
    $V \leftarrow V \cup \{w'\}$
    $E \leftarrow E \cup \{(v, w')\}$

The control strategy discussed in the previous section acts as our 2-point BVP solver, `FLY`$(v,\ w)$. The controller ensures that a control policy exists such that starting from v, the quadcopter converges to an δ-ball around w within a finite time, T.

VIII. NUMERICAL VERIFICATION

The considered workspace measures 12m $\times$ 12m $\times$ 6m. Assuming the altitude is held constant at 3m, the Configuration space is $\mathbb{R}^2$. Obstacles are assumed to be ceiling-high cylinders of radius 1m, similar to pillars. Therefore, in the xy-plane, they are modelled to be circular. The quadrotor is considered to have a volume for the purposes of obstacle avoidance. Collision detection is carried out at small intervals along the edge. $\mathcal{L}_2$-norm is used as the distance metric.

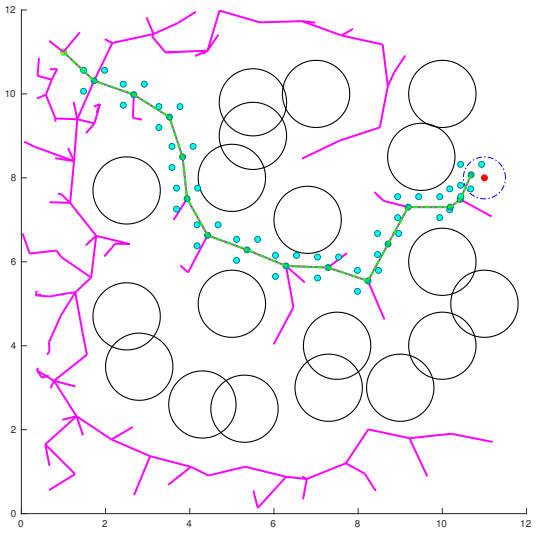

Fig. 20. Simulation: Path determined by RRT-based motion planner when one rotor has failed. *Green - solution branch*, *Magenta - explored tree edges*, *Black - obstacles*

From Fig. 20, it is observed that the quadrotor is able to fly through small passages even though a rotor has failed. However in the case of 2 rotor failure, Fig. 21, the δ-ball of position error increases in size, therefore, the quadrotor is more prone to crashes if it flies close to the obstacles. Hence, it takes a more roundabout route to the goal position. It is also observed that in the latter case, the number of sampled nodes and time taken to determine the path is higher than the first one.

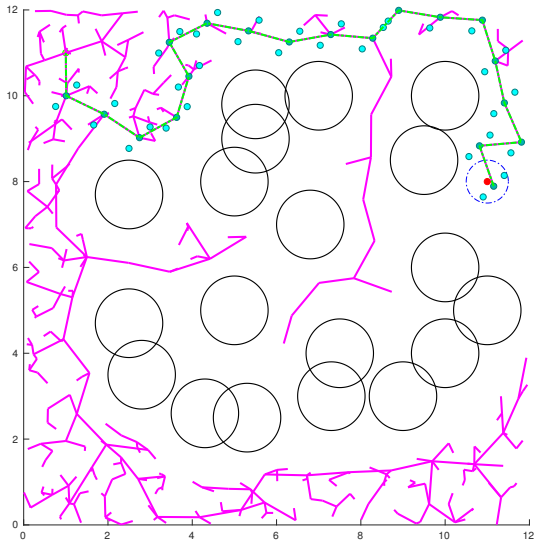

Fig. 21. Simulation: Path determined by RRT-based motion planner when two rotors have failed

As mentioned earlier, in the event of 3 rotor failure, the required thrust exceeds the rotor constraints and therefore, position control is not feasible. Hence, the only solution would be to hover and land safely in such a scenario.

IX. CONCLUSION

This paper addresses the complete problem of path-planning and control of quadrotors in the event of rotor failure. It presented the characterisation of the changes in quadcopter dynamics when one, two or three rotors fail. It elaborated on the synthesis of optimal control strategies to track attitude and position. A cascaded control architecture was proposed and justified. The attitude controller was meticulously designed by considering the equilibrium solutions and system controllability.

This control policy is used as an approximate 2-point BVP solver in RRT-based planner, converging to a δ-ball around the terminal node. Although, RRT doesn't require solving the 2-point BVP, the proposed algorithm, `RRT_FLY`, could be extended to asymptotically optimal sampling based motion planners such as RRT* and RRT#. We intend to incorporate such notion of optimality in the motion planning problem in future work.

Further work needs to be done in studying the effect of the various parameters (controller gains, ϵ, δ) on the system response and devise a systematic approach to tune these parameters depending on the scenarios. Another interesting line of research would be to address the combined problem of path-planning and control instead of the existing hierarchical structure. One possible solution would be to explore the C-space using regions of attraction instead of one-dimensional edges [4]. This is best visualised as funnels around controllable optimal trajectories. Finally, the developed algorithms would be tested on a physical test platform and experimental conclusions would be drawn.

REFERENCES

[1] A. Lanzon, A. Freddi, and S. Longhi, "Flight control of a quadrotor vehicle subsequent to a rotor failure," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 580–591, 2014.

[2] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 45–52.

[3] A. Nemati and M. Kumar, "Modeling and control of a single axis tilting quadcopter," in *2014 American Control Conference*. IEEE, 2014, pp. 3077–3082.

[4] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[5] M. Otte and E. Frazzoli, "Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.

[6] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[7] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[8] M. Geisert and N. Mansard, "Trajectory generation for quadrotor based systems using numerical optimal control," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 2958–2964.

[9] A. P. Peirce, M. A. Dahleh, and H. Rabitz, "Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications," *Physical Review A*, vol. 37, no. 12, p. 4950, 1988.

[10] R. Allen and M. Pavone, "A real-time framework for kinodynamic planning with application to quadrotor obstacle avoidance," in *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 1374.

[11] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and control of mini-flying machines*. Physica-Verlag, 2006.

[12] M. K. M. Jaffar, M. Velmurugan, and R. Mohan, "A novel guidance algorithm and comparison of nonlinear control strategies applied to an indoor quadrotor," in *2019 Fifth Indian Control Conference (ICC)*. IEEE, 2019, pp. 466–471.

[13] R. Tedrake, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," *Avaliable at: http://drake. mit. edu (accessed March 2014)*, 2014.

[14] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, 2011.

[15] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Epfl, Tech. Rep., 2007.

[16] J. J. Kuffner Jr and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *ICRA*, vol. 2, 2000.