

Online decision making and path planning framework for safe operation of unmanned aerial vehicles in urban scenarios

Marcos Quiñones-Grueiro¹, Gautam Biswas², Ibrahim Ahmed³, Timothy Darrah⁴, and Chetan Kulkarni⁵

^{1, 2, 3, 4} *Vanderbilt University, Nashville TN, 37209 USA*

marcos.quinones@vanderbilt.edu

gautam.biswas@vanderbilt.edu

ibrahim.ahmed@vanderbilt.edu timothy.s.darrah@vanderbilt.edu

⁵ *KBR, Inc., NASA Ames Research Center, Moffett Field, CA, 94035*

chetan.s.kulkarni@nasa.gov

ABSTRACT

As the potential for deploying low-flying unmanned aerial vehicles (UAVs) in urban spaces increases, ensuring their safe operations is becoming a major concern. Given the uncertainties in their operational environments caused by wind gusts, degraded state of health, and probability of collision with static and dynamic objects, it becomes imperative to develop online decision-making schemes to ensure safe flights. In this paper, we propose an online decision-making framework that takes into account the state of health of the UAV, the environmental conditions, and the obstacle map to assess the probability of mission failure and re-plan accordingly. The online re-planning strategy considers two situations: (1) updating the current trajectory to reduce the probability of collision; and (2) defining a new trajectory to find a new safe landing spot, if continued flight would result in risk values above a pre-specified threshold. The re-planning routine uses the differential evolution optimization method and takes into account the dynamics of the UAV and its components as well as the environmental wind conditions. The new trajectory generation routine combines probabilistic road-maps with B-spline smoothing to ensure a dynamically feasible trajectory. We demonstrate the effectiveness of our approach by running UAV flight simulation experiments in urban scenarios.

1. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) is increasing at unprecedented rates across across a wide spectrum of applications that include surveillance, package delivery, photography, cartography, remote sensing, agriculture, military mis-

sions, and more¹. The FAA passed regulations in 2016 that authorized the commercial use of UAVs. As adoption and use of these vehicles increase, so does the risk of collisions and crashes that can result in a loss of money, time, productivity, and most important, human lives.

There is an abundance of research on the technical aspects of UAV systems: their design, implementation, stability, operation, and fault diagnostics (Moir & Seabridge, 2012). However, a more holistic approach to ensuring safe operations in a heterogeneous airspace is required to address this multifaceted problem that combines component and system diagnostics, system-level prognostics, airspace safety assurance, flight path risk assessment, as well as trajectory planning and replanning, just to name a few (Clothier & Walker, 2006).

Our overall goals are to support safe operations of package delivery UAVs operating in urban environments. A decision-making framework is developed to maintain system safety during a UAV mission, i.e., flight from a starting point to a destination going through a sequence of pre-determined way points. Our objectives are to minimize the overall risk of mission failure by simultaneously considering: (1) a number of risk factors (e.g., risk of collisions), (2) uncertainties in the environment (e.g., wind gusts), and (3) the operating state of the vehicle (e.g., degradation in the UAV components). In this paper, we develop a *Risk analysis framework* to compute and update risks associated with projected UAV flight paths by considering two potential hazards: (1) collision with static obstacles, and (2) depletion of battery charge below a pre-specified safe threshold. In addition, we also take into account the effects of degradation of system components on overall UAV flight performance. In general, multiple components of a system may degrade at the same time, therefore,

Marcos Quiñones-Grueiro et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://doi.org/10.36001/IJPHM.2021.v12i3.2953>

¹UAV sales are expected to top \$12 billion in 2021 (<https://www.businessinsider.com/drone-technology-uses-applications>)

we apply methodologies for a system-level prognostics that we have developed in past work (Khorasgani et al., 2016).

Ensuring safe UAV flight operations requires careful *flight planning* and *trajectory generation* based on a series of four-dimensional waypoints (latitude, longitude, altitude, and arrival time), while also satisfying a set of constraints that assure safety during flight (e.g., distance from nearest obstacle $\geq 2\text{m}$, battery charge $\geq 10\%$, risk threshold $\leq 15\%$), and optimizing a set of performance parameters (e.g., flight time, power consumption). We assume our package-carrying UAV is a low flying craft that typically flies over city streets between large buildings. Our path planning algorithm assumes the existence of a map of possible routes that UAVs can fly within this urban environment. Trajectory generation requires deriving a *stable* trajectory, defined by a set of waypoints, with the assumption that the UAV, if undisturbed will fly a straight line path between two consecutive way points. The UAV trajectory is smoothed using B-splines at corners, ensuring that the UAV does not have to make sharp turns when it negotiates these corners.

In situations where there are no ambient disturbances, such as wind direction changes or wind gusts, we assume the UAV chooses the minimum distance path between its start and destination points because that consumes the least battery power (energy), which, therefore, involves the least risk. However, wind gusts or change in environmental conditions in the vicinity of where the UAV is flying, may result in a change in flight risk causing the risk values to exceed specified thresholds. In extreme circumstances, the UAV may decide that it is too risky to proceed. In such situations, the UAV executes emergency landing procedures to prevent crashes. We do not deal with contingency management situations in this paper.

The rest of this paper is organized as follows. Section 2 reviews the literature on UAV safety analysis. Section 3 outlines the proposed framework of our work. Section 4 provides a description of our octocopter UAV system model and the test bed for our experimental studies. Section 5 presents our online decision making and path planning framework. Section 6 presents the optimization methods used to solve the trajectory re-planning problems. The results of our experimental case studies run on the octocopter system under different scenarios are presented in Section 7. Finally, Section 8 presents the conclusions of our work and directions for future research.

2. BACKGROUND: SAFETY ANALYSIS, HEALTH MANAGEMENT AND ONLINE PLANNING FOR UAVS

Past work on UAV safety analysis has focused on collision avoidance in the civil airspace for commercial aircraft with recommendations for developing and advancing Traffic Collision and Alert Systems (TCAS) that can detect and minimize these collisions (Kuchar, 2005). Clothier & Walker

(2006) contend that overall safety requirements for UAV systems should be the same as that for human-piloted aviation. They developed a simple simple fatality model to illustrate the impact of different safety objectives on the design and operation of UAV systems. They used comparative examples to highlight the importance of the nature of risk exposure to the type of operation being performed.

Traditionally, contributions from the field of robotics have evaluated risk without considering the state of the vehicle. For instance, data driven approaches to risk analysis have taken into account uncertain weather conditions (Rubio-Hervas et al., 2018) and measurement uncertainty (De Filippis et al., 2011) together with the vehicles kinematics, but they have ignored the fact that a UAV's operating conditions can change during flight. On the other hand, contributions from the field of control systems incorporate trajectory planners that consider the dynamic model of the vehicle (Brown & Rogers, 2016), but do not take into account its state of health for risk analysis. However, in urban scenarios, it is important to consider the interactions between the system's state and the varying environmental conditions (Coutinho et al., 2018).

Computing risk and safety is another area of research that is a key component of ensuring safe UAV operation. Lin & Shao (2020) compute the expected level of safety (ELS) of a path as a function of mean time between failures, the area of exposure in square meters (assuming a ground impact), the population density, and accident severity. Ancel et al. (2017) and Ancel et al. (2019) present a real-time risk assessment framework that quantifies the risks to bystanders for operations in populated areas. They consider three separate modules:

- a Probabilistic Graphical Model based on Bayesian belief networks (BBNs) that estimates the likelihood of pre-determined mishaps using the aircraft health systems status provided as part of the aircraft telemetry downlink. The BBNs are derived from a FMEA of the system and they are parameterized based on the dynamic aircraft health data;
- an Off-Nominal Trajectory and Impact Point Prediction module that provides an expected impact point for the modeled mishap with an associated uncertainty bound; and
- a Casualty Estimation module that employs the impact point and associated uncertainty as well as population distribution data to evaluate injury or fatality to humans on the ground.

This approach, however, does not formally address the dependence between the probability of collision with obstacles and the state of health of the UAV.

Primatesta et al. (2019) proposed an off-line and on-line path planning strategy that considers a dynamic risk map based on the probability of casualties caused by the loss of control of

the vehicle. Their path planning strategy only considers the kinematic constraints associated with the vehicle. Schacht-Rodríguez et al. (2018) present a path-planning strategy that takes into account the available battery power, while also considering the degradation of the battery. Krishnan & Manimala (2020) consider both the minimal path length and the minimal risk of collision for designing a path-planning algorithm suitable for real-time applications. They, however, do not consider the state of health of the UAV in the optimization problem. Benders et al. (2020) propose an adaptive path planning strategy that takes into account performance uncertainties associated with the parameters of the kinematic model of the UAV.

Real-time path planning for UAVs in the context of obstacles as dynamic traffic and geofences is explored in Chatterjee et al (2019). They use RRTs and build on top of the DAIDALUS software by NASA to implement maneuver computation for collision avoidance. To save computation time, the authors propose heuristics for early termination of tree generation for RRTs. Zammit & Van Kampen (2020) compare an A* graph search algorithm with limited look-ahead and intermediate goals, against RRTs for path planning. In their scenario of traversing through openings in walls laid out in a single principal direction, A* outperforms the more random exploration of RRT. This may be because the heuristics for intermediate goal points and the cost to goal are monotonic with respect to the optimal trajectory and its true cost. Performance may change if the environment required moving in directions at obtuse angles to the position vector to the goal. Convex optimization strategies have also been proposed for real-time trajectory planning (Aggarwal & Kumar, 2020). Those approaches allow considering the kinematic constraints associated with the UAV in the path planning method. However, they do not consider the dynamic constraints associated, for example, with faults or the degraded behavior of one or more components of the UAV.

3. PROPOSED FRAMEWORK

Earlier state-of-the-art methods discussed do not take into account the dynamic model of the UAV, the degradation of its components, the obstacle map, and the environmental conditions, and how all of these together may affect the flight performance. The proposed novel approach presented in this work, integrates multiple adverse effects to define the likelihood of mission failure, and, when needed, executes path re-planning as a function of all the above-mentioned elements. Probability of mission failure is calculated based on both the remaining useful life (depending on component-level performance measures, such as the battery state of charge), and the probability of collision (which also depends on the health of the components and the environmental conditions, i.e. the trajectory tracking error will worsen as the motors degrade and/or under increased wind conditions). An analytic frame-

work is implemented, that tracks the degradation rate of individual components, such as the battery and motors from mission to mission, to update the dynamic model of the UAV. Based on that model, we continuously update the predicted remaining trajectory of the UAV to decide when re-planning is required. We make the following assumptions

- We can estimate the state variables and parameters of the UAV and its components within an error bound;
- An online fault detection and diagnosis module is available;
- The obstacle map information is available, this includes a database with semantic geo-referenced data, such as terrain and land-use maps, obstacles, and airspace restrictions;
- The environmental conditions do not change dramatically during flight, i.e., wind gusts might be present but their intensity does not change during flight; and
- There are no communication issues observed between the UAV and a primary cloud-based station.

In future work, we will relax some of these assumptions to consider conditions and a framework that includes more realistic scenarios. WE should also clarify that the objective of the proposed approach is to assess the probability of mission failure, and when that falls below a pre-specified threshold, to re-plan and derive safer trajectories. Therefore, we define the risk of mission failure as: the product of the likelihood of mission failure and the consequences of mission failure. With this proposed framework, we can seamlessly integrate with modules for impact point prediction and casualty estimation developed by (Ancel et al., 2017, 2019) and (Primatesta et al., 2019).

4. UAV SYSTEM DESCRIPTION

In this section, we described the modeling of an octocopter UAV with its components as well as the implemented control system.

4.1. UAV Model

The octocopter airframe dynamics model is based on Newton-Euler equations of motion for a rigid body (Powers et al., 2015). The octocopter is composed of a central hub with 8 arms extending radially and a motor at the end. The derived body forces considered are

$$F_b = F_M + F_D + mgR_{IB}e_z, \quad (1)$$

where $F_b \in \mathbb{R}^3$ is the resulting force acting on the body frame, $F_M \in \mathbb{R}^3$ is the resultant force generated by the motors, $F_D \in \mathbb{R}^3$ is the drag force resulting due to the movement of a UAV through the air, m is the mass of the UAV, g is the gravity acceleration, $R_{IB} \in \mathbb{R}^{3 \times 3}$ is the rotation matrix from the inertial frame to the body frame, and $e_z = [0 \ 0 \ 1]^T$.

The rotation matrix is calculated based on the Euler angles $[\phi, \theta, \psi]$ (Mahony et al., 2012). The order of rotations from the inertial frame to the body frame considered in this work is yaw (Z)-pitch (Y)-roll (X) assuming the UAV is moving forward in the positive X direction.

Motor forces and torques

The eight rotating motors including propellers are used to generate motor forces F_M and torques M_M . The disposition and rotation of the motors is shown in Figure 1. For each motor i , the force and torque generated are given by

$$F_{M_i} = c_T \omega_{M_i}^2, \quad M_{M_i} = c_Q \omega_{M_i}^2, \quad (2)$$

where c_T is the coefficient of thrust and c_Q is the torque constant. Both parameters can be easily estimated from static thrust tests (Mahony et al., 2012). The net force applied to the airframe is the summation of the forces generated by the motors. The net torque acting on the octocopter arises from the aerodynamics (the combination of the produced rotor forces and air resistances) applied to the N-rotor vehicle.

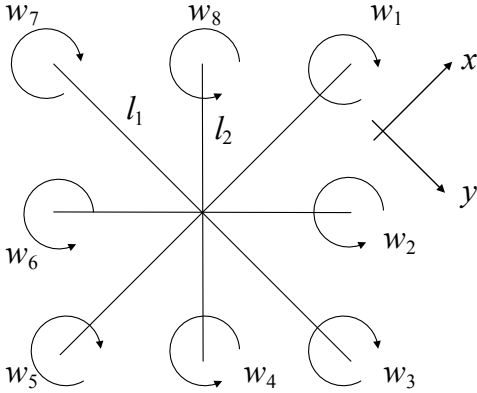


Figure 1. The octocopter seen from above. Showing the numbering of each motor and their rotational directions

Drag force

Form drag is the most common and easily modeled aerodynamic effect. Form drag arises due to the movement of a reference area through a fluid. The general expression of the form drag force is

$$F_{Drag} = -\frac{1}{2} \rho C_D A v_b^2, \quad (3)$$

where ρ is the air density, C_D is the drag coefficient, A the reference area that is perpendicular to the velocity of the object (v_b). The torque generated due to drag is considered negligible in this work.

Wind effect

Wind is a major disturbance source for light aerial vehicles, such as octocopters. The ability to remain stable and minimize wind disturbances is a requirement for the use of UAVs in urban scenarios. The wind velocity vector is defined by

$$\mathbf{v}_w = \mathbf{v}_{w,nom} + \mathbf{v}_{w,rand}, \quad (4)$$

where $\mathbf{v}_{w,nom}$ is the nominal component and $\mathbf{v}_{w,rand}$ is the stochastic component. A valid representation of the latter term can be generated by using Dryden's turbulence model. The wind affects the translation of the octocopter by modifying the drag force (Gougeon et al., 2018). The change in the moment dynamics caused by the wind acting on the propellers of the motors is assumed to be negligible in this work.

4.2. Propulsion system

The propulsion system of the octocopter is formed by a set of BrushLess DC Motors (BLDCM), an Electronic Speed Controller (ESC), and a Lithium Polymer battery. Each motor is powered by the battery through the ESC, which controls the angular speed through a PWM signal. The dynamics of the ESC are neglected in this work. We assume, therefore, that the voltage supply of the battery and the current consumed by the motors are averaged with respect to the duty cycle value of the control signal.

Motor model

A Brushless DC motor is a type of permanent magnet synchronous motor driven by a DC supply voltage. The simplified mathematical model can be described by the following equations (Schacht-Rodríguez et al., 2018)

$$\dot{\omega}_i = \frac{1}{J_m} (K_e i_c - T_{load} - D_f \omega - T_f), \quad (5)$$

$$i_c = \frac{1}{R_{eq}} (v_{DC} - K_e \omega_i), \quad (6)$$

where $R_{eq} = \frac{2}{3} \sum_{j=1}^3 R_j$ is the equivalent electric resistance of the coils, K_e is the back electromotive force constant, ω_i is the angular velocity of the i th BLDCM, T_f is the static friction torque, D_f is the viscous damping coefficient that allows to estimate the dynamic friction torque ($D_f \omega$), J_m is the inertia of the BLDCM, v_{DC} is the input voltage control signal, i_c is the current demanded from the battery pack, and T_{load} represents the torque load generated by the propellers. The relationship between the torque load and the angular velocity is non-linear and can be defined from experimental data according to the dimensions and material of the propellers.

Battery model

The dynamics of a battery cell in the electrical domain can be modeled using an equivalent circuit representation (Plett,

2015). The dynamic equations that characterize the battery behavior are given by:

$$\dot{V}_{SoC} = -\frac{i_c}{Q} \quad (7)$$

$$i_d = \frac{i_c}{R_d C_d} - \frac{i_d}{R_d C_d} \quad (8)$$

$$V_{out} = V_{ocv} - R_d i_d - R_0 i_c, \quad (9)$$

where Q represents the total capacity of the battery, i_c the input current, R_d and C_d are the diffusion resistance and capacitance, i_d the current going through the diffusion resistance, R_0 represents the internal resistance, V_{ocv} is the open circuit voltage, and V_{out} is the output voltage of the battery. The open circuit voltage depends non-linearly on the state of charge of the battery. The function that characterizes such relationship can be experimentally estimated and approximated as a polynomial by using charge and discharge data.

4.3. Navigation system

The standard navigation system of a UAV carries an inertial measurement unit (IMU) and a global positioning system (GPS), with additional augmentation provided by augmented by imaging systems that provide additional information, especially when the GPS signal is lost in urban environments (Conte & Doherty, 2008). A typical IMU includes a three-axis rate gyro, a three-axis accelerometer, and a three-axis magnetometer.

4.4. Control scheme

The octocopter's hierarchical control scheme is shown in Figure 2. This control approach is commonly used for multi-copters since it allows for stabilization of the position and orientation of the octocopter with respect to a trajectory. The low level controls the vehicle attitude, and the top level controls the position along a trajectory, together forming nested feedback loops.

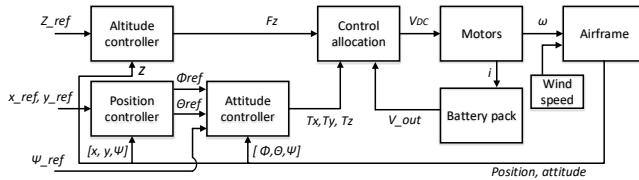


Figure 2. Hierarchical control scheme for the octocopter

The reference trajectory for the octocopter can be defined for each time step in terms of position and yaw angle $[x_t, y_t, z_t, z_t, \psi_t]$. Altitude is controlled through a Proportional Integral Derivative (PID) controller based on the reference input. This controller generates the required force in the z direction. The position controller block estimates, based on the current position

of the vehicle and yaw angle, the reference for the pitch (θ) and roll (ϕ) angles. The attitude controller block is formed by three PID controllers for pitch, roll, and yaw. This block generates the required torque in each direction. The control allocation block transforms the required torques and force into a reference voltage for each BLDC motor of the octocopter based on the maximum voltage that the battery can provide. Finally, the motor block generates the angular velocities according to the motor's dynamics.

4.5. Degradation of Components

The state of health (SOH) of a component describes its actual physical condition in relation to its nominal working condition. Typically, the SOH is described by one or more parameters that are linked to the behavior of the system. A function that aggregates the deviation of these parameter values from their nominal values defines the SOH of the component. Typically, the degradation of the components of the octocopter increases monotonically from mission to mission. The SOH of the components of the octocopter affect its overall performance.

BLDC motors are susceptible to mechanical degradation in the form of bearing wear, and less common electrical degradation in the form of contact corrosion and insulation deterioration (Abramov et al., 2014). An increase in winding resistance results in increased power consumption under load (and decreased consumption under no-load conditions), as depicted in the FMECA chart provided in Kulkarni & Corbetta (2019). The winding resistance change is associated with the electric resistance of the coils (R_{eq}). Mechanical degradation is associated with the static and dynamic friction parameters (T_f and D_f) of the motor model presented above. The SOH of the battery decreases with time due to aging mechanisms caused by its charge/discharge cycles, and damage due to deep discharges. The aging of the battery causes an energy loss leading to capacity loss (a reduction of battery capacity), and the increase of the battery impedance (power fade). Capacity loss is evidenced in the battery model by the reduction of the total capacity parameter Q , and the power fade by the increase of the internal resistance R_0 (Daigle & Kulkarni, 2016; Sierra et al., 2019).

5. DECISION MAKING AND PATH PLANNING APPROACH

As discussed earlier, we develop a risk-based framework for maintaining system safety during a mission by considering a number of risk factors along with uncertainties in the environmental and operational state of the vehicle (UAV). In this paper, We define mission failure, i.e., the loss of the vehicle due collisions with other objects (primarily static objects, such as buildings or towers or moving objects, such as other UAVs) or due to a crash to the ground. The primary causes of mission failure are: (1) adverse weather conditions and/or (2)

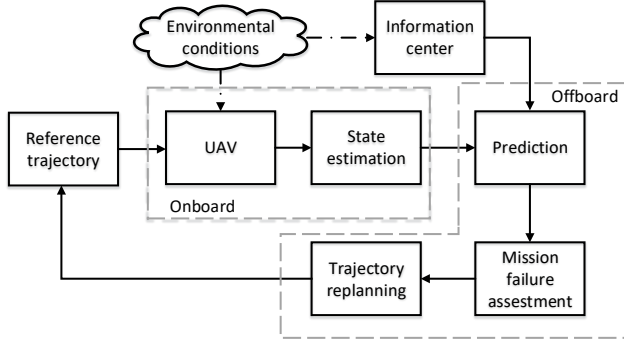


Figure 3. Online re-planning framework

system faults that degrade system performance. Faults may be associated with different components of the UAV, such as the battery, the motors, and the Inertial Measurement Unit.

We assume that the UAV system has a mission plan, defined by a set number of way-points $\Gamma = \{c_1, c_2, \dots, c_n\}$ and a desired cruise speed at the start of its mission. The risk associated with the current flight trajectory, given by the current plan is continually computed taking into account the effect of current conditions, which may include adverse situations, such as a wind gusts (weather-related condition) or component degradation and failures (faults). When the risk of the current trajectory increases above a pre-specified threshold, the *online re-planning algorithm* depicted in Figure 3 is initiated. The algorithm includes four processing modules: (1) state estimation, (2) prediction, (3) mission failure assessment, and (4) trajectory replanning. The *state estimation module* determines the system state and parameters and the model uncertainty. The *prediction module* considers the current state, system parameters, model uncertainty, and information about the environmental variables as initial conditions, and performs stochastic simulations of the remaining trajectory for the current mission plan. The *mission failure assessment module* computes the risk of mission failure for the remaining trajectory, and initiates re-planning if the predicted risk exceeds the pre-specified threshold. Finally, the *trajectory re-planning module*, if invoked, generates a new trajectory for the remaining mission, so that the risk of mission failure is reduced to an acceptable value. If such a trajectory cannot be found, the re-planning algorithm will attempt to find a trajectory that results in an emergency landing to avoid a crash.

Risk computation by stochastic simulation and the replanning algorithms are computationally expensive, and typically, there is not enough processing power on a small octocopter to compute them in a timely manner. Therefore, we assume a two part computational architecture for our UAV safety analysis system. First, an on-board processor performs the monitoring, fault detection and isolation, and state estimation tasks.

The computational intensive predictions of mission failure and replanning are performed off-board using a cloud-based infrastructure that communicates continually with the UAV system. The rest of this section discusses the four modules in greater detail.

5.1. State estimation

In this work, our state estimation primarily focuses on the state variables associated with the vehicle flight trajectory, i.e., the height, attitude, angular velocity, linear velocity and position of the octocopter system in 3-D space. In addition, we have designed state estimators for the primary components of the UAV that need to be monitored for system health, for example, the state of charge and the output voltage of the battery.

We estimate the attitude, velocity, and position of the UAV using complementary filtering techniques (Mahony et al., 2012), or using extensions of Kalman filtering methods (Burri et al., 2015; Yang et al., 2017; Al-mashhadani, 2019). Inertial and visual sensor fusion algorithms can be employed if imaging sensors are available (Corke et al., 2007). State estimation of the motors can be performed using extended Kalman filters. A number of methods, such as unscented Kalman filters and particle filters have been proposed for state and parameter estimation of the battery (Plett, 2015; Sierra et al., 2019). In this work, we consider the use of an Sigma-Point Kalman filter (Merwe et al., 2004).

5.1.1. Sigma-Point Kalman filter

The Sigma-Point Kalman filter (SPKF) is a second-order non-linear state estimation method based on probabilistic inference theory. SPKF approximates the mean and covariance of the posterior distribution of the state variables by using a set of sigma-points. Each Sigma point represents a random state variable formed by concatenating the original state and noise variables as follows: $\mathbf{x}_n^a = [\mathbf{x}_n^T \mathbf{w}_{x_n}^T \mathbf{w}_{y_n}^T]^T$. The sigma points χ_{n-1} are obtained in a deterministic fashion using

$$\chi_n = \{\mathbf{x}_n^a \mathbf{x}_n^a + \gamma \sqrt{P_n} \mathbf{x}_n^a - \gamma \sqrt{P_n}\}, \quad (10)$$

where χ_n represents the set of Sigma points, P_n is the error covariance matrix, and γ is a parameter that determines the spread of the Sigma points. P_n is formed by three sub-diagonal blocks based on the state error covariance matrix P_x , the measurement error covariance matrix R , and the process noise covariance matrix Q . Both R and Q are parameterized offline based on sensor error analysis Gross et al. (2010).

The Sigma points are propagated to estimate the state variables, which are later updated according to the measurements obtained from the system inputs and outputs. Full implementation details can be found in Merwe et al. (2004). We use the unscented Kalman filter as our Sigma-point selection scheme,

and describe the equations used to estimate the position and the battery state of charge next.

5.1.2. Inertial navigation equations

The navigation state vector is formed by the local Cartesian position and velocity as well as the roll, pitch, and yaw angles. The IMU measures the three axis accelerations and angular rates. Since $\mathbf{x} = [x, y, z, v_x, v_y, v_z, \phi, \theta, \psi]$, the navigation update equations in continuous time are written as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = R_{be} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \\ q \cos(\phi) - r \sin(\phi) \\ (q \sin(\phi) + r \cos(\phi)) \sec(\theta) \end{bmatrix}, \quad (13)$$

where $R_{be}(\phi, \theta, \psi)$ is the rotation matrix from the body frame to the inertial/world frame known as Direction Cosine Matrix. The IMU measures the acceleration and angular rates $[a_x, a_y, a_z, p, q, r]$ which are considered the inputs of the filter, and the GPS measures position and velocities $[x, y, z, v_x, v_y, v_z]$ which are considered the output measurements Gross et al. (2010).

5.1.3. Battery model equations

The mathematical model of the battery is given by the set of equations (7)–(9). The relationship between the battery state of charge (SoC) and the open circuit voltage (OCV) for a cell is shown in Figure 4. The measured current is considered the input of the filter and the measured output voltage is considered the output.

5.2. Prediction

The prediction module computes the states of the UAV for the remaining of the reference trajectory at each time step. Uncertainty propagation is performed by sampling from the distribution of the state variables as the trajectory is completed. Considering the state space model of the UAV

$$\mathbf{x}_{n+1} = f(\mathbf{x}_n, \mathbf{u}_n, \vartheta_n, \mathbf{w}_x), \quad (14)$$

$$\mathbf{y}_n = h(\mathbf{x}_n, \mathbf{u}_n, \vartheta_n, \mathbf{w}_y), \quad (15)$$

where \mathbf{x}_n represents the set of variables that characterize the dynamics of the UAV, \mathbf{u}_n is the inputs of the system, ϑ_n is the set of degrading parameters, \mathbf{w}_x and \mathbf{w}_y capture the system

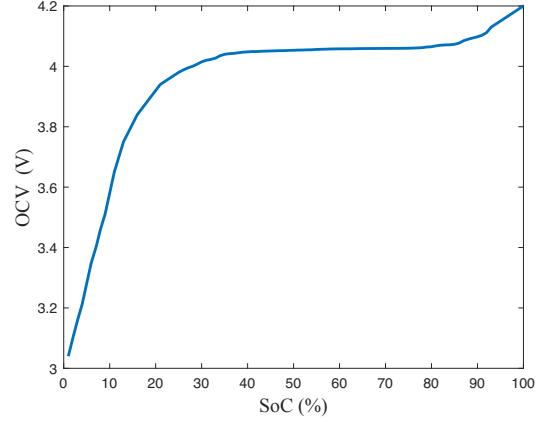


Figure 4. Relationship between battery cell SoC and OCV

model uncertainties and the measurement noise at time step n , respectively.

The distributions of \mathbf{w}_x and \mathbf{w}_y are required to propagate the state of the UAV along the remaining trajectory. The distribution of \mathbf{w}_y is estimated offline based on the precision of sensors. The distribution of \mathbf{w}_x is estimated online based on the state estimators designed for the UAV. More specifically, the process noise associated with the position of the UAV is estimated. This allows us to compute the probability of collision with a given obstacle. The SPKF, discussed previously, has been used in previous work to estimate the position error of the UAV (Merwe et al., 2004). Similarly, the SPKF has been used to estimate the state of charge and voltage of battery packs (Plett, 2006). We, therefore, use the SPKF for process noise estimation of the states of the UAV position and attitude, as well as for the battery. Another component that affects the system dynamics is the wind vector. We assume this information is provided by an external meteorological (i.e., weather forecasting) center. This provides the parameterization of equation 4.

5.3. Mission failure assessment

Risk of mission failure is performed for a predicted trajectory by taking into account the uncertainties associated with the state of the UAV and the measurements. This computation can be performed for the current trajectory, starting from the current time point forward, or a new trajectory generated by the replanning algorithm from the current time point forward till the end of the mission. In this paper, we compute the risk of mission failure as a product of two factors: (1) the risk of collision and (2) the risk of running out of battery power. The risk of mission failure is defined as follows:

$$f_{mfail}(\Gamma) = \max(F_c^{max}(\Gamma), F_{rul}^{max}(\Gamma)) \in [0, 1], \quad (16)$$

where $f_{mfail}(\Gamma)$ is the risk of mission failure, $F_c(\Gamma)$ is the maximum risk of collision with an obstacle, and $F_{rul}(\Gamma)$ is

the maximum risk of crashing because the UAV runs out of battery power along the trajectory Γ . We simplify the solution process for a multi-objective optimization problem, which may not be tractable, replacing it with the worst case scenario risk calculation for the two factors of mission failure. The maximum risk of mission failure is used to determine whether re-planning is required based on a pre-defined failure threshold $\tau_{fail} \in [0, 1]$. We describe how each term of equation 16 is computed next.

5.3.1. Risk of collision with an obstacle

The total risk of collision of a mission is accumulated along a specific trajectory that has a set of known K static obstacles in the environment in the vicinity of the trajectory. The average risk can be calculated according to:

$$Risk_{tot} = \frac{1}{K} \sum_{k=1}^K \int_{\Gamma} f(\Gamma, obst_k) d\Gamma, \quad (17)$$

where $f(\Gamma, obst_k)$ is a function that calculates the probability of collision of the UAV with obstacle k when it follows trajectory Γ . For convenience, we convert the trajectory of the UAV into a discrete form by considering a sampling time. Thus the total risk can be expressed as:

$$Risk_n^{tot} = \frac{1}{KN} \sum_{k=1}^K \sum_{n=1}^N f(\Gamma(n), obst_k), \quad (18)$$

where $\Gamma(n)$ represents the position of the UAV at time point n , and $f(\Gamma(n), obst_k)$ takes the following form

$$f(\Gamma(n), obst_k) = p(D_{obst_k}^n < dsafe_k), \quad (19)$$

where $D_{obst_k}^n$ represents the distance between the UAV and the obstacle k at time n of the trajectory, and $dsafe_k$ represents a minimum distance separation that must be kept between the UAV and the obstacle to avoid an imminent collision. We assume the position of the UAV is a stochastic variable whose distribution is estimated through the state estimator described in Section 5.1. , D_{obst_k} is also a stochastic variable normally distributed whose standard deviation can be inferred from the position variance (estimated through the SPKF). Equation (19) evaluates the inverse cumulative distribution function of D_{obst_k} at $dsafe_k$.

The position of the UAV is represented by the coordinates of its center of mass. Assuming a two dimensional map, we consider a set of static obstacles that can be represented by rectangles defined in terms of their four corner coordinates on a 2D map of the UAV spatial environment, i.e., $obst_k = (x_i, y_i), i = 1, 2$. For a realization of the predicted future trajectory, the distance is computed as $d_{obst_k} = |\xi - \nu_k|$, where ξ is a vector of the coordinates of the UAV and ν_k is the closest vertex of obstacle k . The distance threshold $dsafe_k$ depends on a number of factors that we discuss next.

The total risk of collision provides the basis for offline path planning. We reiterate that re-planning is triggered because we predict that the minimum separation distance required between the UAV and the closest obstacle will be violated for a segment (s) of the remaining trajectory with a computed risk value that exceeds a threshold. This triggers an evaluation of multiple alternate trajectories, and selection of the one that is considered to be the safest in terms of risk and does not violate the minimum separation distance. Therefore, the objective of re-planning is to update the expected risk of the chosen trajectory, as a result of which, we also reduce that probability of collision to an acceptable value.

The maximum risk (and, therefore, probability of collision) for a trajectory is computed with respect to the closest obstacle using the equation (18):

$$Risk_{max} = \max_{n \in N, k \in K} f(\Gamma(n), obst_k). \quad (20)$$

As a summary, in this work, F_c^{max} is equated to $Risk_{max}$, and determines the objective to be minimized by replanning.

Defining a distance threshold

The distance threshold $dsafe_k$ can be established in multiple ways taking into account different factors like the dimensions and orientation of the UAV, errors introduced in the discretization of the obstacle map, the velocity of the UAV as well as communication times. Static operational safety bounds and dynamic operational safety bounds are two concepts implemented for UAV operations. The key difference in implementation of a static operational safety bound is it does not need any specific knowledge about the UAV operating in the airspace. Dynamic operational safety bounds use the velocity information for the UAV and wind data for safety bound calculation (D'Souza et al., 2016; Zhu & Wei, 2016).

Recently, Hu et al. (2020) proposed a dynamic operational safety bound method that also includes more information about the UAV (such as performance data and system updating information taking into account communication standards). A warning zone for collisions form the proposed dynamic operational safety bound. The collision zone is defined by considering the minimum stopping distance required for the UAV to decelerate until it can hover. The warning zone takes into account the sampling time to expand the collision zone. Moreover, the bound considers the vectorized shape of a UAV with different speed components that are parallel and perpendicular to the heading direction. This shows how the isotropic design reduces the capacity of the airspace. It is also interesting to see that their proposed dynamic operational safety bound is generally longer in the direction parallel to the heading direction. This can be attributed to the large speed component along the heading direction.

the DA bound proposed by (Hu et al., 2020) can be integrated into the decision-making and path-planning framework pre-

sented in this paper. This will take into account more realistic scenarios. However, to show the feasibility of the proposed framework, a static bound was used for the experiments reported in this paper.

5.3.2. Risk of crashing due to loss of battery power

The risk of running out of battery power is computed for an existing trajectory or a new trajectory by considering the uncertainties associated with the state of charge of each battery cell in the battery pack. This is then translated to the remaining useful life (RUL) of the UAV as a function of the state of charge of the battery.

The RUL estimation encompasses two distinct but related problems: (1) estimating the current system state and the degradation rates of individual components; and (2) predicting future system performance to derive a system RUL function. To compute the RUL, we follow an analytic framework that tracks the degradation rate of individual components, such as the battery and motors from mission to mission. Based on the degradation parameters estimated at the beginning of a mission, we use methods to compute and predict the state of charge over a specified trajectory. The system-level prediction to compute the system RUL_n distribution function at time instant n defined by:

$$RUL_n = (EOL_n - n)\Delta t, \quad (21)$$

where Δt is the sampling time of the system, $n \in N$ represents the sampling step, and EOL_n represents the end of life of the system defined by:

$$EOL_n = \inf\{z \in Z : z \geq n \ \& \ T(\varphi_z) = 1\}, \quad (22)$$

where $T(\varphi_z)$ is the following system performance threshold function

$$T(\varphi_z) = \begin{cases} 1 & \text{if } 0 \in R(\varphi_z) \\ 0 & \text{if } 0 \notin R(\varphi_z) \end{cases} \quad (23)$$

$R(\varphi_z)$ denotes a set of performance constraint functions that map the performance of the system to a Boolean domain $[0, 1]$. A value of 1 implies that the constraint is satisfied, and a value of 0 implies that it is not. This set of constraints is defined based on the system degradation model but a single constraint is defined in our case based on the state of charge of the battery.

The risk of not completing the mission is calculated at each time step n as:

$$F_{rul}(\Gamma_{ij}) = p(RUL_n < t_{end}), \quad (24)$$

where $t_{end} = N * \Delta t$ represents the time planned for the UAV to reach the final way-point. We calculate F_{rul}^{max} as the maximum of F_{rul} .

Note that the probability distribution of RUL_n , as well as the

accumulated and maximum risk ($Risk_n^{tot}$, $Risk_n^{max}$) associated with the current trajectory are calculated considering the distribution of the system states, nominal and degrading parameters, $p(\mathbf{x}_n, \vartheta_n, \alpha_n | \mathbf{y}_n)$, the distribution of future inputs to the system, and the distribution of the system model uncertainties.

5.4. Trajectory re-planning

In this work, our proposed re-planning strategy does not address the UAV take-off and landing phases. Our re-planning strategy is only applied to in-flight way-point adjustments. We assume the UAV flies at a constant altitude (constant z value), therefore, the flight coordinates of interest for optimization are the two continuous variables that define the x - y motion of the UAV. Once the UAV arrives at the landing spot way-point in the x - y plane, we consider that it enters into a different operating mode, and a different planning strategy is required to implement safe landing. This may involve other sensors, such as visual sensors (Lusk et al., 2019).

The trajectory re-planning module is activated when the mission failure assessment module determines that the risk of mission failure is higher than a pre-specified threshold $\tau_{fail} \in [0, 1]$. Trajectory re-planning is solved as an optimization problem that is constrained by the time to failure along the trajectory. The solution of the re-planning problem depends on whether the risk of mission failure is caused by the collision with an obstacle or the UAV running out of battery power. We formalize the solution to each problem next.

5.4.1. Trajectory re-planning to reduce the risk of collision with an obstacle

The trajectory re-planning routine is activated when the risk threshold τ_{fail} is violated for a segment of the remaining trajectory ($F_c^{max}(\Gamma) > \tau_{fail}$). The re-planning optimization problem can be formalized as follows:

$$\min_{\Gamma_{ij} \in \Gamma_{ij}} (f_c^{max}(\Gamma_{ij})), \quad (25)$$

where Γ_{ij} represents a trajectory between way-point c_i and c_j , Γ_{ij} represents the set of possible trajectories between the two way-points, the function f_c^{max} estimates the maximum probability of mission failure along Γ_{ij} . A sufficient stopping condition for this optimization problem is $f_c^{max}(\Gamma_{ij}) < \tau_{safe}$ with $\tau_{safe} \in [0, 1]$ representing a threshold for the maximum allowed probability of collision. In the context of dynamic systems, the optimization problem above is subject to the following constraints:

- Initial state of the system:

$$\{\mathbf{x}_{n=0}^{ij}, \vartheta_{n=0}^{ij}, \mathbf{w}_x^{ij}\} = \{\hat{\mathbf{x}}, \hat{\vartheta}, \hat{\mathbf{w}}_x\}, \quad (26)$$

where $\hat{\mathbf{x}}$, $\hat{\vartheta}$, and $\hat{\mathbf{w}}_x$ represent the estimated state, degradation parameters, and model uncertainties at $n = 0$.

This constraint defines the initial condition of the system to solve the optimization problem.

- Dynamics of the system:

$$\mathbf{x}_{n+1}^{ij} = f(\mathbf{x}_n^{ij}, \mathbf{u}_n^{ij}, \vartheta_n^{ij}, \mathbf{w}_x^{ij}), \quad (27)$$

This constraint means that the system dynamics must be satisfied between any two way-points i and j to be flown.

- Time constraint:

$$T_{sol} < T_{avail}, \quad (28)$$

This constraint means that a solution must be found in the time frame defined by T_{avail} . We discuss the selection of T_{avail} in the experimental section of this paper.

Note that the latter constraint is defined by the time to collision in the trajectory, which is estimated using the prediction module. Therefore, a trajectory that reduces the risk of collision below an acceptable threshold level must be found in the available time before the collision occurs. Moreover, the expected communication delay has to be considered with the estimated time to collision. This dictates the parameterization of the optimization algorithm to satisfy computational time requirements. Alternatively, if the estimated computation time exceeds the allowable computation time, an alternate landing trajectory must be generated. In other words, if the optimization problem cannot be solved in the given time, the UAV must perform an emergency landing at the closest available landing spot. To accommodate these constraints, we propose a Differential Evolution optimization algorithm Storn & Price (1995) to formalize the path replanning problem as an *anytime* optimization algorithm. Details of our proposed approach are presented in the next Section.

5.4.2. Trajectory re-planning to reduce the risk of running out of battery charge

The trajectory re-planning routine is activated when the risk of running out of battery power is higher than a pre-specified failure threshold ($F_c^{max} > \tau_{fail}$). Re-planning, in this case, involves finding the nearest available emergency landing spot from the current position of the UAV, and deriving a dynamically feasible trajectory to it. For emergency landing, a set of available landing spots and their coordinates should be known beforehand.

The mission time to failure, in this case, depends on the available battery power. Therefore, T_{avail} is approximated by the prediction module. The re-planning is performed as a two-step procedure. In the first step, the shortest trajectory between the current position of the UAV and a chosen landing spot is computed by using the *probabilistic roadmap* path planning algorithm. In the second step, this trajectory is transformed into a smooth and dynamically feasible trajectory by using the B-spline curve approximation method. Additional

details on these two methods are presented in the next section.

6. OPTIMIZATION METHODS

We propose to use different methods to solve the two re-planning tasks presented in the previous section. First, Differential Evolution to reduce the risk of collision with obstacles, and second, a combination of Probabilistic Roadmap path planning with the generation of polynomial trajectories using B-splines to reduce the risk of running out of battery power. We explain the algorithms and implementation details in the following sections.

6.1. Re-planning to reduce the risk of collision

We solve the re-planning optimization task that was formalized as equations (25)–(28) as an inverse problem using Differential Evolution (DE). DE is an optimization algorithm that is applicable to "non-differentiable continuous space functions", (Storn & Price (1995)). A number of studies have compared meta-heuristic algorithms like DE with classic path planning methods, or convex optimization approaches (Aggarwal & Kumar, 2020). The main advantage of this algorithm is that the evaluation of the candidate solutions can be parallelized, and after the first iteration of a single member of the population, we can measure approximately the time it takes to complete an epoch and adjust the maximum number of iterations allowed (based on the time available for re-planning).

The goal of this re-planning task is to update the planned trajectory by modifying the reference coordinates that the UAV must follow. Therefore, the decision variables β for the DE method are C intermediate reference way-points $\Gamma_{ij} = \{c_{i+1}, c_{i+2}, \dots, c_{i=j}\}$ that define the trajectory between the current location of the UAV and the final way-point. In this work, we assume the UAV flies at a pre-defined altitude, therefore, the flight coordinates of interest are the two continuous variables $c_i = x_i, y_i$.

The general idea behind DE is to generate new solutions by varying the properties of the members of the population by repetition of three operators: (1) mutation, (2) crossing, and (3) selection till convergence to an optimal solution (Das et al. (2016)). The solution model is usually expressed with the notation $DE/p/q/r$ (p : type of the mutation vector, q : number of pairs of solutions to be used in order to vary the current solution, and r : distribution function that will be used during the crossover operation). We used the configuration $DE/rand/1/bin$ in this paper, where a random vector ($p = rand$) and the difference of a vector pair ($q = 1$) were applied to generate a mutation vector. The binomial crossover model ($r = bin$) is used for generating a crossover between two solutions. Besides this configuration, the parameters that are adjusted for the DE method are: (1) a matrix that holds

the upper and lower bounds for the values that each decision variable can take; (2) $\Delta \in \mathbb{R}^{2C \times 2}$, the population size that defines the number of candidate solutions to the optimization problem η ; (3) the mutation factor μ ; (4) the crossover rate γ ; and (5) the number of iterations (epochs) for the algorithm α .

We set a crossover rate of 0.9 to ensure a low influence of the current population solution on the newly generated ones, and a mutation factor of 0.6 to allow for the exploration of new solutions. The stopping criteria established are based on the optimization function threshold τ_{safe} and the maximum number of iterations. If the risk threshold is satisfied the algorithm stops. The maximum number of iterations on the other hand is defined by a percentage of the time available to find a new solution T_{avail} . One iteration of the algorithm must be run to define the approximate time it takes the evolution of one generation, such that the maximum number of iterations can be defined properly. Another advantage of this algorithm is that the evaluation of the candidate solutions (which is the operation that demands most computational power) can be run in parallel. The population of individuals is set to 20.

The evaluation of the candidate solutions is implemented as a dynamic simulation of the UAV for its remaining trajectory to derive $f_c^{max}(\Gamma_{ij})$. The larger the trajectory to be re-planned the longer it will take the simulation to converge to “good” solutions. Since we are implementing an online re-planning problem, it is much more efficient to re-plan the segments of the trajectory associated with high risk of collision; in other words, we find a new trajectory from the current point to the next “safe” way point. If the risk of collision is high for multiple segments of the trajectory the re-planning can be applied sequentially to each segment starting from the one closest to the current position of the UAV. In this way, the trajectory re-planning can be dynamically generated while the UAV is flying, but the overall solution, while safe, may not be globally optimal.

6.2. Re-planning to reduce the risk of running out of battery power

The trajectory re-planning routine to reduce the risk of running out of battery power is a three-step procedure. First, a dynamically feasible path between a future reference coordinate in the remaining trajectory and an emergency landing spot must be computed. To accomplish this task, we first generate the shortest collision-free trajectory using Probabilistic Roadmap path planning (PRM) algorithm (Kavraki et al. (1996)) and transform that trajectory into a dynamically smooth trajectory using B-splines (Farin (2014)). Second, we evaluate each possible trajectory by performing a stochastic simulation considering the current UAV and environmental conditions. We select the shortest trajectory that meets the condition $F_c^{max} < \tau_{fail}$.

A probabilistic roadmap is a graph structure, where the nodes

represent *collision-free* way-points and the edges represent a realizable path between these way-points (Kavraki et al. (1996)). The PRM graph generates random samples from the configuration space and tests them for whether they represent feasible paths. The graph is created by connecting the set of random samples typically using the *k-nearest neighbors* method. Only the connections smaller than a predetermined distance in the free space are kept. This is the first phase of the algorithm, which is executed off-line based on the navigation map. In the query phase, the start and goal coordinates are connected to the closest nodes of the graph, and the path is obtained by using Dijkstra’s shortest path algorithm.

The PRM algorithm extends the capability of similar algorithms, such as rapidly-exploring random trees because it does not generate a new graph when a new node is discovered. One advantage of this algorithm is that it is provably probabilistically complete (as the number of samples increases without bound, the probability that the algorithm will find a path if one exists approaches 1). A thorough evaluation of this algorithm is provided in Hsu et al. (2006).

B-spline curves are used for generating a smooth trajectory based on the way-points derived by the PRM method for a desired cruise speed. A B-spline curve can be defined as a piece-wise curve, where each component is of degree k . This method has been widely used in robotic applications because of its desirable properties of convex hull and maintaining continuity up to the $k + 1$ derivative for a curve of degree k . Specifically, we considered cubic B-splines (Farin (2014)) with a desired cruise speed of 1 m/s.

7. EXPERIMENTAL STUDY

We demonstrate the feasibility of the proposed online re-planning framework for two different re-planning scenarios. For these experiments, we operated the GPS and IMU systems at sampling rates of 1 Hz and 100 Hz, respectively. The corresponding accuracy of the position and velocity of these devices were 3 m and 0.1 m/s, respectively. An example of the accuracy of the SPKF state estimator for the estimation of the UAV position is shown in Figure 5. The standard deviation of the state error covariance matrix corresponding to the position estimates is used to define the distribution of $D_{obst_k}^n$ when calculating the risk of collision with obstacles.

Because of the Euler angles formulation, singularity problems might occur in the rotation matrix. This phenomenon is known as gimbal lock (Diebel, 2006). Gimbal lock occurs when two rotation axes align, thus losing one degree of freedom. Under normal operating conditions, the octocopter should not experience such a large pitch. However, to ensure that this phenomenon does not occur in our simulations we saturate the output of the position controller (in the cascade control scheme) to guarantee a smooth pitch reference. This requires smooth reference trajectories and we ensure that by

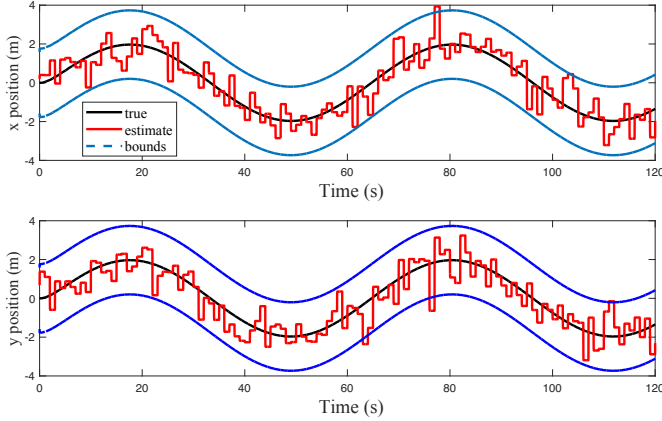


Figure 5. Position state estimates

setting the desired reference speed of the vehicle to 1 m/s when running the B-splines smoothing method. If more aggressive trajectories were to be simulated, we would switch to the quaternion-based representation (Kehlenbeck, 2014).

Figure 6 shows the initial trajectory between the start and goal points chosen by the UAV under nominal wind conditions (indicated by the red and yellow dots in the figure). Figure 7 plots the computed risk of collision for each time step. We considered a $dsafe_k = 1m$ considering the octocopter dimensions and set the maximum allowed risk of mission failure $\tau_{fail} = 0.8$. Therefore, re-planning is not required given that the maximum risk does not go over 0.4.

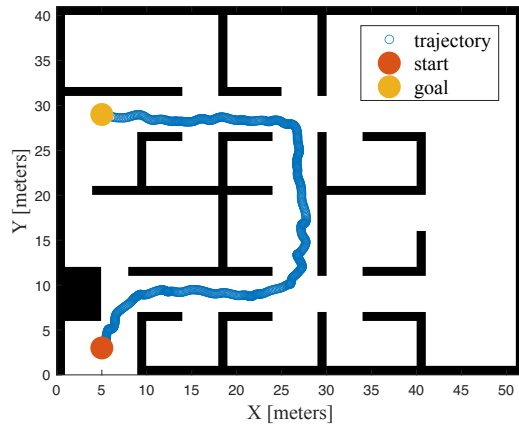


Figure 6. Example of trajectory under nominal conditions

Next, we show an example, where the risk of mission failure went above the threshold because of the possibility of collision. This situation occurs after taking off when a fault occurs on BLDC motor 2 (see Figure 1). The fault consists on the increase in the coil resistance, which results in a loss of power. After the fault occurs, the trajectory predicted by the prediction module and the risk of collision are depicted in Figures

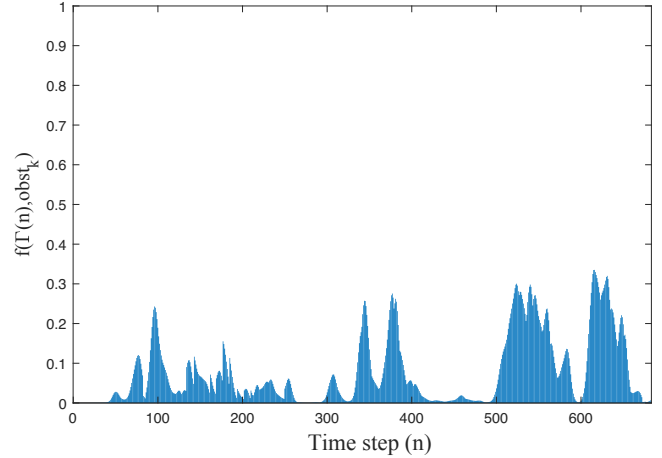


Figure 7. Risk of collision with the closest obstacle at each time step under nominal conditions

8 and 9, respectively. Since the risk threshold is exceeded, re-planning of the trajectory is initiated.

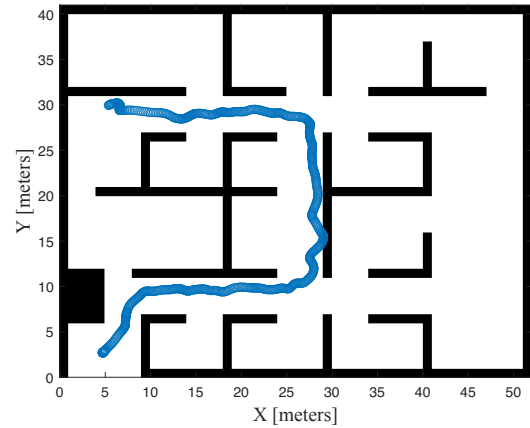


Figure 8. Example of trajectory with motor fault

The differential evolution algorithm is applied sequentially to the reference trajectory coordinates to compute a new trajectory. Several parts of the remaining trajectory are updated as shown in Figure 10, which plots the old and the new trajectory. The result shows that the trajectory is modified in the opposite direction of the obstacles. Figure 11 shows the trajectory followed by the UAV considering the new reference coordinates. The risk associated with this trajectory is shown in Figure 12. As shown, the risk of collision goes below the safety threshold, $\tau_{safe} = 0.6$. The goal of the re-planning strategy is to reduce the risk to acceptable levels such that the probability of collision stays below a certain threshold. Although re-planning should not increase risk (as it occurs between time step 80 and 100), a compromise has to be established when the UAV flights in a degraded condition because

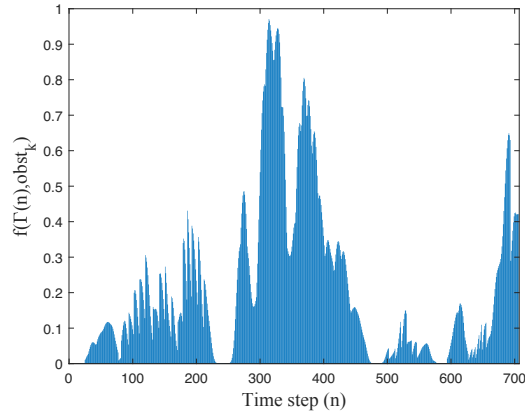


Figure 9. Risk of collision with the closest obstacle at each time step under motor fault

it is not possible to update a segment of the trajectory without affecting the other segments.

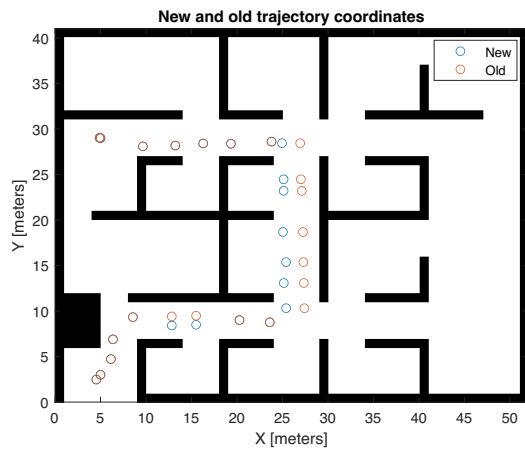


Figure 10. Old and new trajectory coordinates

In the second example, trajectory re-planning occurs when the risk of mission failure exceeds the pre-specified threshold. This increased risk is attributed to low battery charge. An example of the accuracy of the SPKF state estimator for the estimation of the battery voltage and state of charge through a complete discharge cycle is shown in Figure 13. The prediction module predicts the RUL depending on the battery *SoC* by propagating the future inputs and considering the uncertainty estimates at each time step.

The re-planning routine is triggered when the time-to-failure is close to the time required to complete the mission, and the corresponding risk of failure (i.e., crashing) exceeds the predefined threshold. This situation occurs after the UAV has covered part of the mission trajectory shown in Figure 14 (blue trajectory). To avoid a crash, the UAV has to land at the closest emergency landing site. We assume hovering is pos-

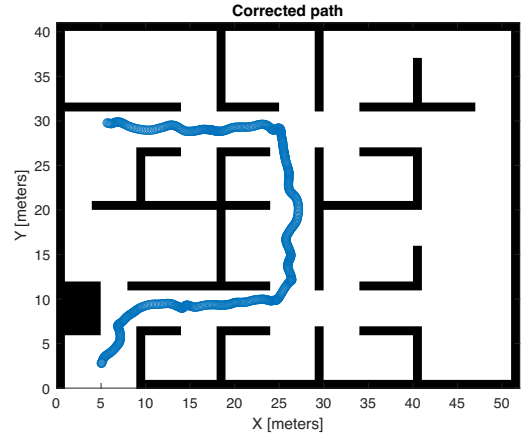


Figure 11. Example of trajectory under motor fault with corrected trajectory

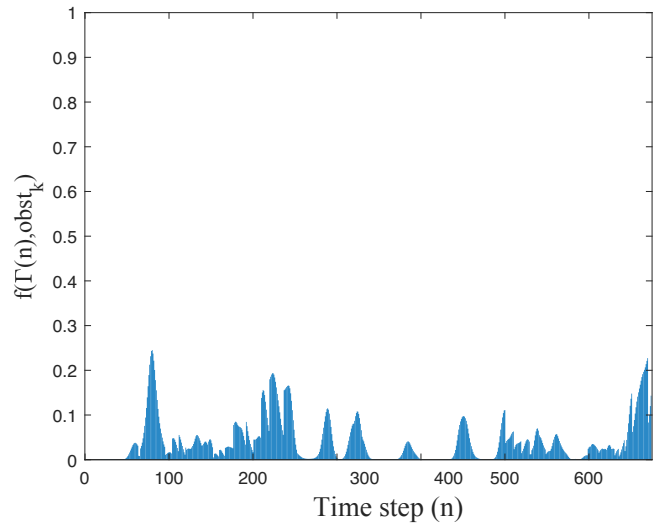


Figure 12. Risk of collision with the closest obstacle at each time step under motor fault with new trajectory

sible, and calculate a new trajectory using the PRM planner (which computes off-board) from the current UAV location to the nearest emergency landing spot. A time-stamped smooth trajectory is obtained by using cubic B-splines on the reference coordinates. The realization of the replanned trajectory is shown in Figure 14 (green trajectory).

Discussion

The goal of the proposed re-planning approach is to reduce the likelihood of collisions to an acceptable level. If this is not achievable then an emergency landing spot has to be identified, and a new trajectory is planned. The allowed computation time T_{avail} depends on the communication latency and the time-to-collision (approximate flying time from the current position of the UAV to the set of continuous way-points

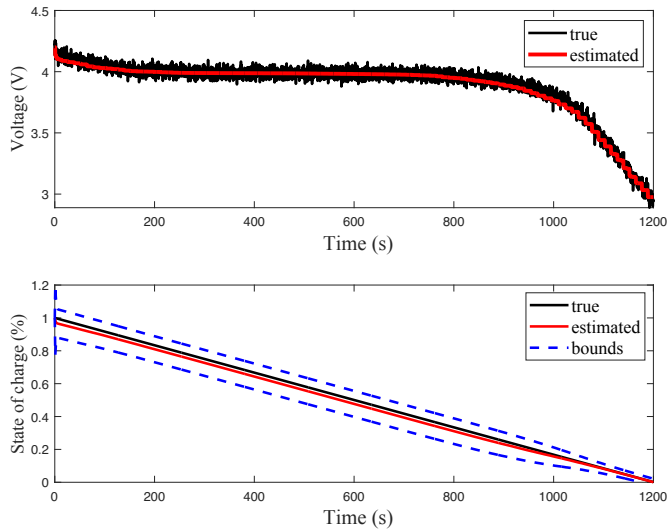


Figure 13. Battery voltage and state of charge estimates during a discharge cycle

during which the probability of collision is not acceptable. Therefore, re-planning for updating the current trajectory and completing the mission will be possible if the segment to be corrected is far enough. Moreover, if the allowed computation time is feasible, achieving the desired minimum probability of collision will depend on the hardware on which DE is running. The primary computation effort of the optimization method can be attributed to the evaluation of the cost function, where the dynamics of the UAV are simulated. This evaluation (for the DE algorithm) can be parallelized using a vector population, where the stochastic perturbation of the population vectors happens independently. If there are segments (between pairs of waypoints) in the future trajectory for which the probability of collision is not acceptable, each segment will be re-planned sequentially starting with the one closest to the current position of the UAV. The larger the segment the larger the computational cost. In this work, the feasibility of the proposed approach is demonstrated, but no validation steps are conducted. This would require real-time experiments. We will conduct them in future work.

8. CONCLUSIONS AND FUTURE WORK

A decision-making and path planning framework to support safe low-flying urban operations of small load-bearing unmanned aerial vehicles (UAV) flights has been proposed in this work. The core of the proposed approach includes four modules: (1) state estimation, (2) prediction, (3) mission failure assessment, and (4) trajectory replanning. The first module comprises a set of estimators for the state variables that characterize the behavior of the UAV. We specifically estimate the pose of the UAV and state of charge of the battery because these are important variables that determine the success of a mission. The prediction module propagates the es-

timated states, considering the uncertainty bounds to determine the behavior of the UAV along the remaining trajectory at each time step. The mission failure assessment module performs risk analysis based on the predicted behavior and activates the trajectory replanning routine if the risk of flight exceeds a pre-defined threshold. The trajectory replanning module triggers one of two possible routines depending on the cause, i.e., (1) a high risk of collision or (2) insufficient battery charge to complete the mission.

We tested the proposed framework using a realistic octocopter model and incorporated environmental (primarily wind) conditions in our analyses. The proposed approach allows us to compute new trajectories that reduce the risk of mission failure to acceptable values. In the future, we plan to investigate more sophisticated planning approaches based on Markov Decision Processes (MDPs). We also plan to integrate a system-level prognosis module Khorasgani et al. (2016) that allows to track the degradation of individual components on the UAV, and determines the RUL of the UAV system based on performance parameters. The system-level prognostics algorithms will also allow us to track UAV state of health across multiple missions.

ACKNOWLEDGMENT

This research has been supported by grant # 80NSSC19M0166 from the NASA Shared Services Center.

REFERENCES

- Abramov, I., Nikitin, Y., Abramov, A., Sosnovich, E., & Bozek, P. (2014). Control and diagnostic model of brushless dc motor. *Journal of Electrical Engineering*, 65(5).
- Aggarwal, S., & Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 149(July 2019), 270–299.
- Al-mashhadani, M. A. (2019). Optimal control and state estimation for unmanned aerial vehicle under random vibration and uncertainty. *Measurement and Control*, 1–8. doi: 10.1177/0020294019866860
- Ancel, E., Capristan, F. M., & Foster, J. V. (2017). Real-time Risk Assessment Framework for Unmanned Aircraft System (UAS) Traffic Management (UTM). In *17th aiaa aviation technology, integration, and operations conference* (p. 3273). doi: 10.2514/6.2017-3273
- Ancel, E., Capristan, F. M., & Foster, J. V. (2019). In-Time Non-Participant Casualty Risk Assessment to Support On-board Decision Making for Autonomous Unmanned Air-

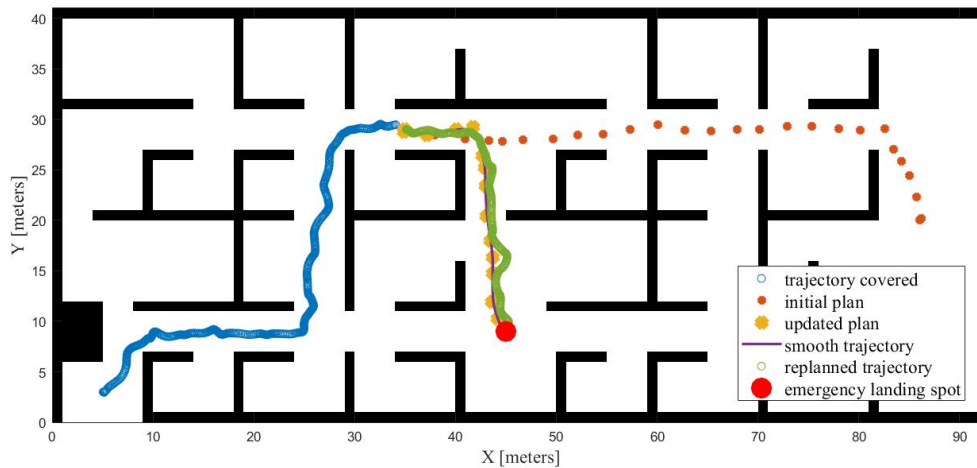


Figure 14. Trajectory re-planning scenario when the risk of running out of battery is high

- craft. In *Aiaa aviation 2019 forum* (p. 3063). doi: 10.2514/6.2019-3053
- Benders, S., Schopferer, S., & Nawrath, A. (2020). In-flight Kinematic Model Parameter Estimation and Adaptive Path Planning for Unmanned Aircraft. In *Aiaa scitech 2020 forum* (p. 0.135). doi: 10.2514/6.2020-0135
- Brown, A., & Rogers, J. (2016). A sampling-based probabilistic path planner for multirotor air vehicles in cluttered environments. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 231. doi: 10.1177/0954410016636912
- Burri, M., Manuel, D., Achtelik, M. W., & Siegwart, R. (2015). Robust State Estimation for Micro Aerial Vehicles Based on System Dynamics. In *Ieee international conference on robotics and automation (icra)* (pp. 5278–5283). Washington, D.C.
- Chatterjee, A., & Reza, H. (2019). Path planning algorithm to enable low altitude delivery drones at the city scale. In *2019 international conference on computational science and computational intelligence (csci)* (pp. 750–753).
- Clothier, R. A., & Walker, R. A. (2006). Determination and evaluation of uav safety objectives.
- Conte, G., & Doherty, P. (2008). An integrated uav navigation system based on aerial image matching. In *2008 ieee aerospace conference* (pp. 1–10).
- Corke, P., Lobo, J., & Dias, J. (2007). An Introduction to Inertial and Visual Sensing. *The International Journal of Robotics Research*, 26, 519–535. doi: 10.1177/0278364907079279
- Coutinho, W., Battarra, M., & Fliege, J. (2018, 04). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*, 120. doi: 10.1016/j.cie.2018.04.037
- Daigle, M., & Kulkarni, C. (2016). End-of-discharge and end-of-life prediction in lithium-ion batteries with electrochemistry-based aging models. *AIAA Scitech Forum*.
- Das, S., Mullick, S., & Suganthan, P. (2016). Recent advances in differential evolution an updated survey. *Swarm and Evolutionary Computation*, 27. doi: https://doi.org/10.1016/j.swevo.2016.01.004
- De Filippis, L., Guglieri, G., & Quagliotti, F. (2011). A minimum risk approach for path planning of uavs. *Journal of Intelligent and Robotic Systems*, 61, 203-219. doi: 10.1007/s10846-010-9493-9
- Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 15–16(58), 1–35.
- D'Souza, S., Ishihara, A., Nikaido, B., & Haseeb, H. (2016). Feasibility of varying geo-fence around an unmanned aircraft operation based on vehicle performance and wind.

- In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)* (pp. 1–10).
- Farin, G. (2014). *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Elsevier.
- Gougeon, O., Nguyen, D.-t., & E, D. S. (2018). Modeling and Control of a Quadcopter Flying in a Wind Field: A Comparison Between LQR and Structured H Control Techniques. *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*(1), 1408–1417. doi: 10.1109/ICUAS.2018.8453402
- Gross, J., Gu, Y., Gururajan, S., Seanor, B., & Napolitano, M. R. (2010). A Comparison of Extended Kalman Filter, Sigma-Point Kalman Filter, and Particle Filter in GPS/INS Sensor Fusion. In *Aiaa guidance, navigation, and control conference* (p. 8332).
- Hsu, D., Latombe, J.-C., & Kurniawati, H. (2006). On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research*, 25. doi: 10.1177/0278364906067174
- Hu, J., Erzberger, H., Goebel, K., & Liu, Y. (2020). Probabilistic Risk-Based Operational Safety Bound for Rotary-Wing Unmanned Aircraft Systems Traffic Management. *Journal of Aerospace Information Systems*, 17(3), 171–181.
- Kavraki, L. E., Svestka, P., Latombe, J. ., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12.
- Kehlenbeck, A. G. (2014). *Quaternion-based control for aggressive trajectory tracking with a micro-quadrotor uav* (Unpublished doctoral dissertation).
- Khorasgani, H., Biswas, G., & Sankararaman, S. (2016). Methodologies for system-level remaining useful life prediction. *Reliability Engineering & System Safety*, 154, 8–18.
- Krishnan, P. S., & Manimala, K. (2020). Implementation of optimized dynamic trajectory modification algorithm to avoid obstacles for secure navigation of UAV. *Applied Soft Computing*, 90, 106168.
- Kuchar, J. K. (2005). Safety analysis methodology for unmanned aerial vehicle (uav) collision avoidance systems. In *Usa/europe air traffic management r&d seminars* (Vol. 12).
- Kulkarni, C., & Corbetta, M. (2019). Health management and prognostics for electric aircraft powertrain. *AIAA*.
- Lin, C., & Shao, P.-C. (2020). Failure analysis for an unmanned aerial vehicle using safe path planning. *Journal of Aerospace Information Systems*, 1-12. doi: 10.2514/1.I010795
- Lusk, P. C., Glaab, P. C., Glaab, L. J., & Beard, R. W. (2019). Safe2Ditch: Emergency Landing for Small Unmanned Aircraft Systems. *Journal of Aerospace Information Systems*, 19(8), 327–329.
- Mahony, R., Kumar, V., & Corke, P. (2012). Modeling, Estimation, and Control of Quadrotor. (August).
- Merwe, R. V. D., Wan, E. A., & Julier, S. I. (2004). Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation. In *Aiaa guidance, navigation, and control conference and exhibit* (p. 5120). Washington, D.C.
- Moir, I., & Seabridge, A. (2012). *Design and development of aircraft systems* (Vol. 67). John Wiley & Sons.
- Plett, G. L. (2006). Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2: Simultaneous state and parameter estimation. *Journal of Power Resources*, 161, 1369–1384.
- Plett, G. L. (2015). *Battery Management Systems Volume 2: Equivalent-Circuit Methods*. Artech House.
- Powers, C., Mellinger, D., & Kumkar, V. (2015). Quadcopter kinematics and dynamics. *Handbook of Unmanned Aerial Vehicles*.
- Primates, S., Guglieri, G., & Rizzo, A. (2019). A risk-aware path planning strategy for UAVs in urban environments. *Journal of Intelligent & Robotic Systems*, 95(2), 629–643.
- Rubio-Hervas, J., Gupta, A., & Ong, Y.-S. (2018). Data-driven risk assessment and multicriteria optimization of uav operations. *Aerospace Science and Technology*, 77, 510 - 523. doi: 10.1016/j.ast.2018.04.001
- Schacht-Rodríguez, R., Ponsart, J. C., García-Beltrán, C. D., Astorga-Zaragoza, C. M., Theilliol, D., & Zhang, Y. (2018). Path Planning Generation Algorithm for a Class of UAV Multirotor Based on State of Health of Lithium Polymer Battery. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 91(1), 115–131.
- Sierra, G., Orchard, M., Goebel, K., & Kulkarni, C. (2019). Battery health management for small-size rotary-wing

- electric unmanned aerial vehicles: An efficient approach for constrained computing platforms. *Reliability Engineering & System Safety*, 182, 166–178.
- Storn, R., & Price, K. (1995). Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11. doi: <https://doi.org/10.1023/A:1008202821328>
- Yang, H., Lee, Y., & Dongjun, S.-y. J. (2017). Multi-rotor drone tutorial: systems , mechanics , control and state estimation. *Intelligent Service Robotics*, 10(2), 79–93. doi: 10.1007/s11370-017-0224-y
- Zammit, C., & Van Kampen, E.-J. (2020). Comparison of a* and rrt in real-time 3d path planning of uavs. In *Aiaa scitech 2020 forum* (p. 0861).
- Zhu, G., & Wei, P. (2016). Low-altitude UAS traffic coordination with dynamic geofencing. In *16th aiaa aviation technology, integration, and operations conference* (p. 3453).