

# An Integrated Risk-Based Strategy for Real-Time Decision Making and Mission Reconfiguration in Unmanned Aerial Vehicles

Jorge Fabry<sup>1</sup>, Ignacio Carvajal<sup>2</sup>, Heraldito Rozas<sup>3</sup>, Marcos Orchard<sup>4</sup>, Ferhat Tamssaouet<sup>5</sup>, Khanh T.P. Nguyen<sup>6</sup> and Kamal Medjaher<sup>7</sup>

<sup>1,2,3,4</sup> *University of Chile, Santiago, Metropolitan Region, 8370451, Chile*

*jorge.fabry@ug.uchile.cl*  
*ignacio.carvajal@ug.uchile.cl*  
*heraldo.rozas@ing.uchile.cl*  
*morchard@u.uchile.cl*

<sup>5,6,7</sup> *Laboratoire Génie de Production, INP-ENIT, Université de Toulouse, Tarbes, 65000, France*

*ferhat.tamssaouet@enit.fr*  
*thi-phuong-khanh.nguyen@enit.fr*  
*kamal.medjaher@enit.fr*

## ABSTRACT

Small unmanned aerial vehicles (UAVs) have been increasingly popular in the last years, being employed in a wide range of applications in diverse areas, including, for instance, military, medicine, and package delivery. These aerial vehicles are commonly energized by rechargeable batteries, and, as a result, their autonomy can be significantly affected by several uncertainty sources, including variable ambient conditions or mechanical mishaps. For this reason, the ability to predict their electric consumption for a determined path is critical to design UAV missions and complete them successfully. Additionally, due to the potential occurrence of unexpected events (e.g., mechanical failures or changes in the weather conditions), airborne implementation of real-time decision-making schemes for mission replanning is of utmost importance. Hence, this paper presents an integrated Risk-based strategy utilizing a State-of-Charge (SOC) prognostics algorithm to quantify the risk associated with a given path in terms of future consumption, intending to make real-time decisions on the UAV operation. More specifically, we consider a UAV mission framework with a discrete set of possible targets, where each one has costs, rewards, and failure risks, which is characterized by calculating the total probability using sample-resampling methods. Then, we seek the optimal choice by employing a Bayesian-Risk inspired approach whose outcome results in a compromise between risks and rewards.

---

Jorge Fabry 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.

## 1. MOTIVATION AND PROBLEM STATEMENT

During the last decade, the utilization of Unmanned Aerial Vehicles (UAVs) in diverse areas has increased significantly, from civil and commercial to military applications. Technical advances and design improvements have allowed manufacturers to offer low-cost UAVs with high maneuverability. Nevertheless, and independently of the specific application domain where the UAV is being used, flight range autonomy still looms as a problem of great importance. In fact, the majority of UAVs are powered by Lithium-Ion batteries; thus, depending on the application, the flight time of such vehicles may be a serious issue that negatively impacts the mission's reliability. To tackle this problem, the coordination and management of the UAV fleets are of utmost relevance. In this regard, since prognostic algorithms have been successfully implemented in the time of discharge prediction problems for Li-Ion batteries (Saha et al., 2011; Saha, Quach, & Goebel, 2012; Kulkarni, Hogge, & Quach, 2018; Sierra, Orchard, Goebel, & Kulkarni, 2019), Prognostic Decision Making (PDM) strategies arise as a suitable platform to design and implement customized solutions for UAV fleets coordination. Consequently, and as a first step, in this paper, we explore the utilization of a real-time PDM scheme to manage the operation of a single UAV. The insights developed on this work will be useful as foundations so that, in future efforts, we can design PDM solutions for UAVs coordination at the fleet level, which is our ultimate goal.

To be more specific, in this article, we explore a PDM scheme oriented to provide, at each execution, a binary decision  $x_t \in \{1, 0\}$ , namely, the UAV "should either proceed to the next



Figure 1. Path plan over O'Higgins park in Santiago, Chile.

waypoint according to the mission plan or return to base". The underlying idea is that the UAV has an initial flight plan (calculated before its departure) with several waypoints, which will be visited sequentially according to the mission path. However, due to changes in the direction and intensity of the wind, air turbulence, or even the occurrence of mechanical failures, the energy consumption along the path could differ significantly from the initial guess. Therefore, computing mid-flight replanning is essential to keep the reliability standards of the mission. To accomplish this, the proposed decision-making algorithm must have low latency so as to guarantee that the decision update is computed before that new step begins.

The underlying motivation of investigating real-time sequential decisions in the UAV domain comes from 3D mapping applications similar to the one depicted in (Yamazaki, Miyazaki, & Liu, 2018), wherein the UAV requires to scan a predefined area, at a constant velocity, following a series of waypoints to reconstruct a virtual three-dimensional map. Figure 1 illustrates an example of a mission plan for a rotatory-wing UAV, where the mission assigns to each waypoint a different reward.

To detail this optimization problem and the solution proposed in this article, the following structure is adopted. Section 2 presents a brief analysis of state of the art on prognostics decision-making algorithms for mission reconfiguration and basic theoretical background on particle-filtering-based

prognostics, which is used to predict the evolution of the battery SOC in the UAV. Section 3 introduces the proposed risk-based methodology for mission reconfiguration. Finally, Section 4 describes the case study that is considered in this research effort, while Section 5 presents the most relevant results obtained. Section 6 summarizes the main conclusions.

## 2. THEORETICAL BACKGROUND

In this section, we first discuss previous advances associated with PDM strategies (Section 2.1), and then review a technique called Particle-Filtering-based prognostic (Section 2.2), which plays a major role in the proposed PDM approach.

### 2.1. Prognostic Decision-Making Algorithms for Mission Reconfiguration

Currently, Lithium-Ion (Li-Ion) batteries are key components from smartphones to UAVs. One of the most relevant variables into the operation of batteries is the State-of-Charge (SOC), which is typically defined as the ratio between the available energy and the total capacity. It is, usually, described as a percentage (Pola et al., 2015). Consequently, knowledge about this state variable is essential for online decision making, for instance, to assess if a path can be traversed or to verify which mission goal(s) can be accomplished.

Both SOC estimation and SOC prognostic strategies are fundamental to estimate the End of Discharge (EOD) time. Unfortunately, as in many other state estimation problems, the SOC is not observable; thus, it has to be inferred from indirect but partially correlated measurements (e.g., the battery voltage, discharge current, or temperature) (Pola et al., 2015; Diaz et al., 2020).

Even more important than estimations and predictions of SOC, it is what to do with these results. It was not until a few years back that literature started offering contributions that aimed at the implementation of Post-Prognostics Decision Making (H. Skima & Bourgeois, 2019) or Prognostics Decision Making (PDM) (Rozas, Munos-Carpintero, Perez, Medjaher, & Orchard, 2018) strategies. It could leverage failure prognostic outcomes to efficiently make decisions about the system operation, such as optimizing preventive and predictive maintenance action scheduling, mission re-planning, or mission redesign. Although the PDM strategies have great potential for optimizing process operation, the pace of recent developments has not coped with the actual needs imposed by concepts such as Industry 4.0.

The implementation of PDM strategies into mission reconfiguration or mission re-planning is certainly one of the most promising applications of these decision-making schemes. In this regard, in (Balaban & Alonso, 2012), the authors investigate PDM problems oriented to Aerospace Domain, discussing requirements and emphasizing the importance of real-

time optimization solvers due to the need for computing decisions on-board. Moreover, they tested proposed solutions in a case study inspired on the usage of a planetary rover. In that perspective, they consider a Probabilistic Policy Generator (PPG), as a policy optimization algorithm, and Dynamic Constraint Redesign (DCR), which is a methodology with roots in the fields of Multidisciplinary Design Optimization and Game Theory. The results shown are interesting in terms of the computational burden for relatively small problems. However, its scalability for even greater problems is not clear. The same research group later provides a comprehensive work (Sweet et al., 2014) in which they illustrate the real-world implementation of PDM decision making strategy for a terrain rover, namely the *K11* planetary rover prototype. This research effort is verified by simulation on a Mobile Robot Testbed ((Balaban et al., 2011), (Balaban et al., 2013)), where the rover is purposely affected by different fault modes in the middle of the mission. Therefore, PDM is implemented to compute mission re-planning as soon as possible, incorporating information obtained from prognostic algorithms. Although results seem to be promising, authors still solely considered a small network, where the decision universe is limited. In (T. Potteiger & Karsail, 2017), the authors offer a prognostic-based scheme to manage the configuration of the battery pack, where a PDM scheme continuously estimates the Remaining Useful Life (RUL) for different possible cell configurations, choosing the one with the highest RUL. In (H. Skima & Bourgeois, 2019), the authors propound the use of a prognostic-based strategy for electric vehicle routing under stochastic traffic conditions. The main idea is to leverage both the traffic model and route information by executing a prognostics algorithm that allows predicting the route performances in terms of travel time and energy consumption.

Among all the aforementioned PDM applications, mission re-planning is probably the most challenging problem since it requires the implementation of real-time failure prognostic modules and optimization solvers. Unfortunately, there is an inherent trade-off between solutions performance and the computational burden. In fact, the most exhaustive optimization solvers, which usually lead to the best solutions, have high computational costs; therefore, their real-time implementations might be highly challenging.

## 2.2. Particle-filtering-based Prognosis

Prognostic schemes endeavor to estimate the probability of failure by predicting, in the long-term, the future evolution of fault indicators. For this purpose, the Monte Carlo (MC) methods offer a great variety of useful tools. However, as mentioned in the first section, one of the requirements of this task is having low algorithm latency, and not all MC-based methods satisfy this requirement. This is why the Particle Filter prognostic is used; in exchange for a reasonable precision loss, a PDF of the system can be estimated with low

latency. Particularly, in the case of particle-filtering-based prognostic algorithms (PPF) (Orchard, 2007), the underlying concept is to model uncertainty propagation in time based on a stochastic state-space model of the faulty system, a probabilistic characterization of future operating profiles, and particle-filter-based estimates of the state probability density function (PDF). In this regard, this paper assumes that the system state is being monitored using a Particle Filter (PF) algorithm (Arulampalam, Maskell, Gordon, & Clapp, 2002).

PF-based estimation algorithms aim to sequentially approximate the posterior PDF of the state by a set of weighted particles. Thus, at time  $k$  the estimate of the posterior PDF of  $x$  is given by:

$$p(x(k)|y(1), \dots, y(k)) = \sum_{i=1}^{N_p} w_i(k) \delta(x(k) - x^i(k)), \quad (1)$$

where  $\{x_i(k), w_i(k)\}_{i=1}^{N_p}$  is a set of weighted-particles,  $N_p$  is the number of particles,  $x_i(k)$  is position of particle  $i$  and  $w_i(k)$  is the weight of particle  $i$ , and  $\delta(x) = 1$  if  $x = 0$  and  $\delta(x) = 0$  otherwise. The estimation process is composed of two major stages: 1) Prediction and 2) Update.

**1) Prediction:** each particle is propagated one time ahead:

$$x^i(k) \sim q(x(k)|x^i(k-1)) \quad (2)$$

where  $q(x(k)|x^i(k-1))$  is the prior conditional distribution of the states at time  $k$  given the state at time  $k-1$ , and is computed using the simulation data.

**2) Update:** once a new measurement is available, it is utilized to update the weight of each particle as follows:

$$\hat{w}_i(k) = w_i(k-1) \cdot p(y(k)|x^i(k)). \quad (3)$$

Finally, the weights are normalized:

$$w_i(k) = \hat{w}_i(k) \cdot \left( \sum_{i=1}^{N_p} \hat{w}_i(k) \right)^{-1}. \quad (4)$$

Let us suppose that the prognostic algorithm is executed at time  $k$  using as initial condition a set of  $N_p$  weighted-particles  $\{x_i(k), w_i(k)\}_{i=1}^{N_p}$ . Then, these particles are propagated according to the state transition equation. In general, to estimate the state PDF at time  $k + \tau$ , for  $\tau \in \{1, \dots, n\}$ , we need to propagate the particles from  $k + \tau - 1$  to  $k + \tau$ :

$$p(\hat{x}(k + \tau)|\hat{x}(k + \tau - 1)) \approx \sum_{i=1}^{N_p} w_i(k + \tau - 1) \cdot \hat{p}(x_i(k + \tau)|\hat{x}_i(k + \tau - 1)). \quad (5)$$

Note that  $\hat{p}(\hat{x}_i(k + \tau)|\hat{x}_i(k + \tau - 1))$  is related to the state transition equations; it demands a characterization of the fu-

tures inputs. An update that has proven particularly advantageous to propagate particles for large prediction horizons (Orchard & Vachtsevanos, 2009) is based on the regularized PF algorithm (Musso, Oudjane, & Le Gland, 2001). By implementing this, instead of updating particle weights in each prediction step, the uncertainty is represented by a re-sampling of the predicted state.

### 3. RISK-BASED STRATEGY FOR REAL-TIME DECISION MAKING AND MISSION RECONFIGURATION

In this section, we will formally explain the proposed Risk-based strategy for real-time decision making and mission reconfiguration.

#### 3.1. Mission Definition: Graph and waypoints

Inspired by the 3D mapping applications (Yamazaki et al., 2018), we consider a mission setup wherein a UAV has to traverse a set of waypoints or nodes  $\mathcal{C} = \{n_1, \dots, n_N\}$  in order to characterize a certain surface. In such a framework, we suppose that before departure an optimization algorithm is executed so as to compute an initial flight plan, so-called initial path  $\mathcal{P}_0 = n'_1, n'_2, \dots, n'_{N'}, n_0$ . Essentially, this path establishes both the order in which the waypoints are visited and the final visited waypoint  $n'_{N'}$  before returning to the base node  $n_0$ . Note that 1) this path could not include all nodes of  $\mathcal{C}$  since the UAV autonomy is limited and that 2) the final waypoint should be the base node.

Unfortunately, due to weather conditions changes or other reasons, the actual energy consumption associated with  $\mathcal{P}_0$  could differ from the initial predicted one; hence, computing mid-flight replanning is essential to keep the reliability standards of the mission. In this context, a real-time procedure is required to periodically decide whether the mission should continue or not. To make this decision, we propose a Risk-based strategy able to quickly replan the mission while taking into account 1) the expected value of the cumulative reward, and 2) the risks associated with both returning to the base and covering the next nodes. The constitutive blocks of this proposal are depicted in Figure 2, each of them will be explained in detail in the next subsections.

#### 3.2. Characterizing the UAV energy consumption

In order to make decisions regarding the UAV operation, a suitable characterization of the UAV consumption is critical because its autonomy can constrain seriously the feasible choices. To describe energy demand along a certain path, we propose a dynamic model that is discretized by the path nodes. Thus,  $x(k)$  denotes the SOC when the UAV is at node  $n_k$ . As a result, the SOC evolution can be expressed as follows:

$$x(k+1) = x(k) - \frac{E(k, \mathcal{X})}{E_c}, \quad (6)$$

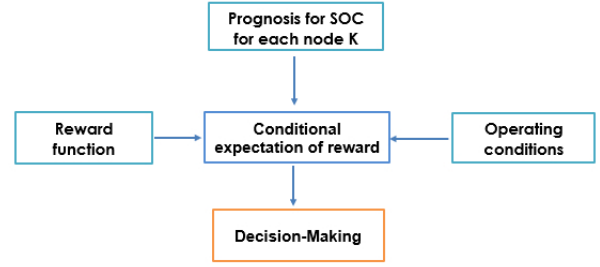


Figure 2. Graphical abstract of the proposed mission re-planning strategy.

where  $E_c$  is the battery capacity expressed in [J]. On the other hand,  $E(k, \mathcal{X})$  corresponds to the energy demanded to travel from node  $n_k$  to  $n_{k+1}$  given the environmental conditions  $\mathcal{X}$ .

Note that, in real-world applications,  $\mathcal{X}$  can include different variables such as wind speed, temperature, and air pressure. Among these variables, wind speed represents a significant source of uncertainty for UAV energy consumption's characterization due to its complex inherent dynamic behavior and stochasticity. In this regard, it is noteworthy that wind speed intensity can be predicted by using both real-time measurements (acquired while airborne) and historical data by using fuzzy models or similar methodologies. The 3D imaging problem, which corresponds to our study case study application, requires a constant speed with respect to the ground so as to take distinct images. Constant turbulence and wind speed are assumed, taking into account that in the future, these results will be used to characterize the operation with a particular wind speed of the defined fuzzy set.

The model presented in (6) enables us to assess the energy consumption of a determined path in advance. However, to do this, we first have to characterize the  $E(k, \mathcal{X})$  whose nature is stochastic. This task is achieved by a simulation-based approach in which we simulate thousands of flights to estimate the distribution of energy demanded conditional to different environmental conditions. Then, by combining these simulation results with a prognostic algorithm, it is possible to evaluate in real-time and probabilistically the energy demand associated with a determined path. Finally, we can derive the risk associated with each decision, which constitutes the basis of our decision-making algorithm.

#### 3.3. UAV Failure condition

To link successfully the failure events with the online variables,  $S_f$  will be defined as follows:

$$S_f(n_0, n_f) = SOC(n_0) - E_{n_0, n_f}, \quad (7)$$

where  $n_0$  is the initial node,  $n_f$  is the final node and  $E_{n_0, n_f}$  is the consumed SOC from the initial node till the last node.

It is important to note that the UAV is assumed to visit each point only once, making no more than one lap. Once this variable is defined, a relationship can be formed with the drone failure imposing  $S_f < T$  as a condition for falling, where  $T$  is the SOC threshold for the UAV safe operation. This condition implies that if  $S_f - T < 0$  at any point of the journey, the drone falls. Then, the probability of failure is given by Equation 8.

$$P(S_f(n_0, n_f) < T | d_{n_0+n_f}, X_i) = P(\mathbb{F}_{n_0, n_f} | d_{n_0+n_f}, X_i) \quad (8)$$

where  $\mathbb{F}_{n_0, n_f}$  is the scenario where the travel from node  $n_i$  to  $n_f$  is unsuccessful,  $d_{i+j}$  is the decision to travel from node  $n_i$  to  $n_j$  and  $X_i$  is the vector containing the different conditions in node  $n_i$ .

#### 3.4. Probability of success $P(\mathbb{S}_{i,j} | d_{i+j}, X_i)$

The probability of success is given by  $P(\mathbb{S}_{i,j} | d_{i+j}, X_i)$ , where  $\mathbb{S}_{i,j}$  is the scenario in where the travel from node  $n_i$  to  $n_j$  is successful. Prognostics algorithms based on Sequential Importance Sampling Re-sampling (SIR) are used to approximate this probability, mainly because of the efficiency of this approach and its reduced computational time. The procedure starts by sampling  $N_p$  particles  $x^n(k) \sim SOC_k$ , where  $SOC_k$  corresponds to the probability density function (PDF) of the battery SOC when the UAV is located at node  $n_i$  ( $n = 1, \dots, N_p$ ). The weights  $w_n(k)$  for each  $x^n(k)$  are computed as in Equation 9 (all particles have the same weight).

$$w_n(k) = N_p^{-1} \quad (9)$$

Each particle then evolves assuming different possible energy consumption scenarios  $E(k)$  at node  $k$ , conditional to  $X_k$ , by simulating Equation 10.

$$x^n(k+1) = x^n(k) - \frac{E(k, X_k)}{E_c}, \quad (10)$$

where  $E_c$  is the capacity of the UAV battery.

This procedure generates a new set  $x'(k+1)$ , with  $N'_p$  particles, from which the probability of failure  $P(\mathbb{F}_{i,j} | d_{i+j}, X_i)$  is obtained by summing the number of samples  $x^n(k+1) \in x'(k+1)$  with SOC values smaller than the SOC threshold  $T$ ; situation that would represent a battery discharge event, and normalizing weights as shown in Equation 11.

$$p(S_f(n_o, n_f) < T | \delta_{i,j}, X_K) = \sum_{i=1}^{N'_p} w_i(k) I(T - x^i(k)), \quad (11)$$

where  $\{x^i(k), w_i(k)\}_{i=1}^{N'_p}$  is a set of weighted-particles,  $N'_p$  is the number of particles,  $x^i(k)$  is position of particle  $i$  and  $w_i(k)$  is the weight of the  $i$ -th particle. The indicator function  $I(x) = 1$  if  $x < 0$  and  $I(x) = 0$  otherwise. Then a new particle population  $x(k+1)$  with  $N_p$  samples is subtracted

and the process is repeated to characterize the evolution of the state PDF in time.

#### 3.5. Conditional Expectation of Reward Function

The Conditional Expectation of Reward Function (CERF) is employed in this proposal as the risk function to assess each possible choice. Recall that our decision is binary: go to the next node or return to the base, but to make it, we will first evaluate a set of  $n_s$  actions associated with the choice of exploring the next node. To be more specific, if the UAV is currently at node  $i$ , the set of possible actions examined is denoted as  $\{d_{i+k}, k = 0, \dots, n_s\}$ , wherein  $d_{i+k}$  means the UAV will return to the base after reaching node  $n_{i+k}$ .

To compute the CERF of each sub-decision, it is necessary to parameterize them in "economic" terms considering both rewards and penalties. Consequently, it is assumed that the action of traveling successfully from node  $i$  ( $n_i$ ) to node  $j$  ( $n_j$ ) yields a deterministic reward denoted  $G_{i,j} \geq 0$ , while losing the UAV represents a potential sunk cost that can be avoided if it "Returns To Landing" (RTL) safely. On the other hand, if the UAV battery gets depleted in mid-flight, it results in a complete loss of the UAV,  $D \geq 0$ , with no chance of recovering (or recycling) any parts.

An action  $d_{i+k}$  could face different events depending on the realizations of future energy consumption. Those events are disjointed and can be clustered in different families. Besides, since they are disjointed, we can thus calculate the CERF of each action by adding the CERF of all event families. This procedure is explained in the following bullets.

- $\mathcal{F}_1$  : The UAV moves from the current node  $n_i$  to an ahead node  $n_j$  and then returns to base safely. Therefore, its corresponding CERF can be described as follows:

$$E(U(\mathcal{F}_1) | d_{i+k}, X_k) = G_{i,k} \cdot P(\mathbb{S}_{i,k} | d_{i+k}, X_i) \cdot P(\mathbb{S}_{k,0} | d_k, X_k), \quad (12)$$

where  $G_{i,j}$  is the summation of the rewards collected from  $i+1$  to  $j$  i.e  $G_{i,j} = \sum_{k=i+1}^j g_k$ , with  $g_k$  being the reward associated to node  $k$ ;  $\mathbb{S}_{i,j}$  is the event of going successfully from  $i$  to node  $j$ ; and  $X_i$  corresponds to a vector with the current wind conditions at node  $i$ .

- $\mathcal{F}_2$  : The battery gets depleted while the UAV is moving from the node  $n_{i+l}$  to either some ahead node or the base,  $l \geq 0$ . In such a scenario, the CERF includes the summation of rewards previously collected and the cost of the UAV loss, then:

$$E(U(\mathcal{F}_2) | d_{i+k}, X_k) = \sum_{j=i+1}^{k-1} G_{i,j} \cdot P(\mathbb{S}_{i,j} | d_{i+j}, X_{j-1}) \cdot P(\mathbb{F}_{j,j+1} | d_{j+1}, X_j) + (G_{i,j}) \cdot P(\mathbb{S}_{i,k} | d_{i+k}, X_i) \cdot P(\mathbb{F}_{k,0} | d_{i+k}, X_k) \quad (13)$$



$$U(\mathbb{F}_{i,j}) = G_{i,j-1} \quad (14)$$

Let  $\mathbb{S}_{i,0}$  be the scenario of going from the current node to the initial node  $n_0$  successfully, the reward related to this decision is the cost of the UAV ( $\mathcal{D}$ ) since the UAV is recovered when RTL safely, as shown in Equation 15.

$$U(\mathbb{S}_{i,0}) = \mathcal{D} \quad (15)$$

Finally, if the UAV battery discharges midair when the drone is returning to the base node, the drone's cost can be salvaged, as shown in Equation 16.

$$U(\mathbb{F}_{i,0}) = 0 \quad (16)$$

To measure the reward associated to different trajectories starting from node  $n_i$ , the conditional expectation is used,

$$\begin{aligned} E(U|d_{i+j}, \mathcal{X}_i) = & [U(\mathbb{S}_{i,j}) + U(\mathbb{S}_{j,0})] \cdot \underbrace{P(\mathbb{S}_{i,j}|d_{i+j}, \mathcal{X}_i) \cdot P(\mathbb{S}_{j,0}|d_{i+j}, \mathcal{X}_j)}_{\text{Probability of an event in } \mathcal{F}_1} \\ + \sum_{k=i}^j & U(\mathbb{F}_{i,k}) \cdot \underbrace{P(\mathbb{S}_{i,k-1}|d_{i+j}, \mathcal{X}_i) \cdot P(\mathbb{F}_{k-1,k}|d_{i+j}, \mathcal{X}_{k-1})}_{\text{Probability of an event in } \mathcal{F}_2} \\ + [U(\mathbb{S}_{i,j}) - U(\mathbb{F}_{j,0})] & \cdot \underbrace{P(\mathbb{S}_{i,j}|d_{i+j}, \mathcal{X}_i) \cdot P(\mathbb{F}_{j,0}|d_{i+j}, \mathcal{X}_j)}_{\text{Probability of an event in } \mathcal{F}_2} \end{aligned} \quad (17)$$

In the aforementioned expressions, three possible scenarios are captured: (i) proceeding to node  $n_j$  and returning safely, (ii) proceeding safely to some other node  $n_k$  (between node  $n_i$  and node  $n_j$ ) but failing in returning safely to the landing point, and (iii) proceeding to node  $n_j$  but failing in returning safely to the landing point. Replacing the reward values in the Equation 17 leads to:

$$\begin{aligned} E(U|d_{i+j}, \mathcal{X}_i) = & (G_{i,j} + \mathcal{D}) \cdot P(\mathbb{S}_{i,j}|d_{i+j}, \mathcal{X}_i) \cdot P(\mathbb{S}_{j,0}|d_j, \mathcal{X}_i) \\ + \sum_{k=i+1}^{j-1} & (G_{i,k}) \cdot P(\mathbb{S}_{i,k}|d_{i+j}, \mathcal{X}_i) \cdot P(\mathbb{F}_{k,k+1}|d_{i+j}, \mathcal{X}_i) \\ + (G_{i,j}) & \cdot P(\mathbb{S}_{i,j}|d_{i+j}, \mathcal{X}_i) \cdot P(\mathbb{F}_{j,0}|d_j, \mathcal{X}_j) \end{aligned} \quad (18)$$

If the decision is to return the UAV to base, then the conditional expectation is computed as shown in Equation 19.

$$E(U|d_i, \mathcal{X}_i) = \mathcal{D} \cdot P(\mathbb{S}_{i,0}|\mathcal{X}_i) \quad (19)$$

### 3.6. Decision making model

Once the expected reward has been computed, we can construct the set  $\{E(U|d_{i+j}, \mathcal{X}_i)\}_{j=\{i+1, \dots, n_f\} \cup \{0\}}$ . Finally, the criterion for choosing the return node is defined as:

$$i^* = \operatorname{argmax}_j (E(U|d_{i+j}, \mathcal{X}_i)), \quad (20)$$

which is equivalent to choosing the route that maximizes the expected reward.

## 4. CASE STUDY SETUP: UAV SIMULATION AND DATA ACQUISITION

The simulation-based testbed was performed by *ArduPilot's Arducopter*. Since this software considers that both the drone model-simulated and *DroneKit* python are employed as a Ground Control Station, including direct communication suppressing *Mavproxy*, it considerably slows down the simulation speed. *Arducopter* allows us to accelerate the simulation whose speed is capped by that of the CPU clock. Default values for the drone parameters were changed to the parameters for 3DR IRIS+ drone obtained from connecting a 3DR IRIS+ drone to Mission Planner, a ground control station. The wind speed value is substituted by that of each particular simulation, and the turbulence is set to 0.1.

Recall that the ultimate goal of the simulation testbed is to characterize statistically  $P(\mathbb{F}_{i,j}|d_{i+j}, \mathcal{X}_i)$ . As wind intensity, traveling angle, and distance were the essential variables, simulations were made considering a drone traveling along a straight line, changing angle, and distance for different wind speeds. For each scenario simulated, the variations of State of Charge ( $\Delta SOC$ ) are collected because they represent themselves the energy consumptions.

The acquired data has values for every possible angle with a resolution of one degree, distances between 500 meters and 1800 meters and for wind intensities of 7 and 6. Effective consumption is defined as  $\Delta SOC$  divided by the distance traveled. The evolution of the State of Charge was computed according to Equation 21, where  $V$  is the UAV battery voltage,  $I$  the battery current,  $t_{sample}$  is the period between two iterations of the proposed algorithm,  $t_{sim}$  is the sampling time within the simulation framework (1 [sec]), and  $E_{crit}$  is the initial total energy which is a constant given the vehicle battery.  $E_{crit} = 202426.858$  in the case of a 3DR IRIS+ drone.

$$SOC(k) = SOC(k-1) - V \cdot I \cdot t_{sample} (t_{sim} \cdot E_{crit})^{-1} \quad (21)$$

As simulations for different wind speeds were executed in parallel computers, with dissimilar CPU clock speeds,  $t_{sim}$  cannot be directly utilized to compare the effective  $\Delta SOC$  between the two simulations. For this reason, the effective  $\Delta SOC$  has been normalized by the simulation time. Figure 3 displays the distribution of the effective consumption.

As it was expected, traveling against the wind translated into higher consumption. Higher wind speeds translate into bigger differences between configurations against or in favor of the wind direction.

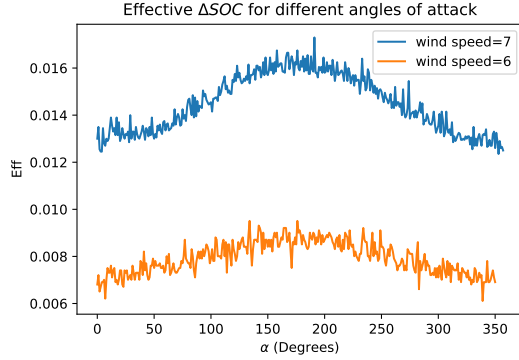


Figure 3. Data acquired from traveling a single constant distance for different wind speeds and angles of attack.

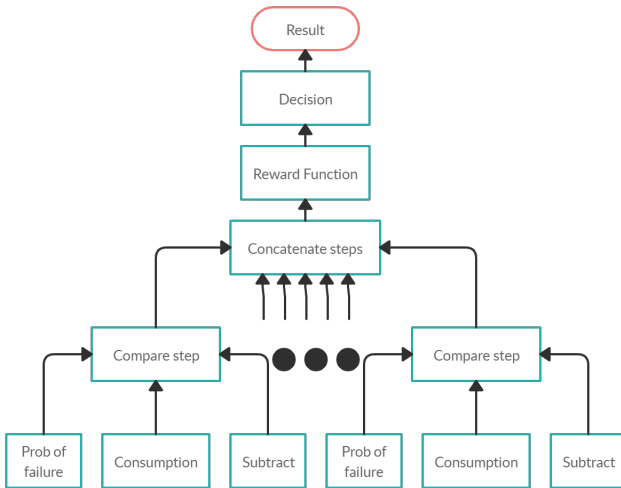


Figure 4. Prognostic Decision-Making Algorithm Design.

Figure 4 presents the flowchart of **Prognostic Decision-Making Algorithm Design**. All modules and functions are described as follows.

- **“Decision” module.** It decides if the UAV should continue by selecting the route that maximizes the expected reward such that if  $E(U|d_{i+j}, \mathcal{X}_i)$  for some  $j$  is larger than  $E(U|d_i, \mathcal{X}_i)$ , then the UAV goes to the next node.
- **“Reward Function” module.** It calculates the expected reward of reaching some node and Return To Landing (RTL) for all the nodes considered within the algorithm prediction horizon. Then, it selects the maximum mission reward from all the possible choices explored by the algorithm.

- **“Concatenate Steps” module.** It computes failure probabilities and rewards for all the routes in the algorithm prediction horizon by concatenating the outcomes of the module “Compare step” for all the routes explored. Every “Compare step” module gives the SOC distribution for a given node, and this distribution is used in the next node as the initial SOC.
- **“Compare step” module.** It produces four outputs: 1) the distribution of the SOC, 2) the probability of failure associated with traveling from a node  $n_i$  till a node  $n_i + 1$ , 3) the reward gained between node  $n_i$  and node  $n_i + 1$ , 4) gives the probability of failure traveling from a node  $n_i$  till a node  $n_0$ . This is done using the modules shown below in Figure 4.
- **“Prob of failure” module.** It calculates the probability of failure, given a probabilistic characterization of the evolution of the battery SOC in time, assuming that the drone will not function if the SOC falls below the threshold  $T$ .
- **“Consumption” module.** Given an angle, wind intensity, and distance to travel, this module returns the distribution of the consumption related to those variables. This is done by creating a new dataset with the given wind angle and intensity values and multiplying the Effective  $\Delta SOC$  by the distance to the next waypoint.
- **“Subtract” module.** It propagates the SOC distribution by subtracting the possible consumption for each particle of the SOC distribution.

## 5. OBTAINED RESULTS

In order to test the proposed algorithm, UAV flights are simulated in a computational engine that uses an accelerated internal clock, which runs 10 times faster than in real-time. Every two seconds, the decision-making strategy is executed, telling whether the drone should continue to the following waypoint (or “node”) or RTL. Design variables for each mission are the cost of the drone ( $D$ ) and the safety threshold ( $T$ ) for the battery SOC (which represents a catastrophic failure). A Particle Filter algorithm is assumed to be used to provide real-time estimates of the battery SOC, as described in Section 2.2. As the wind speed cannot be changed mid-flight in the simulation, the wind speed is assumed constant for now.

In all simulated flights, the proposed strategy achieves to return the UAVs to the landing location with sufficient remaining battery energy. This situation proves that the algorithm can avoid catastrophic failure midair. However, it is essential to conduct sensitivity analyses on the cost  $D$  of the asset (in this case, the UAV) to understand the impact of this variable in the decision-making process. In this regard, Figure 5 shows the dispersion associated with the nodes that were finally reached as a function of the UAV cost. In all cases, the sensitivity in terms of the final decision is computed by vary-

ing the UAV cost, while the SOC threshold  $T = 15$  is kept constant, as well as the rewards for every node. Information on the node that is finally reached by the drone is essential because it clearly shows whether the UAV is recovered. In fact, every drone that reached a node  $n_j$  with  $j < 40$ , was also able to RTL safely. Figure 5 also shows that there is a natural transition between decisions biased towards sacrificing the drone and those that intend to recover it. This transition is highly influenced by the battery SOC as the cost of the UAV decreases. On the other hand, as this UAV cost increases, the decision shifts progressively towards a situation where the strategy aims to recover the drone. Following this logic, it is important to point out that the UAV cost plays a fundamental roll in the decision taken by the drone. For this simulation scenario, the drone gets to a node close to the node 35, and it has to decide whether to continue given that it will fall (without taking into account outliers where node 35 was reached with spare SOC). If the UAV cost is low, the drone will not come back; if it is high, the drone will come back, but in the middle, the drone makes a decision based on the current SOC, leaning towards the bias given by the UAV cost. As shown in Figure 5, the mean node reached by the UAV decreases as its cost increases.

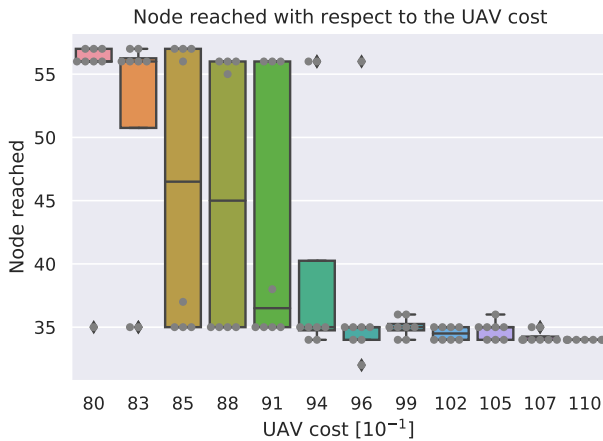


Figure 5. Dispersion of the node reached when moving the UAV cost.

## 6. CONCLUSION

This work presents an integrated Risk-based strategy utilizing a SOC prognostic algorithm to quantify the risk associated with a given path in terms of future consumption, with the aim of making real-time decisions on the UAV operation. More specifically, we consider a UAV mission framework with a discrete set of possible targets, where each one has costs, rewards, and failure risk, which is characterized by calculating the total probability using sample-resampling methods. Future work will focus on validating this methodology on actual UAVs, incorporating a characterization of power consump-

tion for different wind speeds, thus allowing to modify the flight policy depending on the current wind speed and historical distribution data on the flight site. Besides, this work gives a method that allows real-time decision-making considering several factors that condition the UAV energy consumption. However, the number of covered factors is limited. Indeed, crucial elements such as signal noise, component failure, moving wind intensity mean, and moving wind direction are not taken into account. In our future work, we intend to extend the number of factors to be considered by our PDM approach.

## 7. ACKNOWLEDGMENT

This work has been partially supported by FONDECYT Chile Grant Nr. 1170044, the Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008, ANID.

## NOMENCLATURE

$g_i$	node i reward
$G_{i,j}$	accumulative reward from node i to node j
$\mathcal{D}$	drone cost
$\mathbb{S}_{i,j}$	success when traveling from node i to node j
$\mathbb{F}_{i,j}$	failure when traveling from node i to node j
$d_{i+j}$	decision to travel node j from node i
$d_i$	decision to travel node 0 from node i
$\mathcal{X}$	variable vector that influences energetic consumption

## REFERENCES

- Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 174–188.
- Balaban, E., & Alonso, J. J. (2012). An approach to prognostic decision making in the aerospace domain. In *Proceedings of the annual conference of the prognostics and health management society 2012* (p. 396-415).
- Balaban, E., Narasimhan, S., Daigle, M., Celaya, J., Roychoudhury, I., Saha, B., ... Goebel, K. (2011). A mobile robot testbed for prognostics-enabled autonomous decision making. In *Annual conference of the prognostics and health management society* (pp. 15–30).
- Balaban, E., Narasimhan, S., Daigle, M., Roychoudhury, I., Sweet, A., Bond, C., & GoroSpe, G. (2013). Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. *International Journal of Prognostics and Health Management*, 4(1), 87.
- Diaz, C., Quintero, V., Perez, A., Jaramillo, F., Burgos-Mellado, C., Rozas, H., ... Cardenas, R. (2020). Particle-filtering-based prognostics for the state of maximum power available in lithium-ion batteries at electromobility applications. *IEEE Transactions on Ve-*



*hicular Technology.*

- H. Skima, E. D. K. M., C. Varnier, & Bourgeois, J. (2019). Post-prognostics decision making in distributed mems-based systems. *J. Intell. Manuf.*, 30(3), 1125–1136.
- Kulkarni, C., Hogge, E., & Quach, C. C. (2018, jul). Remaining Flying Time Prediction Implementing Battery Prognostics Framework for Electric UAV's. Cincinnati.
- Musso, C., Oudjane, N., & Le Gland, F. (2001). Sequential monte carlo methods in practice. *Springer*, 247–260.
- Orchard, M. E. (2007). *A particle filtering-based framework for on-line fault diagnosis and failure prognosis* (Unpublished doctoral dissertation). Georgia Institute of Technology.
- Orchard, M. E., & Vachtsevanos, G. J. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control*, 31(3-4), 221–246.
- Pola, D. A., Navarrete, H. F., Orchard, M. E., Rabié, R. S., Cerda, M. A., Olivares, B. E., ... Pérez, A. (2015). Particle-filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles. *IEEE Transactions on Reliability*, 64(2), 710-720.
- Rozas, H., Munos-Carpintero, D., Perez, A., Medjaher, K., & Orchard, M. (2018). An approach to prognosis-decision-making for route calculation of an electric vehicle considering stochastic traffic information,. In *Phm soc. eur. conf.* (p. 1-9).
- Saha, B., Koshimoto, E., Quach, C. C., Hogge, E. F., Strom, T. H., Hill, B. L., ... Goebel, K. (2011). Battery health management system for electric uavs. In *2011 aerospace conference* (p. 1-9).
- Saha, B., Quach, C. C., & Goebel, K. (2012). Optimizing battery life for electric uavs using a bayesian framework. In *2012 ieee aerospace conference* (p. 1-7).
- 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.
- Sweet, A., Gorospe, G., Daigle, M., Celaya, J. R., Balaban, E., Roychoudhury, I., & Narasimhan, S. (2014). *Demonstration of prognostics-enabled decision making algorithms on a hardware mobile robot test platform* (Tech. Rep.). SGT Inc. Moffett Field United States.
- T. Potteiger, K. R. P., W. Strayhorn, & Karsail, G. (2017). A dependable, prognostics-incorporated, n-s modular battery reconfiguration scheme with an application to electric aircraft,. In *Aiaa/ieee digital avionics systems conference - proceedings* (Vol. 2017- Septe, p. –).
- Yamazaki, F., Miyazaki, S., & Liu, W. (2018). 3d visualization of landslide affected area due to heavy rainfall in japan from uav flights and sfm. In *Igarss 2018 - 2018 ieee international geoscience and remote sensing symposium* (p. 5685-5688).