

# Computational Architecture for Autonomous Decision-Making in Unmanned Aerial Vehicles

Shankar Sankararaman<sup>a</sup> and Kai Goebel<sup>b</sup>

<sup>a</sup>SGT Inc., NASA Ames Research Center, Moffett Field, CA, USA

<sup>b</sup>NASA Ames Research Center, Moffett Field, CA, USA

## ABSTRACT

This paper presents a computational architecture to facilitate autonomous decision-making under uncertainty for safe operation of drone-like vehicles. The proposed framework is based on identifying and predicting the occurrence of various risk-factors that affect the safe operation of such vehicles and estimating the likelihood of occurrence of these risk-factors. This analysis is then used to select trajectories for the operation of the vehicle. Feasible trajectories are classified into four different categories: nominal and safe, off-nominal but safe, unsafe and abort the mission, and unsafe and ditch the vehicle. An important challenge in the operation of drones is that there are several sources of uncertainty that affect their operation; these sources of uncertainty arise from wind conditions, imprecise future power-demands, inexact future trajectories, etc. Therefore, it is important to develop a decision-making framework that can incorporate all these sources of uncertainty and make decisions that are robust to the presence of such uncertainty. Potential risk-factors such as dynamic obstacles, battery drain, etc. are identified and the likelihood of occurrence of these risk-factors are predicted preemptively and proactively in order to facilitate risk-informed safety-assured decision-making.

**Keywords:** UAV, decision-making under uncertainty, computational architecture, risk

## 1. INTRODUCTION

The importance of unmanned aerial vehicles and systems has steadily increased in the past ten to fifteen years. Both the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA) have shown significant interest in the development of technologies for unmanned aerial vehicles and unmanned traffic management systems.

It is expected that there will be a significant increase in unmanned aerial traffic and therefore, the overall safety of the United States National Airspace System needs to be analyzed carefully. An Unmanned Aerial System (UAS) will have access to civilian air space only when the safety of the airspace, property, and the vehicle itself can be guaranteed. Further, traffic management becomes increasingly complicated in urban environments where low altitude flight control and safety is of critical importance.

Several researchers have been focusing the development of various technologies that would ultimately enable the systematic inclusion of unmanned systems into the airspace.<sup>1</sup> These technologies include sensing and data logging,<sup>2,3</sup> fault tolerant flight control,<sup>4,5</sup> simultaneous localization and mapping (SLAM),<sup>6,7</sup> obstacle detection and avoidance,<sup>8,9</sup> optimal power management,<sup>10,11</sup> path planning and trajectory design,<sup>12,13</sup> searching and tracking,<sup>14</sup> autonomous decision-making,<sup>15,16</sup> unmanned traffic management,<sup>17</sup> etc.

According to Koperdaker,<sup>18</sup> the near-term goal (1-5 years) is to safely enable low-altitude airspace and UAS operations while the long-term goal (10-15 years) is to safely enable massive increases in airspace density and UAS operations. These goals are particularly challenging because there is a significant amount of uncertainty and numerous factors that are constantly and dynamically evolving in urban environments.<sup>19</sup> As a result, developing a methodology for safe, autonomous decision-making is a challenging problem.

---

This research was performed, and this manuscript was written while Shankar Sankararaman was employed with SGT Inc., NASA Ames Research Center. Shankar Sankararaman is presently an independent consultant.  
Email: Shankar Sankararaman - shankar.sankararaman@gmail.com, Kai Goebel - kai.goebel@nasa.gov

Sankararaman and Kalmanje<sup>20</sup> presented a probabilistic framework for decision-making under uncertainty in order to aid autonomous operation of small unmanned aerial systems (UAS). This framework focuses on the identification, assessment, and prediction of various risk-factors (such as obstacle collision, untimely battery drain, etc.) that affect the operation of sUAS, and aids decision-making under uncertainty. While the framework was presented in a mathematical manner, Sankararaman and Teubert<sup>21</sup> investigated prospective software architectures to implement this decision-making framework. This paper integrates several of these individual activities to present the complete architecture that facilitates decision-making in autonomous systems. While the approach is specifically developed for unmanned airborne vehicles, the framework is general enough to be applicable to other autonomous systems.

The rest of this paper is organized as follows. Section 2 discusses the various risk-factors in flight, and explains how to model such risk-factors in order to aid decision-making. Prediction of risk-factors is an important component of decision-making; such prediction can either be performed onboard (for instance, using the Generic Software Architecture for Prognostics, that is developed and open-sourced by NASA Ames Research Center) or remotely (using “Prognostics as a Service”, i.e., Paas, being developed at NASA Ames Research Center). Section 3 uses the prediction of risk-factors to enable decision-making and summarizes the decision-making framework developed by Sankararaman and Krishnakumar.<sup>20</sup> Section 4 discusses the advantages and disadvantages of various architectures for implementing the aforementioned decision-making framework; in particular, onboard versus remote options are considered. Finally, Section 5 concludes the paper and discusses possible directions for further research.

## 2. MODELING RISK FACTORS IN FLIGHT

There are several risk-factors that affect the flight of unmanned systems. Decision-making needs to take into account the occurrence of risk-factors, compute the likelihood of such occurrence in the future, and proactively make plans at the time of decision-making. For instance, one risk-factor could refer to complete discharging of the battery prior to the completion of the trajectory; another risk-factor could refer to the impending collision of the unmanned system against a static/dynamic obstacle.

Consider a given trajectory and a generic time of prediction  $t_P$  at which it is necessary to calculate the likelihood of a particular risk-factor continuously as a function of future time ( $\forall t > t_P$ ). In order to achieve this goal, it is necessary to model the evolution of the UAS continuously as a function of time along with the evolution of external factors related to the risk-factor. For instance, in the case of a collision against a dynamic obstacle, it may be necessary to model the evolution of the position of the UAS continuously as a function of time (based on the planned trajectory), and the anticipated position of the dynamic obstacle (which is typically uncertain if the trajectory of the obstacle is unknown and can only be approximately quantified based on its position and velocity as estimated by the sensors on the UAS).

### 2.1 Modeling the Evolution of State With Respect to the Risk-Factor

Consider the state space model which is used to continuously predict the state of the system, as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1)$$

where  $\mathbf{x}(t) \in \mathbb{R}^{n_x}$  is the state vector,  $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$  is the parameter vector,  $\mathbf{u}(t) \in \mathbb{R}^{n_u}$  is the input vector,  $\mathbf{v}(t) \in \mathbb{R}^{n_v}$  is the process noise vector,  $\mathbf{f}$  is the state equation, and  $t$  is the continuous time variable. Note that the above state vector is not necessarily equal to the aerodynamic state of the UAS (measured in terms of position, attitude, etc.); instead, this state vector is directly related to the risk-factor under consideration. If collision against a dynamic obstacle is a risk-factor, then this state vector contains the position of the UAS. On the other hand, if battery-charge draining is a risk-factor, then this state vector contains the charge of the battery of the UAS. Note that all the quantities in Eq. 1 are uncertain in nature and need to be treated probabilistically.<sup>22</sup>

The state vector at time  $t_P$ , i.e.,  $\mathbf{x}(t)$  (and the parameters  $\boldsymbol{\theta}(t)$ , if they are unknown) is (are) estimated using output data collected until  $t_P$ . Let  $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ ,  $\mathbf{n}(t) \in \mathbb{R}^{n_n}$ , and  $\mathbf{h}$  denote the output vector, measurement noise vector, and output equation respectively. Then,

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)) \quad (2)$$

Typically, filtering approaches such as Kalman filtering, particle filtering, etc. may be used for such state estimation.<sup>23</sup>

Having estimated the state at time  $t_P$ , Eq. 1 is used to predict the future states of the component/system. This differential equation can be discretized and used to predict  $\mathbf{x}(t)$  for all  $t > t_P$ .

## 2.2 Modeling the Risk-Factor

Risk-Factors can be expressed in terms of a binary constraint function  $c^H(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t)) = 1$  that maps a given point in the joint state-parameter space given the current inputs,  $(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t))$ , to the Boolean domain  $\mathbb{B} \triangleq [0, 1]$ . Without loss of generality,  $c^H(\mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t))$  can be written as  $c^H(t)$ ;  $c^H(t) = 1$  implies that the risk-factor is encountered at time  $t$  whereas  $c^H(t) = 0$  implies that the risk-factor is not encountered at time  $t$ .

At any generic time of prediction  $t_P$ , note that the constraint function  $c^H(t)$  associated with each risk-factor is a function of  $t$ . Therefore, the approach needs to forecast all available information until future time  $t$  in order to predict the occurrence of the risk-factor. Thus, it needs all information (states, parameters, and inputs in Eq. 1) between time  $t_P$  and  $t$ .

Typically, there are two quantities of interest, in the context of risk-factor prediction:

1. Time of Occurrence: At any time of prediction  $t_P$ , it is useful to know the future time at which the risk-factor will be encountered. Let  $T^H(t_P)$  (expressed as a probability distribution due to the presence of uncertainties) denote this quantity. This information can be helpful in determining the amount of time remaining so that corrective action may be taken.
2. Likelihood of Occurrence of the Risk-Factor as a function of time: At any time of prediction  $t_P$ , it is also useful to know the likelihood of the occurrence of risk-factor as a function of future time. This likelihood is denoted as  $P_t^H(t_P)$ ; this is a trajectory as a function of future time  $t$  and changes with the time of prediction  $t_P$ .

First, at any  $t_P$ , the time of occurrence of a risk-factor (that is, the future time at which the risk-factor will be encountered) can be written as:

It can be easily seen that  $T^H(t_P)$  depends on the state at time of prediction, future inputs/parameters, etc., which are uncertain in nature; in order to calculate the probability distribution of  $T^H(t_P)$ , it is necessary to systematically propagate the aforementioned uncertain quantities and quantify their effect on the probability distribution of  $T^H(t_P)$ . Second,  $P_t^H(t_P)$ , i.e., the likelihood of the risk-factor at future time  $t$  (predicted at time  $t_P$ ) can be expressed as  $P(c^H(t) = 1)$ .

The computation of both the probability distribution of  $T^H(t_P)$  and the probability  $P(c^H(t) = 1)$  can be accomplished using Monte Carlo sampling-based techniques, analytical techniques based on first-order and second-order reliability methods, or hybrid methods involving machine learning approaches.<sup>22</sup>

While onboard computational resources can be used to compute the aforementioned two quantities, it is also possible to send the relevant information to the cloud which hosts PaaS (Prognostics as a Service), to perform future predictions and perform these computations. These two options will be compared and contrasted against each other later in this paper. by analyzing multiple computational/software architectures

## 3. COMPUTATIONAL FRAMEWORK FOR DECISION-MAKING

This section summarizes the decision-making framework presented earlier by Sankararaman and Krishnakumar.<sup>20</sup> An ideal decision-making algorithm should autonomously work in conjunction with the path planner (that generates trajectories) to identify whether a given trajectory is safe or not. In order to achieve this goal, the decision-making algorithm leverages information available from various sources as shown in Fig. 1, and identifies safe trajectories for real-time flight.

As seen from Fig. 1,<sup>19</sup> a trajectory is classified as follows:

1. Safe

- (a) Nominally safe: The likelihoods of risk-factors are extremely low
- (b) Off-nominal but safe: The likelihood of risk-factors are higher than the nominal scenario, but still low enough to be considered safe

2. Unsafe: The likelihood of risk-factors are considerably high.

The limits for likelihood demarcating (1) the nominally safe scenario and the off-nominally safe scenario; and (2) the safe and unsafe scenarios need to be assigned based on computing the costs/risk associated with each risk-factor.

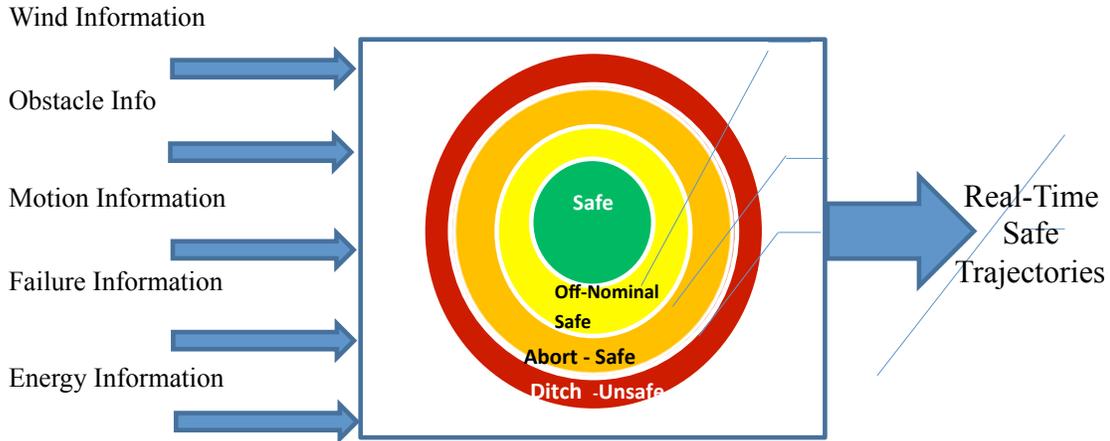


Figure 1. Goal of Decision-Making: Identify Safe Trajectory

If a trajectory is unsafe, then it is necessary to identify whether it is possible to generate a trajectory that can:

1. Abort the mission and return the UAS safely to a landing site; (or)
2. Abort the mission and ditch the UAS without any loss of private and/or public property.

These four different types of trajectories, i.e., nominally safe, off-nominal but safe, abort and return to base, and abort and ditch, are identified in Fig. 1. Note that the scope of this paper is limited to identifying whether a given trajectory is safe or not; further classification and aspects of decision-making will be considered in future work.

### 3.1 Classifying Risk-Factors in Flight

While there are different types of risk-factors that are associated with flights in urban environments, they can be broadly classified into two categories, as shown in Fig. 2.<sup>19</sup>

As seen from Fig. 2, risk-factors may arise simply out of uncertainties (inherent variability, lack of information, etc. due to GPS Denied, degraded sensors, dynamic obstacles, etc.) or due to vehicular performance constraints (such as rapidly draining battery, lack of control, etc.) The decision-making system needs to assess all risk-factors as far as possible, assimilate information from the sensors, and select trajectories. Note that the decision-making is both risk-informed (since it calculates the likelihood of risk-factors along with the associated risk) and safety-assured (selects only those trajectories that are considered “safe”, i.e., the likelihood of a risk-factor is far below a critical limit and hence the operation is considered safe).

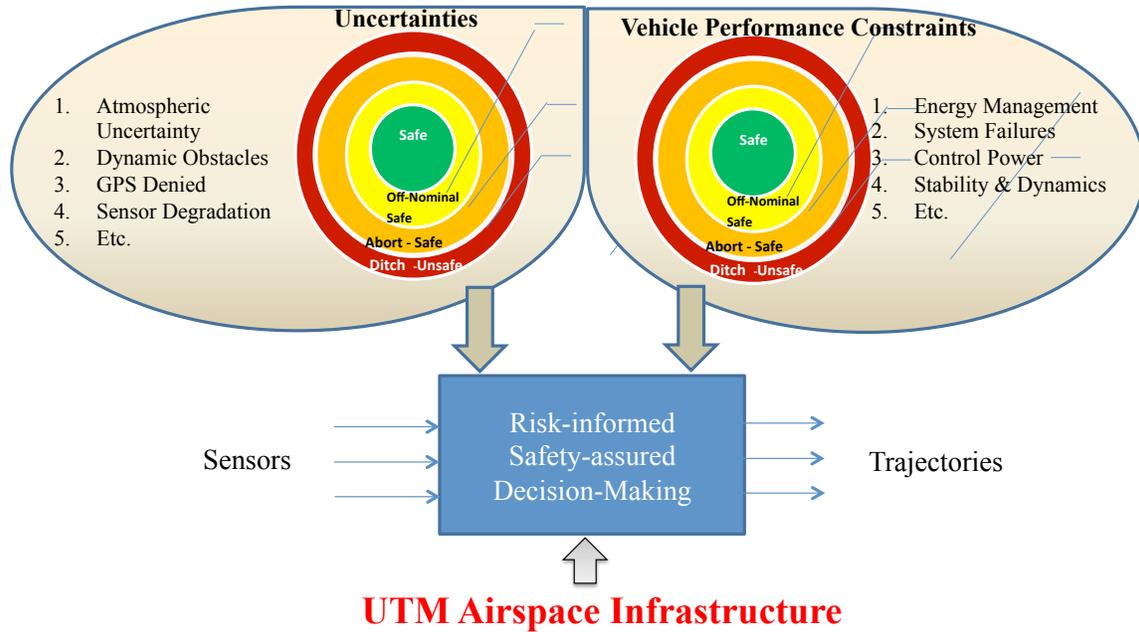


Figure 2. Risk-Factors in Low-Altitude Urban Environment Flight

### 3.2 Decision-Making through Information Fusion

Given a trajectory, and a risk-factor, how should the determine whether the trajectory is safe? Modern reliability analysis<sup>24</sup> defines safety using the so-called limit state function, i.e., a curve of demarcation between a predefined “safe region” and an “unsafe region”.

In simple scenarios, the idea of the limit state can be viewed in terms of capabilities ( $C$ ) and requirements ( $R$ ). When capabilities of a system are more than its requirements, then the system is said to be safe; otherwise, the system is considered to be unsafe. The limit state is then represented by the equation that implies capabilities are equal to requirements  $C - R = 0$ .

In more realistic scenarios, the limit state can be represented as a generic function  $G(\mathbf{X}) = 0$ , where  $\mathbf{X}$  represents the vector of quantities that affect the limit state. In the context of this paper,  $\mathbf{X}$  may potentially include (depending on the risk-factor under consideration) wind information, obstacle information, vehicular information (including motion, dynamics, and properties), energy information, and trajectory information, as shown in Fig. 3. Without loss of generality, the region represented by the curve  $G(\mathbf{X}) > 0$  can be assumed to be the safe region, and the region represented by the curve  $G(\mathbf{X}) < 0$  can be assumed to be the unsafe region.

As mentioned earlier in Section 1, it is likely that elements contained in the vector  $\mathbf{X}$  are all uncertain quantities and hence, these are represented as probability distributions in Fig. 3. It is therefore necessary to compute the probability ( $P(G) < 0$ ), and this probability corresponds to the likelihood of the risk-factor under consideration. It is important to compute this likelihood continuously as a function of future time (starting with the time of prediction) until the end of the trajectory under consideration. Thus, this approach facilitates risk-informed decision-making under uncertainty that incorporates multiple risk-factors.

## 4. COMPUTATIONAL ARCHITECTURES: ONBOARD VS REMOTE

There are three primary functions in a prognostics-enabled decision making architecture: risk factor prediction (prognostics), a decision maker, and a trajectory generator. Each of the three primary functions in prognostics-enabled decision making can be either hosted on-board or remotely. Onboard hosting reduces risk by eliminating dependency on a communication method, but it requires user to host a computer capable of performing complex

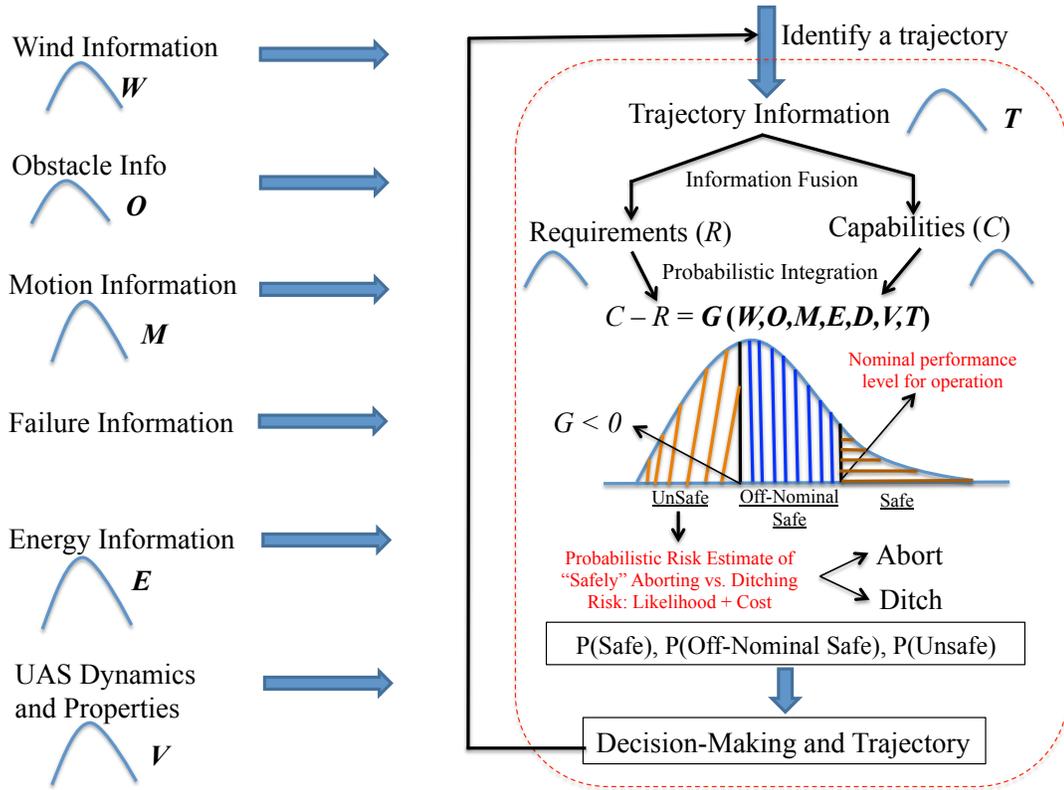


Figure 3. Framework for Decision-Making

prognostics and prognostic decision making activities. This might not be possible because of size, weight, or power constraints (SWaP). Remote prognostics allows users to share resources efficiently, have access to greater computational resources, and upgrade capabilities easily, but communication between the vehicle and the remote server requires time and energy.

There are software architectures exist for performing the risk factor prediction functions on-line or remotely. The Generic Software Architecture for Prognostics (GSAP) is often used to create an on-board prognostics application. GSAP is a general, object-oriented, cross-platform software framework and support library for prognostics technologies. GSAP implements many of the functions and algorithms used in prognostics as part of a prognostics support library. The GSAP framework implements and enforces the prognostic process. Alternatively the Prognostics As-A-Service (PaaS) architecture could be used to support remote prognostics. PaaS is a GSAP-enabled Prognostics Application for performing prognostics remotely, as-a-service. Sankararaman and Teubert<sup>21</sup> identified four different options for performing prognostics and decision-making onboard versus remotely, and discussed in detail how to quantify the efficiency of each option. The various architecture options can be summarized as follows, in the following subsections.

#### 4.1 All On-Board

The first architecture is to host all three primary functions on-board the vehicle. This architecture is illustrated in Fig. 4.

The advantages and disadvantages of this architecture are described below:

##### 1. Advantages

- (a) Less risk through reduced dependence on communications
- (b) Reduced the communication requirements (bandwidth, etc). This reduces the mass and cost of communication equipment.
- (c) Reduced communication time between the three functions

**2. Disadvantages**

- (a) Need to host computer capable of performing calculations on-board
- (b) Increased difficulty in upgrading components

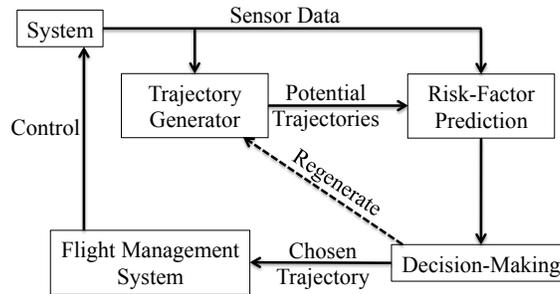


Figure 4. On-Board Architecture for Decision-Making

**4.2 Remote Prognostics**

The second architecture option is to host risk-factor prediction remotely using PaaS, while continuing to host trajectory generation and decision making on-board. This architecture is illustrated in Figure 5.

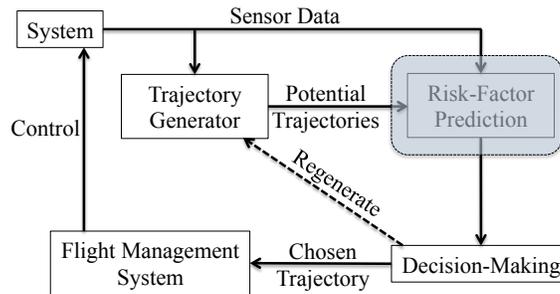


Figure 5. Cloud-based Architecture for DM using PaaS

The advantages and disadvantages of this architecture are described below:

**1. Advantages**

- (a) Partially hosted remotely: This allows resource sharing, and access to more powerful machines
- (b) Somewhat reduced weight, size, and power (SWaP) of on-board computer

**2. Disadvantages**

- (a) Data has to be transmitted every iteration of the Prediction, Decision Making, and Trajectory Generation.

- (b) Network needs: Need to host a network hardware capable of robustly handling communications
- (c) Increased risk: Decision making now becomes dependent on the network connection

For this architecture network delay is significant; this constitutes the time to communicate sensor information to PaaS, time to communicate prognostic results to the decision maker, and time to communicate trajectories to test to PaaS.

### 4.3 Remote Prognostics and Path Generation

The third architecture is to host risk-factor prediction (using PaaS) and path generation remotely, while continuing to host decision making on-board. This architecture is illustrated in Fig. 6.

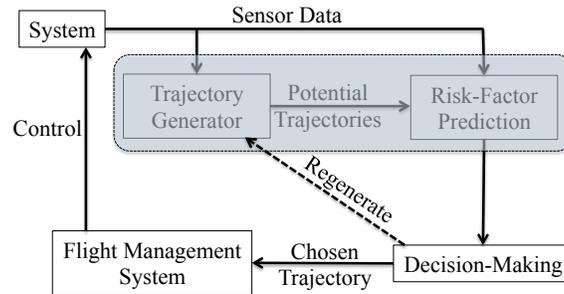


Figure 6. Cloud-based Architecture for DM using PaaS

The advantages and disadvantages of this architecture are described below:

#### 1. Advantages

- (a) More hosted remotely: This allows resource sharing, and access to more powerful machines
- (b) More reduced weight, size, and power (SWaP) of on-board computer

#### 2. Disadvantages

- (a) Data has to be transmitted every iteration of the Prediction, Decision Making, and Trajectory Generation.
- (b) Network needs: Need to host a network hardware capable of robustly handling communications
- (c) Increased risk: Decision making now becomes dependent on the network connection

For this architecture network delay is significant; this delay constitutes the time to communicate sensor information to PaaS, time to communicate prognostic results to the decision maker, and time to decision making instructions to the path generator.

### 4.4 All Remote

The final architecture is to host all critical functions remotely. This architecture is illustrated in Fig. 7.

The advantages and disadvantages of this architecture are described below:

#### 1. Advantages

- (a) Low network costs: Only have to transmit twice: once to receive the sensor data, a second time to send the results of decision making
- (b) Most hosted remotely: This allows resource sharing, and access to more powerful machines

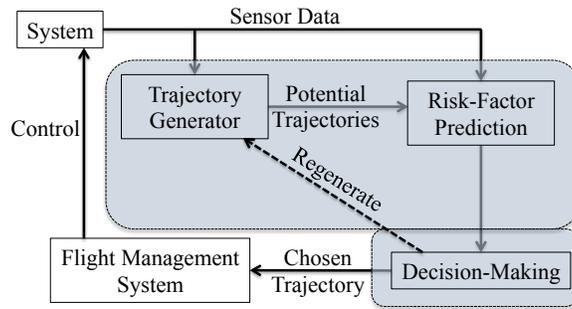


Figure 7. Cloud-based Architecture for DM using PaaS

- (c) Most reduced weight, size, and power (SWaP) of on-board computer

## 2. Disadvantages

- (a) Network needs: Need to host a network hardware capable of robustly handling communications
- (b) Increased risk: Decision making now becomes dependent on the network connection

For this option, the network delay is much lower than the previous option. Network delay has two parts, the time to communicate sensor information to PaaS and the time to decision making results to the aircraft.

## 4.5 Which Architecture to Choose?

The decision of architectures to choose for UAS decision making is highly dependent on (1) the ability of the vehicle to host computers capable of performing these computations; (2) the quality (reliability, bandwidth, latency) of communications; and (3) risk tolerance of the mission.

In all cases there are advantages to hosting operations remotely. These include: shared resources (one server can host capabilities for multiple vehicles), reduced on-board hosting requirements, simplified updating, and increased capabilities. There are also disadvantages, such as increased network requirements, increased program risk due to reliance on networking, and, sometimes, increased step time when network times are greater than function step duration improvements from running remotely. Future research should focus on benchmarking algorithm performance on different class computers and networks to estimate metrics that can quantitatively describe these trade-offs.

## 5. CONCLUSION

This paper presented a computational framework for decision-making under uncertainty, to facilitate the autonomous, safe operation of small drone-like unmanned aerial vehicles. This predictive framework was based on the identification of risk-factors that affect the safe operation of such vehicles, and predicts the occurrence of events related to such risk-factors during the operation of the vehicle. By analyzing various risk-factors, the framework classified possible trajectories into four categories: “nominal and safe”, “off-nominal but safe”, “unsafe and abort the mission”, and “unsafe and ditch the vehicle”. This facilitated the optimal selection of trajectories that can achieve the mission objectives while guaranteeing minimum safety during operation. In order to achieve this goal, the likelihood of occurrence of risk-factors was systematically computed and predicted during the course of operation; the proposed framework was preemptive because it can predict the likelihood of a risk-factor continuously as a function of future operational time, and therefore, identify the future time at which a potential risk-factor may be encountered. Such computation of likelihood also required a systematic integration of the various sources of uncertainty that affect the operation of these unmanned aerial vehicles; this inclusion of uncertainty is particularly important when the focus is on preemptively predicting the future operation (future operations are significantly affected by uncertainty regarding the future conditions) and making changes to a

predetermined trajectory. Finally, the paper also discussed options for performing the prognostics and decision-making computations onboard using the GSAP (Generic software architecture for prognostics) architecture or remotely using the PaaS (Prognostics as a service) architecture.

While this paper presented a computational architecture for decision-making, there are several directions for future research work. It is necessary to develop methods to select trajectories for aborting the mission or ditching the octocopter when safe trajectories or not possible. It is also necessary to account for faults that may occur in the system and incorporate diagnostic information into the decision-making procedure. It is necessary to include multiple risk-factors into the proposed framework and expand the computation of likelihoods; it is also important to incorporate risk measures into the proposed framework. While the present version of implementation focuses on simply predicting when future risk-factors will be encountered, ongoing research is focusing on seamlessly integrating this framework into the trajectory selection/generation procedure. It is also important to continue the investigation of onboard vs remote computations, and understand the advantages and disadvantages in greater detail. Finally, it is important to transform the proposed framework into onboard technology that can be mounted as hardware used on unmanned aerial vehicles, to guide onboard autonomous, safe, operational decision-making.

### Acknowledgment

This work was partly supported by the NASA Ames SAFE50 Center Innovation Fund (CIF) project and the UAS Traffic Management (UTM) Sub-project, under the NASAs Safe Autonomous Systems Operations (SASO) Project, partly supported by PaaS (Prognostics as a Service) sub-project, under the Convergent Aeronautics Solutions Project of the Aeronautics Research Mission Directorate, and partly funded by a NASA-DHS inter-agency agreement. This support is gratefully acknowledged. The authors would also like to thank Dr. Krishnakumar Kalmanje and Christopher Teubert at NASA Ames Research Center for their valuable inputs and contributions to research discussions.

### REFERENCES

- [1] McAree, O. and Chen, W.-H., “Artificial situation awareness for increased autonomy of unmanned aerial systems in the terminal area,” *Journal of Intelligent & Robotic Systems*, 1–11 (2013).
- [2] Corke, P., Hrabar, S., Peterson, R., Rus, D., Saripalli, S., and Sukhatme, G., “Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle,” in [*Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*], **4**, 3602–3608, IEEE (2004).
- [3] Sharp, C. S., Shakernia, O., and Sastry, S. S., “A vision system for landing an unmanned aerial vehicle,” in [*Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*], **2**, 1720–1727, IEEE (2001).
- [4] Coombes, M., McAree, O., Chen, W.-H., and Render, P., “Development of an autopilot system for rapid prototyping of high level control algorithms,” in [*Control (CONTROL), 2012 UKACC International Conference on*], 292–297, IEEE (2012).
- [5] Ducard, G. J., [*Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*], Springer Science & Business Media (2009).
- [6] Hening, S., Ippolito, C. A., Krishnakumar, K. S., Stepanyan, V., and Teodorescu, M., “3D LiDAR SLAM Integration with GPS/INS for UAVs in Urban GPS-Degraded Environments,” in [*AIAA Information Systems-AIAA Infotech@ Aerospace*], 0448 (2017).
- [7] Caballero, F., Merino, L., Ferruz, J., and Ollero, A., “Unmanned aerial vehicle localization based on monocular vision and online mosaicking,” *Journal of Intelligent and Robotic Systems* **55**(4-5), 323–343 (2009).
- [8] Sinopoli, B., Micheli, M., Donato, G., and Koo, T. J., “Vision based navigation for an unmanned aerial vehicle,” in [*Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*], **2**, 1757–1764, IEEE (2001).
- [9] Shim, D. H., Chung, H., and Sastry, S. S., “Conflict-free navigation in unknown urban environments,” *Robotics & Automation Magazine, IEEE* **13**(3), 27–33 (2006).

- [10] Kim, K., Kim, T., Lee, K., and Kwon, S., “Fuel cell system with sodium borohydride as hydrogen source for unmanned aerial vehicles,” *Journal of Power Sources* **196**(21), 9069–9075 (2011).
- [11] Marsh, R., Vukson, S., Surampudi, S., Ratnakumar, B., Smart, M., Manzo, M., and Dalton, P., “Li-Ion batteries for aerospace applications,” *Journal of power sources* **97**, 25–27 (2001).
- [12] D’Souza, S. N., “Developing a Generalized Trajectory Modeling Framework for Small UAS Performance in the Presence of Wind,” in [*AIAA Information Systems-AIAA Infotech@ Aerospace*], 0447 (2017).
- [13] Yang, G. and Kapila, V., “Optimal path planning for unmanned air vehicles with kinematic and tactical constraints,” in [*Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*], **2**, 1301–1306, IEEE (2002).
- [14] Chakrabarty, A., Morris, R. A., Bouyssounouse, X., and Hunt, R., “An integrated system for autonomous search and track with a small unmanned aerial vehicle,” in [*AIAA Information Systems-AIAA Infotech@ Aerospace*], 0671 (2017).
- [15] Ollero, A. and Maza, I., [*Multiple heterogeneous unmanned aerial vehicles*], Springer Publishing Company, Incorporated (2007).
- [16] Ruff, H. A., Narayanan, S., and Draper, M. H., “Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles,” *Presence: Teleoperators and virtual environments* **11**(4), 335–351 (2002).
- [17] Jang, D.-S., Ippolito, C. A., Sankararaman, S., and Stepanyan, V., “Concepts of airspace structures and system analysis for uas traffic flows for urban areas,” in [*AIAA Information Systems-AIAA Infotech@ Aerospace*], 0449 (2017).
- [18] Kopardekar, P. H., “Unmanned Aerial System (UAS) Traffic Management (UTM): Enabling Low-Altitude Airspace and UAS Operations,” Tech. Rep. NASA/TM2014218299, NASA Ames Research Center, Moffett Field, CA 94035, USA (April 2014).
- [19] Krishnakumar, K., Ippolito, C., Kopardekar, P., Melton, J., Stepanyan, V., Sankararaman, S., and Nikaido, B., “Safe Autonomous Flight Environment (SAFE50) for the Notional Last “50 ft” of Operation of “55 lb” Class of UAS,” in [*AIAA SciTech Conference, Grapevine, Texas, USA*], AIAA (2017).
- [20] Sankararaman, S. and Kalmanje, K., “Towards a computational framework for autonomous decision-making in unmanned aerial vehicles,” in [*AIAA Information Systems-AIAA Infotech@ Aerospace*], 0446 (2017).
- [21] Sankararaman, S. and Teubert, C., “Prospective architectures for onboard vs cloud-based decision making for unmanned aerial systems,” in [*Annual Conference of the Prognostics and Health Management Society*], (2017).
- [22] Sankararaman, S., “Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction,” *Mechanical Systems and Signal Processing* **52**, 228–247 (2015).
- [23] Sankararaman, S., Daigle, M. J., and Goebel, K., “Uncertainty quantification in remaining useful life prediction using first-order reliability methods,” *Reliability, IEEE Transactions on* **63**(2), 603–619 (2014).
- [24] Hohenbichler, M. and Rackwitz, R., “First-order concepts in system reliability,” *Structural safety* **1**(3), 177–188 (1983).