Automated Emergency Landing System for Drones

SafeEYE Project

Bektash, O. M.; Pedersen, Jacob Naundrup; Gomez, Aitor Ramirez; la Cour-Harbo, Anders

Published in:
2020 International Conference on Unmanned Aircraft Systems (ICUAS)

DOI (link to publication from Publisher):
10.1109/ICUAS48674.2020.9214073

Publication date:
2020

Document Version
Early version, also known as pre-print

Citation for published version (APA):
Bektash, O. M., Pedersen, J. N., Gomez, A. R., & la Cour-Harbo, A. (2020). Automated Emergency Landing System for Drones: SafeEYE Project. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 1056-1064). [9214073] IEEE. International Conference on Unmanned Aircraft Systems (ICUAS) https://doi.org/10.1109/ICUAS48674.2020.9214073

Automated Emergency Landing System for Drones: SafeEYE Project

O. Bektash*, Jacob Naundrup Pedersen, Aitor Ramirez Gomez and Anders la Cour-Harbo

***Abstract*— Automated emergency response systems have been the focus for development of more reliable and robust safety systems, from simpler ones to the most complex. Such systems have specific requirements, such as high reliability, real-time response, and performance. For drones, they can be designed to allow compliance standards, track safe places for landing and provide an easier development for operational process. The automated response for increasing drone safety focuses on the system health for detecting failures that can lead to vehicle accidents. Given this outlook, this paper presents the SafeEYE project, which was initiated to develop and commercialise an automated emergency landing system for larger ($>$ 7 kg) drones. The system consists of a small embedded computer, mounted on a drone, that keeps track of safe places to land, or even crash, as well as the health state of the drone. When there is a failure condition, the device can terminate the flight with the least probability of fatalities. This means a significantly reduced risk for automated, typically Beyond Visual Line of Sight, operations. Therefore, SafeEYE has the potential to become a safety enabler for many applications, including farming, inspection, transportation, search and rescue. With the risk mitigation ability, the project aims at achieving formal approval of the Danish authorities and abroad. SafeEYE is planned to be manufactured as a standalone unit, provided first through drone technology suppliers and later to service providers and manufacturers of autopilots.**

I. Introduction

In 2019 EU regulation [1], [2] has been published to ensure safe, sustainable, and secure drone operations across Europe to protect the citizens' safety and privacy while enabling the less restricted drone operations. With the new comprehensive set of regulations including technical, as well as operational provisions for drones, one can reasonably expect the difficulties for drone operations. Accordingly, new drone systems will have to be individually reliable, allowing the authorities and end-users to prevent particular drone failures. EU legislation for unmanned aircraft has also adopted the proposal of Joint Authorities for Rulemaking on Unmanned Systems (JARUS) for three categories of "Open" which is on the limitations and operational rules, "Specific" and "Certified" which are on the risk assessments to be made. Such categorisation is mainly risk-based and the assessments must address both air and ground risks, either collision with another flying object or collision with humans and critical infrastructures [3] .

As like the other aerospace applications, decision-making and response systems in drones include the selection of different actions ranging from the lower levels such as selecting values in a subsystem or defining the allowable movement range to the mission level such as changes in vehicle route [4]. In critical situations where the drone experiences in-flight anomalies, having an automated response provided by system prognostics and diagnostics can ensure a safe and reliable mission outcome. It is also crucial for drone users to manage the flight risks associated with failures of their operational assets. As the current technologies are often able to identify when system degradation starts, it is possible to identify the way that the drones are failing. Determining how the systems can no longer perform their intended function can provide the possibility for drone operators to get permission to make "Beyond Visual Line of Sight" (BVLOS) flights and thus create more jobs for drone pilots, inspection companies, and other businesses.

Similar to other vehicles with safety risks, drones suffer from uncertainties in operations and hence most flight plans are highly conservative in their nature. This is combined with increased societal acceptance and perceptions of risks as well as the complexity of human behaviour, operational roles and human machine interface [5]. As a result, reliable drone response systems, which can estimate in real-time health status and the remaining useful life (RUL), draws increasing attention. In the long term, SafeEYE aims to fulfil this "missing link" by fully integrating drones as part of the aviation infrastructure with a safe and reliable operational profile and increased societal acceptance. The target for the project is the value inflexion point where SafeEYE has achieved operational status and formal acceptance as a risk mitigation device having a strong focus on offering a new and much needed service to the end-users. This is done by combining a series of existing and well-proven technologies on data analysis and processing, where measurements during flight is fused with pre-flight data to determine in real time how and where to best descent a malfunctioning drone.

A. Related Work

As there has been an increasing number of regulations and a need to allow the safe operation of unmanned aircraft [8], [9], [10], [11], there is an effort to address the challenge of risk assessment for unmanned aircraft flights. Even though much has been borrowed from the manned aviation [12], the later studies acknowledge the fundamental risk differences for unmanned aircraft, and define safety objectives by using historical analysis of human-piloted aviation accidents. In [13], a simple ground fatality expectation model is introduced with an attempt to define the safety objective variations on the design and operation of unmanned aircraft flight. Other works also refer to the concepts and definitions of

Automation and Control, Department of Electronic Systems, Aalborg University, Fredrik Bajers Vej 7C 9220 Aalborg Denmark
YM27JV@es.aau.dk, jnp@es.aau.dk, arg@es.aau.dk, alc@es.aau.dk
* publons.com/researcher/3373343

preliminary steps of risk analysis to define the meanings and characteristics, as well as the main causes and consequences. To define hazards, risks, and associated classifications of UAS (unmanned aircraft systems), the techniques in these works used functional hazard assessment (FHA) and/or failure modes and effects analysis (FMEA) [14], [15], [16]. These concepts and definitions for hazard classifications were applied in preliminary assessments for UAS with the purpose of conducting the preliminary analyses to explore potential risks unique to unmanned aviation and how those could be classified.

While some studies in the literature mainly address safety challenges by focusing on aircraft and their associated components required for the deployment, other approaches consider third-party risk associated with UAS operations. In particularly, the risk to people on the ground, who are not involved in the operations [17], [18]. Similarly, efforts have been devoted to integrate UAS by removing the threats to human safety from mid-air collisions, as well as the ground impacts [19]. Within the same context, the methodologies proposed by [20] assess the risk of operating UAS in populated areas by allowing users to estimate bystander collisions based on data, such as satellite imagery and census information, as well as the potential aircraft failure rates from manufacturer specifications and experimentally provided data.

In [21] introduces similar methods to enable safety for UAS operations in civilian airspace, and also validated their model using fatality rates caused by crashes monitored from historical data. This work also revealed that several studies have tried to predict the bystander casualty rate caused by UAS operations. Clearly obvious in the initial considerations were those explicitly relating to third party risks around airports such as crash, individual risk, and societal [24]. In the following studies, [23] introduced the risk management of unmanned flights over inhabited areas and outlined the need for an objective awareness of the related risks, and [22] presented a point-based tool that evaluates small UAS by rewarding practices considering complex flights over populated areas for extended time and BVLOS operations.

In order to determine if any fault is occurring or about to occur, some sort of prediction must be made. This could be done by Predictive Maintenance and Prognostics and Health Management (PHM) methods. These could be tripping some safety threshold such as a simple univariate limit like pressure, temperature limits, current, or even a setting to any physical quantity that is regarded critical for safe operations [25]. An example might be time to critical event conditions such as remaining time to battery depletion in drone, or a set of features from condition monitoring data with no obvious physical meaning, but correlated with unsafe conditions. Such methods have been the subject of a number of publication, such as [27] that introduced a novel end-of-discharge estimation framework for electric UAS based on a Bayesian inference driven prognostic model. In fact, there are numerous UAS prognostic works in the literature on how to conceptually approach the challenge of safety. In a following work, [26] presented such a prognostic model with a model parameter augmented Particle Filtering framework to explore UAS battery behaviour under the potential load uncertainties. Further, [28] dealt with prognostic decision making problems with complex dynamics and non-linear degradation processes, and applied their model to UAS mission planning. In [29], an approach was proposed for estimating component degradation as exemplified on a battery of an UAS electric propulsion. Their work was model-based in which the technique was linked to the internal operation of the battery and validated by data. For the similar problems of battery state health estimation, [30] addressed diagnostics and prognostics challenges of Lithium-Polymer (Li-Po) batteries for UAS by utilising several discharge voltage histories information for a data-driven approach of a Hidden Semi Markov model variant.

Even though the literature on UAS prognostics has primarily focused on battery state estimation, the health management carries the potential risk of failing to identify the overall safety risks of drones. Without accurately understanding the inclusive health pattern, the detection and diagnosis of UAS degradation would be unfeasible. Alternatively, the drone safety could be realised as a complex set of operational features that are derived using advanced analytic from condition monitoring data. An example of this is a set of features from a vibration power spectrum [25].

B. Current Method

The focus of SafeEYE project is developing, demonstrating, and commercialising an automated emergency landing system for larger (>7 kg) drones. Thus, tracking of safe landing zones, along with the health state of the drone, will be the primary methods in the development of the framework. The project will conceptually be divided into three steps as

1) Detection of malfunction (**Detect**),
2) Find a good landing spot (**Find**), and
3) Planning and guidance for landing (**Land**)

This is visualised in Figure 1. These steps can be considered chronological in terms of how they are applied when SafeEYE is in action on a drone operation.

The first step (Detect) mainly focuses on both monitoring the health status and estimating the time at which the drone can no longer perform its desired function. During flight, SafeEYE monitors the ground below the drone and when it is estimated the flight cannot be continued with sufficient safety margins, one of the previously found useful ground locations is selected for landing or crashing (Find). The third step is then to either guide the drone to a quick landing, or in severe cases, to terminate the flight by shutting down the motors to initiate a ballistic descent into the selected crash area (Land). The three steps are described in more detail in the following section.

II. Methods

This section first provides background information on the SafeEYE framework. Then the applied methods are

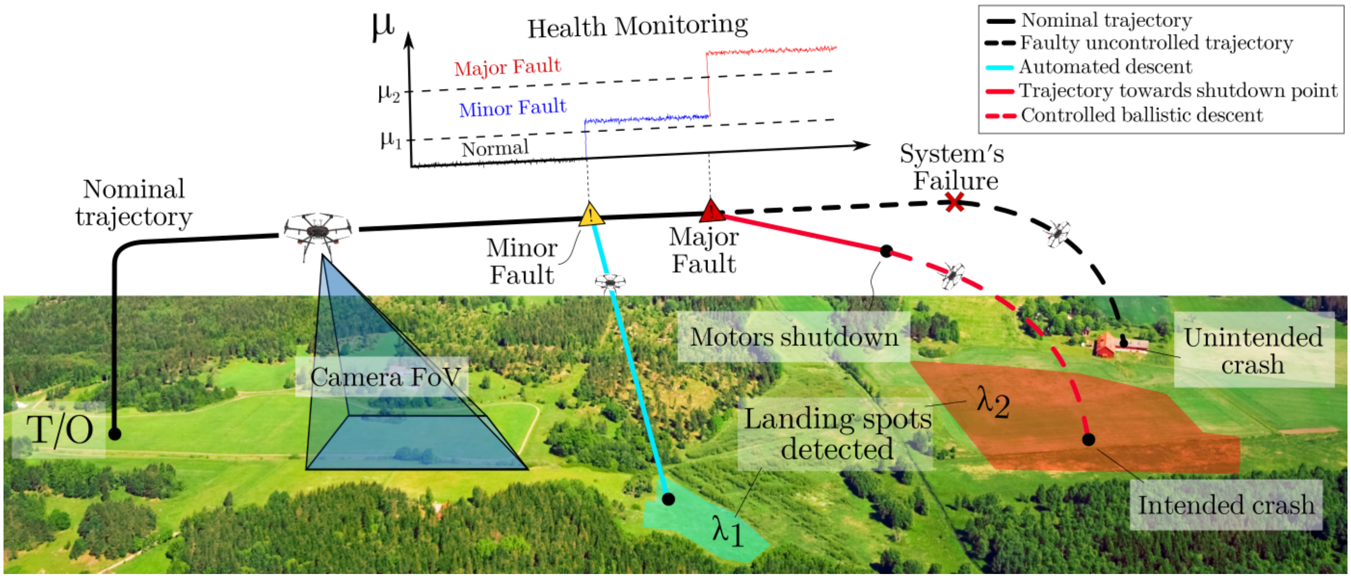


Fig. 1: SafeEYE functionality during an operation: The aircraft takes off, and monitors the ground during flight. If a minor fault is detected, a landing is attempted, and if a severe fault is detected, a motor shutdown and crash is conducted. In both cases the decision is taken based on the health monitoring and the knowledge of the ground area in the vicinity of the aircraft.

described, and the development of a comprehensive approach for safe operations is laid out. These approaches fall under three main categories: vibration analysis and time series estimation for predictive maintenance (Detect), convolutional neural networks designed to recognise visual patterns for appropriate landing areas (Find), and mathematical modelling for landing spot optimisation and the guidance system (Land).

The SafeEYE project has been running for about 18 months, and the effort has primarily been on Find, which is almost completed (and presented in a separate publication), while the Detect and Land stages are currently in development. For the Detect step, initial algorithm development is made with a subset of historical vibration data monitored from flights from a concurrent project, where high vibration data represents a potential failure condition. Although the proposed models are framed by monitored data from previous projects and artificially created data with the help of algorithms, it is planned that there will be access to some hours of flight data prior to first installation of SafeEYE on the drone, and these stages will be validated by actual condition monitoring information and data. The Find step is far in development and verified with drone operations. A brief review of these results for this step and the relevance to the safety objectives are presented below. The third step, Land, has been assumed to take the outputs from the two first steps to build two decision-making algorithms. On the one hand, a mathematical function is first proposed in order to select the most suitable spot to land among the ones identified and stored by Find. On the other hand, a simple rule based only on the output of Detect is initially made to choose the proper descent method. Planned flight tests are expected to provide insight on the performance and viability of these algorithms, as well as for the two emergency descent methods which are still in ongoing research.

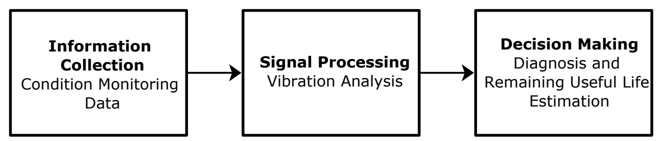


Fig. 2: Major steps in predictive maintenance stage

A. *Detection of Malfunction*

The output of the Detect step is a recommendation on emergency actions based on the data monitored through operations. To properly establish the malfunction detection, the procedure is first to deal with data acquisition by processing sampling signals obtained from real world operations, which is processed to obtain useful relevant information to drones' operational health status. This practice mainly involves monitoring historical data of condition in machinery with an attempt to identify a major change which may be indicative of a developing failure or fault. This is a significant component of predictive maintenance, diagnosis and prognosis to estimate both the current health status and the failure time, at which the drone cannot complete its desired functions. In general, the estimations can be made by realising the current and the historical system conditions that will have direct affect on the future behaviour of the system [31]. Thereupon, data acquisition is directly associated with signal processing and decision making (see Figure 2). Since the estimations for malfunction detection are statements about an uncertain events (mostly in the future), the main approach in this step is concerned with basic acceptance regarding the characteristics of degradation. The following assumptions for the predictive maintenance and diagnosis parts of the program.

- SafeEYE will be developed on the same drone as the test environment where the framework is controlled with regard to the performance and checked that the project requirements are met with reference to functionality, along with other requirements by the team developing the framework before the it meets the end user. However, the project has assumptions that planned malfunctions will be tested on lab environment and disposable devices.
- Initial algorithm development will be made with a subset of data previously taken from another project. These data contains high vibration and represent a potential failure condition.
- SafeEYE will be able to store flight data and features regarding malfunction detection.
- For the developed algorithms to work properly on a drone, it needs to have a few hours of flight data before it becomes operational.
- It is accepted that SafeEYE might be prone to uncertainties and reduced performance on first hours of flights.

SafeEYE continuously monitors the flight with the goal of finding initial failures and taking necessary actions before a catastrophic failure occurs. Therefore, its main function of the first step is to determine any potential system degradation. In general, the assessment of system health, criticality and reliability is performed to determine significant failure modes by using sensors to monitor specific components for degradation and the initiation of potential failure modes. Sensor data can demonstrate the changes in system components and provide feature vectors (time series).

Vibration sensors are often credited for sensing the mechanical degradation of system components in advance of a catastrophic failure. Common to all degradation models is the exponential characteristics and behaviour of the fault evolution such as $\exp(at^b)$, where a and b are model parameters which are case specific [32]. Accordingly, the health is written as $h(t) = 1 - \exp(at^b)$. However, each system starts at a distinct operational performance level "d" which is a case specific initial wear degradation point in the wear-space (each operational case is observed with some non-zero initial wear degradation that is unique for each observation), and maintains a distinctive "h" pattern, $h(t) = 1 - d - \exp(at^b)$.

For better understanding and interpretation of the collected data, the fault detection stage is concerned with this health degradation by analysing the monitored signals, diagnosing when an initial fault has occurred, pinpointing the fault type with a lower probability, and finally estimating remaining useful flight time. Here, the decision-making steps in to recommend efficient policies of diagnostics and prognostics. While diagnostics deal with the detection, isolation and identification of the fault, prognostics are concerned with the prediction of future failures and the remaining life time of the system. The main difference between the two is the nature of their analyses. Diagnostics focus on the posterior and current health, whilst prognostic applications are based on remaining service life referring to the time left before observing a fault given the current system condition, and the past operational profile [33], that is,

$$\mathrm{RUL} = t_\mathrm{f} - t_\mathrm{c} \mid t_\mathrm{f} > t_\mathrm{c}, Z(t_\mathrm{c}) \ ,$$

where RUL is the remaining useful life up to the failure time t_f, and $Z(t_\mathrm{c})$ is the condition profile (past operational profile) up to the current time t_c. For the remaining useful life estimate we have

$$E[t_\mathrm{f} - t_\mathrm{c} \mid t_\mathrm{f} > t_\mathrm{c} t, Z(t_\mathrm{ct})] \ .$$

For diagnostic and remaining useful life estimate, two different potential approaches are to be used:

- A pattern recognition technique that indicates a failure, and
- an analysis of the discrepancy between sensor readings and tolerable limits.

Such methods can be broadly classified into three main categories according to the way that they participate in estimation and prediction: physics-based, knowledge-based and data-driven approaches. Since SafeEYE is mainly designed to process monitoring data, a data-driven approach is used that proposes the determination of precursors to fault and remaining time by considering historical records and estimation outputs from operational data. This will also provide solutions for the long term decision making challenges such as "System Health Check Prior to Mission (or also go/no go poll)". Therefore, this step is planned to deal with flight controllers monitoring drone systems that are queried for operation and readiness status before a flight can proceed.

Another subject of this step is the confidence of predictive maintenance that relies on the accurate processing of historical training data. This is a state of being certain either that the estimation is correct or most effective. In practice, the capability would be used to schedule maintenance or assist in assets management. End users will perform management of the individual drones; therefore, they might require an intuitive, basic display that conveys information on: Current system health, remaining service life, and "confidence" in the RUL estimation. When the model used for RUL prediction, it is planned to be both consistent and indicative of a satisfying estimation. Along with developing the basics of confidence estimations, the predictive maintenance application will call for fielded usage towards its maturation. This is where a stringent performance evaluation comes in to exploit the significance of confidence concept. Currently, SafeEYE project proposes the standard predictive maintenance definitions and consistent interpretations

B. Landing Recognition

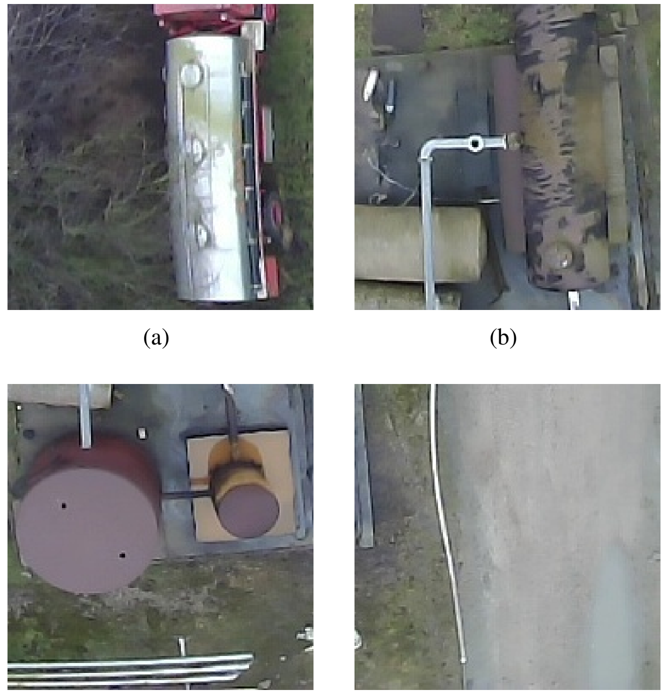
The landing recognition method proposed in the project is a machine vision approach for the detecting areas of interest that can facilitate a landing or a crash without our human injury or fatality. Since this method will be running on a small embedded computer, the chosen method is a convolutional neural network model that is trained to recognise landing spots in images captured by an built-in camera.

The data set used to trained neural network is a collection of images taken from the camera mounted on SafeEYE lab. These images are from various test flights where the primary purpose was to provide images of a variety of surfaces. In Figure 3 a sample image is shown. Such images

Fig. 3: An image taken by SafeEYE showing a variety of surfaces. See section III for details on the test flight.

require a categorisation process in which objects and features are recognised, differentiated, and understood. To classify landing locations, a script is used to split up these images into smaller frames as shown in Figure 4. These frames will then be classified in two classes, suitable for landing and not suitable for landing. Each images gives 78 frames of 180 x 180 pixels. For the initial training of the neural network, 2850 different frames are used to define the classification problem, 1011 landing locations and 1839 no-landing locations. This data set has been deemed sufficient at this time, and is therefore used in the following classification applications.

The convolutional neural network consists of an input layer of split images and an output layer of ”landing fields” and ”not landing fields”, as well as multiple hidden layers. To form the output layer, the criteria for deciding whether a landing site is good or not is based on a manual visual inspection of the frames. It has been chosen to determine green areas like grass fields as a landing site where variations in colour are limited. Therefore images with rocks, shadows, trees, houses etc. are excluded, thus labelled as ”not landing fields”. Examples of both categories from the same test flight are shown in Figure 5.



(a) (b) (c) (d)

Fig. 4: Image parts (frames) taken from the image in Figure 3.

Fig. 5: Frames showing examples of (top row) ”landing locations”, and (bottom row) ”not landing locations”.

The most essential building block in neural networks is the single-input neuron structure which is formed by three definite functional operations of input (x), weight (w) and the transfer function (f). For such a single-input neuron, the general equation is denoted by [6]

$$f(net), \quad \text{where } net = \sum_{i=1}^{n} w_i x_i \tag{1}$$

In classical feed-forward neural networks, the neurons in the input layer are connected to output neurons in the next layer. This forms a fully-connected layer. Nevertheless, in a convolutional neural network model the fully-connected layers are not employed until the very last layer(s) in the network. Therefore, they form a neural network model that swaps in a specially designed “convolutional” layer in exchange for “fully-connected” layer for at least one of the layers in the structure [34]. A nonlinear activation function is then applied to the output of the convolutions. The process of convolution continues until the end of the network and the application of one or two fully-connected layers in which the output classifications are obtained.

In SafeEYE, such a convolutional model, namely LeNet architecture [35], is used due to its computationally efficient structure. It is straightforward and small (in terms of memory footprint), making it perfect for teaching the basics of the proposed model. LeNet consist of multiple layers of convolution, pooling and activation followed by a fully-connected layer, activation function and another fully-connected layer, and at the end, a soft max classifier.

$$\begin{aligned}\text{In} &\Rightarrow \text{Conv} \Rightarrow \text{Act} \Rightarrow \text{Pool} \Rightarrow \text{Conv} \\ &\Rightarrow \text{Act} \Rightarrow \text{Pool} \Rightarrow \text{FC} \Rightarrow \text{Act} \Rightarrow \text{FC} \end{aligned} \quad (2)$$

For the activation, the LeNet architecture in SafeEYE uses a unit employing the rectifier, also called a rectified linear unit (ReLU) [36], defined as $f(x) = \max(0, x)$. ReLU tends to outperform alternative activation functions and it is widely acknowledged in deep neural network applications [37].

C. Planning and Guidance for Landing

The last phase of SafeEYE's operations is to actually perform landing. Hence, the procedures described in this section conceptually take place after a fault has been detected by the Detect step. The primary goal is to bring an emergency situation to a landing (or crash in the worst case) of the drone in a desired location. Areas recognised and classified by the Find step as a "landing field" are stored in the SafeEYE's computer, and subsequently analysed as described in the following.

In order to land the drone, two decisions need to be made: Where to land and how to land. To keep the framework simple, the questions are considered independently. First a solution for the problem of where to land is proposed, and subsequently, a solution for how to descend is proposed. Different approaches can be tackled to select between appropriate landing locations, such as *if/else* control commands and/or another neural-network. The simplicity of using *if/else* commands can easily lead to undesired emergency landing performances, whereas a neural network model may require data for training that are not easily available. Rather, the choice of landing location is based on a cost function that optimally selects among the stored landing areas based on three parameters.

- Current urgency level (from Detect step): $\mu \geq 0$,
- Safeness of the landing area (from Find step): $\lambda \in [0, 1]$
- Normalised distance to landing area: $\bar{R} \in [0, 1]$

A function, $J_p : \mathbb{R}^3 \to \mathbb{R}$, is proposed as a cost function to quantify the suitability of landing a faulty drone in the landing area p, where $p \in \mathbb{N}$ is an index used to identify a landing area from the ones stored. The function J_p provides a cost for each of the areas recognised and stored, by relating the three parameters μ, λ_p and $\bar{R}_p$ for each landing area p.

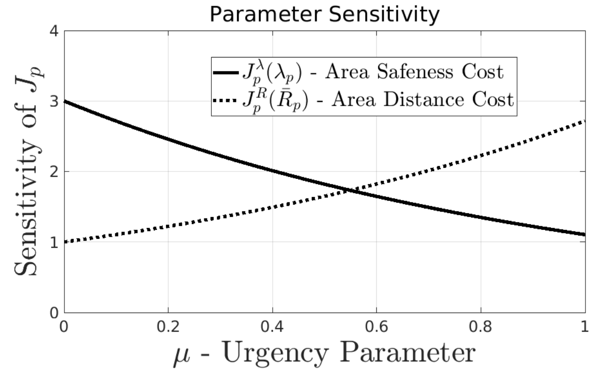


Fig. 6: Sensitivity of J_p to the parameters λ_p and $\bar{R}_p$ while increasing the urgency level μ from 0 to 1.

Note that all three parameters are independent from each other, and are bounded differently, as listed.

The intention is to have a function that provides a certain cost to an arbitrary landing area p based on its balanced quality. Subsequently, the cost given to spot p changes according to the emergency level μ, i.e. the urge to land. For instance, if a malfunction is associated with a low urgency level, it is assumed that forcing the drone to travel further to reach a safer landing spot is not considered critical. Whereas a high urgency level would prioritise a closer spot, at the expense of a lower safeness to land at that spot. This is motivated by the concept that with a high urgency level, a failure occurring during a traversal to the best, perhaps far away, landing spot is considered high risk.

Following the criteria described above, the general structure of the proposed function J_p is

$$J_p(\mu, \lambda_p, \bar{R}_p) = w^\lambda(\mu) J_p^\lambda(\lambda_p) + w^R(\mu) J_p^R(\bar{R}_p) \quad (3)$$

Since λ_p and $\bar{R}_p$ are independent parameters, a cost is associated to each of them by means of individual functions, $J_p^\lambda(\lambda_p)$ and $J_p^\lambda(\bar{R}_p)$. Each tackles only one parameter, and then is introduced linearly to the general function J_p. Subsequently, two weights, $w^\lambda(\mu)$ and $w^R(\mu)$, multiply each individual cost in order to increase or decrease the value of the individual costs to prioritise safeness or closeness. The value of these weights change with the severity of the situation, and henceforth, are defined as functions of μ.

A way to visualise J_p is imagining a 1×1 manifold, initially shaped by the individual functions $J_p^\lambda(\lambda_p)$ and $J_p^\lambda(\bar{R}_p)$. The manifold is then softly reshaped from its initial configuration according to the urgency level of the emergency situation, i.e. according to the weight functions, $w^\lambda(\mu)$ and $w^R(\mu)$. It is assumed, thus, that in a nominal state ($\mu \ll 1$) the weights are not affecting the value of the cost function J_p, i.e. $w^\lambda(0) = w^R(0) = 1$. And when the level of urgency rises the weights modify the cost J_p in a manner corresponding to the criteria that has been previously described. To illustrate this, the sensitivity of an example of cost function J_p has been taken as a function of the urgency level μ, which looks like shown in Figure 6. It can be seen that the value of J_p associated to a low urgency situation is provided mostly by the safety parameter λ_p, whereas the distance of the landing area provides low values to the general function J_p. However, this relation is swapped by increasing enough the urgency level μ, which matches with the desired behaviour described before. Henceforth, functions $J_p^\lambda(\lambda_p)$, $J_p^\lambda(\bar{R}_p), w^\lambda(\mu)$ and $w^R(\mu)$ will be sought, such that (3) behaves like, or similar to, Figure 6. The selected area to land is, therefore, the one which gives the lowest cost value of J_p, meaning it is the most suitable location to land given the current emergency level.

In order to perform the landing procedure towards the selected area, three methods will be considered (the first being the least aggressive to perform and the last the most aggressive).

- Auto-pilot landing procedure.
- SafeEYE automated descend.
- Ballistic descend.

The choice of which procedure to choose depend strictly on the severity of the situation, namely, on μ. Therefore, thresholds on the value of μ will be studied in subsequent research in order to determine exactly when is it appropriate to choose either method.

III. Testing and Results

SafeEYE has been mounted, integrated, and flown in a number of test setting on a DJI Matrice 600. This section briefly describes some of the results from these test flights.

A. Test flights

A number of test flights have been conducted with the SafeEYE, primarily to collect data and to test the Find step. Most prominently, three campaigns have been conducted with a total of approximately 12 hours of flight. Two sites in Denmark have been used, one is an emergency responders training facility at Rørdal, Aalborg (which is where the images in Figures 3, 4, and 8 are from), and the other is a military training compound consisting of about 30 empty houses. This areas was also used for test flights with home guard personnel acting as city residents. These sites have been used, because they have both urban and rural ground textures with legally being an urban area, and because they can reasonably be closed for public access. In addition, the military area has an airspace restriction zone, which allowed for operations up to 150 m. The images in Figure 5 is from this site.

The data collected during operations consists of 10,000's of images, vibration data, attitude and flight trajectories, and on-the-fly estimation results.

B. Results

As the project has completed the Find step, the image recognition module has been tested with imagery from flights, and results are briefly presented in this section. The initial image data set is from the test flight at Rørdal in Aalborg, Denmark. From the first tests using the model from Section II-B, it is concluded that the model is over-fitting due to the gap between training and validation loss which means

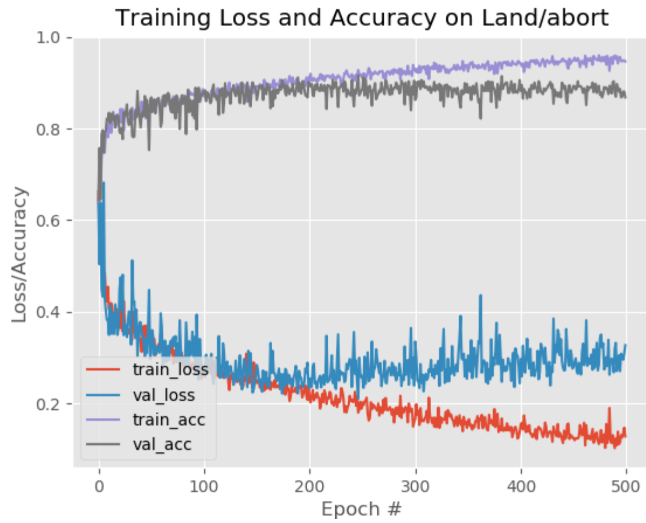


Fig. 7: Illustration of the learning curves for the LeNet classifier with a dropout layer.

that the network has over-memorised the training examples. Thus, it has not been adequately trained to generalise to the new situations. It is acceptable to have a gap between the training loss and validation loss curves, however the validation loss should not be constantly increasing as witnessed in the testing. Therefore, a dropout layer has been added and the new training loss and the accuracy has been increased as seen in Figure 7. Here, the gap between train- and val- loss is reduced. The results of classification test is shown in Figure 8 in which the classifier is able to correctly classify all the frames. As seen by the high accuracy of the classification in Figure 8, the proposed method performs well at classifying correct landing locations. A further point to be mentioned is that the model is able to classify the frames in a reasonable time which is intended for the small computer mounted on drone. The LeNet model built in the training script only takes up 15 MB of memory. Therefore, it is a rather small model which is a very desired as it will not fill up the RAM on the device.

IV. Conclusion

In this work, an automated emergency landing system for larger drones is proposed for flight safety. SafeEYE is designed to be an external attachable device capable of working on any larger drone, no matter what its physical characteristics are. The project is planned take control of the guidance tasks to perform a fast and safe autonomous landing when a fast emergency landing is required. Being in charge of such actions means that the responsibility for casualties lies on SafeEYE, hence a high degree of certainty and safeness regarding the landing is required.

The preliminary test outcomes show that the algorithm developed for landing recognition has been effective for detecting areas of interest for the development of project. Therefore, it has been demonstrated that the framework has a potential to overcome uncertainties in operations with a safe profile and increased acceptance. Additionally, the proposed

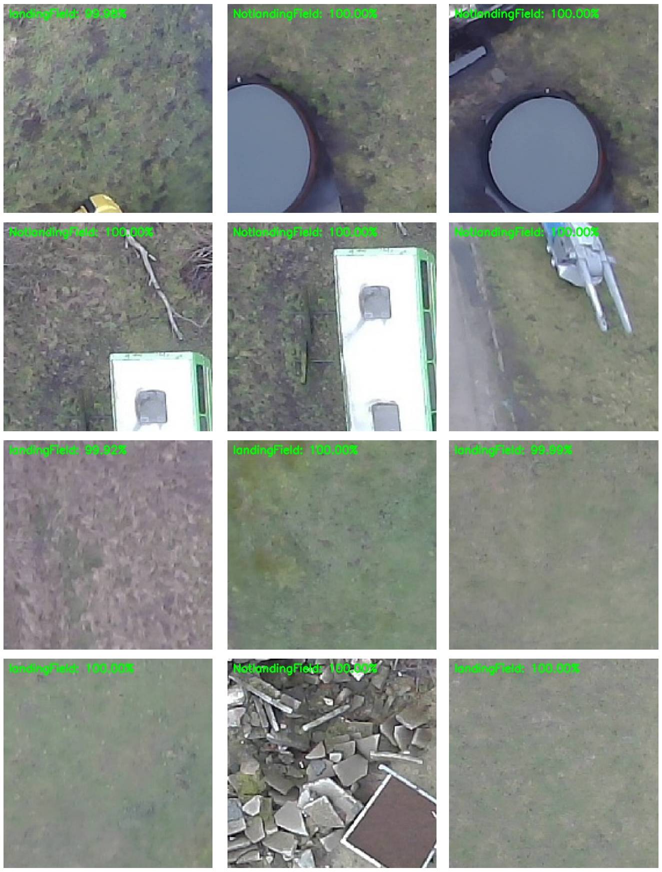


Fig. 8: Results from the classifier. The green text shows if it is considered a landing spot or not, and the percentage the confidence in the classification.

stages of ”fault detection” and ”landing guidance” show that a strategic implementation of subsystems can improve the overall operational reliability. The framework, therefore, has a great potential of improvement in drone safety.

Acknowledgement

We would like to thank Jesper Andersen (CEO & Founder at SenseAble) for his support and assistance. We would also like to extend our thanks to Simon Jensen (Assistant Engineer, Department of Electronic Systems, Aalborg University) for his help in drone operations.

References

[1] EU, Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft (Text with EEA relevance.), Official Journal of the European Union, vol. 2019, May, 2019.

[2] EU, Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems, Official Journal of the European Union, vol. 2019, March, 2019.

[3] la Cour-Harbo, Anders. ”The value of step-by-step risk assessment for unmanned aircraft.” 2018 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2018.

[4] Balaban, Edward, and Juan J. Alonso. ”A modeling framework for prognostic decision making and its application to UAV mission planning.” Annual conference of the prognostics and health management society. 2013.

[5] Susini, Alberto. ”A technocritical review of drones crash risk probabilistic consequences and its societal acceptance.” Lnis 7 (2015): 27-38.

[6] Lippmann, Richard P. ”An introduction to computing with neural nets.” Artificial neural networks: theoretical concepts. IEEE Computer Society Press, 1988.

[7] Hagan, Martin T., Howard B. Demuth, and Orlando De Jesús. ”An introduction to the use of neural networks in control systems.” International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal 12.11 (2002): 959-985.

[8] Hagan, Martin T., and Howard B. Demuth. ”Neural networks for control.” Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251). Vol. 3. IEEE, 1999.

[9] ICAO. Cir 328 AN/190 Unmanned Aircraft Systems (UAS) Circular (2011).

[10] CASA. ”AC 101-1 (0)–Unmanned Aircraft Rockets–Unmanned Aerial Vehicle (UAV) Operations, Design Specification, Maintenance and Training of Human Resources.” Advisory Circular, July (2002).

[11] CAA. CAP 722 Unmanned Aircraft System Operations in UK Airspace: Guidance. 2015.

[12] La Cour-Harbo, Anders. ”Quantifying risk of ground impact fatalities of power line inspection BVLOS flight with small unmanned aircraft.” 2017 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2017.

[13] Clothier, Reece A., and Rodney A. Walker. ”Determination and evaluation of UAV safety objectives.” (2006): 18-1.

[14] Hayhurst, Kelly J., Jeffrey M. Maddalon, Paul S. Miner, George N. Szatkowski, Michael L. Ulrey, Michael P. DeWalt, and Cary R. Spitzer. ”Preliminary considerations for classifying hazards of unmanned aircraft systems.” (2007).

[15] Denney, Ewen, Ganesh Pai, Corey Ippolito, and Ritchie Lee. ”An integrated safety and systems engineering methodology for small unmanned aircraft systems.” In Infotech@ Aerospace 2012, p. 2572. 2012.

[16] Di Rito, Gianpietro, and Francesco Schettini. ”Impacts of safety on the design of light remotely-piloted helicopter flight control systems.” Reliability Engineering & System Safety 149 (2016): 121-129.

[17] Ancel, Ersin, Francisco M. Capristan, John V. Foster, and Ryan C. Condotta. ”Real-time risk assessment framework for unmanned aircraft system (UAS) traffic management (UTM).” In 17th AIAA Aviation Technology, Integration, and Operations Conference, p. 3273. 2017.

[18] Bertrand, S., N. Raballand, F. Viguier, and F. Muller. ”Ground risk assessment for long-range inspection missions of railways by UAVs.” In 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1343-1351. IEEE, 2017.

[19] Lum, Christopher, and Blake Waggoner. ”A risk based paradigm and model for unmanned aerial systems in the national airspace.” Infotech@ Aerospace 2011. 2011. 1424.

[20] Lum, Christopher, Kristoffer Gauksheim, Chris Deseure, Juris Vagners, and Tad McGeer. ”Assessing and estimating risk of operating unmanned aerial systems in populated areas.” In 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference and 19th AIAA Lighter-Than, p. 6918. 2011.

[21] Melnyk, Richard, Daniel Schrage, Vitali Volovoi, and Hernando Jimenez. ”A third-party casualty risk model for unmanned aircraft system operations.” Reliability Engineering & System Safety 124 (2014): 105-116.

[22] Burke, David A., Charles E. Hall, and Stephen P. Cook. ”System-level airworthiness tool.” Journal of Aircraft 48.3 (2011): 777-785.

[23] Clothier, Reece A., Rodney A. Walker, Neale Fulton, and Duncan A. Campbell. ”A casualty risk analysis for unmanned aerial system (UAS) operations over inhabited areas.” (2007).

[24] Evans, A. W., Foot, P. B., Mason, S. M., Parker, I. G., & Slater, K. ”Third party risk near airports and public safety zone policy.” (1997): 220.

[25] Goebel, Kai, Brian Smith, and Anupa Bajwa. ”Ethics in Prognostics and Health Management.” International Journal of Prognostics and Health Management, 2019

[26] Saha, Bhaskar, Cuong C. Quach, and Kai Goebel. ”Optimizing battery life for electric UAVs using a Bayesian framework.” 2012 ieee aerospace conference. IEEE, 2012.

[27] Saha, Bhaskar, Edwin Koshimoto, Cuong C. Quach, Edward F. Hogge, Thomas H. Strom, Boyd L. Hill, Sixto L. Vazquez, and Kai Goebel.

"Battery health management system for electric UAVs." In 2011 Aerospace Conference, pp. 1-9. IEEE, 2011.
[28] Balaban, Edward, and Juan J. Alonso. "A modeling framework for prognostic decision making and its application to uav mission planning." Annual conference of the prognostics and health management society. 2013.
[29] Goebel, Kai, and Bhaskar Saha. "Prognostics applied to electric propulsion UAV." Handbook of Unmanned Aerial Vehicles (2015): 1053-1070.
[30] Eleftheroglou, Nick, Dimitrios Zarouchas, Theodoros Loutas, Sina Sharif Mansouri, George Georgoulas, Petros Karvelis, George Nikolakopoulos, and Rinze Benedictus. "Real time diagnostics and prognostics of UAV lithium-polymer batteries." Proceedings of the Annual Conference of the PHM Society. Vol. 11. No. 1. 2019.
[31] Goebel, K., Vachtsevanos, G., and Orchard, M. E. (2013). Prognostics. NASA, Prognostics - Intelligent Systems Division.
[32] Saxena, Abhinav, Kai Goebel, Don Simon, and Neil Eklund. "Damage propagation modeling for aircraft engine run-to-failure simulation." In 2008 international conference on prognostics and health management, pp. 1-9. IEEE, 2008.
[33] Jardine, Andrew KS, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance." Mechanical systems and signal processing 20.7 (2006): 1483-1510.
[34] Rosebrock, Adrian. Deep Learning for Computer Vision with Python: Starter Bundle. PyImageSearch, 2017.
[35] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86, no. 11 (1998): 2278-2324.
[36] Hahnloser, Richard HR, Rahul Sarpeshkar, Misha A. Mahowald, Rodney J. Douglas, and H. Sebastian Seung. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit." Nature 405, no. 6789 (2000): 947-951.
[37] LeCun, Yann. "Yoshua Bengio, and Geoffrey Hinton." Deep learning. nature 521.7553 (2015): 436-444.