Integration of Condition Information in UAV Swarm Management to increase System Availability in dynamic Environments

Lorenz Dingeldein[1]

[1] *Institute of Flight Systems and Automatic Control, Darmstadt, 64289, Germany*
dingeldein@fsr.tu-darmstadt.de

ABSTRACT

The approach of prognostics and health management (PHM) focuses on the real-time health assessment of a system under its actual operating condition and even extending this by the prediction of the future state based on up-to-date system information. This pursues the aim to derive more advanced maintenance or asset deployment strategies in order to keep the operation of the system safe and reliable. In this context, the outcome of a PHM system is often used as a decision support. For a high fidelity system where the actual state is considered at every timestep and a decision is executed immediately based up on this information, Reinforcement Learning (RL) becomes a tool to find an optimized solution. Therefore the paper presents a methodology that integrates health and operational data into a RL approach in order to derive immediate operational strategies for lower degradation and higher safety and reliability. The approach is evaluated on the basis of a swarm of unmanned aerial vehicles (UAVs) that performs a complete-area path-coverage (CAPC) mission. It can be shown that the integration of health information as well as environmental data describing dynamic operating conditions lead to lower degradation and result in more reliable operations of the swarm while achieving a more flexible mission performance compared to pre-divided swarm-missions. Varying states are also taken into account, which emphasises this approach to be a highly dynamic PHM system application.

1. INTRODUCTION

To avoid fatal incidents, safety and reliability are two major objectives for developments in the aviation industry (Tumer, 2011). While safety refers to system operations without causing harm or damage to people, property or the environment, reliability focuses on the ability of a system to perform its intended functions without failure or degradation over time (Stapelberg, 2009). The latter is the motivation to develop PHM functionalities where system states are predicted in order to derive decisions for maintenance strategies to increase reliability, availability and save costs. It is evident that usage leads to some level of degradation. Implementing a specific usage strategy can mitigate this degradation, resulting in extended system functionality and improved operational reliability, even in systems that have already experienced degradation. This approach is called prescriptive maintenance which complements PHM approaches by utilising their outputs, namely state detections and remaining useful life (RUL) predictions. While traditional PHM approaches try to extend system usage through a more precise calculation of the RUL, prescriptive maintenance guarantees for a reliable usage of systems that already show remarkable degradation (Marques & Giacotto, 2019). Combining usage specific degradation with a PHM based condition assessment is subject of this paper, which provides a prescriptive maintenance strategy using reinforcement learning.

The system used in this paper to implement the condition-based operational optimisation is a UAV swarm. The high-level mission goal of the swarm is the CAPC which applies to time-critical reconnaissance missions and also covers a common problem definition in the field of multi-agent (MA) robotics. The following reasons emphasize the suitability of this system and use-case for the developed approach:

- **Mission reliability:** System functionality of every swarm member needs to be guaranteed in order to be able to fulfill high level mission goals of a complete area coverage. While operational capabilities of a single UAV are limited, a swarm has the advantage of achieving more challenging mission goals in shorter time. The time-factor is crucial, for example, in search and rescue missions or forest fire observations.
- **Autonomy:** UAVs do not have a pilot on board. This means that various functionalities have to be automated. System state detection, as part of a PHM approach, is a decisive one in order to guarantee for reliable system functionality.
- **Redundancy:** Every individual swarm member represents a redundancy in the swarm structure. Individual

tasks can be distributed reasonable within the whole swarm in order to define a specific usage based upon environmental conditions. This guarantees a flexible task assignment that allows for optimization strategies.

In current research literature, the aspects of the integrated approach developed in this paper are considered separately. The basic idea of using PHM for a dynamic reliability assessment is described by the authors from (Heier, Mehringskötter, & Preusche, 2018). The paper emphasises the connection of PHM and reliability topics to develop decision support tools. The authors in (Bougacha & Varnier, 2020) use PHM as a driver for decision support. They pursue the primary goal of achieving higher reliability, availability and operational safety. Especially health and RUL indicators are utilized from the PHM approach in order to establish a decision-making process. Early approaches of in-mission decision making based on system states are discussed in (Andersson et al., 2015) and (Alighanbari, 2004). The latter even includes changes within the environment where the UAVs need to react to. In addition not only one UAV is part of the mission but a swarm of UAVs is considered. Data-driven approaches and machine learning techniques were not as easy accessible and developed as they are nowadays, leaving potential for the problems presented in these papers. A more recent consideration can be found in (Darrah, Quiñones-Grueiro, Biswas, & Kulkarni, 2021) where they use an online state observation to update parameters that optimize the prognosis for specific mission profiles. The better prognostic performance can than be used to derive more precise decisions but the focus of this paper is on a single UAV.

While the previously mentioned literature deals with linking PHM approaches with reliability, the following literature analysis focuses on deploying multi-agent swarm operations in a digitized environment. A baseline for multi-agent path-coverage is shown in (Cho, Park, Park, & Kim, 2021) where different grid-based map representations are compared. Even though hexagonal grids show certain advantages, such as increased navigation capabilities, the use of a cubic grid based map representation seems to be a suitable choice for the CAPC mission. Another approach for efficient swarm applications is described in (Mahmoud Zadeh, Yazdani, Elmi, Abbasi, & Ghanooni, 2022) and focuses on data acquisition. This approach could be interesting when deploying the swarm management approach from this paper in the real world and a good concept for data acquisition is needed. Nevertheless it describes the possibilities of UAV swarms. No CAPC is performed, but the distance between UAVs for better sensor measeruments is taken into account and considered as a useful approach. In (Radzki et al., 2021) travel uncertainties for a complete UAV fleet get determined. The result is used to optimize the usage of a UAV fleet but no in-mission decisions are made. This approach rather solves a scheduling problem than dealing with in-mission decisions to react on environmental conditions and system changes.

In order to make dynamic in-mission decisions, the approach of this paper uses reinforcement learning. This allows a large amount of heterogeneous data to be taken into account, which can change spontaneously in a sequential simulation. The successful application of reinforcement learning to a similar problem statement can be seen in the following literature. Using RL to control a swarm of buoys is described in (Kouzehgar, Meghjani, & Bouffanais, 2020). The goal is also the CAPC mission but input data differs in contrast to the deployment of UAVs. More comparable is the approach in (Puente-Castro, Rivero, Pazos, & Fernandez-Blanco, 2022) where a CAPC mission is performed with UAVs. The focus lies on the high level mission goal and enables the identification of relevant parameters for the coverage task. In addition (Xiao, Wang, Zhang, & Cheng, 2020) propose an approach to solve a CAPC task as well. No considerations of external factors or systems states are integrated into the approach but it helps to get an overview to solve the high level mission goal of CAPC. The closest approach is presented in (Theile, Bayerlein, Nai, Gesbert, & Caccamo, 2020) where power limitations of UAVs are integrated into a RL approach. The CAPC mission is specified as the target, but in fact more of a path-finding algorithm is implemented, which appears to be too permeable for reconnaissance missions and power limitations do not reflect the link between usage and degradation.

This paper uses the individual results from the literature stated above to develop an integrated solution for the condition-based organisation of a UAV swarm for the CAPC task using reinforcement learning. The general approach, including the experimental setup, is presented in Section 2. The results of the approach, applied to the described use case, are presented in Section 3 and analysed in Section 4. The paper concludes with a summary and an outlook on future work in section 5.

2. METHODOLOGY

To solve the task of a CAPC mission performed by multiple UAVs with respect to their health condition, a reinforcement learning method is implemented in python using RLlib (Liang et al., 2018). RLlib is only one of many possibilities to implement RL, whereby the following aspects are favourable for this paper:

- It is open-source
- It allows the integration of own simulation-environments
- It contains the option of implementing multi-agent reinforcement-learning (MARL) approaches

For the learning algorithm the Proximal Policy Optimization (PPO) is chosen. To date, this is the only MARL-capable algorithm in RLlib, so the implementation in comparable libraries should also be considered.

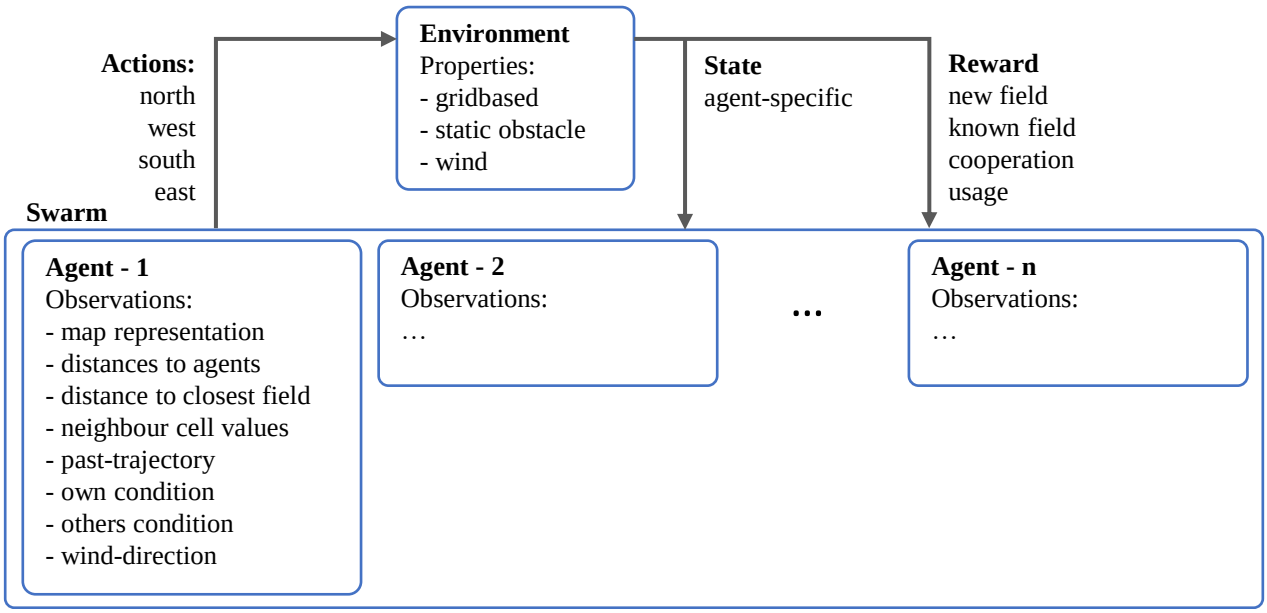

Figure 1. Framework for the multi-agent reinforcement-learning algorithm with consideration of system condition and environmental influences

The MARL method is similar to the standard RL framework where an agent is interacting with an environment through choosing actions and consequently receiving a reward. The basic principle of the MARL method with respect to the given task is shown in Figure 1. The methodology is based on the single-agent RL approach described in (Wiering & Van Otterlo, 2012) and extended with specifications to implement a MARL specific model. As it is the task to cover a certain area through creating a trajectory based on the movement decisions into the four main directions, a squared grid based environment is beneficial. This presupposes that a search is carried out along the search path with a certain radius, whereby simplified squares are assumed for coverage of a certain area that has been observed through a fly-over. While a third dimension could be used for deconfliction, it is neglected in this case to simplify the complexity of the system. The focus is on optimising a UAV swarm so that not only one agent interacts with the environment, but the actions of several UAVs are orchestrated and used as input for the environment. The action space thus becomes a vector that represents the four main directions of possible movements into north, east, south and west direction for every agent that takes part in the mission. The observation for every agent is derived from the state of the environment and takes agent-specific information into consideration. The reward rates the agents behavior and thereby helps the RL-algorithm to learn and successively improve the accumulative reward that is gathered in one mission. Every component of the MARL model is described in more detail in the following sections.

2.1. Environment Design

The MARL approach assumes multiple agents that cooperate together and interact with an environment through the execution of actions. The action of an individual agent changes the environment and in reverse calculates a reward that is fed back to the agent. In order to be aware about the actions to take the agent receives a state representation from the agent's point of view in form of observations. An environment model is necessary to provide a realistic and dynamic interaction between the environment and the agent, allowing the MARL model to learn and improve its decision-making through trial and error for a lot of training runs, which is ultimately understood as training process.

The use-case specified environment design is based on a grid based representation of a search field, that needs to be covered by a swarm of agents, the UAVs. The visualization of the environment used in this paper is shown in Figure 2 and subsequently described in detail.

The cell shape is square. This leads to four primary directions of movement, where the distance from the center of one cell to the center of its neighboring cells remains consistent. Cells that have been discoverd are displayed in beige, not visited cells are colored in green. Cells that are obstacles are colored as bricks and the wind direction is indicated as yellow arrows left and on top of the searchfield-cells. Assuming that the UAV proceed with a constant speed, the travelling distance of one timestep within the environment model is constant, which is also fits to the square shaped cells. When an UAV moves from one cell to another, it increments the value assigned to

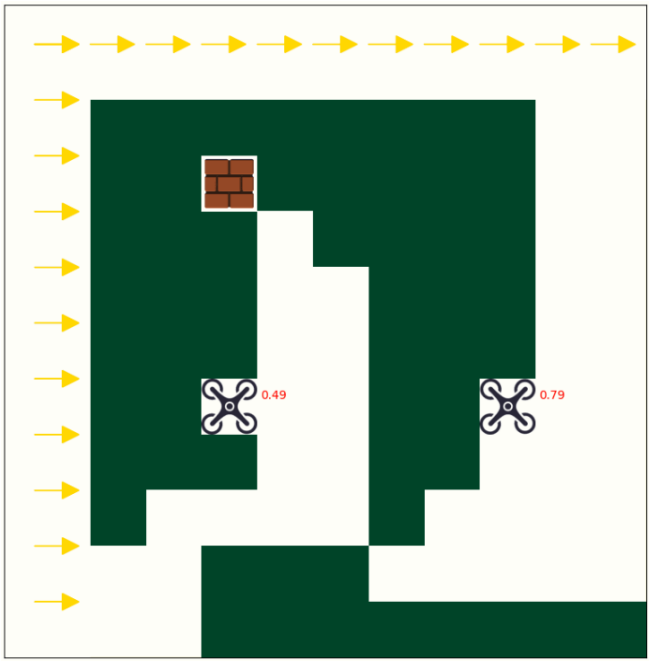

Figure 2. Environment design for a UAV-Swarm CAPC-Mission with external factors and varying system conditions

the cell by one, representing the number of visits to that cell. The highest value within the map is ten, which is utilized both for static obstacles and as a termination condition if an UAV visits a single cell too often. As an observation, which is described in detail in subsection 2.1.2, the UAV draws the map information and fortifies it to a potential map. In addition to the search field representation, the environment also incorporates a wind simulation. The wind simulation is used to specify the main influence on the system usage. A detailed description of the influence of wind on UAVs can be found in (Wang, Wang, Ali, Ting Ting, & Wang, 2019). It is assumed that the UAV is able to fly at constant speed under any wind condition. This results in a different power demand, depending on the direction of movement of the UAV and the prevailing wind direction. Higher power consumption means greater stress on the components and therefore increased degradation. The swarm configuration enables a system management where degraded UAVs take over the coverage of the search field crosswind and are thus exposed to lower degradation, while intact UAVs can take over more demanding trajectories against the wind and can absorb higher degradation without noticeable increasing the risk. The risk mitigation can be derived from the general assumption of degradation taken from (Kim, An, & Choi, 2017). While at the beginning of a system life the degradation is mainly characterised through wear represented through a linear progression, the degradation grows exponentially at the end of a system life, which results in higher chances for unexpected system failures.

Within the environment design wind is considered as constant while the direction of wind can change between the missions. Generally missions are treated individually so that the condition of one UAV gets defined at the beginning of a mission as well. The condition is chosen arbitrarily between 0.1 and 1, representing UAVs with high degradation when a low value is chosen and UAVs with a good condition if a value is chosen that is close to one.

The upcoming sections explain the design of the remaining parts of the MARL approach: the action space, the observation space, and the reward function. Additionally, it covers how missions are initialized and outlines the experiments conducted to assess RL-algorithm performance.

2.1.1. Action Space

The grid-based representation of the search field enables movements along the four main directions, from cell center to cell center, ensuring an equal traveling distance. The actions are defined as a single value from zero to three, representing the four main directions. The movement than is executed on a global scale, which means that based on the chosen value from the movement vector, the UAV moves to the north, west, south or east. In contrast, an UAV-centred approach could be chosen, which changes the direction of flight depending on the chosen action previously. In this case, four actions would also be conceivable, one value for continuing to fly straight ahead, one value each for a left or right turn and one value for reversing the direction of flight. In the remainder of the paper, however, the global approach is pursued further.

2.1.2. Observation Space

The UAVs draw an self-centered observation from the environment after changing it with their action. The observation space contains the following:

- **Map representation**: The UAV gets a matrix that counts the visits of the fields. In addition other UAVs as well as obstacles are highlighted with a value of the maximum allowed visits for one run. The latter is used for the termination condition and is described in more detail in section 2.1.4.
- **Own position**: The UAV gets its current position after it moved. Because of the two dimensional characteristics of the environment the position is represented globally as a xy-position within the environment grid.
- **Distances to UAVs**: The UAV gets the distances in number of cells in xy-direction to all other UAVs that are operating for one mission.
- **Field distances**: The UAVs gets the distances in number of cells in xy-direction to the closest field that does not count any visit from any UAV.
- **Surrounding**: With the surrounding data the UAVs get a representation of the environment based upon their cur-

rent position. With a size of four by one the surrounding matrix contains the values of the local map representation based on the UAVs position into the four main direction. If the UAV operates close to the border of the search field, values that exceed the search field are represented with a value of ten which is equal to the value of obstacles or other UAVs which should not be visited and lead to a mission termination.

- **Movement history**: The movement history is a vector of the length of five which contains the direction decisions of the UAV in a chronological sequence. For every step of the UAV within the environment the last value of the sequence is deleted, the sequences and a the direction that the UAV moved during this step is added as the first entry to the vector.
- **Own condition**: The UAV needs to know its own condition to compare it with the condition of the other UAVs. It is a normalized value between zero and one and saved as a scalar in the observation space.
- **Others condition**: Due to the same reason as before the observation space of a single UAV also contains all the conditions of the other UAVs that are participating in the swarm mission.
- **Wind information**: As the wind information mainly is responsible for the usage and degradation of the UAV it is also integrated into the observation space as a two dimensional, directional vector.

Based on the observations of the environment and the UAVs behavior the RL-algorithm is able to coordinate the UAVs movements with respect to dynamic environmental states. The goal is to reduce intensive usage for UAVs with bad condition, which gets then compensated by the UAVs that are in good condition. This results in an overall lower degradation according to (Kim et al., 2017) where it is stated that the degradation of a system increases over usage time in two steps, firstly linear and afterwards exponentially. Further more it reduces the risk of sudden system failures which occur with a higher chance to the end of life of a system and therefore decreases mission risk and increases mission reliability. To allow the UAV to optimize its decisions it receives a reward after every step according to section 2.1.3. The UAVs also exchange and communicate information about their position and condition, enhancing their decisions even further, establishing swarm intelligence.

2.1.3. Reward Function

In order to get the UAV to perform as desired, it receives a reward based upon its decision and the changes that occur within the environment at every step and at the end of the mission. The major goal in the mentioned use-case is to completely cover a designated area. The sub task consists of the efficient coverage of this area with respect to the systems usage that in combination with the degradation state has an impact on mission reliability. Therefore the reward can be divided into two types. The step-wise reward that is applied at every step on every UAV and a sparse reward (Hare, 2019) that is applied at the end of a mission.

The step-wise reward consists out of a positive reward for visiting unseen cells of the environment. For every new cell the UAV receives a reward of $R_{new\_visit} = 1$. That causes the reward to increase linear for the exploration of new fields. This motivates the UAV to search for isolated cells. The UAV receives a reward of $R_{already\_visited} = -1$ if the cell was already visited. This encourages to discover new cells as fast as possible.

To support cooperative search the UAVs get an additional positive reward R_{coop} every step if there is more than one UAV active to complete the mission. This means that the swarm has to organize itself based on the environment representation and without crashing into an obstacle, another UAV or the boundary of the searchfield in order to achieve this reward. The formula for R_{coop} is as follows:

$$R_{coop} = \begin{cases} 1 & n_{UAVs\_active} > 1 \\ 0 & else \end{cases} \tag{1}$$

The reward function incorporates both the UAVs' conditions (AC) and usage conditions by comparing the direction of movement with the wind direction. It is assumed that flying against the wind (headwind) is more energy consuming than flying crosswind or with the wind (tailwind). Therefore an angular comparison of wind direction and movement direction is made and energy cost (EC) for the manoeuvre gets calculated as following:

$$EC = \begin{cases} 0 & Tailwind \\ 0.5 & Crosswind \\ 1 & Headwind \end{cases} \tag{2}$$

It is only possible to use this energy cost if a constant speed is assumed. This is also helpful for integrating the system behaviour into a reinforcement learning environment. This simplification is made in order to focus on and analyse the interactions of degraded systems and environmental conditions within the multi UAV environment. To link environmental conditions and degradation, the reward is conditionally calculated as follows:

$$R_{usage} = \begin{cases} 1 & AC > 0.5 \; and \; EC = [0,1] \\ -5 & AC > 0.5 \; and \; EC = 0.5 \\ -1 & AC < 0.5 \; and \; EC = [0,1] \\ 5 & AC < 0.5 \; and \; EC = 0.5 \end{cases} \quad (3)$$

It shows that R_{usage} is dependent on the wind direction and the UAV condition. An UAV in good condition is meant to fly against the wind. This means that the UAV has to fly with the wind after a certain amount of steps in wind direction in order to not leave the search field (which leads to a mission termination). For this reason, it is not possible to differentiate between flying with and against the wind. Accordingly the flight movements of the UAV in bad condition must inevitably take place increasingly in crosswinds to reduce the proportion of movement directions against the wind.

The cumulative reward values calculated for each UAV at every step are aggregated over an episode which stands for a mission until a termination criteria is met. Initially, rewards are determined individually for each UAV, and at the episode's end, the total rewards across all UAVs are summed up. The end of an episode is initiated by predefined termination conditions. An additional reward known as the sparse reward is introduced alongside the termination condition, both of which will be further detailed in the following subsection.

2.1.4. Termination Conditions

Termination conditions are necessary to end an episode which is equivalent to a mission. They can be triggered if the mission task is fulfilled, the UAV's behavior leaves specified boundaries or to prevent inefficiency where the episode is trapped in an infinite loop. With the problem at hand, the termination conditions are chosen as follows:

- **Completely covered:** The UAVs were able to visit every cell of the designated search field at least once. For completing the task the UAVs do not get a negative reward. This can also be interpreted as a sparse reward that motivates the UAV to perform the task as efficient as possible with regard to the coverage performance. If an UAV is not active at the end of an episode, it gets a negative reward as described in the crash termination condition.
- **Inefficient search:** The episode gets cancelled if one of the cells within the search field gets visited more than ten times. In that case the sparse reward is -100 minus the number of unexplored fields of the search field. This reward applies for every UAV of the swarm that is still active at that time. Otherwise the crash termination. Otherwise, crash termination has already been applied.
- **Crashes:** It is classified as a crash if an UAV shares the same cell with another UAV, an obstacle or if it leaves the search field. In that case the sparse reward is calculated the same way as it is calculated for the inefficient search and the crashed UAV stops exploring the search field.

The primary objective of the MARL approach is to maximize the accumulation of rewards within a single episode such that the reward function significantly determines the behaviour of the UAVs. Within section 4 the effects of changing the reward function will be discussed in detail.

2.1.5. Initialization

Certain initial conditions must be defined to start the simulation. This includes:

- **Number of UAVs:** The primary focus of the RL-algorithm pertains to the optimization of the concurrent operation of multiple UAVs. The parameter dictating the swarm size can be specified during the initialization phase.
- **Starting location of UAV:** The UAVs are meant to fly to the designated search field, therefore their starting position is always at the boarder of the search field. To maintain a certain distance to each other, every UAV starts from another side of the search field, representing different UAV bases and approaching directions.
- **UAV condition:** The condition of the UAV is determined through a random selection process within the interval of 0.1 to 1, with precision of two decimal places.
- **Map representation:** A map in form of an array, representing the search field coverage, that counts the visits of each cell. The map is adjusted during the course of the mission as described in section 2.1.
- **Wind direction:** An initial wind direction is defined in form of a two dimensional vector.
- **Obstacle position:** While the UAVs can be understood as moving obstacles, fixed obstacles are also defined within the initialization phase as high values in the map representation.

The parameters during initialization are adaptable to specific requirements, facilitating the experimentation and evaluation of the RL-algorithm across diverse scenarios. The next section provides detailed explanations on how the parameters are set up for the experiments conducted in this paper.

2.2. Design of Experiments

A Monte Carlo simulation was run to assess the capability of reinforcement learning in optimizing specific relationships, particularly focusing on the dynamic management of UAV degradation in response to varying environmental conditions during the execution of a CAPC mission. The experiments are set up almost with the same parameters but are randomized with the regard to the following parameters:

- UAV starting position

- Obstacle location
- Wind direction
- UAV condition

The training parameters such as number of episodes, batch size and RL-algorithm are chosen as it is proposed by the documentation of the Python library of RLlib. For the evaluation, the trained RL-algorithm that achieved the best result is used, which can be determined by analysing the average reward of the learning curve (3.1). The experiment consists out of 100 runs. The metrics used for the evaluation of the experiment is described in the following section.

2.3. Evaluation metrics

Two metrics are used for the evaluation of the experiments. The first metric counts the cells with an equal number of visits using the following pythonic algorithm:

Algorithm 1 Evaluation of Coverage Performance

```
cell visits = [0 for visits in range(0, max(visits))]
for cell in searchfield do
    if cell is not obstacle then:
        cell visits[cell in searchfield(visits)] += 1
```

The list of cell visits is then visually represented and should give evidence about the coverage performance of the trained RL-algorithm. The visualization can be seen in Figure 4 for the coverage performance of a completely trained RL-algorithm where the results for 100 missions are summarized with the help of errorbars. The goal is to avoid multiple visits of cells which shortens the mission time for complete area coverage.

The second metric compares the movement decisions made by the trained RL-algorithm based on the UAVs condition. The evaluation is performed using the following formula:

$$UAV\ Wind\ Load = \begin{cases} Headwind & WD \angle MD = 180° \\ Crosswind & WD \angle MD = \pm 90° \\ Tailwind & WD \angle MD = 0° \end{cases} \tag{4}$$

The case differentiation of loads the UAV experiences based on the wind is determined by calculating the angle between the wind direction (WD) and the direction of movement (MD). The UAV wind load can be linked to the UAV state and can thus be visualised in a bar chart (see Figure 5). By comparing the frequency of movement decisions in connection with the UAV state, it is possible to assess whether the RL learner has learnt to use the UAV swarm as efficiently as possible with a focus on the system-state.

3. Results

The following section presents the results from the experiment described in 2.2. First, the overall training process is pictured. This is followed by the results of the coverage performance, which enable the evaluation of the first sub-task of the RL approach. The results of the second sub-task are presented in the last subsection, showing an evaluation that focuses on the cooperation and degradation of the UAVs within the RL approach.

3.1. Learning performance

The goal of the reinforcement learning process is to increase the average reward successively over the number of training iterations. While a supervised learning approach compares the produced output of a network with labeled data and backpropagates the error, RL does not need labeled data and it is producing training data within the learning process. The reward function helps to choose the actions that lead to the best reward. This is not only considered at every single step within one training episode, but also at the end of one episode to increase the cumulative reward. The reward function used in this paper (described in subsection 2.1.3) should establish a multi-UAV cooperation to perform a complete area path coverage with respect to degradation that results from the individual system usage. The average reward achieved by the RL approach over the training process can be seen in Figure 3.

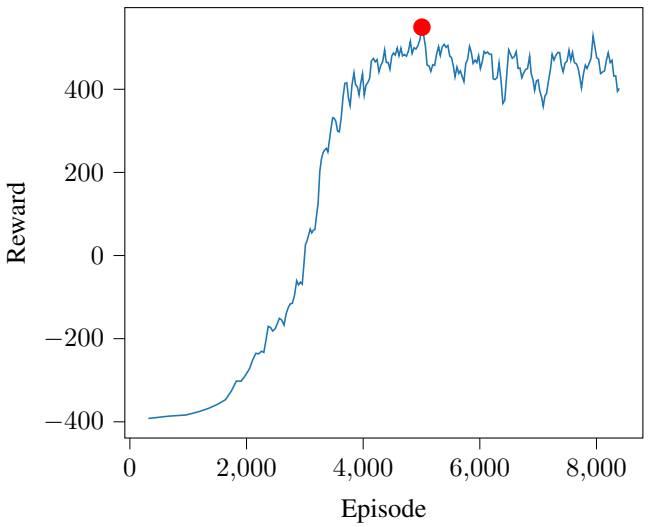

Figure 3. Cumulative training reward achieved by the RL approach over the training process

The training process starts with a high negative reward, which is comprehendable because the movement of the UAV is arbitrary and due to the starting position at the boarder of the search field, the UAV leaves the search field quite often at the very beginning of an episode, resulting in a high penalty and

low gathered reward for covering new fields and moving with usage considerations. The UAV then learns to not directly leave the search field but rather fly in a straight line until it reaches the opposite boarder. The average reward per mission increases slightly, which can be seen at the beginning of the reward curve. Subsequently the UAV learns to move in the right wind direction, paying respect to its own condition. It also learns to change direction at the boarder of the search field, resulting in a much higher average reward. This can be seen from the exponential increase in the reward curve. Afterwards it is harder for the UAV to consider the movement of the other UAVs, still it is able to optimize its movement pattern with respect to wind, information about the rest of the swarm and surrounding map data. The increase in the average rewards achieved per episode decreases again, whereby the reward curve reaches a saturation point. The convergence behavior at the end of the training does show instability, which can be explained by varying coverage and cooperation performance. Nevertheless, it can be concluded from the amount of the reward at the end of the learning process and the consideration of the reward function that the UAVs can achieve the first sub-goal in co-operation, namely searching the search field with slightly varying performance. Using the metrics that are described in 2.3 the coverage performance is discussed in more detail in the next section, as well as the level of cooperation where the developed metrics give more insight about the RL-algorithm performance.

3.2. Coverage performance

The primary goal is the CAPC. Only if the UAV is able to fulfill this kind of mission the cooperation performance with respect to the swarm condition can be compared and evaluated. To evaluate the coverage performance not only the complete coverage is considered but also the effectiveness of the coverage through counting the number of visits per cell. However, because this ideal solution conflicts with a search that takes environmental and systems conditions into account, coverage performance varies slightly at the end of the mission and cells of the search-field are visited more than once. Nevertheless the MARL approach is able to complete cover the search field area 92% of the time. This is not ideal but enough to evaluate the RL-algorithm performance. The result of the cell visit counts in order to evaluate the coverage performance can be seen in Figure 4.

The figure shows the number of cell visits on the x-axis and the frequency of occurrence of cells with the number of visits (from the x-axis) for a completed mission on the y-axis. The RL-algorithm was completely trained according to Figure 3. To get a representative behavior of the trained RL-algorithm 100 missions were performed for evaluation. The display with error bars clearly shows that the coverage performance is in a very good range. This is illustrated by the very low number of missions in which fields with zero visits

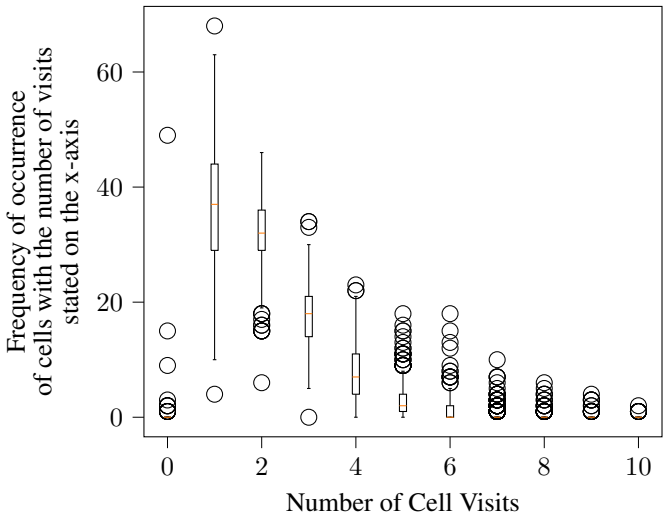

Figure 4. Evaluation of coverage performance by counting the number of fields with the same number of visits

remain at the end of a mission. The fields with zero visits are also categorised exclusively as outliers. A lot of outliers are also visible at fields with a high number of visits, which is also beneficial, because it means that the UAV learns that it should visit a single field as little as possible. This statement is confirmed by the highest value for single field visits. Overall, the distribution of field visits takes the form of a Weibull distribution that is used to describe the frequency of wind speed. Weibull distributions are also often used to describe the lifetime of technical components. Both aspects, namely wind and system lifetime are present within the presented framework and it is remarkable to see that the trained UAV shows such a behavior. Further analyses of the relationship between UAV behaviour and the Weibull distribution are pending.

3.3. Cooperation and degradation evaluation

The secondary goal of the trained RL approach is to coordinate multiple UAVs such that they are utilized according to their condition. This should encourage a usage suitable deployment of the swarm members in order to avoid sudden system breakdowns and increase reliability for the whole mission. To evaluate system usage with respect to environmental conditions, the number of movement decisions depending on the wind direction where the UAV conditions differ at least about 0.5 is counted. The result can be seen in Figure 5.

The barchart shows the decision of the UAVs with bad condition in blue and the decisions of the UAVs with good condition in orange. Only the values for which both UAVs were active are used, as otherwise cooperation is not possible. Furthermore, the values are normalised so that they can be easily compared with each other. It can be seen that the weaker UAV

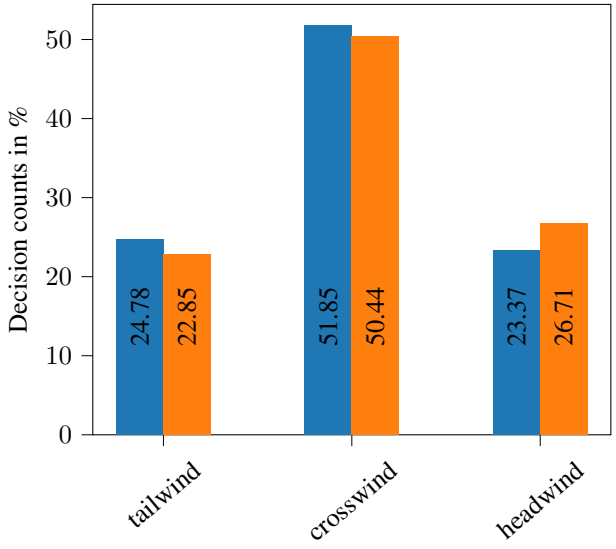

Figure 5. UAV decisions within a cooperative environment

chooses to move more into the direction in which it experiences crosswind. This part equally escapes the movement decisions against and with the wind. On one hand this suits the reward function. On the other hand this leads to less degradation for the weak UAV where it avoids moving against the wind. Choosing the major moving direction crosswind also avoids the UAV to fly against the wind after travelling a long distance with tailwind. Defining the right reward function is sometimes contradicting which gets discussed in the next section.

4. Discussion

During the implementation of the MARL approach, challenges arose with regard to the reward design and the superimposed objectives within the MA mission, which will be discussed below.

4.1. Reward Design

The reward design is very sensitive to minor changes. Also the weighting of the reward significantly changes the behavior of the UAVs. Not all intuitive rules for the reward achieve the desired effect, as the RL-algorithm incorporates the numerical values directly into its learning process. This is also partly dependent on the environment design. As shown in Section 3.3, no reduced degradation can be achieved by flying with the wind, as this inevitably requires flying against the wind from the search field boundary onwards. Another example is a weighted negative reward for multiple cell visits. It could be assumed that if not only a constant negative reward is used for cells visited several times, but the negative reward is multiplied by a factor derived from the number of visits per cell, better UAV behaviour is achieved. However, this is not the case, as the UAV is restricted in its free movement across the search field. Reaching an unvisited field directly would be associated with an increased negative reward. In order to find the right reward policy, the paper used a trial and error approach, so that there is further potential for optimisation at this point. This can also be realised through a different environment design that is connected with the reward assignment.

4.2. Superimposed Objectives

The MARL approach in this paper combines two goals, which creates a conflict between objectives. Both goals can only be achieved if compromises are made with regard to the individual goals. On the one hand, this complicates the reward design that comes into play at the end of a step. On the other hand, it makes evaluation methods more difficult. As this paper is a proof of concept and the assessment of performance is not the main focus, the topic of detailed evaluation should be the subject of further work.

5. Conclusion

This paper presents a MARL approach to solve a CAPC mission under the consideration of dynamic system states and other external factors which places a stress on the deployed systems. The topic dealt with is motivated by the reference to current research topics and specified by analysing the relevant research literature. A generalised methodology is derived that allows state and environment data to be integrated into a MARL approach. This approach allows individual UAVs to communicate with each other and perceive their surroundings as they navigate through the environment. The emphasis lies on designing the reward function, as it serves as the primary driver influencing the behavior of the UAVs, which is intended to utilize the swarm members in a resource-saving manner as an approach for optimisation.

A drone reconnaissance mission is used as a practical example to apply all components of the generalized methodology. The RL-algorithms performance is then evaluated regarding the learning process and the RL-algorithm performance. It can be stated that the completely trained RL-algorithm is able to solve the superimposed objectives of covering the complete area under consideration of the varying system state of the UAVs and a varying wind direction as an external factor. Through the integration of system condition and external loads through wind, the system usage is the main parameter that gets optimized. It turns out that due to the conflicting goals and the associated reward function, the behaviour of the UAVs follows a compromise. While the coverage performance decreases slightly, a more energy-efficient use of the drone swarm can be observed. With that, the methodology is able to recover from sudden system failures and guarantee a more reliable mission fulfilment. This extends existing approaches from current research literature through a highly dynamic in-mission decision process. In addition, a much freer

mission design is made possible by dispensing with segmentation of the search field.

With the promising results of this paper, an in-depth analysis of RL-algorithm performance based on relevant parameters is pending. Such an analysis can provide further insight about the design of the reward function and thus help to design the MA system for a desired behaviour.

NOMENCLATURE

AC	agent condition
$CAPC$	complete-area path-coverage
EC	energy consumption
MA	multi-agent
$MARL$	multi-agent reinforcement-learning
PHM	prognostics and health management
PPO	Proximal Policy Optimization
RL	reinforcment learning
RUL	remaining useful lifetime
UAV	unmanned aerial vehicle

REFERENCES

Alighanbari, M. (2004). *Task assignment algorithms for teams of uavs in dynamic environments* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Andersson, K., Bang, M., Marcus, C., Persson, B., Sturesson, P., Jensen, E., & Hult, G. (2015). Military utility: A proposed concept to support decision-making. *Technology in society*, *43*, 23–32.

Bougacha, O., & Varnier, C. (2020). Enhancing decisions in prognostics and health management framework. *International Journal of prognostics and health management*, *11*(1).

Cho, S.-W., Park, J.-H., Park, H.-J., & Kim, S. (2021). Multi-uav coverage path planning based on hexagonal grid decomposition in maritime search and rescue. *Mathematics*, *10*(1), 83.

Darrah, T., Quiñones-Grueiro, M., Biswas, G., & Kulkarni, C. S. (2021). Prognostics based decision making for safe and optimal uav operations. In *Aiaa scitech 2021 forum* (p. 0394).

Hare, J. (2019). Dealing with sparse rewards in reinforcement learning. *arXiv preprint arXiv:1910.09281*.

Heier, H., Mehringskötter, S., & Preusche, C. (2018). The use of phm for a dynamic reliability assessment. In *2018 ieee aerospace conference* (pp. 1–10).

Kim, N.-H., An, D., & Choi, J.-H. (2017). Prognostics and health management of engineering systems. *Switzerland: Springer International Publishing*.

Kouzehgar, M., Meghjani, M., & Bouffanais, R. (2020). Multi-agent reinforcement learning for dynamic ocean monitoring by a swarm of buoys. In *Global oceans 2020: Singapore–us gulf coast* (pp. 1–8).

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., … Stoica, I. (2018). Rllib: Abstractions for distributed reinforcement learning. In *International conference on machine learning* (pp. 3053–3062).

Mahmoud Zadeh, S., Yazdani, A., Elmi, A., Abbasi, A., & Ghanooni, P. (2022). Exploiting a fleet of uavs for monitoring and data acquisition of a distributed sensor network. *Neural Computing and Applications*, 1–14.

Marques, H., & Giacotto, A. (2019). Prescriptive maintenance: Building alternative plans for smart operations. In *The 10th aerospace technology congress.*

Puente-Castro, A., Rivero, D., Pazos, A., & Fernandez-Blanco, E. (2022). Uav swarm path planning with reinforcement learning for field prospecting. *Applied Intelligence*, *52*(12), 14101–14118.

Radzki, G., Bocewicz, G., Golińska-Dawson, P., Jasiulewicz-Kaczmarek, M., Witczak, M., & Banaszak, Z. (2021). Periodic planning of uavs' fleet mission with the uncertainty of travel parameters. In *2021 ieee international conference on fuzzy systems (fuzz-ieee)* (pp. 1–8).

Stapelberg, R. F. (2009). *Availability and maintainability in engineering design*. Springer.

Theile, M., Bayerlein, H., Nai, R., Gesbert, D., & Caccamo, M. (2020). Uav coverage path planning under varying power constraints using deep reinforcement learning. In *2020 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 1444–1449).

Tumer, I. (2011). System health management: With aerospace applications. In *Chapter* (Vol. 8, pp. 129–142). John Wiley and Sons United Kingdom.

Wang, B. H., Wang, D. B., Ali, Z. A., Ting Ting, B., & Wang, H. (2019). An overview of various kinds of wind effects on unmanned aerial vehicle. *Measurement and Control*, *52*(7-8), 731–739.

Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, *12*(3), 729.

Xiao, J., Wang, G., Zhang, Y., & Cheng, L. (2020). A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning. *IEEE Access*, *8*, 33511–33521.

BIOGRAPHY

Lorenz Dingeldein received his M.Sc. in mechanical and process engineering from Technische Universität Darmstadt, Germany, in 2019. Since then he has been working at the Institute of Flight Systems and Automatic Control (FSR). As a Research Associate, he has been involved in several projects with a strong focus on Prognostics and Health Management. He focuses on the development of condition based asset management of multi agent systems in dynamic environments.