Robotica: (2021) volume 39, pp. 483–502. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000521

ARCog: An Aerial Robotics Cognitive Architecture

Milena F. Pinto†*, Leonardo M. Honório‡, Andre L. M. Marcato‡, Mario A. R. Dantas‡, Aurelio G. Melo‡, Miriam Capretz¶ and Cristina Urdiales§

†*Department of Electronics, Federal Center for Technological Education of Rio de Janeiro (CEFET-RJ), Rio de Janeiro, Brazil*
‡*Department of Electrical Engineering, Federal University of Juiz de Fora (UFJF), Juiz de Fora, Brazil*
¶*Department of Electrical and Computer Engineering, Faculty of Engineering, Western University, London, Canada*
§*Department of Electronics Technology, University of Málaga, Málaga, Spain*

(Accepted May 28, 2020. First published online: July 16, 2020)

SUMMARY
Efficient algorithm integration is a key issue in aerial robotics. However, only a few integration solutions rely on a cognitive approach. Cognitive approaches break down complex problems into independent units that may deal with progressively lower-level data interfaces, all the way down to sensors and actuators. A cognitive architecture defines information flow among units to produce emergent intelligent behavior. Despite the improvements in autonomous decision-making, several key issues remain open. One of these issues is the selection, coordination, and decision-making related to the several specialized tasks required for fulfilling mission objectives. This work addresses decision-making for the cognitive unmanned-aerial-vehicle architecture coined as ARCog. The proposed architecture lays the groundwork for the development of a software platform aligned with the requirements of the state-of-the-art technology in the field. The system is designed to provide high-level decision-making. Experiments prove that ARCog works correctly in its target scenario.

KEYWORDS: Unmanned aerial vehicle; Intelligent systems; Autonomous mission; Decentralized architecture; Decision-making.

1. Introduction

The vast number of applications using unmanned aerial vehicles (UAVs), in recent years, has caused rapid growth in research related to this technology, which is explained by the aircraft's ability to perform cost-effective activities with flexibility and reduced risks to human life. The demand for more complex missions has increased UAV capabilities. Besides, this demand generates platforms with a high degree of autonomy to carry out tasks simultaneously, with less human intervention.[1] The integration of complex algorithms and methods working onboard the aircraft system has developed many highly specialized solutions to specific applications.[2] Solutions for particular tasks are usually limited by environmental understanding where specific algorithms can quickly achieve results. Examples of such a task include the search and rescue (SAR) mission, activity recognition, and monitoring.

Many specialized systems, such as in refs. [3–5], are working on optimal solutions for specific problems. However, a few of those, refs. [1] and [6], are integrated to advance the cognitive field and

* Corresponding author. E-mail: milenafaria@ieee.org

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

solve general issues underlying the aircraft's operation, such as decision-making. Cognitive architecture organizes the way that the data and information of an agent flow, which is important because the structure supports the emergence of intelligent behavior. Cognitive research breaks the problem down and makes assumptions about the existence of low-level data interfaces, such as sensors and actuators. Besides, it should be focused on the information about those and about abstracting the hardware. The critical question comes in designing an architecture that can be used as a foundation where all issues related to cognitive aspects can be represented, built, and solved incrementally. Examples of such problems are knowledge representation, decision-making, and robotics.

Many works and methods have tackled the problem of defining decision-making and knowledge representation from different angles. However, there are remaining improvements needed to advance this field of research. These improvements concern the practical implementation of a goal-oriented architecture, capable of task generation, scheduling, coordination, and decision-making. Thus, this work highlights the importance of addressing the development of practical methods for inserting artificial cognition in UAVs. The focus is on the structural organization and suggested approaches to break down mission goals into tasks and perform decision-making related to these tasks to allow autonomous operation.

This research also introduces a hybrid structure capable of managing multiple aircraft while still allowing deployment and operation with a single UAV. The multiple aircraft coordination is based on a fog computing structure to enable a flexible organization and the use of UAV teams. The architecture is also evaluated in a real-world, unstructured, outdoor scenario to show its feasibility and effectiveness. The remainder of the paper is organized as follows: Section 2 presents a brief review of the related works highlighting state-of-the-art cognitive and aerial architectures beside these research contributions. The ARCog architecture and its foundations are detailed in Section 3. Section 4 shows the simulations and experiments with a proper discussion of the results. The concluding remarks and suggestions for future improvements are conducted in Section 5.

2. Related Work and Main Contributions

Robotic systems require a combination of several specific and interrelated building blocks to reach autonomous operation capabilities. Moreover, a strict organization is essential for interoperation and the emergence of intelligent behavior. However, this integration leads to new challenges, that is, the adaptability for the execution of unique problems and scalability, among others. In this context, cognitive architectures are capable of mitigating those issues by speeding up development. This section intends to capture the current state-of-the-art cognitive systems and their characteristics. Moreover, the related works will show the proposed improvements to include new advances and to make cognitive architectures suitable for UAVs.

2.1. General cognitive aspects

An important aspect when comparing different methodologies is the applied criterion, which may vary according to the application. Some parameters can be used as a common point to evaluate the architectures. In this sense, this subsection will establish those parameters and evaluate this work proposition and the literature background around them. Examples of relevant cognitive aspects are the ability to store and represent knowledge, process natural language, decision-making, problem-solving, and the possibility of real-time and real-world application.

An important feature during the interaction between the system and the operator is the aptitude for natural language representation. There are extensions to add this feature in the general propositions. For example, NL-SOAR[7] is an improvement of SOAR that can comprehend and generate outputs in natural language. This extension can be used in both simulated and real-time environments. Another equivalent improvement was made in the architecture CHREST[8] and employed in the work of Insurralde et al.[9] In Ball et al.,[10] the proposition ACT-R[11] was designed to support natural language. The architecture proposed in this paper uses symbolic knowledge for data exchange among the main blocks. These implementations support symbolic processing. All required nonsymbolic methods such as computer vision are enclosed to result in symbolic outputs. Thus, the reasoning performed within the architecture can be fully understood while analyzing the data flow.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

The data representation aims at allowing decision-making and learning. Thus, the system is able to learn through previous experience. The capability of the system to learn from its experience can further improve its aptitude for responding to future events. Besides, this reduces the work required at the programming stage. Tweedale[12] presents a review of the general methodology employed by SOAR and ACT-R in the decision-making process. This work uses expert knowledge and situational memory combined with rule-based logic to produce reasoning, which is similar to the method presented in Seleckỳ et al.[1] Note that in many of the works analyzed here, there is no description of experiments. More recent works, like Sampedro et al.,[13] have shown some centralized decision-making, although without much generalization of the decision-making process.

2.2. Aerial robotics systems

Several parts of the control problem are well studied and defined in the literature, such as the design of nonlinear position and attitude control.[14] However, there is still space for new methodologies to combine several techniques to reach autonomous operation capabilities. The work of Manière[15] highlighted the importance of developing architectures for the robotics field. Architectures make robotics systems more accessible to design and implement. Besides, they can break down the problem into manageable parts that are simpler to test and integrate. These parts can be split into operational parts, such as high, middle, or low levels.

High-level decisions in SAR applications are related to target detection, recognition,[16,17] and team coordination.[18] However, more general decision-making is not the usual scope in this context. For instance, the presence of a decision-maker enables the operation of the UAV in the most dynamic environments even without central planning or management. The aircraft can respond optimally to any environmental changes. The most used decision-making algorithms are artificial neural networks (ANNs), expert systems (ES), and evolutionary computation (EC). ANN presents the most impressive results. A drawback is that it requires a massive amount of training data. In decision-making, ANN has been applied in several scenarios.

Many studies have applied the case-based reasoning (CBR) algorithm as a decision support system.[19] The reason is because CBR strives to find solutions for new problems based on correspondences of old solutions. This approach uses specific knowledge of previous situations that can be adapted instead of relying only on general knowledge. Thus, this reasoning method is suitable for decision support systems with a complex and dynamically variable domain. This work uses the CBR method for decision-making with the capability of working autonomously in trained tasks, such as in ref.[20]

Several kinds of research have created approaches for building intelligent systems. The Aerostack[6] is one of the most prominent solutions which presents a modular organization for command and control of autonomous, heterogeneous teams. Moreover, this approach provides incremental development and verification. As a drawback, this work is centered on sensorial data, and it does not define a specific model for data storage. Besides, the architecture does not deal with human-in-the-loop (HITL), team coordination, and fully autonomous decision-making.

The proposed IVCA of Insaurralde et al.[9] enables collaboration among multiple air vehicles. The IVCA is generic, but it aims at the study of aircraft working cooperatively. This work is based on ontologies, and there is no specification about hardware. Seleckỳ et al.[1] showed a complete UAV architecture for autonomous operation centered more on sensor data. However, this work does not implement learning ability, and the hardware requirements depend on work extensions.

Based on the presented developments, the multilayer architecture proposed in this research work is comprised of perceptual, structural, and reasoning levels to develop an autonomous system. The architecture works with online data processing aboard an aircraft. The architectural approach of using blocks presents a model for easy interchange between algorithms, by replacing the specific block. The use of robotic operative system (ROS) also allows for Flight Control Unit (FCU) hardware interdependence. Table I presents the architectural differences among all cited aerial architectures as well as the highlighted advantages of the proposed architecture.

2.3. Main contributions

Cognition is a very active topic across multiple research fields, such as linguistics, psychology, artificial intelligence, and robotics. Recent advances in the cognitive science of different fields

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

Table I. Structural differences among the main cited architectures.

	Features			
Main architectures	**Practical showing architecture operation**	**Multi-agent support**	**It provides insight into learning from human decisions**	**User decision support**
ARCog	Yes	Yes, it applies fog to coordinate the agents	Yes, it can learn the user's decisions	Yes, thought CAD
FFIDUAS [1]	Yes	Yes, it is centralized on the cloud	No, does not define this feature	No, does not define this feature
IVCA [9]	No	Yes; however, does not define the control structure	No, does not define this feature	No, does not define this feature
STRL [21]	Yes	No, only a single agent design is presented	No, does not define this feature	No, does not define this feature
Proactive [22]	No	Yes, it is centralized on the Cloud	No, does not define this feature	No, does not define this feature
Aerostack [6]	Yes	Partially, as it does not define explicit coordination among the agents	No, does not define this feature	No, does not define this feature

suggest that human cognition is the result of the hierarchical structures interacting sequentially to produce increasing complexity.[23,24] In practical approaches, cognition can be combined with the latest advances in computer vision, fog-computing, and HITL, in order to improve results.

An essential contribution of this work, coined as ARCog, is the definition of the components of an aerial robotic cognitive architecture. The proposed architecture organization lays the groundwork for the development of a software platform aligned with the current requirements of state-of-the-art technology, such as semiautonomous navigation capability, environment perception, and decision. The focus on cognition means that the system is designed to be capable of high-level decision-making, such as understanding, interpretation, and goal-oriented reasoning. This research work contribution can be summarized as follows:

- Development of a differentiated architecture to support decision-making and the emergence of intelligent behavior for UAVs in a hybrid structure that can be used individually or in cooperation, organized through a fog-cloud approach.
- Development of a system that operates in a decentralized fashion with embedded decision capability, centralized coordination, and teamwork.
- An implementation proposition for key features, for example, image recognition, scene comprehension, and decision-making, to enable autonomous decision-making along with experiments to show the architecture's capability.

3. The ARCog Architecture

This section presents the general aspects of the proposed multilayer architecture for semiautonomous aerial vehicles, called ARCog. The main objective is to build an architecture oriented by goals and tasks that should be capable of easy adaptation in different scenarios and situations. The architecture should be capable of learning about the operational environment instead of being pre-programmed every time a new situation arises.

Learning ability and cognition encoding in a real environment is a complex task due to many different variables and constraints. This requires an organized approach to make this process effective. The organization proposed in this research is based on Bloom's Taxonomy Approach of Learning Domains,[15,25] where the architecture layers are hierarchically arranged from a low level of perception to a high level of cognition. This approach makes it easier to define goals, to design modularly,

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

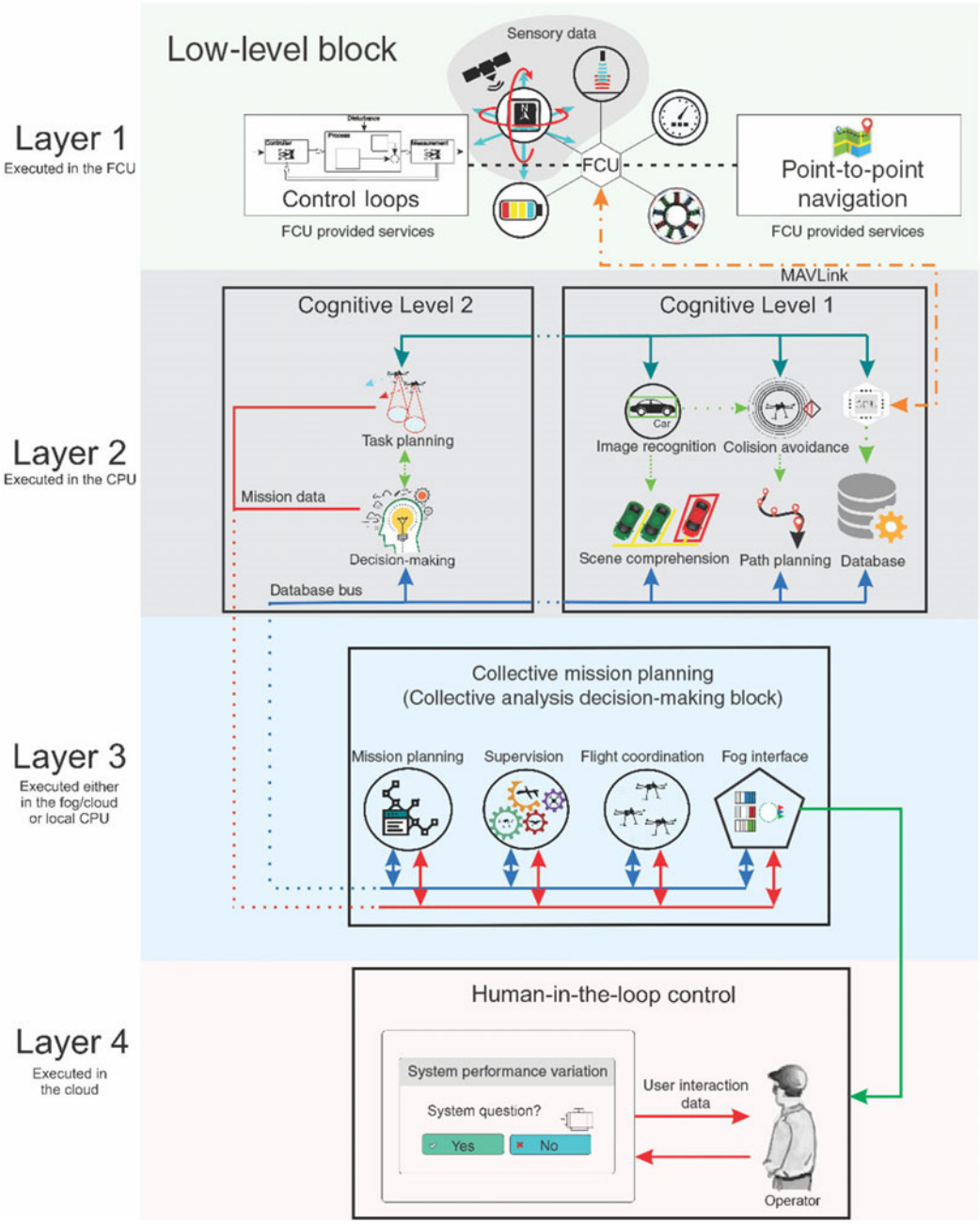


Fig. 1. Logic diagram of the ARCog.

and to tackle problems related to specific blocks. The processing of the data produces information and knowledge at increasing levels of complexity.

This research also addresses a solution for limited computational capability aboard the aircraft by the employment of a fog-cloud computing approach. The architecture is designed to support the deployment of fog nodes to execute algorithms locally, close to the UAVs, reducing the computational workload on them and optimizing the required bandwidth.

The ARCog organization is a central point for achieving the objectives of improving understanding and boosting performance. Figure 1 exhibits the logic diagram. The data flow vertically among layers while the interlayer communication occurs in a hybrid fashion, where some data can also flow horizontally.

The low-level block, presented in Layer 1, is composed of the hardware interface (e.g., IMU, GPS, and ultrasonic sensor) and primary control loops to keep the UAV guided with a stable flight.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

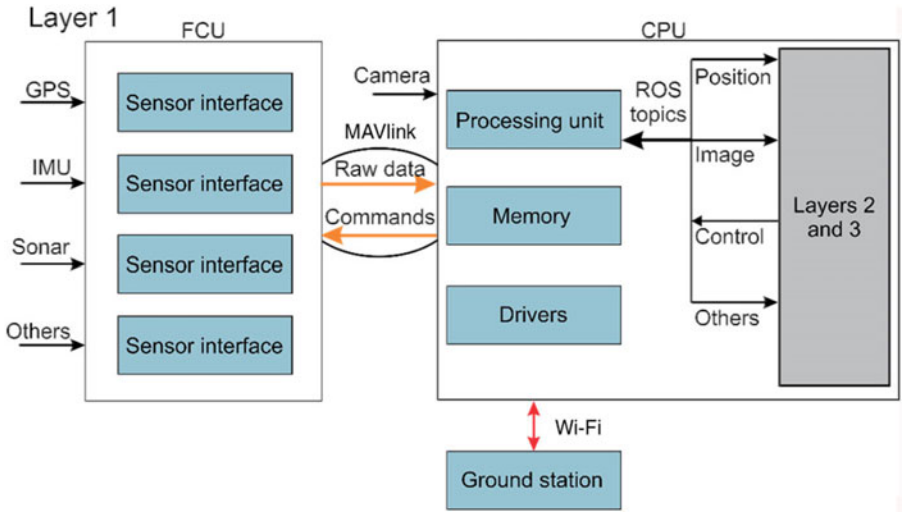


Fig. 2. Low-level block and the relationship with ARCog Layers 2 and 3 blocks.

Besides, it also contains data acquisition to the next levels. This level should be seen as a hardware abstraction layer. Note that there is only one hard requirement, the existence of a proper interface for Cognitive Level 1 on Layer 2. The second block (i.e., Cognitive Level 1) performs image recognition, data storage, scene comprehension, and basic heuristics (e.g., path planning). The autonomous decision-making and individual mission planning should appear at Cognitive Level 2, which uses the information generated at the previous layers for high-level decision-making. At the end of this process, the collective mission planning should be capable of collective analysis, decision-making among all aircraft, strategic plans, comprehension of the system goals, and performing of social skills, that is, dialogue between UAV and the human operator. This interaction is given by the HITL inserted through a human–machine interface (HMI). The architecture can send data to an external operator, presented in Layer 4, for additional processing. Moreover, the ARCog can receive the necessary information through the human agent for changing its mission or the analysis of objects that do not have an appropriate recognition.

The architecture executes multiple processes in a multitasking operating system through ROS communication methods. The ROS was chosen as a meta-operating system since it is an open-source platform and has a collection of software libraries and tools. This operating system assists in robot application development and promotes code reuse. The mechanisms provided by ROS are the publish–subscribe schemes and request–replay services. The multitask operation is essential to execute higher and lower processes with different frequencies. For example, the path planning algorithm[26] demands real-time execution and more computational cost, while other blocks, such as supervision, can be executed with soft real-time requirements.

3.1. Hardware abstraction layer

This layer is the closest to the hardware, and it is equipped with basic features to acquire the required data to feed the other layers. This layer should allow control, based on high-level decisions without human intervention. Note that most of the current UAVs generation already contains powerful microcontrollers capable of reading different sensors. Moreover, they can control flight parameters to maintain a stable flight and to perform point-to-point navigation[27] from USB or serial interfaces. These features are particularly useful because they abstract the hardware interface, enabling the focus on the cognitive aspects of the system. Figure 2 details the model organization.

The FCU is one of the central components. The inputs of the block are sensor data, and the FCU contains all described basic features. Other resources may be required by specific applications and should be provisioned accordingly. For example, another essential component is the Control Processing Unit (CPU). The CPU gathers the data from additional sensors (e.g., camera) required by the upper cognitive levels. This should have a required power processing to execute the algorithms and enough power to allow the onboard application, the CPU, to exchange data with the FCU through

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

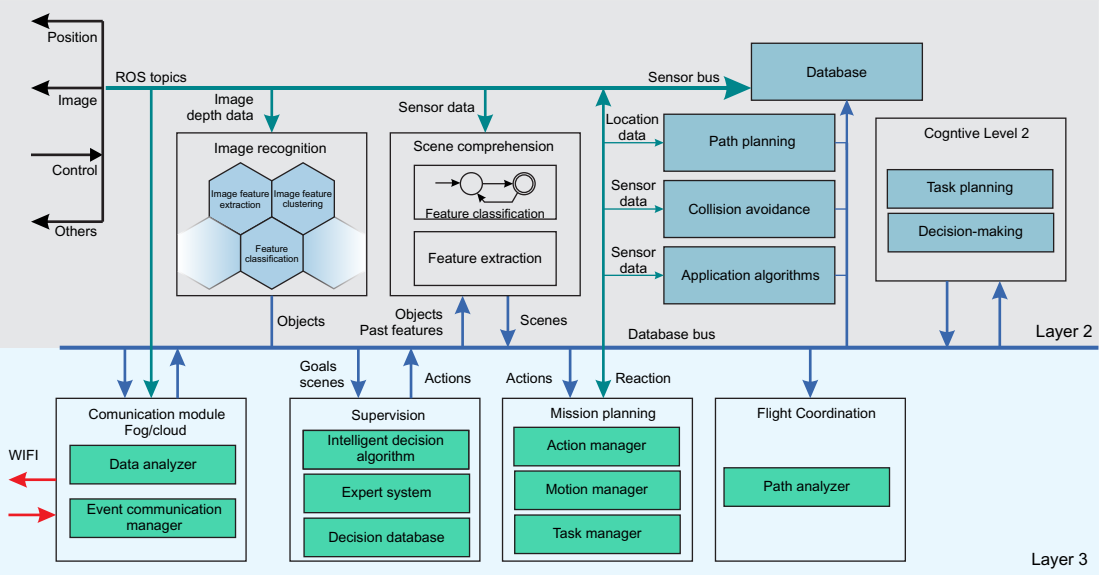


Fig. 3. General description of the layers and their relationships with the ARCog.

the MAVlink protocol. The data exchange utilizes ROS interfaces. This interface is composed of named information buses called topics.

The CPU is also responsible for the interface between aircraft (air to air, A2A) and between aircraft and the ground station (GS; air to ground, A2G). The CPU captures data from sensors that are not supported by the FCU (e.g., images). Besides, it executes all codes related to the higher levels and provides the storage required by the architecture.

3.2. Cognitive layers

The previous layer has detailed the necessary services at the FCU, sensory inputs, and the interface between the abstraction of the hardware layer and the cognitive structures. The cognitive structure of Layers 2 and 3 has been designed based on Bloom's Taxonomy of Learning Domains to produce a hierarchical structure related to the task generation, scheduling, execution, and supervision.[28] This process results in a formation centered on the cognitive process. Bloom's Taxonomy has been successfully applied in many educational fields to structure courses, learning goals, and activities.[29] This architecture provides an insight into the building of human knowledge.

For this reason, the taxonomy can also be used to organize the architecture's blocks, where the knowledge is created and improved from the input sensory data up to the complex thinking of goals and decisions. The cognition is strictly linked with the capability of perception, reasoning, and action. The proposed structure sits right over the sensory-motor blocks of the UAV (low-level block). The architecture is organized in such a way that, at each layer, the data are processed to produce information and knowledge at an increasing level of complexity. Figure 3 details the layers and the relationship with the rest of the ARCog. The inputs of the blocks come from sensor data published by the MAVros.

These layers have the desired performance, behaviors, and goals that should be used to generate tasks. Layer 1 contains methods that are more related to data processing, such as path planning, object recognition, and scene comprehension. Other applications may require different features. For example, marker-based localization and servo control could be appropriately accommodated at this layer. This block also presents an image recognition algorithm to extract information from the environment. Other methods represented in this figure are individual mission planning and collision avoidance,[30] which were positioned to symbolize common algorithms related to the system operation.

An important concept, applied in this work, is the scene. A scene is a data set containing the content present in the UAV camera visual field, such as objects and actors and their positions, along with all other available sensory input. This is a loose definition, and other works may make use of different features.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

The scene comprehension in Fig. 3 extracts essential relationships from the scene data, analyzing the information from sensors and outputting the relationships between them. For example, the system can obtain the changes in the position between object and person to determine if a given person is holding an object or if there is an object exchange. Several methods can be applied to fulfill this function. Despite the efficiency and easy programming, the nonsymbolic method neural network is not appropriate for the specific application of this work. This issue was previously discussed in Section 2. Nonsymbolic methods do not allow appropriate evaluation of the logic used to extract essential data. Thus, grammar-based methods have a substantial advantage concerning other propositions once their outputs are correlated to the human experience. This research uses an automaton for building scene comprehension. The main advantage of using this kind of method for aerial application is the explicit organization of the pieces of information, and it can remove the redundancies to extract the relationship for high-level analysis. However, a few other methods could be applied with some degree of equivalency. Cognition Level 2, in Fig. 1, that is, Reasoning Block, takes the information from the scene comprehension block to produce a partial view of the UAV world. This incomplete scene is compared to a database of cases for decision-making and task planning. The system builds the case model based on the background knowledge and the previously collected cases. The authors in ref. [20] reported a detailed explanation. The usual methods for autonomous decision-making are related to systems limited to extensive training over a big set of data, which means that current techniques are very far from presenting general intelligence to operate a broader range of topics.[31] In this work, the CBR is applied, based on the built scene, through sensory data, image recognition, automata organization, and by the last-taken decisions; that is, this layer executes the CBR algorithm each time a new completed scene is saved in the database. The main CBR's functionalities are to evaluate the scene and to make decisions about the scene objects and actors related to the mission objectives.

The first cases used for decision-making are fed into the system in a learning phase. However, they can be adjusted online through the inputs of the supervision block in Fig. 1. In this architecture, local decision-making does not learn or output direct actions. Instead, the decision system learns the broader decisions, and the resultant chain of operations has to be further evaluated; that is, CBR's decisions are later processed by task planners to transform these pieces of information into proper inputs to other algorithms. For example, if the UAV decides to follow an object, the motion planning must determine the object position to start the path planning for fulfilling the new decision. This process simplifies the implementation of the CBR. Each case in the CBR database contains an action that will be outputted if the situation is similar enough. The initial case base should be included at the development stage. They contain basic actions. Examples of such activities are takeoff and landing when requested by the supervision, and returning to home when the target is found, among others. The learning happens when the UAV receives the inputs from the user. Here, the learning can occur in a supervisory fashion, requiring the human operator to specify any behavior that the architecture should learn to avoid the UAV learning undesired behaviors.

Cognitive Level 3, that is, the Collective Cognition Decision Making Block, is responsible for the generation of tasks made by the mission planning team organization and individual tasks attribution performed by the flight coordination block. Moreover, this layer monitors the tasks' execution and interacts with the HITL for supervision. The reason that this layer is in a separated block is that the functionalities of this layer are usually encoded in a centralized system or a fog computing architecture. As the architecture can be used for an individual or multiple UAVs in a hybrid way, the fog structure has the ability to support features for cooperative work. The mission planning has the duty of ensuring a proper share of mission goals among the available aircraft. Flight coordination has to assure that different UAVs' paths do not overlap with each other in any location in order to guarantee flight safety.

The autonomous decision-making enables the UAV to respond to the environmental changes and tasks required during the mission execution. In the Cognitive Level 3 block, the task planning determines the flight parameters for executing the tasks and other related activities performed by the aircraft during its flight. The decision-making should take decisions related to path planning, interventions, and actions with respect to environmental changes. Besides, it should also be capable of improving the decisions during the flight. Note that these proposed blocks can be implemented with different algorithms and methods. The methods' choice can profoundly impact the performance of the embedded application.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

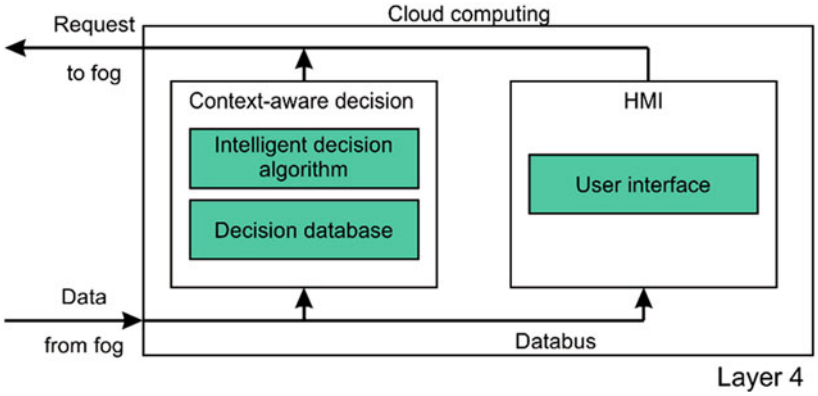


Fig. 4. Details of the HITL control.

3.3. HITL layer

Activities with long-term data storage and human supervision are the responsibility of the last layer in the cloud. The human operator accesses the information by an HMI unit, and the other one, the context-aware decision-making (CAD) unit, adapts the various aircraft information to ease the operator's action. Every time the operator intervenes, the HMI stores that particular action along with the data presented in the unit. Since the next time the same set of aircraft data is received, the architecture can prompt the operator with actions to be taken or even start them automatically.

In the case where the UAV did not respond for a given amount of time, the CAD can decide to alert the operator or take another action based on a database of decisions, such as action plans. In the same example, the CAD could help the operator by presenting an action plan for this particular task, for example, sending a command to the unresponsive UAV to return to the initial place.[32] Figure 4 details the HITL layer.

This level of automation requires methods that rely on direct programming. For instance, the heuristics would be complicated to design and to encode. Thus, this research proposes the CBR method along with a learning methodology. This learning happens every time that the human intervenes in the HMI in a different situation. Then, the CBR stores this new case in the database. However, if the CAD finds a low similarity conflicting case, it prompts the user with a question about a proper expected action to update the database. For example, if a target is found outside of a search perimeter, the similarity may be lower concerning other cases in the database. In this situation, the user may decide to intervene and to track the target to get closer. This intervention will trigger a learning event. As a result, the case base will be updated. As can be noticed, the CBR inside the CAD block deals with the data from several UAVs (e.g., sensory data, detected actors, objects, and actions) and past decisions. Figure 5 shows the data hierarchy in the proposed implementation.

The similarity of the incoming cases needs to be defined, which means that there is a necessity to measure how the incoming cases are compared against the database. This process assumes that the cases do not contain any grammatical errors like a misspelling, which is essential because this simplifies the assumptions necessary when building the similarity function. Equation (1) calculates the similarity for the CAD. The parameter $\phi(s_d, s_c)$ is the similarity between a symbolic value in the test case s_d and the reference case in the database s_c. This parameter is responsible for retrieving the information in a determined table. The variable r_c is a real value in the test scene, and r_d is the value in the reference case. The quotient r_c/r_d measures the similarity between them. The similarity between the test case and the database is $sim(c, d)$, the $\omega_o \subset (0, 1)$ is the weight of an object type in the CRB field, and the normalized value of the similarity $sim_N(c, d)$ is assumed as in Eq. (1).

$$sim_N(c, d) = \frac{\left(\sum \phi(s_d, s_c) \cdot \omega_o + \sum \frac{(r_c \cdot \omega_o)}{r_d}\right)}{max(sim(c, d))} \tag{1}$$

Once the similarity is calculated for the whole database, the two most similar cases are recovered, that is, $sim_{N1}(c, d)$ and $sim_{N2}(c, d)$. After that, if the difference in similarity between the recovered cases is lower than a reference threshold τ_s, that is, $sim_{N1}(c, d) - sim_{N2}(c, d) < \tau_s$, the outcome with lower implications in the current UAV plan is chosen.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

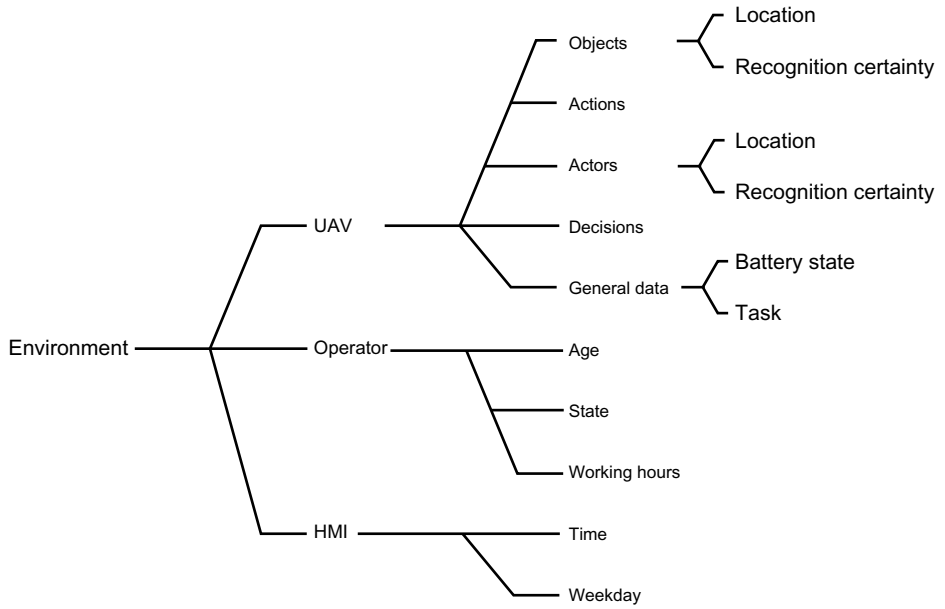


Fig. 5. CAD data hierarchy.

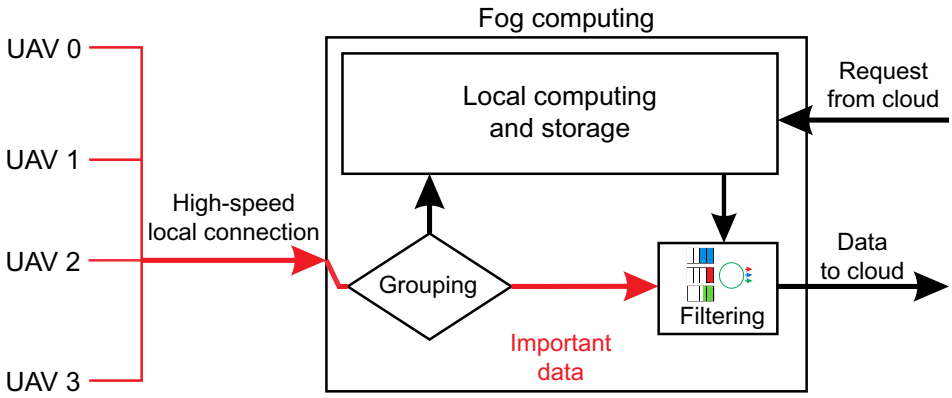


Fig. 6. Fog structure for the ARCog.

3.4. Fog-cloud computing

The required bandwidth and the data redundancy with several UAVs may be impractical for data transmission. Some applications are incredibly susceptible to delays.[33] For example, SAR is generally executed at remote locations with low communication resources. Therefore, this research work proposes the application of a fog block to overcome throughput and latency limitations as shown in Fig. 6. This figure illustrates the data exchange between fog and cloud computing. This research considers that a UAV assigned as a head coordinator is in charge of the fog-cloud computational offloading. The head coordinator analyzes the incoming data and chooses to process locally or to send it to the cloud. Information sent to the cloud will be grouped into chunks before being transmitted to the cloud. This work also considers that each UAV has the embedded architecture proposed by this work. More details are provided by the authors in ref. [34].

4. Environment and Experimental Results

This section describes a few experiments as a representative example of ARCog. The output environment is based on a SAR mission with two decision scenarios. In the first one, the UAV faces various scenes, and different decisions are expected to evaluate decision capability and accuracy. In the second scenario, the decision-making block onboard the UAV is tested when it encounters a target. The main objectives of the missions can be summarized as follows:

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

- Testing the capability of self-localization while performing the mapping of the target position in a predetermined area.
- Identifying targets and informing the human operator.
- Determining the final point of the mission when a predetermined goal is found.

In the experiment, only a selection of key blocks is essential to a single UAV fully automated operation. These blocks are described along with their suggested algorithm. For image recognition, the architecture uses the bag-of-words (BOW) method.[35] This method extracts descriptors from the incoming image, and a trained classifier (i.e., Support Vector Machine) verifies the descriptors' correspondence concerning known objects. If a given object is detected, the symbol that represents the object and its recognition certainty are sent to the scene comprehension.

The scene comprehension encodes the sensors' pieces of information into symbolic information, along with the objects and their interactions. This output will be sent to the CBR inside the UAV. The CBR will make decisions based on the UAV task. For example, if the mission has to find a given object, and the object is spotted, the CBR can decide whether to finish the mission.

The fog node has not been implemented in these experiments. In this way, the UAV decisions and other scene data will be sent directly to the GS. In the GS, the CAD will perform user assistance and supervision. In this scenario, the decisions happen in two steps over the UAV's data. The first step is to evaluate the case decisions against the information of all grouped UAV's data, for example, all actors detected by all UAVs. This process allows the CBR to assess decisions that happen due to the collective result of the system operation, for example, the mission ends when the system finds a given number of subjects. In the second step, the system evaluates each UAV data individually, which helps to take actions in cases related to a single aircraft, for example, sending a second UAV to an area if a given aircraft stops responding for a specified period. This block also uses the human decisions to learn through the actions taken by the operator's interaction with the HMI.

4.1. Description

Figure 7 shows the UAV used in the experiments. The architecture was tested using an onboard computer equipped with an Intel atom z8785G with four cores and 2GB of RAM. This equipment was chosen due to its low power consumption and fanless operation capability. The centralized computer is the i7 7700hq with 8GB of RAM. The UAV has a Logitech full HD camera, a Pixhawk 2.4.8 32-bit flight controller with an ArduPilot firmware and NEO-8M GPS. In case the system is not capable of processing high-resolution images, the downscaling algorithm improves sharpness and contrast, reducing images up to 720p resolution. The image resolution reduction allows the processing of at least five frames per second. This processing rate is the originating event that drives another process, such as feature extraction and scene comprehension. Others, such path planning, happen much faster once the data from FCU are pushed through the system at 50 Hz.

The architecture was built over the Ubuntu platform using ROS Kinect as a base for the development. The communication among the UAVs and the GS was made over Wi-Fi to simplify the implementation and development. However, this technology can be replaced by other methods to meet the requirements of specific applications.

Figure 8 highlights the inter-block communication in the experiment among some of the associated ARCog blocks to exemplify architecture operation. This figure does not contain every interaction due to the complex nature of the data flow. Instead, it only presents key interactions for illustrative purposes. This communication segment shows the mission starts from the HMI and the corresponding events subjected to the detection triggers in the decision-making process. It is possible to observe that during mission execution, several blocks are in parallel communication, such as the supervision and sensor data processing. For example, in the diagram, the first action of the supervision is to acknowledge path change. In sequence, the second action regards the processing of UAV findings. Then, the supervision block forwards this information to the operator. Note that some parts of the architecture are event-driven; that is, these parts are only enabled to work when a trigger is activated, which is the case for decision-making. Once a new scene is available, the decision-making evaluates and takes a decision. However, other functionalities could be dynamical; for example, path planning may have a fast response due to environmental changes.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press



Fig. 7. UAV used in the experiments.

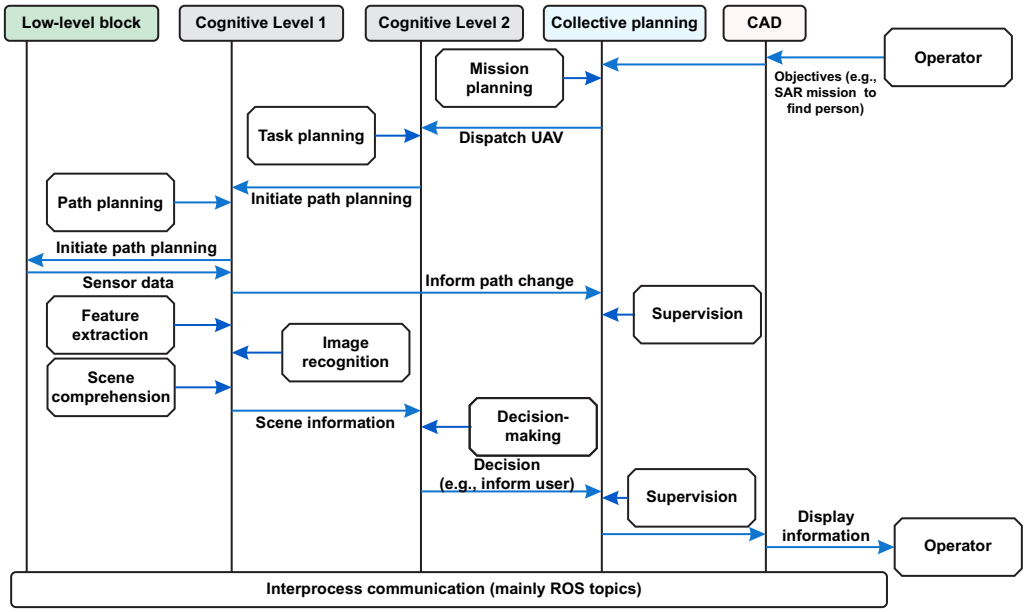


Fig. 8. Inter-block communication among the ARCog.

In SAR, even a few meters of error allow rescue teams to localize the target. Thus, the object localization does not require high accuracy. The object localization is then performed based on the UAV GPS position and the object position in the image. A few tests were conducted to evaluate the HMI. The proposed scenarios were designed to assess the system capacity for responding in different conditions. The experiments were performed with the UAV flying in a predefined trajectory at the height of approximately 6–10 meters. Figure 9 illustrates the mission execution and path followed by the aircraft during the implementation of the mission.

Table II presents the expected system actions in seven scenarios. The data consist of the video recorded by the UAV along with the sensors' measurements. The HMI responses in these determined scenarios are available on https://github.com/ARCog/ARCog/wiki. All codes and interfaces were developed in a Linux-compatible environment, aiming at an easing integration with ROS. The rosbag containing the data from the UAV used in the video is available at the previous link.

4.2. Results and discussion

Figure 10 details the HMI screen during scene 4 with the following presented information: (1) CBR analysis; (2) UAV flight details; and (3) recorded videos and images. In this figure, the HMI displays an image of scene 4 where the target is recognized. Besides, the decision-making output informs the

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

Table II. Scenes for the experiments.

Scenes	Car	People	Expected action
1	Present	Present	Inform
2,3	Present	Absent	Register
4,5	Absent	Present	Inform
6	Absent	Absent	None
7	Absent	Present	Land

Table III. CAD decision accuracy for each event.

Class		Decision accuracy (%)
Expected action: Inform (scenes 1, 4, 5)	Positive	90
	Negative	95
Expected action: Register (scenes 2, 3)	Positive	85
	Negative	95
Expected action: None (scene 6)	Positive	86
	Negative	96

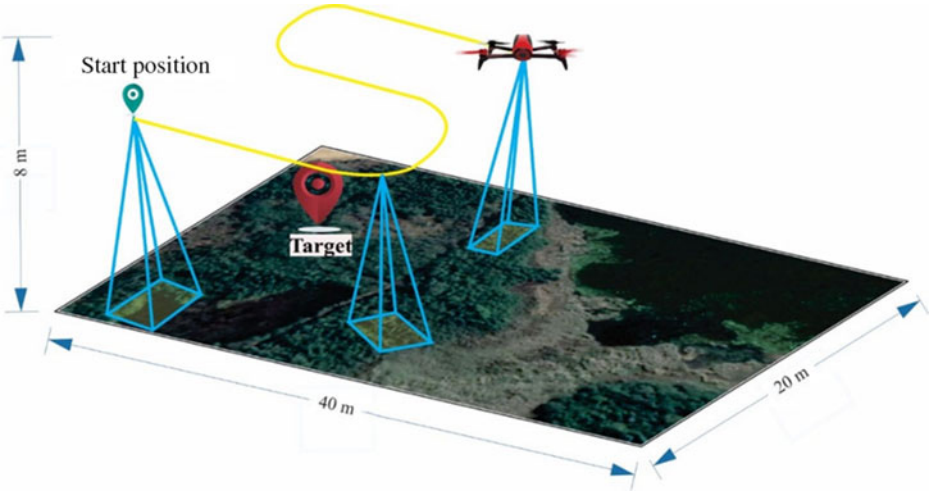


Fig. 9. Mission execution and trajectory followed by the aerial robotic system in the entire SAR mission.

GS of the identified object. The interface also assists the operator in visualizing the necessary data from the UAV and in receiving feedback about the system detection and new cases.

The image recognition is the unit responsible for determining what is present in a given scene. Thus, this algorithm performance has a significant influence on the operation's success. However, despite its importance, this work does not propose improvements to it, and the results related to the recognition algorithm should be seen as guidance for the general system performance. For each video, a segment comprised of a few seconds was selected. This sequence contains data before the UAV finding, up to when it leaves the target. In this data segment, high-level decision-making was analyzed against the expected results. A positive value (i.e., true positive) in the decision is when the target is present and detected. However, when the target is not present, and its presence is not recognized, the event is negative (i.e., true negative). Table III shows this result.

Another strong aspect of this kind of mission is the target location. Therefore, the UAV should record the positions where the target is recognized to ensure adequate planning for the ground rescue team when reaching the target. These next results were designed to show this behavior. Figure 11 exhibits the mission execution representation and the trajectory followed by the aerial vehicle.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

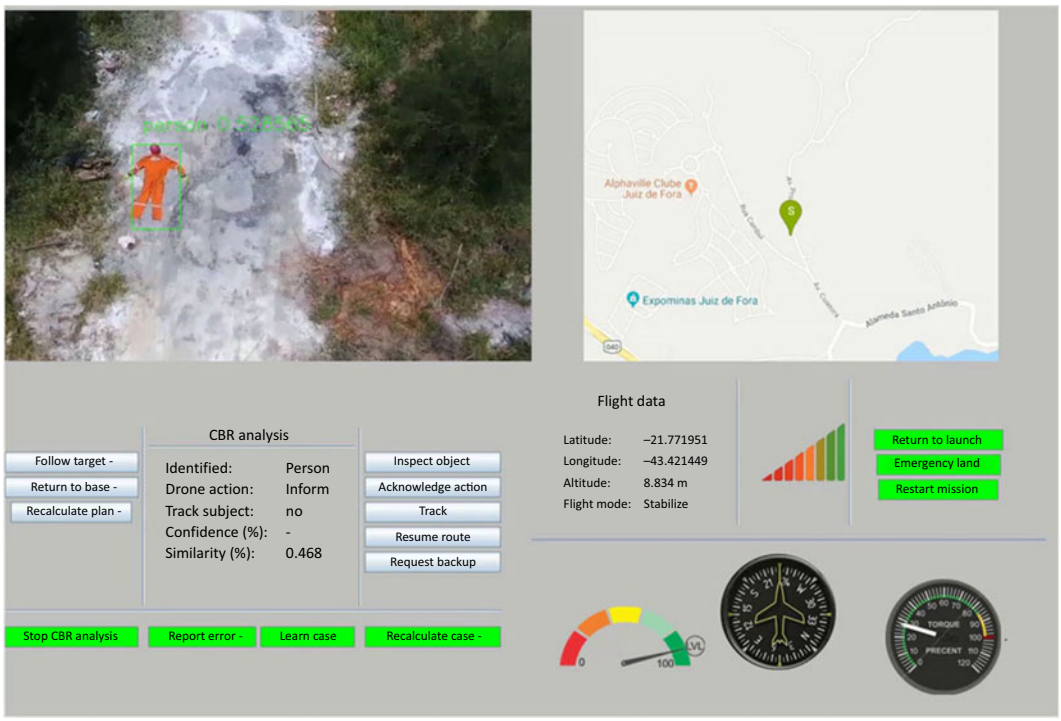


Fig. 10. HMI interface.

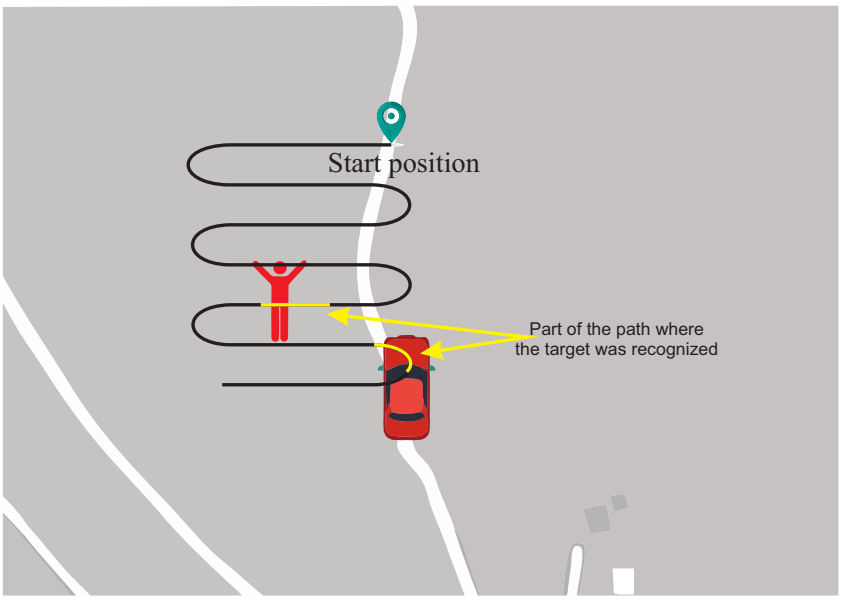


Fig. 11. Mission execution representation and trajectory followed by the aerial vehicle in the SAR mission.

Figure 12 presents the real data from scene 5, where the blue line is the UAV odometry during the flight, and the red represents the part of the path where the target was recognized.

In some missions, the UAV could be in charge of delivering equipment to the target, such as the first aid kit. To show this action, a sequence of cases with the expected behavior was inserted in the decision-making. Firstly, if the target was recognized with low certainty, the UAV had to change path planning and get closer to the goal to increase it. If the target was still identified and had higher recognition confidence, the UAV had to change path planning and land somewhere close to the target. Figure 13 shows this behavior and the key points where the decisions were taken. After landing, the UAV has to take decisions to take off and resume the mission.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

Table IV. Average software performance during a mission.

Parameter	**Value**
CPU Usage (%)	95
RAM Usage (%)	85
Mean number of ROS topics	31
Mean number of messages per second	136
Number of frames per second	2

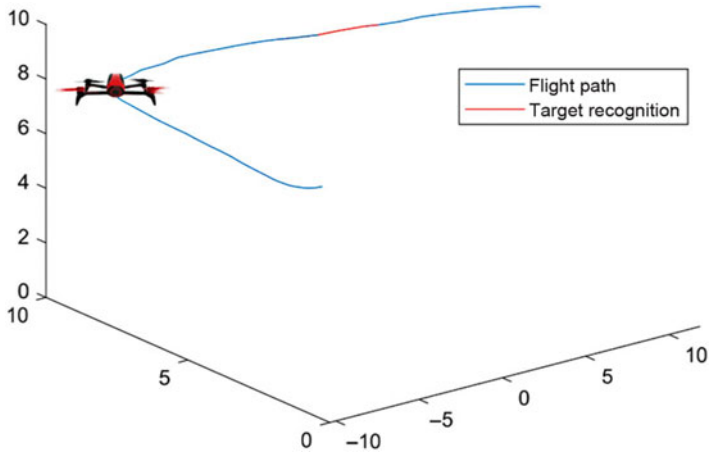


Fig. 12. Part of the real trajectory executed by the aerial vehicle in the SAR mission.

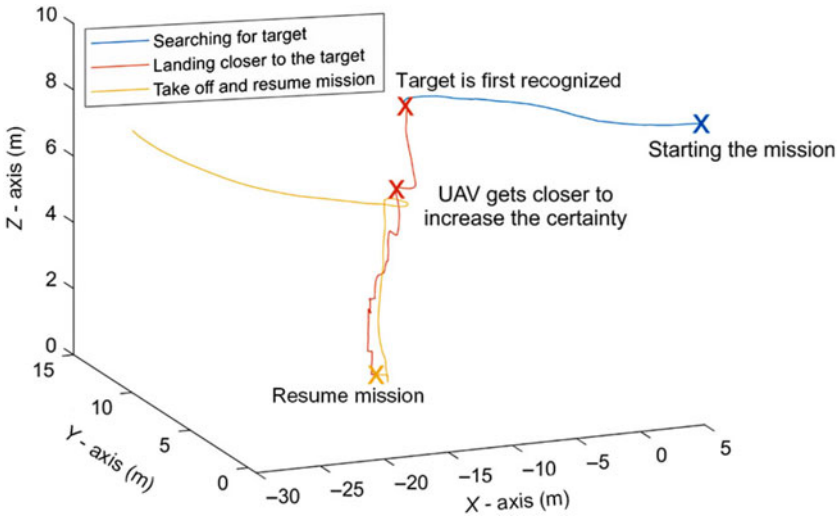


Fig. 13. Part of the real trajectory executed by the aerial vehicle in the SAR mission.

To allow for a better understanding of the behavior described in Fig. 12, a sequence of events was compiled showing key points. Figure 14 illustrates these events. When the recognition certainty, when the target was first recognized, was lower than 75%, the UAV reduced its flight height and finally landed close to the target.

A few parameters of the software performance can be extracted during the mission, such as the ones in Table IV. They allow the understanding of the minimal processing power required by the onboard part of the ARCog. These values show that hardware improvements could benefit the system.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

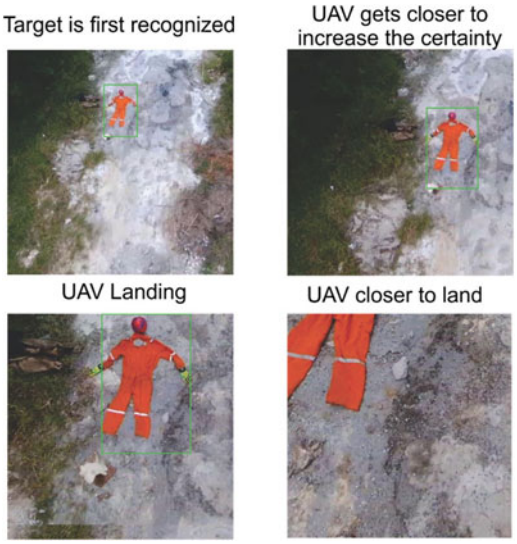


Fig. 14. Part of the real trajectory executed by the aerial vehicle in the SAR mission.

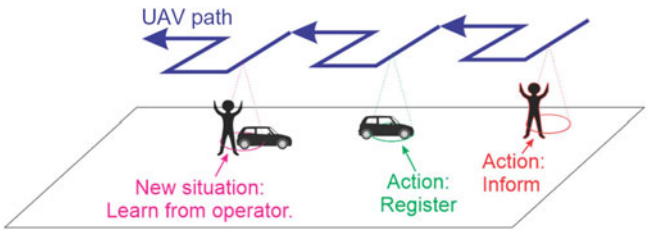


Fig. 15. UAV behavior and training.

4.2.1. Decision-making learning. The learning process of decision-making starts with an offline training stage. In this stage, a trained operator classifies the information and selects the proper decision for the UAV for each input scene in real time. The operator will take these decisions based on the image feedback from the UAV and its sensors, deciding upon the expected actions. This training will provide the basic operation framework for the UAV. Figure 15 illustrates a scene with a set of different behaviors. In the first two scenes, the UAV has a trained case in the database to use. In the last scene, the architecture can adapt the action using the similarity measurement or learn the expected outcome with the human operator interaction with the HMI.

After training in the new situation, the UAV keeps the previous experience to deal with future similar situations. The performance obtained in the experiments has shown good results with the UAV actions, and the mission outcomes followed the expectation. Every time that the UAV faces a new type of mission, the adaptation for previously known cases may result in an inadequate response. However, the performance of the expected behaviors improves very fast as the system learns. Figure 16 shows the behavior of the learning process. In Fig. 16(a), the UAV is in a determined path, and a target enters its Field Of View. The expected outcome would be to track. Note that this case appears for the first time. So, the operator has to interact and request tracking. In Fig. 16(b), the UAV starts tracking immediately upon target recognition.

4.2.2. Multi-agent support. A simulated experiment was designed to evaluate multi-agent support. In that experiment, a multi-vehicle simulation with a Gazebo software was set up. In this simulation, a server runs two UAVs instances, and two other virtual machines are used to execute the onboard control instances as represented in Fig. 17. This intends to allow a more scalable layout and avoid computational bottlenecks.

The experiment is designed for a SAR mission; that is, the UAVs have to find a potential target. The Gazebo simulation environment consists of an enormous scene, where a single target was randomly spawned to accomplish this task. After this, the user defines the search space and starts the mission.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

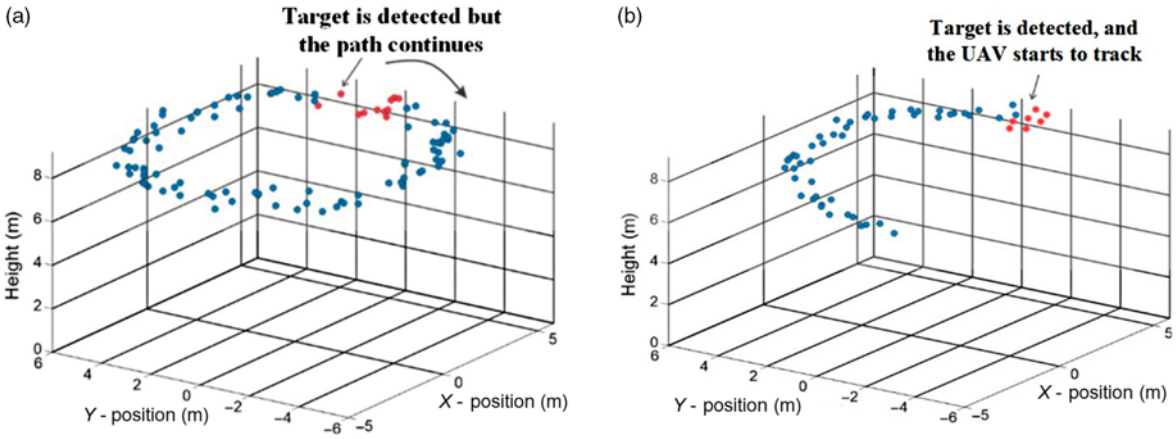


Fig. 16. Learning process for the tracking action. (a) UAV detects the target but continues the path. (b) Target is detected but the UAV starts to track it.

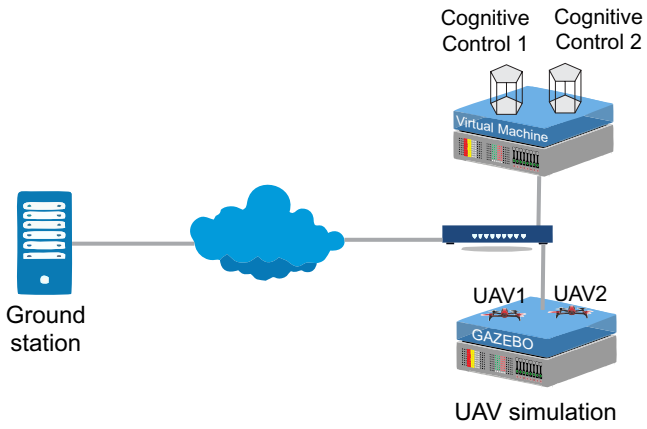


Fig. 17. Simulation layout.

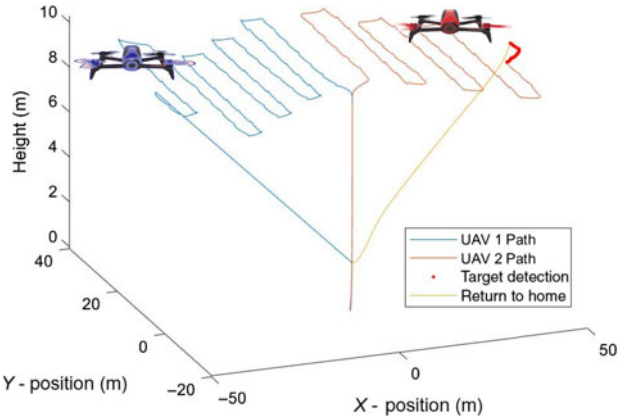


Fig. 18. Multiple UAVs trajectory.

The mission stops when the UAV finds a target. Then, both aircraft have to return to the home point. Figure 18 presents the both UAVs' path. The red color illustrates that the target is recognized. At this point, the system requests both UAVs to return home and to end the mission, as shown in Fig. 19. Those figures only have shown the potential for the use of UAV in scenarios with multiple vehicles, and further experiments in the real-world scenario have to be performed to validate the architecture fully.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

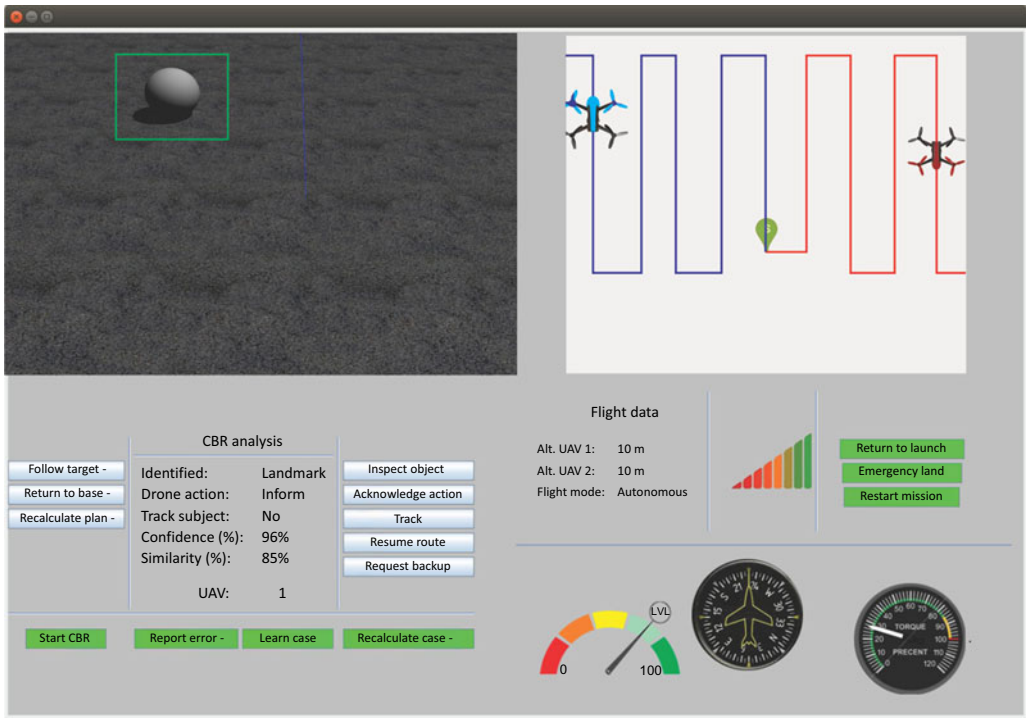


Fig. 19. HMI for two UAVs at a target recognition point.

5. Conclusions and Future Work

Robotic systems require a combination of several specific and interrelated building blocks to reach autonomous operation capabilities. Many open-source architectures for UAVs have been developed to guarantee a fully autonomous operation. However, in most of the cases, the autonomy level is limited and restricted by the application. Therefore, this research has described an intelligent software architecture for UAVs with an optimized definition of the blocks. The architecture lays the groundwork for the development of a software platform aligned with the current requirements of state-of-the-art technologies. The proposed architecture is designed to have the capability of high-level decision-making, such as understanding, interpretation, and goal-oriented reasoning. A simpler version of the architecture was firstly presented in ref. [20]. Compared with that initial publication, the main contribution of this research work is the capability of supporting the latest advances in technologies and the development of an organized paradigm to support decision-making and high-level cognition. Besides, this paper describes the justification of each functional block and its organization within the architecture. A few experiments were conducted to demonstrate the capabilities of the proposed architecture.

The architecture is based on Bloom's Taxonomy of Learning Domains. The layers can be hierarchically arranged from low-level perception to high-level cognition by a modular design to tackle problems related to specific blocks. Processing data produces information and knowledge at increasing levels of complexity. The several layers in the architecture are capable of abstracting hardware, and they encapsulate the data flow from perception to reasoning.

The experiments confirm the successful use of the architecture to be applied in semiautonomous missions. The UAV presented some degree of autonomy in the decision-making process when making use of the proposed data flow along with CBR. The main limitation of the current architecture is related to the ability of the scene comprehension block to make sense of the incoming data, which means that for more complex scenarios, this may have to recognize human action, more complex objects, and interaction among groups of people.

In terms of evaluation, albeit that the proposed ARCog architecture considers global missions, our assessment was focused on a fully semiautonomous outdoor SAR mission. In the current development stage, the architecture only performs semiautonomous missions, given that several events are still in the evaluation phase. Nevertheless, more experimentation will lead to an increase in autonomy

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

toward supporting fully autonomous missions. Thus, additional experiments should include more complex scenarios and different mission goals. These experiments will evaluate the architecture capabilities for an extended range of situations, including multiple aircraft cooperation, hazardous environments, and UAV failure. Other future works may also include the use of the architecture in mobile robot systems to tackle similar existing problems.

Acknowledgments

The work reported in this paper was performed as part of an interdisciplinary research and development project undertaken by UFJF. The authors acknowledge the financial funding and support of the following companies: CAPES, CNPq, INCT-INERGE, BAESA, ENERCAN, and FOZ DO CHAPECÓ, under supervision of ANEEL - The Brazilian Regulatory Agency of Electricity, through Project number PD 03936-2607/2017. The authors also would like to thank CEFET-RJ.

References

1. M. Seleckỳ, M. Rollo, P. Losiewicz, J. Reade and N. Maida, "Framework for Incremental Development of Complex Unmanned Aircraft Systems," *Integrated Communication, Navigation, and Surveillance Conference (ICNS), 2015* (IEEE, 2015) pp. J3–1.
2. Y. Du, C. W. de Silva and D. Liu, "A Multi-agent Hybrid Cognitive Architecture with Self-awareness for Homecare Robot," *2014 9th International Conference on Computer Science & Education (ICCSE)* (IEEE, 2014) pp. 223–228.
3. M. Pérez-Ortiz, J. M. Peña, P. A. Gutiérrez, J. Torres-Sánchez, C. Hervás-Martínez and F. López-Granados, "Selecting patterns and features for between- and within- crop-row weed mapping using uav-imagery," *Expert Syst. Appl.* **47**, 85–94 (2016).
4. F. Ropero, P. Muñoz and M. D. R-Moreno, "Terra: A path planning algorithm for cooperative UGV-UAV exploration," *Eng. Appl. Artif. Intell.* **78**, 260–272 (2019).
5. X. Sun, C. Cai, J. Yang and X. Shen, "Route evaluation for unmanned aerial vehicle based on type-2 fuzzy sets," *Eng. Appl. Artif. Intell.* **39**, 132–145 (2015).
6. J. L. Sanchez-Lopez, M. Molina, H. Bavle, C. Sampedro, R. A. S. Fernández and P. Campoy, "A multi-layered component-based approach for the development of aerial robotic systems: The aerostack framework," *J. Intell. Robot. Syst.* **88**(2–4), 683–709 (2017).
7. C. Van Dang, T. T. Tran, Y.-B. Shin, K.-J. Gil, D. Ngo, B.-S. Kang, G.-D. Lee and J.-W. Kim, "Application of a soar-ROS agent on finding the correct object," *International Conference on Advanced Science and Technology and Advanced Signal Processing*, Hanoi, Vietnam, (2017) vol. 143, pp. 151–155.
8. F. Gobet and P. C. Lane, "Constructing a standard model: Lessons from chrest," *2017 AAAI Fall Symposium Series* (2017).
9. C. C. Insaurralde, "Service-oriented agent architecture for unmanned air vehicles," *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)* (IEEE, 2014) pp. 8B1–1.
10. J. T. Ball, "Advantages of ACT-R over Prolog for Natural Language Analysis," *Proceedings of the 22nd Annual Conference on Behavior Representation in Modeling and Simulation* (Citeseer, 2013).
11. J. G. Trafton, L. M. Hiatt, A. M. Harrison, F. P. Tamborello II, S. S. Khemlani and A. C. Schultz, "ACT-R/E: An embodied cognitive architecture for human-robot interaction," *J. Human-Robot Interact.* **2**(1), 30–55 (2013).
12. J. W. Tweedale, "A review of cognitive decision-making within future mission systems," *Procedia Comput. Sci.* **35**, 1043–1052 (2014).
13. C. Sampedro, H. Bavle, J. L. Sanchez Lopez, R. Suarez Fernandez, A. Rodriguez Ramos, M. Molina and P. Campoy Cervera, "A Flexible and Dynamic Mission Planning Architecture for UAV Swarm Coordination," *Proceedings of 2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (ETSI_Informatica, 2016).
14. T. P. Nascimento and M. Saska, "Position and attitude control of multi-rotor aerial vehicles: A survey," *Ann. Rev. Control.* **48**, 129–146 (2019).
15. E. Coste-Maniere and R. Simmons, "Architecture, the Backbone of Robotic Systems," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1 (IEEE, 2000) pp. 67–72.
16. D. Erdos, A. Erdos and S. E. Watkins, "An experimental UAV system for search and rescue challenge," *IEEE Aerosp. Electron. Syst. Mag.* **28**(5), 32–37 (2013).
17. J. Gu, T. Su, Q. Wang, X. Du and M. Guizani, "Multiple moving targets surveillance based on a cooperative network for multi-UAV," *IEEE Commun. Mag.* **56**(4), 82–89 (2018).
18. B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robot. Autom. Lett.* **3**(2), 1025–1032 (2018).
19. A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.* **7**(1), 39–59 (1994).
20. M. F. Pinto, A. G. Melo, A. L. Marcato and C. Urdiales, "Case-Based Reasoning Approach Applied to Surveillance System Using An Autonomous Unmanned Aerial Vehicle," *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)* (IEEE, 2017) pp. 1324–1329.

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press

21. S. Emel'yanov, D. Makarov, A. I. Panov and K. Yakovlev, "Multilayer cognitive architecture for UAV control," *Cognit. Syst. Res.* **39**, 58–72 (2016).
22. P. Stütz, G. Hummel, M. Kaiser, A. Schulte, P. Climent-Pérez, P. Remagnino, D. Lund, A. Magzoub, Y. Tsado, J. Gozdecki, K. Loziak, "UAV Integration Aspects within the Proactive Network," *World of UAV International Conference WoUCON 2015* (2015).
23. J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Front. Neural Circuits* **10**, 23 (2016).
24. Y. Cui, S. Ahmad and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," *Neural Comput.* **28**(11), 2474–2504 (2016).
25. N. E. Adams, "Bloom's taxonomy of cognitive learning objectives," *J. Med. Lib. Assoc. JMLA* **103**(3), 152 (2015).
26. H. Jiang and Y. Liang, "Online path planning of autonomous UAVs for bearing-only standoff multi-target following in threat environment," *IEEE Access* **6**, 22 531–22 544 (2018).
27. J. Kwak and Y. Sung, "Autonomous UAV flight control for GPS-based navigation," *IEEE Access* **6**, 37 947–37 955 (2018).
28. J. Uhrmann and A. Schulte, "Concept, design and evaluation of cognitive task-based UAV guidance," *J. Adv. Intell. Syst.* **5**(1), 145–158 (2012).
29. M. Barak and M. Assal, "Robotics and stem learning: Students' achievements in assignments according to the p3 task taxonomy-practice, problem solving, and projects," *Int. J. Tech. Des. Edu.* **28**(1), 121–144 (2018).
30. N. Gageik, P. Benz and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access* **3**, 599–609 (2015).
31. T. Das, "Intelligent techniques in decision making: A survey," *Indian J. Sci. Tech.* **9**(12), 1–6 (2016).
32. M. Cummings, "Operator interaction with centralized versus decentralized UAV architectures," **In:** *Handbook of Unmanned Aerial Vehicles* (Springer, 2015) pp. 977–992.
33. E. Baccarelli, P. G. V. Naranjo, M. Scarpiniti, M. Shojafar and J. H. Abawajy, "Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study," *IEEE Access* **5**, 9882–9910 (2017).
34. M. F. Pinto, A. L. M. Marcato, A. G. Melo, L. M. Honório and C. Urdiales, "A framework for analyzing fog-cloud computing cooperation applied to information processing of UAVs," *Wireless Commun. Mobile Comput.* **2019**, 1–14 (2019).
35. F. Zeng, Y. Ji and M. D. Levine, "Contextual bag-of-words for robust visual tracking," *IEEE Trans. Image Process.* **27**(3), 1433–1447 (2018).

https://doi.org/10.1017/S0263574720000521 Published online by Cambridge University Press