JOURNAL
AerospaceLab

ONERA
THE FRENCH AEROSPACE LAB

Issue 15
September 2020

DOI: 10.12762/2020.AL15

Artificial Intelligence and Decision Making

Publisher
Stéphane Andrieux

Editor in Chief
Francis Dupoirieux

Production
ONERA Scientific
Information Department

On line
www.aerospacelab-journal.com
Webmaster ONERA

Contact
E-mail: aerospacelab@onera.fr

Produced by
ONERA - BP 80100
Chemin de la Hunière
et des Joncherettes
91123 PALAISEAU CEDEX
France
www.onera.fr

ISSN: 2107-6596

Artificial Intelligence and Decision Making

Stéphane Herbin
(ONERA)

Jean-Loup Farges
(ONERA)

E-mail: stephane.herbin@onera.fr

DOI: 10.12762/2020.AL15-00

Artificial Intelligence (AI) is currently an inescapable keyword in computer science given its predicted huge contribution to the global economy [1] or to the whole society [2, 3], as argued in many recent white papers or reports. Its techniques are expected to provide efficient ways to deal with heterogeneous and voluminous numerical environments and data, help or even improve decision making, and automate complex functions. The aeronautics, spatial and defense domain (ASD) is impacted by this evolution [4].

AI is a research field with a complex history and numerous areas. It is customary to divide these into two trends:

- Formal and logical, which rely on models, knowledge representation and solvers;
- Empirical, which rely on data, statistical estimation and inference.

Although this last trend mainly drives what is sometimes referred to as the third wave of AI, with Machine Learning playing the key role of a general design principle, it cannot fully solve all problems: data can be rare and costly in the ASD domain, where the requirements of reliability and predictability are often very high.

The present issue of the Aerospace Lab journal contains several examples of AI research, from rather specific studies to more general position papers or surveys, which exemplifies these two traditions and, eventually, their possible combination.

With regard to the representation and use of knowledge, the contribution "*Semantic Mediation for Dynamic Fusion of Observations and Sensor Data*" addresses the problem of fusing sensor data and natural language messages generated by humans for entity tracking and identification in dynamic environments. The framework relies on a generic ontology, which provides a uniform vocabulary for data from human observations and sensor outputs, with different levels of granularity. Specifically, a semantic mediation layer is used to fuse information from both sensors and humans.

"*A survey on chronicles and other behavior detection techniques*" is a survey of information extraction from data flows. It treats this area from a logical perspective and describes logical approaches from information flow processing, knowledge representation and reasoning perspectives. It describes event calculus, Etalis system, chronicle and SQL-based approaches, together with their specific operators for combining and detecting events.

Making plans is a typical intelligent activity often addressed using formal AI methods. The paper "*Collaborative Common Path Planning in Large Graphs*" studies the two-agent collaborative path planning problem, where the agents are incentivized to move together along the same path as much as possible by scaling down the duet cost function when they move together. It proposes A*-based algorithms and provides heuristics to guide the search. The paper presents an assessment of the algorithms on grids of different sizes and with different cost reductions. Another article: "*Planning for space telescopes: survey, case studies and lessons learnt*" presents the application of AI planning and scheduling techniques for optimizing the operations of satellites whose mission is to observe celestial objects. Several mission-planning tools are presented and three case studies are tackled using a constraint-based optimization and operations research approach. Future work directions are highlighted: development of generic mission-planning tools, management of uncertain events and definition of a centralized mission-planning concept for several telescopes.

The Multi-agent architecture is a traditional model of formal AI methods. Considering a class of linear multi-agent s*ystems, "Robust consensus seeking via a multi-player nonzero-sum differen*tial *game*" studies the problem of consensus seeking in the presence of an exogenous signal, possibly representing a disturbance. It formulates the robust consensus-seeking problem as a nonzero-sum differential game, characterizes solutions, and presents simulations of examples to illustrate the resulting performances: one example concerns consensus among agents described by single-integrator dynamics and the other concerns formation flight of unmanned aerial vehicles. A multi-agent approach is also used to coherently manage embedded sensors in the *article "RAMSES: a multi-agent architecture to design a multi-sensor system deployed on the next generation of airborne* platform". This paper focuses in particular on multi-sensor systems embedded in remote piloted aircraft systems and proposes a scenario related to those systems.

The Machine Learning dimension of AI is discussed in three articles: "*Challenges in certification of computer vision based systems for civil aeronautics*" addresses the problem for modern computer vision techniques to comply with current certification standards. By describing two prototypical pipelines, for visual odometry and for scene understanding, it outlines the challenges faced when applying current certification guidelines to non-deterministic systems and machine-learning-based algorithms. The article "*Scaling up information extraction from scientific data with deep learning*" focuses on applications of Deep Learning (DL) for real life use cases where the size of the experimental data is an obstacle for understanding a physical phenomenon and showing how such techniques allow experimental work to get rid of uninteresting, repetitive and time-consuming tasks. Finally, the article "*Recent examples of deep learning contributions for earth observation issues*" focuses on the impact on Earth Observation (EO) practices of using learning methods for remote sensing image analysis. It presents DL particularities and challenges for analyzing EO data, highlights differences between EO and computer vision problematics and covers the main modalities of EO sensor data and the principal aspects of EO image processing: co-registration, image enhancement, classification, object detection, parameter retrieval and multi-temporal analysis

References

[1] *Artificial Intelligence the Next Digital Frontier?* McKinsey and Company Global Institute. https://www.mckinsey.com/~/media/McKinsey/Industries/Advanced%20Electronics/Our%20Insights/How%20artificial%20intelligence%20can%20deliver%20real%20value%20to%20companies/MGI-Artificial-Intelligence-Discussion-paper.ashx, 2017.

[2] *White Paper on Artificial Intelligence: a European Approach to Excellence and Trust.* https://ec.europa.eu/info/files/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en, 2020.

[3] C. VILLANI - *For a Meaningful Artificial Intelligence: Towards a French and European Strategy*. Conseil national du numérique. https://www.aiforhumanity.fr/pdfs/MissionVillani_Report_ENG-VF.pdf, 2018.

[4] *Artificial Intelligence Roadmap. A Human-Centric Approach to AI in Aviation.* European Union Aviation Safety Agency. https://www.easa.europa.eu/sites/default/files/dfu/EASA-AI-Roadmap-v1.0.pdf, 2020.

V. Dragos
(ONERA)

S. Gatepaille
(Airbus Defence & Space)

E-mail: valentina.dragos@onera.fr

Semantic Mediation for Dynamic Fusion of Human Observations and Sensor Data

DOI: 10.12762/2020.AL15-01

This paper addresses the problem of combining human observations and sensor data for entity tracking and identification in dynamic environments. The complexity of the track-and-detect task for realistic applications requires dynamic fusion of sensor data and observations, and a semantic mediation approach is adopted. Moving targets are detected and classified based on sensor data. Soft data in the form of short messages are automatically processed to identify relevant information, to be associated with entities detected by sensors. While sensor data provide rows of numerical features, observations convey finer descriptions of entities and contextual information that is intuitively included by human sources when reporting. A fusion system accommodates both sensor and soft input, and provides a unified framework for their effective integration. The system relies on semantic mediation to combine observations and sensor data and uses ontologies to create a bridge between two complementary representations of the same situation.

Introduction

In dynamic environments, where entities of interest can be not only mobile, but also geographically dispersed, versatile and unpredictable, sensing devices come close to their limits of perception. Human sources are then a key feature to be considered, since they can provide finer input about entities.

Traditionally, fusion systems handled sensor and soft data fusion as two distinct problem sets, based on the intuition that track-and-detect applications seemed to be well supported by fusion of sensor data, while human reports and open sources are more suitable to analyze asymmetric threats in urban areas. The following example shows that the sensor vs. soft data dichotomy is out of date, and makes the case for a unified approach. Let us consider a convoy of vehicles that illegally crosses the border between two countries. As with any illegal crossing, an alarm is triggered and the entity is tracked by a sensor-fusion system. While the convoy approaches a city, one of the vehicles enters the urban area and losing its track is highly possible, since urban terrain has a very dense traffic and affects visibility and line-of-sight communications. Once track loss occurs, any eyewitness sighting or testimonies from bystanders during their day-to-day activities can be of interest and the track-and-detect problem requires not only sensor based fusion, but rather a joint analysis of sensor and soft data. The task does not require qualified field analysts, but rather selecting people who are close to the incident and who may be opportunistic sources and provide meaningful input.

This paper presents a semantic approach to combine soft and sensor data for entity tracking and identification. A general architecture was developed for information fusion, which creates a situation using sensor data and enriches this situation by taking into account observations. The reminder of the paper is organized as follows: Section 2 presents related work on sensor semantics. Architecture of the system and fusion cycles are discussed in Section 3. Section 4 presents approaches developed to process soft and sensor data. Semantic mediation is discussed in Section 5, along with the ontology created to support the overall approach and a practical illustration for entity identification. The last section concludes with a summary and directions for future work.

Related work

The prevalence of heterogeneous systems and their use in applications ranging from health-care to urban traffic monitoring, or security and surveillance, are accompanied by an increase in the heterogeneity of devices connected and formats of data collected, processed and shared. For such systems, semantic technologies offer a way to manage heterogeneity, by providing a common interface to disparate sensors, combining their formats into a unified one and building a coherent view of data.

Several research efforts have been conducted to build ontologies modeling sensors and sensor data, in order to develop techniques augmenting their output and approaches combining heterogeneous sources, as discussed hereafter. Modeling sensor ontologies is a major direction investigated to represent sensor location and supply, along with accuracy, type and frequency of observations in a machine-readable form. SSN is an ontology created by the Semantic Sensor Networks Incubator Group[1] offering descriptions of four related perspectives: sensor, observations, system and property [5]. The model focuses on what and how can be sensed, systems of sensors and their deployment, and the description of observations made at entity level or for particular properties. Sensor Web Enablement (SWE) is a major initiative of the Open Geospatial Consortium (OGC)[2] aimed at improving the capacity to discover relevant sensor data on the Web, through standardized interfaces and specifications, and creating the Semantic Sensor Web, an infrastructure that enables the interoperable usage of sensors by providing services for discovery, accessing and identification based on sensor output augmented with spatial, temporal, and topic-specific semantic metadata [4]. A complement of the Semantic Sensor Web, Sensor Linked Data is a paradigm introduced by Janowicz and colleagues [13] to add interlink as a new challenge. The authors created a proxy delivering observations as Linked Data, connecting them with other data sources, and using ontologies and reasoners for observation alignment.

More related to the work presented in this paper, various research efforts consider the existing bridge between symbolic knowledge representations and raw data collected and measured by sensors. The concept of semantic perception is at the core of approaches developed to upgrade sensor output by attaching semantics to sensor data for enhanced interpretation [12]. The work of Jung [15] investigates the implementation of semantic annotation procedures for sensor streams. Roda and Musulin present in [24] an ontology-based framework to support intelligent data analysis of sensed data, while the construction of a complex situation using multi-layer ontologies is discussed in [22]. The BeAware framework [23] improves situation assessment by using ontologies and rules for spatial and temporal reasoning. Following the same line, graph inferences are used in [21] to combine information provided by heterogeneous sources for security applications. Ontology mapping is used in [14] to create a bridge between two representations of the world: the set of features, as sensed by sensors, and the set of objects, as viewed by humans.

Semantic mediation in sensor networks is addressed by Malewsky [17], who developed a semantic framework to improve matchmaking on sensor measurement and operating capabilities. A modularized Sensor Mediation ontology aligned to the SSN is at the core of the solution, and sensing entities and their attributes are modeled as instances of this ontology. Taking a step further, Bakillah and colleagues present in [2] a semantic mediation service that can support context-aware semantic mapping of sensor outputs and is adaptable to the dynamic of sensor metadata. The system integrates a set of empirical rules and rule-based reasoning mechanisms for semantic mediation.

From a different perspective, the rise of the Internet of Things (IoT) offers a wide application area for fusion methods able to discover potential knowledge from large amounts of perceptual information [9]. For this emergent application context, semantic-based solutions have been used to develop semantic-enhanced solutions for information retrieval [19], [20] to offer a shared understanding for event matching [11] or to support an integrated service platform in smart cities [18].

A more detailed discussion on pitfalls and challenges of heterogeneous fusion is presented in [7].

The research presented in this paper is in line with approaches developed for semantic mediation. While a generic ontology is used to provide uniform descriptions of both data extracted from observations and sensor output, reasoning mechanisms combine those augmented results into a finer description of entities.

A framework for heterogeneous fusion

Entity tracking and identification

Detection and classification of entities has a long tradition and extensive literature. Knowing exactly where the entity is, eventually who that entity might be, and monitoring its trajectory in real-time, has already attracted a lot of interest from both academia and industrial communities.

The main problem of track-and-detect in realistic applications is the combination of the sensor-level detection reports and human observations. Track-and-detect is performed under dynamic conditions: trajectories of entities can be out of reach for sensors, and human observations arrive on an irregular basis. Not surprisingly, results are impacted by the quality of the sensor algorithms for detection and identification, as well as by the ability to efficiently combine sensor output and human reports. In other words, the system should rely upon a mediation layer allowing for the sensor and human reports to share as precisely as possible the meaning of the information conveyed.

Definition of a situation and situation assessment

To avoid terminological confusion, in this work the term entity refers to vehicles, persons, or convoys in the real world. The outcome of the fusion is a situation assessment, to be provided to men in the field involved in operations or to commanders in tactical and operational headquarters.

Each entity is described as a vector of features, which, according to the sensor data used in the fusion process, provides the position and kinematics of the entity, its type and also relations to contextual information, such as geographical features (roads, airways) or to other entities in the situation. An entity is described as a set of

[1] https://www.w3.org/2005/Incubator/ssn/
[2] http://www.opengeospatial.org/

states, representing the knowledge of this entity at any moment in time during the surveillance task. Entity state gathers the estimated features and additional information related to traceability and information assessment, such as state likelihood, for instance.

Let E_i be an entity, having a set of states ES_k, with $ES_k = \left(t_k, K_k, Tr_k, A_k\right)$, a time stamped vector of features, composed of the knowledge K_k including kinematics, nature and additional properties, the traceability Tr_k to observations used to produce K_k and the assessment A_k of K_k, represented as a probability, a likelihood or even as a simple confidence score. Entity states can be built upon sensor-based data and soft observation reports: this only depends on the ability of the algorithms to associate these observations with a given entity. A situation of n entities is defined as the union of the set $E_{p,p\in\{1,\ldots,n\}}$ and the set of $p+q$ collected observations $\left\{O^{sensor}_{i,i\in\{1,\ldots,p\}}, O^{soft}_{j,j\in\{1,\ldots,q\}}\right\}$, some of which could be false alarms, or inaccurate or misleading reports.

Situation assessment combines information from multiples sources to reason about several entities over a range of time horizons. Often the situation is described by a collection of tracks, where a track is a temporal sequence of entity states and it is generally developed for individual or group entities, such as persons or vehicle convoys.

General architecture and cycles of dynamic fusion

The framework developed for heterogeneous fusion was designed to support field practitioners with the ability to select among various technological blocks or to implement new ones in a dynamic environment that demands innovative solutions for increasingly complex challenges. Figure 1 shows the general architecture of the framework.

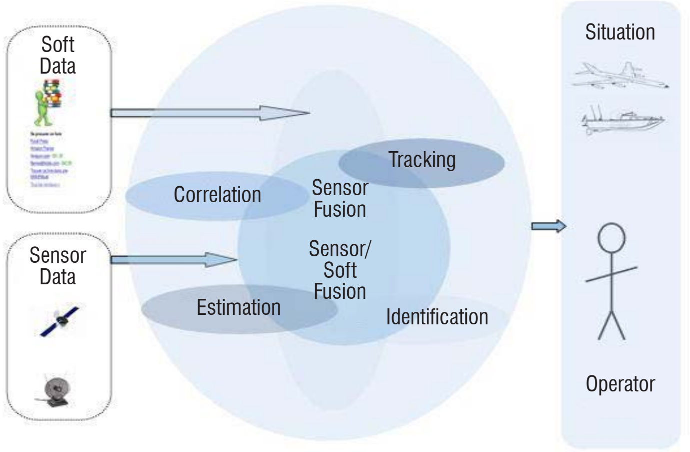


Figure 1 – General architecture for information fusion

For this work, the types of sources available include ground moving target indicator (GMTI), infrared and visible light imaging (IMINT), and signal intelligence (SIGINT) sensors. All of these sensing modalities generate elements that can be described by mathematical and numerical or symbolic representations (e.g., using a universe of discourse), and serve as inputs to automated processing procedures. Sensor measurements result in observations of objects, for which they provide information about properties like location, speed or signal characterization when these objects are electromagnetic emitters. Soft information derived from human or open source is fundamentally different in that its content is often more qualitative and requires additional context elements for complete human interpretation. As shown in Figure 1, heterogeneous fusion is carried out by means of two information fusion cycles. The core is a sensor-based kernel that provides several processes for entity correlation and tracking, along with estimation of their states.

The kernel implements a short-time classical tracking algorithm, since data are provided by sensors on a regular frequency. The outcome is a situation, whose entities are described by their spatio-temporal coordinates and their kinematics. At this stage, the type of entities is also estimated but only using sensor-based data.

The second layer of this architecture enriches the situation by integrating soft-data elements on a stream and irregular basis, as they become available. The enrichment is aimed at refining the states of entities by adding supplementary attributes, such as allegiance and military or civilian nature. Heterogeneous fusion can be considered as a long-time fusion cycle, triggering specific processes as soft-data observations arrive. Those processes first provide matching mechanisms to assign soft-data observations to entities of the situation and then perform fusion strategies in order to combine elements of entity states with items extracted from soft data. The approach is implemented by using a generic development and execution framework [16] providing a collection of basic algorithmic building-blocks for information fusion.

Processing and fusion of sensor data and observations

Extraction of features from sensor data

The features that can be extracted from sensor data, although limited, are heterogeneous due to the different types of deployed sensors.

GMTI sensors are radars with specific signal processing leveraging the Doppler effect which can provide information about moving targets, mainly related to their kinematic state (location and speed in the direction of the sensor) and sometimes some classification information from signal analysis, limited to rough classes of identification, such as rotating objects (e.g., helicopter blades), tracked vehicles such as Tanks, or wheeled vehicles. The location is provided in range and azimuth and possibly in elevation for a 3-D radar, with associated imprecision in each of these dimensions, which can be quite large for azimuth information. The speed is also partially retrieved due to the fact that only one-dimensional information can be acquired, related to the line of sight between the radar and the detected object.

IMINT information is related to imagery or video acquired in diverse wavelengths (visible, or infrared), which may be further exploited through an automated extraction and tracking device and annotated by an operational user. Some sensors are also able to perform tracking on a given object, thus providing track information about the object with location and speed attributes. The operator can then complement this information with precise classification information.

SIGINT information is related to either the detection and localization of specific emitters (e.g., radars) or the interception of communication information between several actors. Through the use of several receivers or a maneuvering receiver, the location of emitters and specific technical information can be extracted.

From the set of items sent by these various types of sensors, a correlation scheme can be set up based mainly upon kinematic information to perform the right association between these detections, so there are a limited number of duplicated tracks related to the same real entity. From this association, a combined estimation of attributes is performed, which leads to a more precise kinematic information and, if available, a rough estimate of the type and hostility of the entity. Different methods, using mainly Bayesian, evidence theory or heuristic techniques, are involved to combine classification results from these detections.

Identification of properties from soft data

Processing of soft data identifies properties of entities within natural language messages. Messages also have a heterogeneous content, and can provide insight on different aspects, such as entity location, and evolution. The methods developed extract binary association attribute-value from messages, which can be easily modeled as properties of entities and integrated into entity states. Attributes specify the type (vehicle, bus, person, etc.), allegiance (foe, friend, neutral) or nature (civilian, military, insurgent, etc.) of entities.

Attributes are identified from soft data by using a text-mining approach based on collocation identification. Collocations are associations of words that co-occur frequently within the same sentence, whether because the meanings of words are related to each other (e.g., vehicle- road, car-driver) or because the two words make up a compound noun (car stop, subway station). The extraction algorithms focus on collocations composed of two words, also called bi-grams. The method developed to extract collocation is shown in Figure 2.

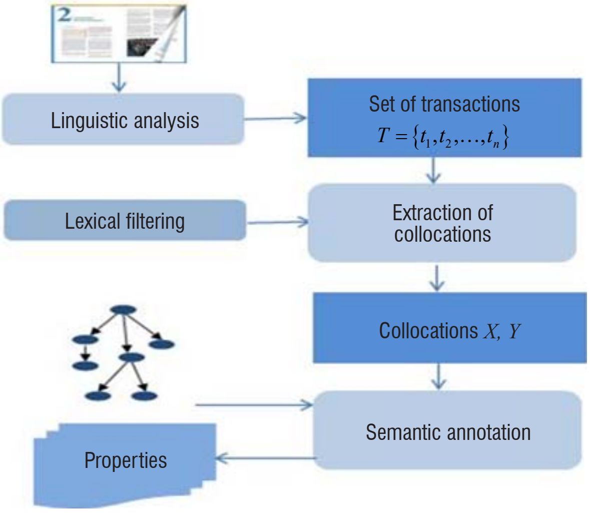


Figure 2 – Processing of soft data

Linguistic analysis reprocesses documents to split texts and filter stop words, and perform lexical normalization. Lexical normalization identifies and removes lexical heterogeneities, which appear as the same type of information provided by different lexical forms, and concern namely: date/ time expression (11 November 2011 vs. 11/11/2011); currency; geographical coordinates; metric units (m vs. inch); expression of quantifiers (two vs. 2 vehicles) and abbreviations (poss. vs. possible). Linguistic analysis also concerns the identification of sentence boundaries and performs tokenizing, while filtering a list of stop words. The output of this phase is a set of transactions, where each transaction is a sentence whose items are words. The set of transactions is the input for the next step. Collocations are generated by using a window placed over a sentence, such that two words are analyzed at a time by moving the window from the first to the last word of the sentence, see Figure 3.

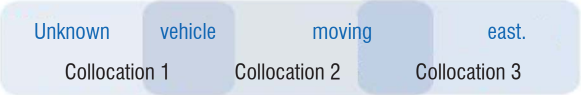


Figure 3 – Extraction of collocations

Because simply taking the entire list of collocations captures an excess of extraneous and incoherent information, additional processing is needed to filter relevant word associations thanks to semantic annotation.

Semantic annotation is performed automatically, using procedures based on lexical similarities, which associate a real number with a pair of words. Lexical similarity offers a measure of the degree to which two words are similar and are used to label a collocation by ontological concepts. The hierarchy of concepts allows the *attribute* part of the association to be retrieved. Thus, given a concept C considered as *value*, the associated *attribute* is identified as the least specific concept subsuming *Property* and generalizing C, as shown in Figure 4.

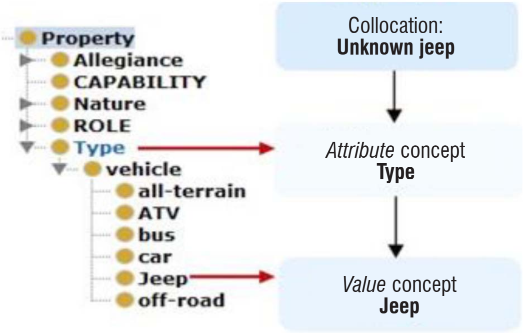


Figure 4 – Semantic annotation

For instance, the collocation *unknown bus* will be matched to the set (*Bus*, *Bus*, *Type*), since *unknown* is not assigned to a concept, while the *insurgent vehicle* is annotated by (*Vehicle*, *Vehicle*, *Type*) and (*Insurgent*, *Insurgent*, *Allegiance*), since both *Vehicle* and *Insurgent* are concepts of the ontology.

At the end of this phase, annotations of collocations are generated in the form of tuples:
$A_i=(W_i,C_i,T_i)$ where A_i is the annotation of item i, W_i is a word, part of a collocation, C_i, is a concept assigned to W_i by lexical similarities, and T_i is the category of C_i, as identified by inferences.

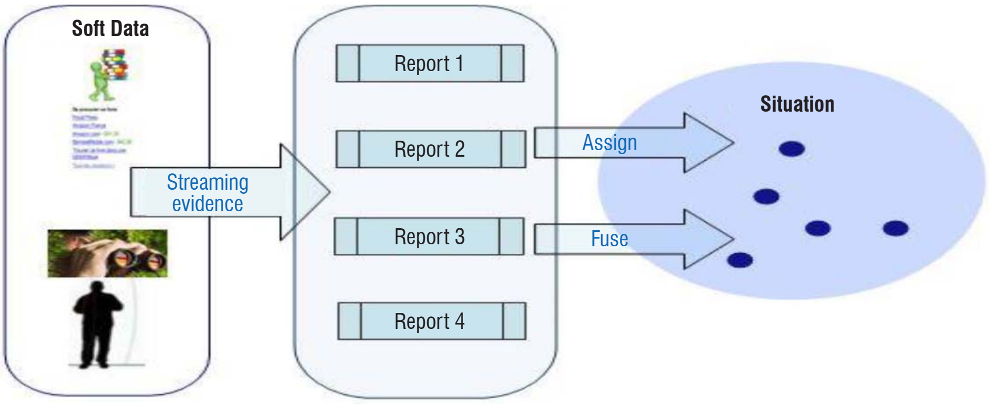


Figure 5 – Combining soft and sensor data

Integration of sensor data and observations

The overall solution for human observations and sensor data integration is summarized in Figure 5 and consists of the assignment of observations sent by humans to entities created by sensor data processing. Soft data used are brief human reports, conveying information about entities in the field. Among those entities, some could be of interest, already detected and eventually tracked by sensors. Information extracted from incoming soft reports will not be considered for tracking purposes; instead, it will help human-operators to have a better description of the situation.

Assignment of human observations to entities of the situation is carried out in the light of spatio-temporal correlation. Given that observations are associated with a timestamp and have specific locations, this method first estimates a correlation coefficient to describe the probability of a human observation to be assigned to e_i , an entity of the situation. The current states of e_i along with its previous states are taken into account for this estimation, since soft observations are not necessarily synchronized with the current situation. Results of this estimation are then ranked and the observation maximizing the value is selected and added to the set of observations associated with e_i .

This combination of soft and sensor data is supported by the fusion architecture.

Semantic mediation

Semantic mediation relies on using a domain ontology to describe data semantics, and implementing semantic annotation procedures to associate sensor output and human observations with corresponding elements of the domain ontology along with reasoning mechanisms for data integration.

While spatio-temporal association enriches the overall situation by adding a set of human observations, semantic mediation is used at entity level to enable the fusion of sensor data and observations.

Semantic mediation is implemented as a process allowing data provided by different types of sources to be combined and a domain ontology is at core of this process, as shown in Figure 6.

The role of the mediation process is to integrate relevant observations to sensor inferred entities according to a shared semantics modeled by a domain ontology and to master the gap between low-level features and richer conceptual descriptions of each entity. Ontologies, as introduced by Gruber [10], are formal domain descriptions defined as: $O=\left(C,R,H^C,H^R\right)$ having: C a set of concepts, R a set of relations, H^C and H^R hierarchies defining a partial order over the set of concepts and relations, respectively. The ontology used for this work was created by using a top-down approach. The development began with a preliminary conceptualization, where a list of high-level

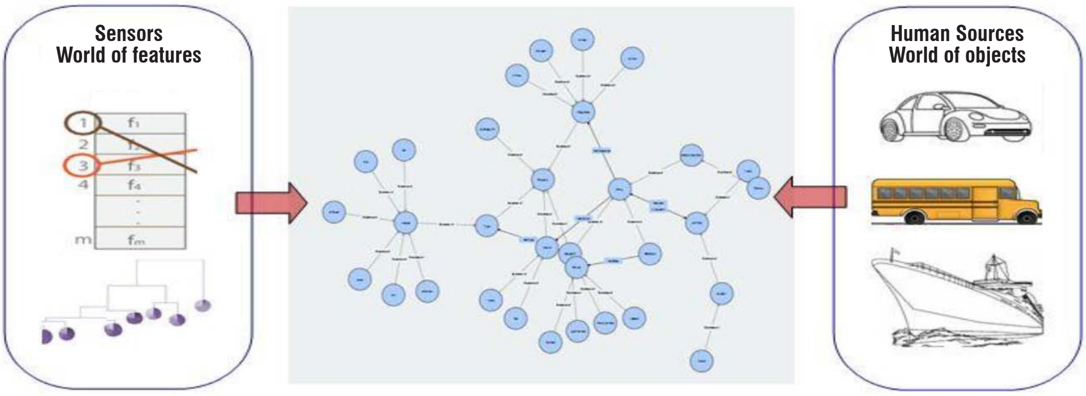


Figure 6 – Semantic mediation

concepts was identified by using the MIM Information Model (MIM)[3]. The hierarchy of concepts was iteratively enriched by adding new classes. The result is a domain ontology composed of 31 concepts, see Figure 7.

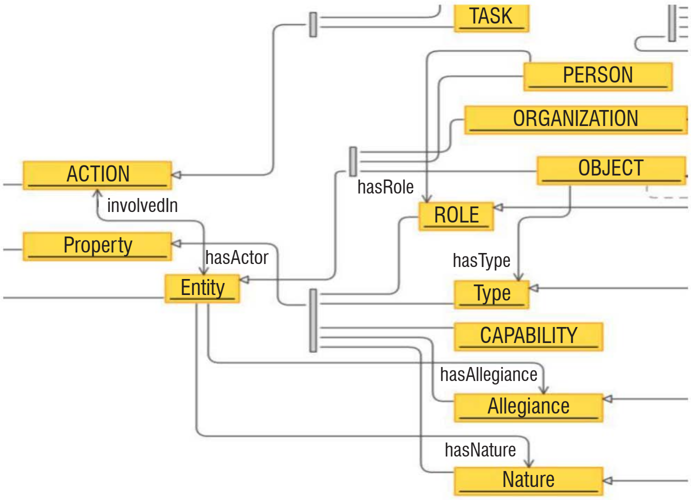


Figure 7 – Semantic mediation

Main classes are *Entity*, *Action* and *Property*. The ontology also has 6 object properties, to make explicit interactions between classes. Thus, hasActor (Event, Entity) models the association of an Entity with an Event and its reversed relation is involvedIn (Entity, Event). The four remaining relations associate entities and their properties: hasRole (Entity, Role), hasType (Entity, Type), hasNature (Entity, Nature) and hasCapability (Entity, Capability). All developments and testing were carried out using Protégé[4], and the ontology is represented in OWL DL, a description logic [1] sub-language of OWL [6].

Refinement of entity states

In the surveillance problem described in this paper, complementary and overlapping inputs exist. Sensor processing has the capacity to identify the type of entities, which is then added to the entity state in the form of a tuple attribute, value-of-attribute, for instance type, bus.

Sensors classify entities based on their measured features and by using some supervised methods for classification, which provide a limited number of categories. Besides having a limited number of categories, sensor processing also has its own detection limitations, and more subtle aspects such as the allegiance of entities are out of reach for their sensing capabilities.

Fusion of sensor and soft data is twofold: first, complementary properties extracted from human reports are added to states of entities; second, the type of entities is updated by taking into account the type of entities as stated by sensor processing and the type as extracted from soft data. In order to update the type, reasoning mechanisms are used to combine attributes of entities. More specifically, given that operators are interested in having a more precise description of entities, reasoning procedures identify the most specific concept of both type labels. This concept is then used to describe the entity, as illustrated in Figure 8, where the final state of the entity highlights the type bus, as a concept more specific than vehicle.

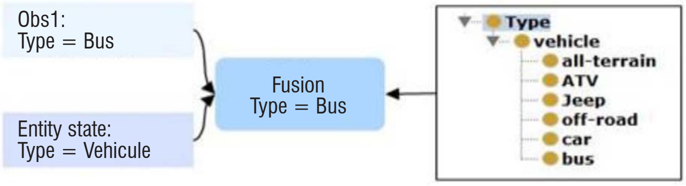


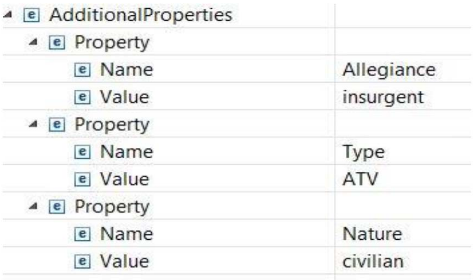
Figure 8 – Inference for type refining

Observations can be noisy, incomplete and sometimes irrelevant, and the inference mechanism fails to identify the most specific concept of both type labels. In this case, fusion provides inconclusive results. Inconclusive inference is due to contradictions between sensor reports and observations, or accidental associations of observations with entities. When successful, the result of fusing soft and sensor data at entity level is a more specific identification of the type of entities, and the enrichment of their state thanks to additional properties that cannot be inferred by sensor processing.

Practical validation and evaluation metrics

An experimental track-and-detect scenario was adopted to provide a valid proof-of-concept of the fusion system. The scenario involves a multi-sensor multi-tracking task with a network of sensors, several observers and a main information fusion functionality.

A total of 20 observations sent by operators on the ground were used, in addition to 204 GMTI reports, 12 COMINT reports, 7 ELINT reports and 5 IMINT reports. At entity level, properties extracted from soft data describe allegiance (friend, foe, insurgent), type and nature (civilian, military), as shown in Figure 9.



AdditionalProperties	
Property	
Name	Allegiance
Value	insurgent
Property	
Name	Type
Value	ATV
Property	
Name	Nature
Value	civilian

Figure 9 – Processing observations: type, nature, allegiance

Sensor data is provided as formatted reports integrating position, time, type and potential subtype of an observed entity. Some additional features may also be present, such as vehicle color or even the vehicle identification number. Observations are in the form of structured reports, having a natural language paragraph to summarize information collected by human sources. After feature extraction from text and fusion of items, the type of entity is updated (*Tank*) and its hostility is identified (HO). Over time, 102 entities are detected and tracked in the situation. Although using an experimental scenario offers a basis to evaluate whether or not the system meets its objectives, a formal evaluation is needed to estimate the impact of using semantic mediation to support heterogeneous fusion. Since mediation affects both entity states and the overall situation, *information*

[3] https://www.mimworld.org/portal/projects/welcome/wiki/Welcome
[4] https://protege.stanford.edu/

gain and *quality of service* introduced in [3] to characterize situation assessment are metrics able to quantify this impact.

Information gain is a criterion intended to capture the value added to entity states after updating their descriptions by using semantic inferences. Information gain is the ability of the system to provide improvements, and its values can be assessed by taking into account the number of additional properties added to entities and the quality of their type.

Information gain is defined assuming that changing the state of entities by integrating observations improves the description of entities, as follows:

$$InfoGain = \frac{N_c}{N} \quad (1)$$

where N_C is the number of entities whose states are affected by observations, and N is the overall number of entities of the situation. Values of information gain are between 0, when observations are not related to entities of the situation, and 1, when ideally states of all entities are updated.

Quality of service is a criterion used to characterize the quality of situation assessment and encompasses aspects related to timeliness, uncertainty of the overall picture, and quality of individual descriptions at entity level. Nevertheless, uncertainty can also arise at the situation level, since soft and sensor data can provide contradictory information items.

Quality of service takes into account the number of failures due to inconclusive fusion:

$$QoS = \frac{U}{N_c} \quad (2)$$

where N_C is the number of entities whose states are affected by observations, and U is the number of valid inferences. QoS ranges between 0, when all inferences for type refinement are inconclusive, and 1, when they are valid.

The values of information gain and quality of service at the end of the scenario are 0.047 and 1, respectively. For this experimentation, all inferences for type refining are valid and low values of information gain are directly related to the small number of entities affected by incoming observations.

Conclusion and future work

This paper tackles challenges arising when combining sensor data and human observations in dynamic environments, and argues that semantics provide a basis to augment results provided by sensors, facilitating their fusion with items extracted from observations. More particularly, the authors describe how semantic annotation of both soft and sensor items allows the implementation of reasoning mechanisms and improves the overall situation to be presented to human operators. From a practical standpoint, a unified fusion framework allows the integration of sensor rows and human observations, and offers specific procedures to extract information in an unsupervised way from sets of numerical values and textual reports. Assuming that results are compliant with an ontological description, reasoning mechanisms are then applied for the automated mediation of information items extracted from heterogeneous data. While enriching description of entities, the approach provides a way to adapt to an evolving context, since ontology-based models can be consistently modified to keep peace with the latest evolutions in the field. The long-term vision underlying this research is to enable on-the-fly integration of human observations in various systems already capable of processing sensor data, and the expected result is a roadmap towards a semantically-enabled heterogeneous fusion. The main difficulty is the implementation of reasoning mechanisms flexible enough to match features extracted from human observations and sensor data. In the short term, the use of semantics to identify, not only entities, but also relationships, is currently under analysis [8] ■

References

[1] F. BAADER - *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[2] M. BAKILLAH, S. H. LIANG, A. ZIPF, M. A. MOSTAFAVI - *A Dynamic and Context-Aware Semantic Mediation Service for Discovering and Fusion of Heterogeneous Sensor Data*. Journal of Spatial Information Science, 2013(6), 155-185, 2013.

[3] E. BLASCH, P. VALIN, E. BOSSE - *Measures of Effectiveness for High-Level Fusion*. Information Fusion (FUSION), 2010 13th Conference on (pp. 1-8). IEEE, July 2010.

[4] A. BRÖRING, J. ECHTERHOFF, S. JIRKA, I. SIMONIS, T. EVERDING, C. STASCH, R. LEMMENS - *New Generation Sensor Web Anablement*. Sensors, 11(3), 2652-2699, 2011.

[5] M. COMPTON, P. BARNAGHI, L. BERMUDEZ, R. GARCÍA-CASTRO, O. CORCHO, S. COX, V. HUANG - *The SSN Ontology of the W3C Semantic Sensor Network Incubator Group*. Web semantics: science, services and agents on the World Wide Web, 17, 25-32, 2012.

[6] M. DEAN, G. SCHREIBER, S. BECHHOFER, F. VAN HARMELEN, J. HENDLER, I. HORROCKS, L. A. STEIN - *OWL Web Ontology Language Reference*. W3C Recommendation February, 10, 2004.

[7] V. DRAGOS, K. REIN - *Integration of Soft Data for Information Fusion: Pitfalls, Challenges and Trends*. Information Fusion (FUSION), 17th International Conference on (pp. 1-8). IEEE, July 2014.

[8] V. DRAGOS, S. GATEPAILLE, X. LEROUVREUR - *Refining Relation Identification by Combining Soft and Sensor Data.* Information Fusion (FUSION), 19th International Conference on (pp. 2139-2146). IEEE, July 2016.

[9] F. WANG, L. HU, J. ZHOU, J. HU, K. ZHAO - *A Semantics-Based Approach to Multi-Source Heterogeneous Information Fusion in the Internet of Things*. Soft Computing, 21(8), 2005-2013, 2017.

[10] T. R. GRUBER - *A Translation Approach to Portable Ontology Specifications*. Knowledge acquisition, 5(2), 199-220, 1993.

[11] S. HASAN, E. CURRY - *Approximate Semantic Matching of Events for the Internet of Things*. ACM Transactions on Internet Technology (TOIT), 14(1), 2, 2014.

[12] C. HENSON, A. SHETH, K. THIRUNARAYAN - *Semantic Perception: Converting Sensory Observations to Abstractions*. IEEE Internet Computing, 16(2), 26-34, 2012.

[13] K. JANOWICZ, A. BRÖRING, C. STASCH, S. SCHADE, T. EVERDING, A. LLAVES - *A Restful Proxy and Data Model for Linked Sensor Data*. International Journal of Digital Earth, 6(3), 233-254, 2013.

[14] A. L. JOUSSELME, V. DRAGOS, A. C. BOURY-BRISSET, P. MAUPIN - *Same World, Different Words: Augmenting Sensor Output Through Semantics.* Information Fusion (FUSION), Proceedings of the 14th International Conference on (pp. 1-8). IEEE, 2011.

[15] J. J. JUNG - *On Sustainability of Context-Aware Services Among Heterogeneous Smart Spaces*. J. UCS, 16(13), 1745-1760, 2010.

[16] X. LEROUVREUR, F. DAMBREVILLE, V. DRAGOS - *Principles of a Unified Framework for Heterogeneous Information Fusion and Assessment.* Military Communications and Information Systems Conference (MCC), 2013 (pp. 1-8). IEEE, 2013.

[17] C. MALEWSKI, A. BRÖRING, P. MAUÉ, K. JANOWICZ - *Semantic Matchmaking & Mediation for Sensors on the Sensor Web*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(3), 929-934, 2014.

[18] M. RYU, J. KIM, J. & YUN - *Integrated Semantics Service Platform for the Internet of Things: A case Study of a Smart Office.* Sensors, 15(1), 2137-2160, 2015.

[19] X. Su, J. Riekki, J. K. Nurminen, J. Nieminen, M. Koskimies - *Adding Semantics to Internet of Things*. Concurrency and Computation: Practice and Experience, 27(8), 1844-1860, 2015.

[20] F. ZHAO, Z. SUN, H. JIN - *Topic-Centric and Semantic-Aware Retrieval System for Internet of Things*. Information Fusion, 23, 33-42, 2015.

[21] V. CARLETTI, R. DI LASCIO, P. FOGGIA, M. VENTO - *A Semantic Reasoner using Attributed Graphs Based on Intelligent Fusion of Security Multi-Sources Information*. International Workshop on Activity Monitoring by Multiple Distributed Sensing (pp. 73-86). Springer, Cham, 2014.

[22] F. P. PAI, L. J. YANG, Y. C. CHUNG - *Multi-Layer Ontology Based Information Fusion for Situation Awareness*. Applied Intelligence, 46(2), 285-307, 2017.

[23] N. BAUMGARTNER, S. MITSCH, A. MÜLLER, W. RETSCHITZEGGER, A. SALFINGER, W. SCHWINGER - *A Tour of BeAware – A Situation Awareness Framework for Control Centers*. Information Fusion, 20, 155-173, 2014.

[24] F. RODA, E. MUSULIN - *An Ontology-Based Framework to Support Intelligent Data Analysis of Sensor Measurements*. Expert Systems with Applications, 41(17), 7914-7926, 2014.

AUTHORS

Valentina Dragos is a research scientist and member of the Department of Information Modeling and Systems at ONERA, the French Aerospace Lab. Valentina received her Masters and PhD degrees in Computer Science from Paris V University, and her research interests are related to topics in the field of symbolic artificial intelligence, natural language processing, semantic technologies and automated reasoning. Since joining ONERA in 2010, Valentina has contributed to national and European projects focused on crisis management, maritime awareness and cyber terrorism. Applications of her research include: semantic interoperability for command and control systems, engineering of symbolic data for situation assessment, and exploration of social media and cyberspace.

Sylvain Gatepaille joined Airbus Defence and Space in 1995, and he currently holds an Airbus senior expert position in the data analytics and information fusion domain in the Intelligence division. His current research interests are related to heterogeneous data fusion, tracking and classification, as well as data analytics software development for anomaly and misinformation detection. Sylvain has contributed to various projects, including national, European, EDA and NATO projects, in the field of information fusion. He was an active member of the NATO work group dedicated to data fusion standardization, and he has published various articles in scientific conference publications and journals. Sylvain is currently the technical leader of the FusionLab toolset provided by Airbus Defence and Space.

R. Kervarc
(ONERA)

A. Piel
(CEA LIST, Executable Language
Engineering and Optimisation
Laboratory R&D Department)

E-mail: romain.kervarc@onera.fr

DOI: 10.12762/2020.AL15-02

A Survey on Chronicles and other Behavior Detection Techniques

Until recently, the processing of a rapidly changing dataflow used to be very costly in terms of computation duration. Thus, the extraction of semantic information, requiring complex correlations of events in a temporal pattern, was not possible in real time. Computer performance has sufficiently improved to now allow such processing to take place, with, obviously, a very broad range of interesting applications. Various underlying formal frameworks exist to perform this kind of analysis, and this paper is aimed at reviewing the various (families of) such formalisms, and at comparing them according to a series of general features. It also provides a detailed example of the kind of analysis that can be performed with these formalisms in aeronautics, where the recent evolution of air traffic management, the potential introduction of unmanned aircraft in general air traffic, and various other recent trends form a general context pointing toward a much wider use of dataflow exchanges.

Introduction

Over the past years, the handling of rapidly changing dataflows at a semantic level has attracted a lot of interest. Indeed, while the semantic processing of a large, fast-pace flow of data used to be too costly in terms of computation, thus obliging to a choice between online syntactic processing and offline semantic processing, computer performance now allows semantic information to be extracted on-the-fly from a large dataflow.

This is, of course, a quite interesting feature in a very broad spectrum of fields, as is evidenced by many recent applications of artificial intelligence. Aeronautics is a particularly interesting field for these dataflow extraction techniques: indeed, while exchanges between pilots and control used to rely mostly on radio, nowadays a large flow of data is exchanged between them. The potential introduction of unmanned aircraft and the recent evolution of air traffic management also point toward a large flow of data exchanges, to which agents of the system have partial access (each having different sensors to track the underlying events), and the introduction of reasoning and semantic processing of these events is a valuable assistance for the pilots and controllers involved.

Traditionally, information extraction from dataflows has been roughly classified into two families of approaches:

- on the one hand, *information flow processing* (IFP): these approaches focus on efficiently handling dataflows by treating incoming information on the fly and providing extracted information in real time;
- on the other hand, *knowledge representation and reasoning* (KRR): these approaches focus on complex reasoning abilities, but perform well mostly on data that changes in low volumes at low frequency.

Stream Reasoning is a multidisciplinary approach to this issue, which encompasses both families and is aimed at combining their respective benefits by enabling complex reasoning about rapidly-changing information flows.

Knowledge-representation and reasoning approaches are based on temporal logic, belief revision, changing vocabularies and evolving ontologies. They find lots of applications within the context of the Semantic Web and allow very complex reasoning tasks. However, there are two drawbacks with regard to these for handling rapidly-changing dataflows. First, the tools involved generally rely on strong combinatorics, and are often not able to scale up to high-frequency dataflows. Second, typical Semantic Web architectures generally crawl and cache information, which is not a robust approach in the case of high-frequency dataflows, since the crawled and cached information would become obsolete too quickly.

Information flow processing is rooted in the so-called Data Stream Management Systems (DSMS), which historically stem from Data Base Management Systems (DBMS). As their name indicates, DBMS are intended to manage databases; i.e., persistent data, where updates are infrequent and where information is extracted through user-made queries. DSMS try to accommodate, within this transient framework, continuously updating data: instead of handling queries that are run just once and of returning a comprehensive answer, DMSM continuously run *standing queries* and return partial answers that are updated on the fly as new data arrives.

Even if they do not seem to have much in common (DBMS handle persistent data by executing ad-hoc queries just once, while DSMS handle transient data by continuously running generic queries), both approaches share a common background and, in particular, process data through transformations based on a relational algebra (e.g., selection, aggregation, joining, etc.). Thus, DSMS can be described, in general, as having very good performance in terms of efficiency, but a rather limited expressivity.

To overcome this limitation, other approaches have been developed in various communities, in which a notion of "distributed system" existed. These approaches have a common characteristic, which is that they consider incoming information in the dataflow not *per se,* but rather as a notification of events occurring in the real world, and are aimed at reconstructing the higher-level behavior of which these events are a trace, mostly through filtering and combinations. In this sense, they are pretty much inspired by the publish-subscribe model that is commonly found in distributed systems: on top of the usual publish-subscribe system, where events are considered separately from the others, they build a more expressive subscription language that allows complex event patterns involving (much) more than one event to be considered. These approaches are referred to by the generic designation of *complex event processing* (CEP). While traditionally classified as part of the IFP family, some CEP techniques have reached reasoning abilities that are comparable to some KRR approaches.

CEP techniques, due to the variety of applications and associated specific needs, exist in a broad variety. This paper is aimed at comparing several CEP frameworks, namely:

- Event Calculus: an approach based on situation calculus, but dealing with local rather than global events;
- ETALIS: an approach based on logical programming and aimed at combining temporal properties with database querying;
- Chronicles: a generic term encompassing various systems based on event signatures;
- Other approaches that are not strictly CEP, but rather based on active databases, DSMS, or KRR.

This paper is organized as follows. First of all, we describe the various important features that can be used to distinguish the various formal frameworks that exist in the CEP community. Then, we discuss the compared merits of the techniques (or technique families) described above, according to these features. We then illustrate the interest of the approach within an aeronautic context, considering an example. Finally, we conclude this survey by providing a brief overview of the various domains where CEP-related techniques have been used with success.

Important features

As stated above, there are many different approaches to stream reasoning and they fulfil the various needs of a broad variety of applications. This section lists the various features that can be used as distinctive criteria for different stream-reasoning approaches.

Language-related features

Any stream-reasoning technique relies upon a formal language that is used to describe the behaviors to be identified. As is frequently the case, expressivity generally results from a trade-off:

- on the one hand, high expressivity is desirable in order to be able to finely describe the behaviors to be detected, and to distinguish behaviors that have very similar traces in terms of observable events;
- on the other hand, higher expressivity inevitably implies a more complex recognition process, and thus less-efficient computation.

In [33] (Section 3.8), Cugola and Margara list operators that are commonly found in the constructs of most CEP frameworks, notably:

- Sequence: two patterns following each other, generally with the sole condition that the first pattern must have been completely recognized before the recognition of the second one starts (some frameworks add that the ending point of the first one must coincide strictly with the starting point of the second one);
- Disjunction: either one of two patterns must be present; when dealing with a rich event (with valued attributes), this can lead to complications because, unless both patterns in the disjunction contain the same event attributes, it may no longer be possible to reason about these attributes;
- Conjunction: it is worth noting that the conjunction cannot be reduced with the two previous operators, since when the elements of conjunction are not elementary events, but rather involve more than one event, conjunction allows intertwined behavior which sequences do not;
- Iteration, which can be parameterized by the number of iterations (which is a parameter relating to the structure of the behavior);
- Negation: this operator is generally tricky when dealing with a flow of information. Indeed, there has to be some form of boundary on the part of the flow, in which a negation must be detected in order to be able to yield effective detections. Otherwise, there will always be a possibility that a later event may trigger the recognition of the negated behavior, and hence invalidate the recognition of its negation. Therefore, many authors prefer to refer to *absence* (implicitly on a bounded support) rather than negation;
- Temporal constraints: while many other temporal logics allow interval properties (e.g., Duration Calculus in [26]) to be expressed, the formalism of Allen's 13 relations [1] is a generally accepted reference in the domain of CEP (see [5, 51, 69]), since it exhaustively considers all possible arrangements between two time intervals;
- Parameter value constraints: the parameters here are parameters of elementary events or of higher-order behaviors.

In addition to the extent to which each of these operators can be expressed, another trait related to the underlying language is the question of whether an open or closed syntax should be used; i.e., whether to allow meaningless formulae or not.

Besides expressivity, an important not-unrelated issue in stream reasoning is how behaviors to be detected can be extracted from experts having an operational knowledge of the behaviors of interest. Within this context, a concise and readable language, where syntactical changes are easily associated with their semantics, is clearly desirable, but again there is a trade-off here, since a very-high readability could lead to an extreme oversimplification of the language, and thus to reduced expressivity.

Recognition-related features

The amount of information contained in recognition is also an important feature, as since recognitions are, in principle, done with a purpose, which implies processing that may require being able to return to the triggering events in the flow. In this sense, recognition information can constitute *evidence* of the recognition, and the nature of this evidence may depend on the application: for example, in some applications, one may be only interested in knowing that the behavior of interest occurred at least once. In other applications, e.g., telecommunication network monitoring [40], a single occurrence of a given behavior is generally not significant, whereas its repetition is. When dealing with security and safety, it is generally necessary to investigate all instances of hazardous behaviors, and not just any. However, another trade-off arises, since the identification of multiple recognitions requires keeping track of all possible recognition starts, and of all intermediate recognitions, thus having an adverse effect with regard to the efficiency of the recognition.

Thus, two main issues arise with regard to recognition: historization and multiplicity.

As explained above, historization is a feature where events are considered as a trace of the behaviors that have been recognized, and recognitions contain the necessary information to return to the events that triggered it.

As to multiplicity, a classification of recognition contexts with respect to this issue has been proposed in [33]:

- The "recent" context: only the most recent occurrence of an event initiating a recognition is kept (each pattern to be recognized is associated with a unique instance of its initiating event at any point during the processing of the flow);
- The "chronicle" context: occurrences are managed in a FIFO way, with the oldest occurrences being used first and discarded as soon as they have been used;
- The "cumulative" context: all event occurrences are stored but, whenever a pattern is recognized, all event occurrences involved are discarded;
- The "continuous" context: all events are stored and can always be used.

Flow-related features

An important feature of any stream-reasoning framework is the way in which it deals with the event flow, and, indeed, which assumptions it makes with regard to it, which determine what kinds of flows it is able to handle. An ideal flow would be a unique flow, totally and strictly ordered: while this kind of flow would probably be handled effectively by most recognition algorithms, it may be possible, depending on the context of the study, to consider other kinds of flows, and notably:

- Distributed flows: the flow may not be centralized, but rather be made of several distributed subflows – in this case, the ordering between events provided by different subflows could be a problem for behavior patterns depending on event order (e.g., sequence), especially if they are not timestamped by a synchronous clocka;
- Partially-ordered flows: events in the flow are not totally ordered, which is a generalization of the previous case and leads to the same problems;
- Non-strictly ordered flows: events may arise simultaneously; this is a case that may have unexpected side-effects in situations where event order is important for the behavior;
- Delayed flow: events may arise late, or their occurrence date may be corrected *a posteriori* (which happens, e.g., in cases where events may be revised following the failure of a transaction) – in general, the date on which the event occurs differs from the date on which it is entered into the reasoning system, which requires specific mechanisms to handle events properly.

In addition, another important aspect of the flow is the time model that it uses. Time is generally linear, but can be discrete, either with a fixed pace or with variable granularity, or continuous. The time model may also not exist: this is the case in most DSMS, where there is no time model and events are considered only with an order – but this has, of course, a negative impact on the expressivity of the language.

Uncertainty-related features

In real cases, lots of uncertainties appear naturally:

- With regard to event dates (or order);
- With regard to event attributes;
- With regard to the events themselves (whether they really occurred or not);
- With regard to behavior parameters;
- With regard to behavior structures.

Depending on the uncertainties considered, specific mechanisms have to be considered, either in the description language or in the recognition algorithm itself.

Self-reference features

Recognitions can be self-referent and create events at various points of the recognition process. The most frequent occurrence of this feature is when the recognition of a behavior triggers a new event that is added to the flow. This yields many issues, since such systems are intrusive: in particular, the flow depends on the monitored activities. In extreme cases, recognitions may also have an effect on the monitored activities, where new activities to be recognized are dynamically added into the flow, leading to a retroaction loop that is difficult to manage, both theoretically and practically.

Event Calculus

Event Calculus (EC) is a formal framework allowing events and actions to be represented and reasoned upon in the form of an executable logical program. It is aimed at determining time-evolved values for logical propositions (the so-called *fluents*).

EC was introduced by Kowalski and Sergot in [50]. Its name is derived from Situation Calculus; the difference between these two frameworks being that it deals with local, rather than global, events:

the purpose of this change is to avoid the frame problem for the sake of efficiency. EC claims to provide a formal analysis of the concepts involved; i.e., events and actions. It can be expressed using Horn clauses, to which, consistently with the logical programming foundation of the approach, a notion of negation by failure is added (thus introducing a closed-world assumption).

The founding principles of EC are the following:

- Events can be processed in any order, not necessarily in relationship with their order of occurrence, since the past and future are considered symmetrically;
- Events can be concurrent and are not necessarily punctual (an elementary event can have a duration);
- Updates are possible, but only if they are additive: they can add information but never remove information;
- The dates of events are not particularly relevant, whereas their relative order is.

Many EC dialects exist, some of which allow the handling of delayed actions or continuous state changes, such as, e.g. [58, 57], an interval-based work built with reaction rules; these approaches are catalogued in [54].

An interesting modular approach can be found in the EC dialect developed by Artikis et al.: this dialect allows a low-level event to be composed into high-level complex behaviors, using the predicates found in Table 1 to express temporal constraints, with an underlying linear time model. These predicates are defined relying upon axioms, some of which may be independent from the application domain. The formalism is quite expressive, and allows constraints, whether temporal or not, to be expressed and contains a form of absence, in the form of a situation where a given behavior must not occur within a certain time interval [9]. High-level behaviors can be defined using punctual events (through predicate `happensAt`) or fluents, initially using predicates such as `initiatedAt`, `holdsFor`, etc.

Predicate	Intuitive meaning
`happensAt`(E, T)	Event E occurs at time T.
`initially`$(F = V)$	Fluent F has value V at time 0.
`holdsAt`$(F = V, T)$	Fluent F has value V at time T .
`holdsFor`$(F = V, I)$	I is the list of all maximum time intervals over which F has value V.
`initiatedAt`$(F = V, T)$	A time interval where F has value V starts at time T.
`terminatedAt`$(F = V, T)$	A time interval where F has value V ends at time T.
`union_all`(L, I)	I is the list of all maximum time intervals resulting from the union of all intervals in List L.
`intersect_all`(L, I)	I is the list of all maximum time intervals resulting from the intersection of all intervals in List L.
`relative_complement_all`(I', L, I)	I is list It of maximum time intervals minus each set of intervals in List L.

Table 1 - Main predicates of Event Calculus

The issue of behavior extraction and writing is studied in [13, 8]. Indeed, writing activities in the EC framework is tedious and error-prone, hence the idea of developing an automated process to generate definitions from temporal data. Thus, the authors use a learning method based on abductions and inductions to infer the behaviors to be recognized.

As explained in the introduction to this article, a major issue for stream reasoning is whether behaviors can be recognized or not in real time. The algorithm in this EC dialect uses a system query method: the reasoning is not performed gradually, but rather on demand, whenever a high-level activity is queried [10]. Thus, in order to perform an online analysis, it is necessary to constantly make queries: without a cache, this implies starting computations over again each time. Moreover, one of the principles of AC is that the order in which events occur further increases the complexity of the computation. In [28], Chittaro et al. introduce a version of EC called Cached Event Calculus (CEC), an implementation managing a cache memory to reduce the complexity of the process. However, CEC has no pre-emption mechanism and it accepts the processing of events with an earlier date than already processed events.

Therefore, recognition times increase gradually as low-level events occur, and after the computation becomes too time-expensive to keep up in real time. Artikis et al. attempted to address this issue in [12], where they introduce RTEC (Run-Time reasoning Event Calculus), an efficient YAProlog[1] implementation of their EC. Their program is also based on successive queries, with a cache memory preserving maximum intervals computed for the HoldsFor predicates of each fluent.

In addition, in order to address uncertainty, several stochastic approaches of EC have been developed. Artikis et al. extended the formalism of EC in [65] by means of Markov logical networks (MLN) [38], combining first-order logic with the probabilistic semantics of Markov networks. In [64] they also provided another extension, this time to probabilistic logical programming, using Prob-Log [49]. This way, they addressed the issue of incorrect low-level event detections by adding confidence indices to the events in the flow. The uncertainty here is limited to event uncertainty: in particular, it is not possible to handle incompletely specified behaviors. Moreover, the authors admit that online recognition is not possible in this formalism, which is corroborated in a recent work by Rincé et al. [63], who showed that, for a whole class of problems, the local search algorithms necessary in MLN and ProbLog perform poorly due to the structural characteristics of the problem.

Another approach for uncertainty handling is introduced in [14]. It is orthogonal to that in [64] in the sense that they may be combined. It relies upon the use of various event sources to determine their likelihood, with an auto-adaptation system based on the behavior recognition process itself: complex behavior definitions are written to identify the uncertainty domains and react accordingly: when the uncertainty becomes significant, the system may ignore events over a certain time interval, or even momentarily discard an event source. In [15], the authors add crowdsourcing to this framework, in order to make decisions when discrepancies between sources become significant.

ETALIS

Event-driven Transaction Logic Inference System (ETALIS)[2] [5, 2] is a CEP language, the syntax and semantics of which allow reasoning simultaneously on temporal assertions and on stable or evolving knowledge (rules, facts, ontologies, encyclopedic data, etc.). Its processing engine allows behaviors to be analyzed online.

[1] http://www.dcc.fc.up.pt/~vsc/Yap/

[2] available in open-source at http://code.google.com/p/etalis/

ETALIS is a logical programming language, and its syntax is defined by rules, the main constructs of which are shown in Table 2. The underlying time model is linear, dense, but countable (i.e., $\mathbb{Q}$), and low-level events may be instantaneous events, as well as events with a duration: events are dated by time intervals $[T_1, T_2]$ (with $T_1 = T_2$ in the case of instantaneous events). The language has a high expressiveness and contains:

- all of Allen's 13 interval relations;
- constraints on event properties;
- a rather limited notion of absence within the framework of the sequence;
- two precisely distinguished kinds of conjunction (in series and in parallel);
- recursive behavior definitions, allowing, for example, the definition of a function accumulating a value over a sequence of events.

A first formal declarative semantic approach is provided in [3, 5], where event patterns (i.e., behaviors) are defined by induction in the manner of model theory. A recognition is a couple (q_1, q_2), with q_1, $q_1 \in$ Q delimiting the necessary and sufficient time interval for the recognition (its support). Other than this support, in which they must all have been encompassed, information pertaining to the events triggering the recognition is not kept: there is no possibility of historization, and multiplicity is limited to the cases in which the supports of the multiple recognitions are distinct.

Constructs	Intuitive meaning
p WHERE t	Behavior p has been recognized and the term t is valued to true.
q	This corresponds to the absolute instant q (for any $q \in \mathbb{Q}$).
$(p).q$	Behavior p has been recognized and lasts exactly q, with $q \in \mathbb{Q}$.
p_1 SEQ p_2	Behavior p_1 is strictly followed (in time) by behavior p_2.
p_1 AND p_2	Behaviors p_1 and p_2 have been recognized, without any temporal constraint.
p_1 PAR p_2	Behaviors p_1 and p_2 have been recognized in parallel; i.e., they overlap in time.
p_1 OR p_2	Either one of both behaviors has been recognized.
p_1 EQUALS p_2	Both behaviors have been recognized over the exact same time interval.
p_1 MEETS p_2	Both p_1 and p_2 have been recognized, and the last recognition instant for p_1 exactly matches the first recognition instant of p_2.
p_1 DURING p_2	Behavior p_1 has been recognized within the recognition of p_2.
p_1 STARTS p_2	The recognition interval of p_1 is an initial segment of the recognition interval of p_2.
p_1 FINISHES p_2	The recognition interval of p_1 is a final segment of the recognition interval of p_2.
NOT$(p_1).[p_2, p_3]$	Behaviors p_2 and p_3 have been recognized in this order, without any occurrence of p_1 strictly contained between both in time.

Table 2 - Main constructs of ETALIS [3]

The ETALIS recognition system is implemented in Prolog. This implementation relies on an operational semantics defined using logic programming rules. The complex behaviors to be recognized are are broken up into intermediate events called *goals*. ETALIS compiles complex behaviors into a set of rules allowing Event-Driven Backward Chaining, which allows an online recognition process. Two types of rules result from the compilation:

- rules creating the goals to be recognized, in order to progress in the recognition of a complex behavior, in the form of an event and the expectation of another event: `goal`$\left(b^{[-,-]}, a^{[T_1,T_2]}, ie_1^{[-,-]}\right)$ means that when a (potentially complex) behavior a has been recognized over interval $[T_1, T_2]$, the system expects an event b to recognize behavior ie_1;
- rules creating intermediate events or event patterns: these check the database to determine whether a certain goal already exists, and, if this is the case, trigger the event that has been recognized by the goal: if `goal`$\left(b^{[T_3,T_4]}, a^{[T_1,T_2]}, ie_1^{[-,-]}\right)$ is in the database, then event $ie_1^{[T_1,T_4]}$ is triggered and propagated if it is an intermediate event, or is used to trigger an action if it is one of the complex behaviors sought.

Rules of the latter type also allow goals that are obsolete and not needed anymore to be suppressed from the database.

In other terms, the underlying recognition structure is a binary tree. However, the equivalence of both these semantics has not been proven.

As to recognition multiplicity, ETALIS allows the following event consumption policies: recent, chronicle, and "free" (i.e., without any restriction). However, the declarative aspect is lost with any policy other than free, which means that the rule-evaluation order ceases to be neutral.

The performance of ETALIS is also assessed on a so-called Fast Flower Delivery use case [43].

ETALIS also handles delayed events [44] through two additional rule types:

- `goal_out`$\left(a^{[-,-]}, b^{[T_3,T_4]}, ie_1^{[-,-]}\right)$, expressing that Event b has been received and that an Event a having occurred *before* b is expected to finalize the recognition of ie_1.
- `if goal_out(...)` and $T_2 < T_3$, expressing that if an event a indeed occurs at $T_2 < T_3$, then event $ie_1^{[T_1,T_4]}$ is triggered.

This algorithm does not have adverse effects on the efficiency of the recognition for events occurring on time. However, it requires a specific procedure to free memory by suppressing `goal_out` rules after a while. Reference [44] explains that, due to practical reasons (probably a matter of recognition efficiency by preventing rule overload) this functionality has not been implemented: therefore, multiplicity is lost. To handle delayed events in the case of an absence, ETALIS also allows the handling of *revised* events [4]: new rev goals are introduced to suppress revised goals.

Chronicles

Chronicles are a family of formal languages developed to formally describe an event signature and, as such, provide a framework for CEP.

Dousson's chronicles

A chronicle language was introduced in [45], and developed mainly by Dousson *et al.* [39, 40, 41]. Within this framework, a chronicle is somehow a partial order of observable events in a certain context. Together with the language comes an efficient online recognition process enabling the analysis of a flow of timestamped events that do not necessarily arrive in their order of occurrence, with the possibility of triggering actions or producing events at a date defined in relation to the dates of the events having caused the recognition.

In [39]: a *chronicle model* is presented as a set of formulae or temporal schemas defining how the association of several *observable* events can lead to a new *deduced* event, and a set of constraints is given. A chronicle is thus a set of events together with contextual and temporal constraints. In this approach, the time model is discrete, totally ordered, and precise enough to take into account the observed events. In more recent works, Dousson *et al.* [41] associate attributes with events that can change their values: chronicles are represented by constraint graphs, with events as nodes, and the edges are labelled with integer intervals that represent time constraints.

This framework has also been used and adapted by Subias and Boufaied [18, 46] to various contexts, but always with discrete time. In his PhD work [42], Vu Dũ'o'ng applies Dousson's work to telecommunication network diagnosis through alarm correlation.

Here is a broad idea of how chronicles are expressed in Dousson's formalism. They are multi-sets of events with additional constraints expressed as time intervals (which may contain negative values, meaning that the events occur in reverse order than that specified) that must be fulfilled by pairs of events. For instance (see Figure 1), a chronicle may be $ABCD$ where the interval between A and C must be within $[-3, 2]$, the interval between A and D must be within $[4, 6]$, and the interval between D and B must be within $[-1, 4]$. Each event in the multi-set has to be mapped exactly once to an event of the flow, and the mapping must be consistent with the constraints.

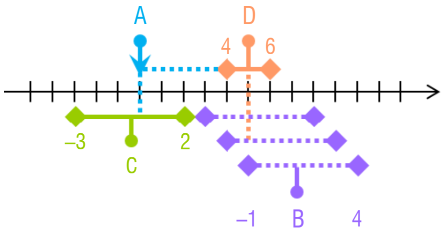


Figure 1 - An example of Dousson's Chronicle: $C \xleftarrow{[-3,2]} A \xrightarrow{[4,6]} D \xrightarrow{[-1,4]} B$

The expressivity of Dousson's formalism is rather low. In particular, the fact that chronicles can combine various intermediate patterns that may or may not share elementary events cannot be expressed. For instance, it is not possible to write a chronicle of the form $(ABC)\&(DBE)$, that would be recognized if an event B occurs between A and C, and an event B (possibly the same, but not necessarily) occurs between D and E. This cannot be expressed in Dousson's formalism, where it is necessary to specify when designing the chronicle whether there is a single B or two distinct B. In addition, absence is difficult to account for in Dousson's work. Moreover, the issue above regarding shared events between subchronicles also applies for absence.

ONERA chronicles

In the formalism of ONERA chronicles, events are represented as ordered pairs comprising an event name and a real number (its occurrence *date*). The underlying time model is linear and continuous. These events can be endowed with information, called *attributes*, which are ordered pairs of an *attribute name* and a *value*.

Attributes are a very expressive feature of the language: an event can have any number of attributes and, given that recognitions are built upon events, new attributes can be computed, named and associated with recognitions, so as to be used at a higher level.

The chronicle language is built by induction, using, among others, four constructs expressing the sequence, the conjunction, and the disjunction of two behaviors, as well as the absence of a given behavior during another behavior. These constructs have been presented in [20, 21].

In addition to this, ONERA chronicles express all of Allen's 13 relations, as well as constraints on the durations of behaviors and a few additional constructs, such as a change of state and a derived event associated with the instant of recognition completion. Moreover, the chronicle language allows reasoning on *event attributes*: a predicate can express desired constraints on manipulated attributes. Constraints on attributes and attribute creations can be added at each level, and a notion of the evaluation context allows attributes to be handled properly in constructs where not all subchronicles are present in the recognition of a chronicle (typically, absence and disjunction).

The notion of chronicle recognition, originally [20, 21] based on a notion of a set of events leading to it, has therefore been replaced by a tree-based notion: indeed, an event model set does not retain the information specifying which event led to the recognition of which sub-chronicle, which becomes an issue when properties are expressed over event attributes. Consider, for example, Chronicle $C = (A\ B)\ \&\ A$. Some recognitions of C may be due to two distinct events a (a denotes an instance of A). In a set formalism, these two events are undistinguishable, so it is impossible to determine which *a* led to the recognition of sub-chronicle $A\ B$, and which led to the recognition of the single A. Not only is this information lost, but this also affects the combinatorics, since the number of recognitions depends on this information: if this information is kept, two events a lead to two different recognitions of C depending on the distribution[3].

Continuous time is managed through the use of a look-ahead function $T_C(\varphi, d)$ providing a future date until which the system does not need to be re-examined, since the recognition set would not have changed until then. Indeed, the systems considered here are asynchronous, and this function provides the next time when it will be necessary to check for the completion of a given chronicle with delays.

More details on the theoretical framework of chronicle recognition are presented in [60], but their general form is that of a triplet (C, P, f), where:

- $C \in \mathfrak{X}$ is a *chronicle formula* (see below);
- $P \in \mathfrak{S}$ is a predicate symbol;
- $f \in \mathfrak{T}(\mathfrak{P}, \mathcal{V})$ is an attribute transformation.

[3] Note that there is also one additional recognition for each event a leading to the recognitions of both sub-chronicles.

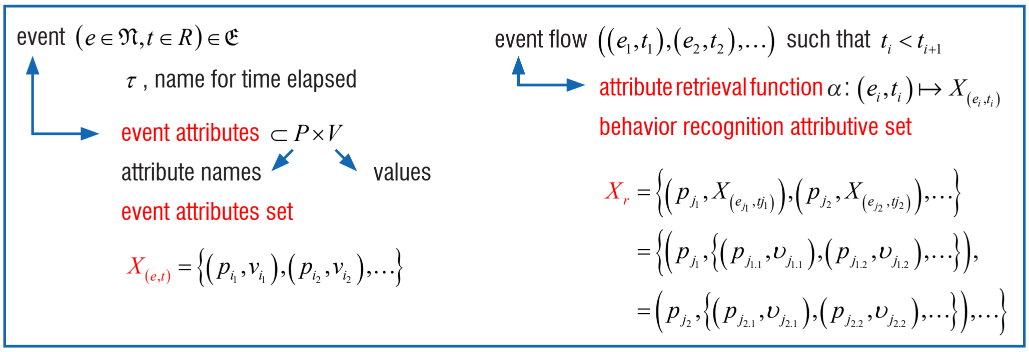

event $(e \in \mathfrak{N}, t \in R) \in \mathfrak{E}$
τ , name for time elapsed
event attributes $\subset P \times V$
attribute names
values
event attributes set
$X_{(e,t)} = \{(p_{i_1}, v_{i_1}), (p_{i_2}, v_{i_2}), \dots\}$
event flow $((e_1,t_1),(e_2,t_2),\dots)$ such that $t_i < t_{i+1}$
attribute retrieval function $\alpha : (e_i,t_i) \mapsto X_{(e_i,t_i)}$
behavior recognition attributive set
$X_r = \{(p_{j_1}, X_{(e_{j_1}, tj_1)}), (p_{j_2}, X_{(e_{j_2}, tj_2)}), \dots\}$
$= \{(p_{j_1}, \{(p_{j_{1.1}}, \upsilon_{j_{1.1}}), (p_{j_{1.2}}, \upsilon_{j_{1.2}}), \dots\}),$
$= (p_{j_2}, \{(p_{j_{2.1}}, \upsilon_{j_{2.1}}), (p_{j_{2.2}}, \upsilon_{j_{2.2}}), \dots\}), \dots\}$

Figure 2 - Events and attributes for chronicles

$\mathfrak{X}$ is inductively defined together with two notions of *contexts*, which are functions from $\mathfrak{X}$ to $\mathfrak{P}$ (an evaluation context $\mathcal{C}_e$ and a resulting context $\mathcal{C}_r$):

simple event: If $A \in \mathfrak{N}$, then $(\mathbf{A}, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(A,P,f) = \{\lozenge\}$, et $\mathcal{C}_r(A,P,f) = \mathcal{C}_e(A,P,f)$;

sequence: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathbf{C_1}\ \mathbf{C_2}, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1\ C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, et $\mathcal{C}_r(C_1\ C_2, P, f) = \mathcal{C}_e(C_1\ C_2, P, f)$;

conjunction: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathbf{C_1} \,\&\, \mathbf{C_2}, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\&\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, $\mathcal{C}_r(C_1 \,\&\, C_2, P, f) = \mathcal{C}_e(C_1 \,\&\, C_2, P, f)$;

disjunction: $(\mathbf{C_1} \| \mathbf{C_2}, P, f) \in \mathfrak{X}$,$\mathcal{C}_e(C_1 \| C_2, P, f) = \mathcal{C}_r(C_1) \cap \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \| C_2, P, f) = \mathcal{C}_e(C_1 \| C_2, P, f)$;

absence: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $((\mathrm{C}_1) - [\mathrm{C}_2], P, f) \in \mathfrak{X}$, $\mathcal{C}_e((C_1) - [C_2], P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r((C_1) - [C_2], P, f) = \mathcal{C}_r(C_1)$;

meets: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 \,\text{meets}\, \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{meets}\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \,\text{meets}\, C_2, P, f) = \mathcal{C}_e(C_1 \,\text{meets}\, C_2, P, f)$;

overlaps: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 \,\text{overlaps}\, \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{overlaps}\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \,\text{overlaps}\, C_2, P, f) = \mathcal{C}_e(C_1 \,\text{overlaps}\, C_2, P, f)$;

starts: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 \,\text{starts}\, \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{starts}\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \,\text{starts}\, C_2, P, f) = \mathcal{C}_e(C_1 \,\text{starts}\, C_2, P, f)$;

during: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 \,\text{during}\, \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{during}\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \,\text{during}\, C_2, P, f) = \mathcal{C}_e(C_1 \,\text{during}\, C_2, P, f)$;

finishes: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 \,\text{finishes}\, \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{finishes}\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \,\text{finishes}\, C_2, P, f) = \mathcal{C}_e(C_1 \,\text{finishes}\, C_2, P, f)$;

equals: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 \,\text{equals}\, \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{equals}\, C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 \,\text{equals}\, C_2, P, f) = \mathcal{C}_e(C_1 \,\text{equals}\, C_2, P, f)$;

lasts δ: If $\delta \in \mathbb{R}_*^+$, then $(\mathrm{C}_1 \,\text{lasts}\, \delta, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{lasts}\, \delta, P, f) = \mathcal{C}_r(C_1)$, and $\mathcal{C}_r(C_1 \,\text{lasts}\, \delta, P, f) = \mathcal{C}_e(C_1 \,\text{lasts}\, \delta, P, f)$;

at least δ: If $\delta \in \mathbb{R}_*^+$, then $(\mathrm{C}_1 \,\text{atleast}\, \delta, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{atleast}\, \delta, P, f) = \mathcal{C}_r(C_1)$, and $\mathcal{C}_r(C_1 \,\text{atleast}\, \delta, P, f) = \mathcal{C}_e(C_1 \,\text{atleast}\, \delta, P, f)$;

at most δ: If $\delta \in \mathbb{R}_*^+$, then $(\mathrm{C}_1 \,\text{atmost}\, \delta, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{atmost}\, \delta, P, f) = \mathcal{C}_r(C_1)$, and $\mathcal{C}_r(C_1 \,\text{atmost}\, \delta, P, f) = \mathcal{C}_e(C_1 \,\text{atmost}\, \delta, P, f)$;

then δ: If $\delta \in \mathbb{R}_*^+$, then $(\mathrm{C}_1 \,\text{then}\, \delta, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \,\text{then}\, \delta, P, f) = \mathcal{C}_r(C_1)$, and $\mathcal{C}_r(C_1 \,\text{then}\, \delta, P, f) = \mathcal{C}_e(C_1 \,\text{then}\, \delta, P, f)$;

naming: If $x \in \mathfrak{P} \setminus \{\lozenge\}$, then $(\mathrm{C}_1 \to \mathrm{x}, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 \to x, P, f) = \mathcal{C}_r(C_1)$, $\mathcal{C}_r(C_1 \to x, P, f) = \{x, \lozenge\}$;

cut: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 ! \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 ! C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 ! C_2, P, f) = \mathcal{C}_e(C_1 ! C_2, P, f)$;

change of state: If $\mathcal{C}_e(C_1) \cap \mathcal{C}_e(C_2) = \{\lozenge\}$, then $(\mathrm{C}_1 !! \mathrm{C}_2, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(C_1 !! C_2, P, f) = \mathcal{C}_r(C_1) \cup \mathcal{C}_r(C_2)$, and $\mathcal{C}_r(C_1 !! C_2, P, f) = \mathcal{C}_e(C_1 !! C_2, P, f)$;

recognition event: $(@\mathrm{C}_1, P, f) \in \mathfrak{X}$, $\mathcal{C}_e(@C_1, P, f) = \mathcal{C}_r(C_1)$, and $\mathcal{C}_r(@C_1, P, f) = \mathcal{C}_r(C_1)$.

Chronicles allow for multiplicity and historization of recognitions. They also allow the gradual recognition of behaviors in real time, as events flow. The need for multiplicity and historization prevents the use of simple finite-state automata (see [17]), but a first recognition tool called Chronicle Recognition System (CRS/ONERA) was developed at ONERA in the late 1990s in [19], based on duplicating automata so as to comply with performance and inter- operability requirements. A colored Petri net model was also developed, implementing chronicle recognition for the initial operators – sequence, conjunction, disjunction, and absence – and its adequacy has been proven (see [23, 21]). A new recognition tool has also been developed in the form of a C++ library called Chronicle Recognition Library (CRL)[4], which can be easily used for real-world critical applications. Its algorithms are directly based on the formal semantics of the chronicle language, therefore the recognitions produced by CRL are considered t be adequate by construction. The efficiency of CRL is also ensured by a validity window mechanism that eliminates obsolete initiated recognitions after a time specified by the user. Applications of CRL are presented in [24, 22].

Other approaches

Other families of approaches to stream reasoning are at the two far ends of the spectrum of SR techniques: DSMS and KRR. We illustrate them in this section through two representative examples: CQL (a DSMS framework) and LARS (based on KRR and answer set programming).

Continuous Query Language

Continuous Query Language (CQL) [6, 7] is a language based on the database query language SQL, extended with streams as additional data sources. In CQL, a *stream* is viewed as a bag of elements in the form $\langle c,t \rangle$, where c is a tuple and t is a timestamp; a *relation* maps timestamps to bags of tuples. To make these concepts compatible, the operational semantics of CQL relies on three kinds of operators:

- Relation-to-Relation operators contain usual SQL operators to manipulate relations;
- Stream-to-Relation operators apply window functions to the input stream to create a relation for recent tuples;
- Relation-to-Stream operators translate back a relation into a stream for the output of continuous queries.

There are interesting parallels between CQL and our approach; in particular, the fact that CQL has operational semantics where evaluation is performed stepwise as a query is evaluated. However, being based on SQL-like queries, CQL handles the stream by filtering, joining and aggregating data in a deterministic way, and does not allow for abstractions, constraints, complex negation, and non-determinism.

Compared to CQL, our approach allows additional abstraction and reasoning features, including the absence (which is a form of complex negation) and temporal modalities. Both features are particularly important in our approach, as evidenced by the various levels of reasoning in aerospace case studies that we have treated (cf. Table 5.9 in [60]): indeed, intermediate (i.e., Level-2) and interest (i.e., Level-3) chronicles contain temporal modalities (e.g., at least, @, and !! in Level-2 chronicles, as well as in Level-3 chronicle NoClearanceToTakeOff(*ID*), as well as absences with additional correlations to event attributes; e.g., Level-3 chronicle NoFrequencyToTakeOff(*ID*).

LARS

LARS [16] consists of two languages: *LARS formulae* extend propositional logic with generic window operators and additional controls to handle temporal information, and, on top of this, *LARS programs* extend Answer Set Programming (ASP) with rich stream-reasoning capabilities. It is aimed at targeting AI applications in a streaming context, such as diagnosis, configuration, or planning.

Fragments of LARS have been implemented in several experimental prototypes [16], based on different realization principles, but they either lack efficiency or are restricted to specific LARS programs, in particular with restrictions on the use of negations.

In contrast to chronicles, LARS semantics is based on time points. Nevertheless, as stated in [16], when comparing LARS and ETALIS, it is possible to represent intervals in LARS and thus partially capture the notion. However, this representation is unable to take into account overlapping intervals (for a same formula): indeed, LARS assigns atoms to a single timeline by an evaluation function, so it can encode intervals only by assigning atoms to consecutive time points. Adjacent or overlapping intervals for the same atom cannot be distinguished and, worse still, merge into a single larger interval, which is incompatible with our objective of multiple recognitions.

An application to aeronautics

To illustrate the interest of these techniques in the aeronautic field, we present an example of a hypothetical unmanned aircraft inserted into general air traffic, as described in Figure 3.

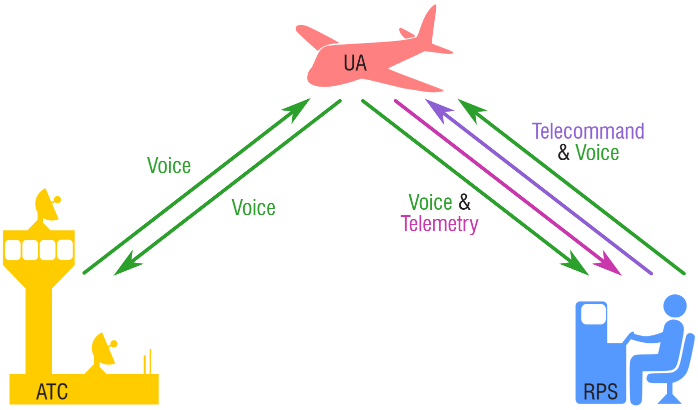


Figure 3 - Schematic representation of the three-agent system

This is a global problem that raises many interesting issues, and implies interactions between the aircraft, its pilot (on the ground) and Air Traffic Control (ATC). We focus here on a potential hazard, which is the loss of the telecommand (TC) link, meaning that the pilot is unable to transmit orders to the unmanned aircraft. In such a case, there has to be a predetermined course of action (e.g., return to base, pursue current route, land at the nearest airport, etc., – which one exactly is not relevant here). If such a loss occurs, it is obviously important that all three agents (ATC, pilot, and aircraft) share the same understanding of the situation, so that, in particular, both of the human actors act consistently.

[4] CRL has been deposited at the French Agency for Program Protection and is available under the GNU LGPL license.

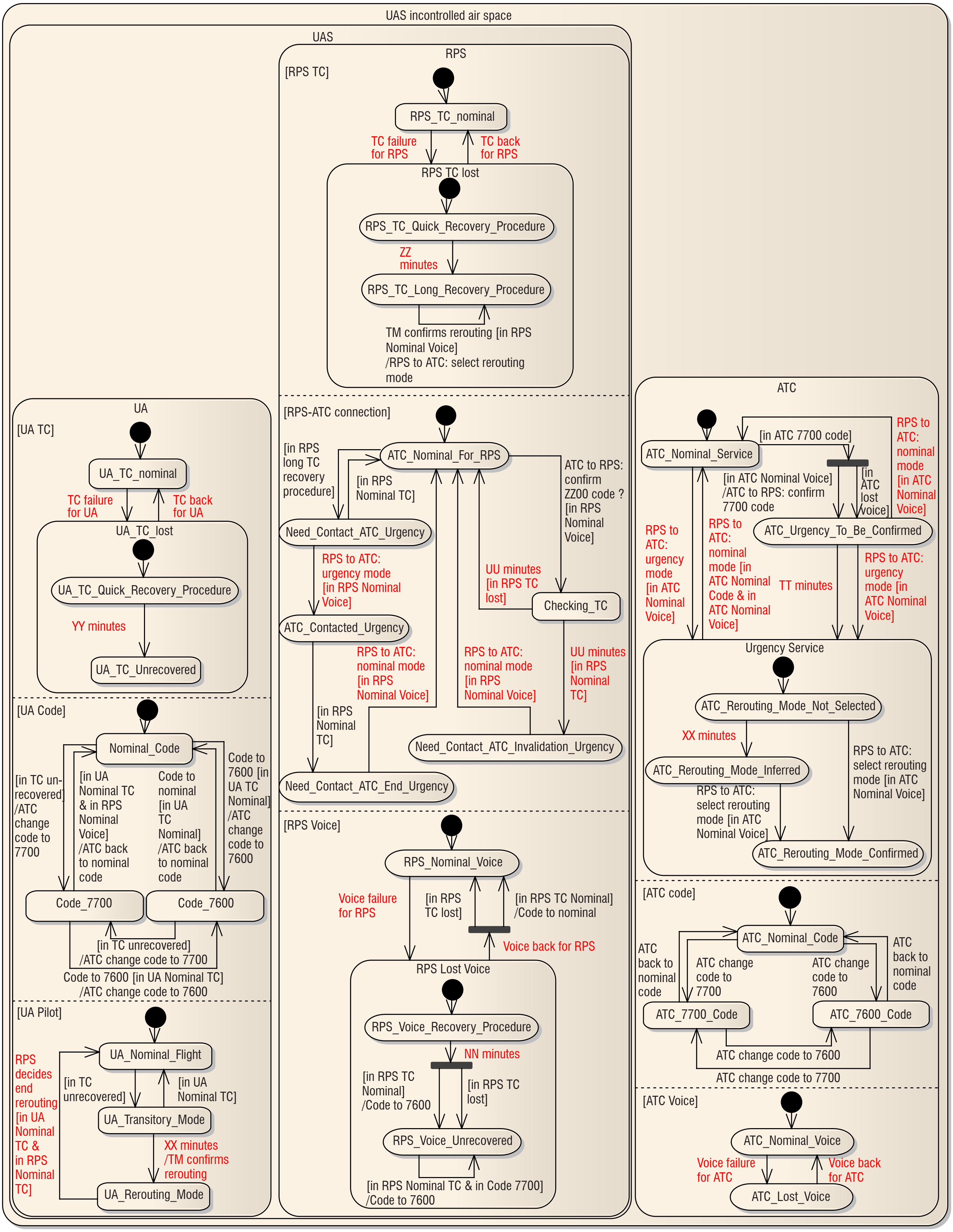

UAS incontrolled air space
UAS
RPS
[RPS TC]
RPS_TC_nominal
TC failure for RPS
TC back for RPS
RPS TC lost
RPS_TC_Quick_Recovery_Procedure
ZZ minutes
RPS_TC_Long_Recovery_Procedure
TM confirms rerouting [in RPS Nominal Voice] /RPS to ATC: select rerouting mode
[RPS-ATC connection]
ATC_Nominal_For_RPS
[in RPS long TC recovery procedure]
[in RPS Nominal TC]
ATC to RPS: confirm ZZ00 code ? [in RPS Nominal Voice]
Need_Contact_ATC_Urgency
RPS to ATC: urgency mode [in RPS Nominal Voice]
UU minutes [in RPS TC lost]
Checking_TC
ATC_Contacted_Urgency
RPS to ATC: nominal mode [in RPS Nominal Voice]
RPS to ATC: nominal mode [in RPS Nominal Voice]
UU minutes [in RPS Nominal TC]
[in RPS Nominal TC]
Need_Contact_ATC_Invalidation_Urgency
Need_Contact_ATC_End_Urgency
[RPS Voice]
RPS_Nominal_Voice
Voice failure for RPS
[in RPS TC lost]
[in RPS TC Nominal] /Code to nominal
Voice back for RPS
RPS Lost Voice
RPS_Voice_Recovery_Procedure
NN minutes
[in RPS TC Nominal] /Code to 7600
[in RPS TC lost]
RPS_Voice_Unrecovered
[in RPS Nominal TC & in Code 7700] /Code to 7600
UA
[UA TC]
UA_TC_nominal
TC failure for UA
TC back for UA
UA_TC_lost
UA_TC_Quick_Recovery_Procedure
YY minutes
UA_TC_Unrecovered
[UA Code]
Nominal_Code
[in TC un-recovered] /ATC change code to 7700
[in UA Nominal TC & in RPS Nominal Voice] /ATC back to nominal code
Code to nominal [in UA TC Nominal] /ATC back to nominal code
Code to 7600 [in UA TC Nominal] /ATC change code to 7600
Code_7700
Code_7600
[in TC unrecovered] /ATC change code to 7700
Code to 7600 [in UA Nominal TC] /ATC change code to 7600
[UA Pilot]
UA_Nominal_Flight
RPS decides end rerouting [in UA Nominal TC & in RPS Nominal TC]
[in TC unrecovered]
[in UA Nominal TC]
UA_Transitory_Mode
XX minutes /TM confirms rerouting
UA_Rerouting_Mode
ATC
ATC_Nominal_Service
[in ATC 7700 code]
RPS to ATC: nominal mode [in ATC Nominal Voice]
[in ATC Nominal Voice] /ATC to RPS: confirm 7700 code
[in ATC lost voice]
RPS to ATC: urgency mode [in ATC Nominal Voice]
RPS to ATC: nominal mode [in ATC Nominal Code & in ATC Nominal Voice]
ATC_Urgency_To_Be_Confirmed
TT minutes
RPS to ATC: urgency mode [in ATC Nominal Voice]
Urgency Service
ATC_Rerouting_Mode_Not_Selected
XX minutes
ATC_Rerouting_Mode_Inferred
RPS to ATC: select rerouting mode [in ATC Nominal Voice]
RPS to ATC: select rerouting mode [in ATC Nominal Voice]
ATC_Rerouting_Mode_Confirmed
[ATC code]
ATC_Nominal_Code
ATC back to nominal code
ATC change code to 7700
ATC change code to 7600
ATC back to nominal code
ATC_7700_Code
ATC_7600_Code
ATC change code to 7600
ATC change code to 7700
[ATC Voice]
ATC_Nominal_Voice
Voice failure for ATC
Voice back for ATC
ATC_Lost_Voice

Figure 4 - State diagram of telecommand loss

However, each actor has access only to a partial subset of information, from which they deduce the status of the TC link. Hence, their reasoning can be modelled through chronicles and this can be used to detect inconsistencies.

Here, low-level observable events will schematically be the actions performed by each agent on the system or their changes of state, which are modelled according the state diagram in Figure 4 representing the protocol followed by each agent. The diagram is distributed between the three agents: UA (Unmanned Aircraft), RPS (Remote Pilot Station), ATC. Each agent is broken down into sub-systems. A sub-system represents a functionality controlled by the agent, or a specific knowledge that it may possess about the overall system situation; e.g., the RPS Voice sub-system describes whether the RPS is aware of a potential radio communication loss. Each sub-system is a set of states that have to be followed in a specific order regarding the system evolution. Arrows between states describe this order and the necessary conditions to trigger a state change, and may possibly be associated with a specific action to be performed by the agent. This information is labelled on the arrows in three different parts: *event*, *condition* and *action,* and written `event [condition]/action`.

The formalism can then be used to detect undesired behaviors, such as:

- *Incoherent ATC Voice*: the transponder code emitted by the UA starts indicating code 7600 to Air Traffic Control, which means that there is a voice failure, but the controller has not realized this, and this is expressed by the fact that the diagram does not switch to `ATC Lost Voice`.
- *Incoherent flight mode UA/ATC*: after a fault that has been solved, the UA has switched back to a nominal flight but ATC remains in an urgency service.

Once one of these behaviors is detected, its origins have to be determined. If the cause is due to faulty behavioral guidelines, then the model has to be corrected, and, otherwise, if the source is human, it should be planned to trigger alarms warning the pilot and/or the air traffic controller of the situation.

These behaviors are represented by the following chronicles:

- *Incoherent ATC Voice*

(`to_ATC_Nominal_Code to_ATC_7600_Code` then 5) – [`to_ATC_Lost_Voice`]

- *Incoherent flight mode UA/ATC*

(`from_UA_Nominal_Flight`
((`to_UA_Nominal_Flight` then 10) – [`from_UA_Nominal_Flight`]))
–[`to_ATC_Nominal_Service`]

The possibility, once a hazardous behavior has been recognized, of determining its origin is provided by the properties of the chronicle framework.

Conclusion

While no single Stream-Reasoning approach can claim to be able to tackle all possible uses of Stream Reasoning, the overview of methods presented in this paper shows that handling a rapidly changing dynamic dataflow with elaborate reasoning is now feasible. These methods find applications in a very broad spectrum, which includes:

- the detection of inconsistencies between pilot and air traffic control in a scenario where an unmanned aircraft may lose its telecommand [24] (see above), with chronicles;
- the supervision and analysis of hazardous situations using an unmanned aircraft to assist police services [47, 35, 36, 34], with chronicles;
- various medical applications, including heart monitoring [30, 25, 62, 37, 61, 27], using chronicles together with learning techniques;
- management of alarms for the detection of cyber-intrusions [56], with Dousson's chronicles;
- Web-service diagnostics [59, 31, 52], with chronicles;
- public transportation quality assessment [48, 68], with EC (project PRONTO);
- video-surveillance [66, 13, 11, 9], with EC (project CAVIAR);
- social media analysis [66];
- assistance in decision-making during air combat [29], with chronicles;
- network supervision and monitoring management[67], with chronicles;
- supervision of a gas turbine in a petrochemical plant [55] and supervision of a milk factory [53], with chronicles;
- characterization of human activities [32], with chronicles ■

References

[1] J. F. ALLEN - *Maintaining Knowledge about Temporal Intervals*. Commun. ACM, Pages 832-843, 1983.

[2] D. ANICIC - *Event Processing and Stream Reasoning with ETALIS*. PhD thesis, Karlsruhe Institute of Technology, 2011.

[3] D. ANICIC, P. FODOR, S. RUDOLPH, R. STÜHMER, N. STOJANOVIC, R. STUDER - *A Rule-Based Language for Complex Event Processing and Reasoning.* Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems (RR 2010), Pages 42-57. Springer, 2010.

[4] D. ANICIC, S. RUDOLPH, P. FODOR, N. STOJANOVIC - *Retractable Complex Event Processing and Stream Reasoning*. Proceedings of the 5th International Conference on Rule-Based Reasoning, Programming, and Applications, Pages 122-137. Springer, 2011.

[5] D. ANICIC, S. RUDOLPH, P. FODOR, N. STOJANOVIC - *Real-Time Complex Event Recognition and Reasoning – A Logic Programming Approach*. Applied Artificial Intelligence, 26(1-2):6-57, 2012.

[6] A. ARASU, S. BABU, J. WIDOM - *Cql: A Language for Continuous Queries over Streams and Relations.* International Workshop on Database Programming Languages, Pages 1-19. Springer, 2003.

[7] A. ARASU, S. BABU, J. WIDOM - *The cql Continuous Query Language: Semantic Foundations and Query Execution*. The VLDB Journal, 15(2):121-142, 2006.

[8] A. ARTIKIS, O. ETZION, Z. FELDMAN, F. FOURNIER - *Event Processing under Uncertainty*. Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, Pages 32-43. ACM, 2012.

[9] A. ARTIKIS, G. PALIOURAS - *Behaviour Recognition Using the Event Calculus*. Artificial Intelligence Applications and Innovations III, Pages 469-478. Springer, 2009.

[10] A. ARTIKIS, G. PALIOURAS, F. PORTET, A. SKARLATIDIS - *Logic-Based Representation, Reasoning and Machine Learning for Event Recognition.* Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems. ACM, 2010.

[11] A. ARTIKIS, M. SERGOT, G. PALIOURAS - *A Logic Programming Approach to Activity Recognition*. Proceedings of the 2nd ACM International Workshop on Events in Multimedia, Pages 3-8. ACM, 2010.

[12] A. ARTIKIS, M. SERGOT, G. PALIOURAS - *Run-Time Composite Event Recognition*. Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, Pages 69-80. ACM, 2012.

[13] A. ARTIKIS, A. SKARLATIDIS, G. PALIOURAS - *Behaviour Recognition from Video Content: A Logic Programming Approach*. International Journal on Artificial Intelligence Tools, 19(02):193-209, 2010.

[14] A. ARTIKIS, M. WEIDLICH, A. GAL, V. KALOGERAKI, D. GUNOPULOS - *Self-Adaptive Event Recognition for Intelligent Transport Management.* Big Data, 2013 IEEE International Conference on, Pages 319-325. IEEE, 2013.

[15] A. ARTIKIS, M. WEIDLICH, F. SCHNITZLER, I. BOUTSIS, T. LIEBIG, N. PIATKOWSKI, C. BOCKERMANN, K. MORIK, V. KALOGERAKI, J. MARECEK, *et al.* - *Heterogeneous Stream Processing and Crowdsourcing for Urban Traffic Management.* Proceedings of the 17th International Conference on Extending Database Technology (EDBT 2014). Athens, Greece, 2014.

[16] H. BECK, M. DAO-TRAN, T. EITER - *LARS: A Logic-Based Framework for Analytic Reasoning over Streams*. Artificial Intelligence, Pages 16-70, 2018.

[17] O. BERTRAND, P. CARLE, C. CHOPPY - *Chronicle Modelling Using Automata and Coloured Petri Nets.* 18th International Workshop on Principles of Diagnosis (DX-07), Pages 229-234, 2007.

[18] A. BOUFAIED, A. SUBIAS, M. COMBACAU - *Détection distribuée par reconnaissance floue de chroniques*. Journal Européen des Systèmes Automatisés, 40(2):233-259, 2006.

[19] P. CARLE, P. BENHAMOU, F.-X. DOLBEAU, M. ORNATO - *La reconnaissance d'intentions comme dynamique des organisations*. 6èmes Journées Francophones pour l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA'98), 1998.

[20] P. CARLE, C. CHOPPY, R. KERVARC - *Behaviour Recognition using Chronicles*. Proc. 5th IEEE International Symposium on Theoretical Aspects of Software Engineering, Pages 100-107, 2011.

[21] P. CARLE, C. CHOPPY, R. KERVARC, A. PIEL - *Behavioural Analysis for Distributed Simulations*. 19th Asia-Pacific Software Engineering Conference (APSEC), 2012.

[22] P. CARLE, C. CHOPPY, R. KERVARC, A. PIEL - *Handling Breakdowns in Unmanned Aircraft Systems.* 18th International Symposium on Formal Methods (FM) - Doctoral Symposium, 2012.

[23] P. CARLE, C. CHOPPY, R. KERVARC, A. PIEL - *A Formal Coloured Petri Net Model for Hazard Detection in Large Event Flows.* 20th Asia-Pacific Software Engineering Conference (APSEC), 2013.

[24] P. CARLE, C. CHOPPY, R. KERVARC, A. PIEL - *Safety of Unmanned Aircraft Systems Facing Multiple Breakdowns*. 1st French Singaporean Workshop on Formal Methods and Applications (FSFMA), 2013.

[25] G. CARRAULT, M.-O. CORDIER, R. QUINIOU, F. WANG - *Temporal Abstraction and Inductive Logic Programming for Arrhythmia Recognition From Electrocardiograms.* Artificial Intelligence in Medicine, 28(3):231-263, 2003.

[26] Z. CHAOCHEN, C. A. R. HOARE, A. P. RAVN - *A Calculus of Durations*. Information Processing Letters, 40(5):269-276, 1991.

[27] L. CHITTARO, M. DOJAT - *Using a General Theory of Time and Change in Patient Monitoring: Experiment and Evaluation*. Computers in Biology and Medicine, 27(5):435-452, 1997.

[28] L. CHITTARO, A. MONTANARI - *Efficient Temporal Reasoning in the Cached Event Calculus.* Computational Intelligence, 12(3):359-382, 1996.

[29] S. CORADESCHI, T. VIDAL - *Accounting for Temporal Evolutions in Highly Reactive Decision-Making*. Fifth IEEE International Workshop on Temporal Representation and Reasoning, Pages 3-10, 1998.

[30] M.-O. CORDIER, C. DOUSSON - *Alarm Driven Monitoring Based on Chronicles.* 4th SafeProcess, Pages 286-291, 2000.

[31] M.-O. CORDIER, X. LE GUILLOU, S. ROBIN, L. ROZÉ, T. VIDAL - *Distributed Chronicles for On-line Diagnosis of Web Services*. 18th International Workshop on Principles of Diagnosis (DX-07), Pages 37-44, 2007.

[32] D. CRAM, B. MATHERN, A. MILLE - *A Complete Chronicle Discovery Approach: Application to Activity Analysis.* Expert Systems, 29(4):321-346, 2012.

[33] G. CUGOLA, A. MARGARA - *Processing Flows of Information: From Data Stream to Complex Event Processing.* ACM Computing Surveys (CSUR), 44(3):15, 2012.

[34] P. DOHERTY, G. GRANLUND, K. KUCHCINSKI, E. SANDEWALL, K. NORDBERG, E. SKARMAN, J. WIKLUND - *The Witas Unmanned Aerial Vehicle Project*. ECAI, Pages 747-755, 2000.

[35] P. DOHERTY, J. KVARNSTRÖM, F. HEINTZ - *A Temporal Logic-Based Planning and Execution Monitoring Framework for Unmanned Aircraft Systems.* 6th International Conference on Recent Advances in Intrusion Detection (RAID'03), 2009.

[36] P. DOHERTY, J. KVARNSTRÖM, F. HEINTZ - *A Temporal Logic-Based Planning and Execution Monitoring Framework for Unmanned Aircraft Systems*. Autonomous Agents and Multi-Agent Systems, Pages 332-377, 2009.

[37] M. DOJAT - *Realistic Model for Temporal Reasoning in Real-Time Patient Monitoring*. Applied Artificial Intelligence, 10(2):121-144, 1996.

[38] P. DOMINGOS, D. LOWD - *Markov Logic: An Interface Layer for Artificial Intelligence.* Synthesis Lectures on Artificial Intelligence and Machine Learning, 3(1):1-155, 2009.

[39] C. DOUSSON, P. GABORIT, M. GHALLAB - *Situation Recognition: Representation and Algorithms.* International Joint Conference on Artificial Intelligence (IJCAI), Pages 166-172, 1993.

[40] C. DOUSSON - *Extending and Unifying Chronicle Representation with Event Counters*. Proceedings of the 15th European Conference on Artificial Intelligence, ECAI'2002, Lyon, France, July 2002, Pages 257-261, 2002.

[41] C. DOUSSON, P. LE MAIGAT - *Chronicle Recognition Improvement Using Temporal Focusing and Hierarchization*. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Pages 324-329, 2007.

[42] T. VU DU'O'NG - *Découverte de chroniques à partir de journaux d'alarmes, application à la supervision de réseaux de télécommunications.* PhD thesis, Institut National Polytechnique de Toulouse, 2001.

[43] O. ETZION, P. NIBLETT - *Event Processing in Action*. Manning Publications Co., 2010.

[44] P. FODOR, D. ANICIC, S. RUDOLPH - *Results on Out-of-Order Event Processing.* Proceedings of the 13th International Conference on Practical Aspects of Declarative Languages, Pages 220-234. Springer, 2011.

[45] M. GHALLAB - *On Chronicles: Representation, On-Line Recognition and Learning.* KR, Pages 597-606, 1996.

[46] H.-E. GOUGAM, A. SUBIAS, Y. PENCOLÉ - *Timed Diagnosability Analysis Based on Chronicles.* 8th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes SAFEPROCESS'2012, Pages 1256-1261, 2012.

[47] F. HEINTZ - *Chronicle Recognition in the WITAS UAV Project, a Preliminary Report*. Swedish AI Society Workshop (SAIS2001), 2001.

[48] P. KAARELA, M. VARJOLA, L. P. J. J. NOLDUS, A. ARTIKIS - *Pronto: Support for Real-Time Decision Making*. Proceedings of the 5th ACM International Conference on Distributed Event-Based System, Pages 11-14. ACM, 2011.

[49] A. KIMMIG, B. DEMOEN, L. DE RAEDT, V. SANTOS COSTA, R. ROCHA - *On the Implementation of the Probabilistic Logic Programming Language Problog*. Theory and Practice of Logic Programming, 11(2-3):235-262, 2011.

[50] R. KOWALSKI, M. SERGOT - *A Logic-Based Calculus of Events*. New Generation Computing, 4(1):67-95, 1986.

[51] K. KUMAR, A. MUKERJEE - *Temporal Event Conceptualization*. Proceedings of the Tenth IJCAI Conference, 1987.

[52] X. LE GUILLOU, M.-O. CORDIER, S. ROBIN, L. ROZÉ, *et al.* - *Chronicles for On-Line Diagnosis of Distributed Systems*. Proceedings of the European Conference on Artificial Intelligence (ECAI), Pages 194–198, 2008.

[53] A. MÕHALLA, E. CRAYE, S. C. DUTILLEUL, M. BENREJEB - *Monitoring of a Milk Manufacturing Workshop Using Chronicle and Fault Tree Approaches*. Studies inInformatics and Control, 19(4):377-390, 2010.

[54] R. MILLER, M. SHANAHAN - *The Event Calculus in Classical Logic – Alternative Axiomatizations.* Electronic Transactions on Artificial Intelligence (http://www.etaij.org), 4, 1999.

[55] R. MILNE, C. NICOL, M. GHALLAB, L. TRAVE-MASSUYES, K. BOUSSON, C. DOUSSON, J. QUEVEDO, J.AGUILAR, A. GUASCH - *TIGER: Real-Time Situation Assessment of Dynamic Systems*. Intelligent Systems Engineering, 3(3):103-124, 1994.

[56] B. MORIN, H. DEBAR - *Correlation on Intrusion: An Application of Chronicles*. 6th International Conference on Recent Advances in Intrusion Detection (RAID'03), Pages 94-112. Springer, 2003.

[57] A. PASCHKE, M. BICHLER - *Knowledge Representation Concepts for Automated SLA Management.* Decision Support Systems, 46(1):187-205, 2008.

[58] A. PASCHKE, A. KOZLENKOV, H. BOLEY - *A Homogeneous Reaction Rule Language for Complex Event Processing*. Workshop on Event driven Architecture, Processing and Systems, 2007.

[59] Y. PENCOLÉ, A. SUBIAS - *A Chronicle-Based Diagnosability Approach for Discrete Timed-Event Systems: Application to Web-Services.* Journal of Universal Computer Science, 15(17):3246-3272, 2009.

[60] A. PIEL - *Reconnaissance de comportements complexes par traitement en ligne de flux d'évènements.* PhD thesis, Université Paris 13 / ONERA, 2014.

[61] F. PORTET - *Pilotage d'algorithmes pour la reconnaissance en ligne d'arythmies cardiaques*. PhD thesis, Université Rennes 1, 2005.

[62] R. QUINIOU, L. CALLENS, G. CARRAULT, M.-O. CORDIER, E. FROMONT, P. MABO, F. PORTET - *Intelligent Adaptive Monitoring for Cardiac Surveillance.* Computational Intelligence in Healthcare 4, Pages 329-346. Springer, 2010.

[63] R. RINCÉ, R. KERVARC, P. LERAY - *On the Use of WalkSAT-Based Algorithms for MLN Inference in Some Realistic Applications*. Proceedings of the 30th International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems, Pages 121-131, 2017.

[64] A. SKARLATIDIS, A. ARTIKIS, J. FILIPPOU, G. PALIOURAS - *A Probabilistic Logic Programming Event Calculus*. Journal of Theory and Practice of Logic Programming (TPLP), 15(2):213-245, 2015.

[65] A. SKARLATIDIS, G. PALIOURAS, G. A. VOUROS, A. ARTIKIS - *Probabilistic Event Calculus based on Markov Logic Networks.* 5th International Conference on Rule-Based Modeling and Computing on the Semantic Web, Pages 155-170. Springer, 2011.

[66] N. STOJANOVIC, A. ARTIKIS - *On Complex Event Processing for Real-Time Situational Awareness.* 5th International Conference on Rule-Based Reasoning, Programming, and Applications, Pages 114-121. Springer, 2011.

[67] A. SUBIAS, E. EXPOSITO, C. CHASSOT, L. TRAVE-MASSUYES, K. DRIRA - *Self-Adapting Strategies Guided by Diagnosis and Situation Assessment in Collaborative Communicating Systems.* 21st International Workshop on Principles of Diagnosis (DX 10), Pages 329-336, 2010.

[68] M. VARJOLA, J. LOFFLER - *Pronto: Event Recognition for Public Transport.* 17th ITS World Congress, 2010.

[69] K. WALZER, M. GROCH, T. BREDDIN - *Time to the Rescue - Supporting Temporal Reasoning in the Rete Algorithm for Complex Event Processing.* Database and Expert Systems Applications, Pages 635-642. Springer, 2008.

AUTHORS

Romain Kervarc graduated from the *École Normale Supérieure de Lyon* and obtained a Ph.D. in formal logic in 2007. Since then, he has been a full-time researcher at ONERA, where he is the head of research unit "Modelling and Engineering of Distributed Systems and Software". His research interests include system modelling, formal methods, complex event processing, temporal logic and mixed logical and stochastic approaches, with a particular focus on the application of such method to actual industrial systems.

Ariane Piel graduated from Université Paris 13 in 2014 where she received her Ph.D. in computer science. She also holds a M.Sc. in mathematical logic and foundations of computer science. From 2011 to 2016, she was a Ph.D. student and then a post-doctoral researcher at ONERA – The French Aerospace Lab, in the Department for System Design and Performance Evaluation. Since 2016, she has been a full-time research fellow at CEA-LIST. Her research interests are focused on logic and formal methods for the conception and evaluation of systems.

A. El Fallah Seghrouchni
(Sorbonne Université)

L. Grivault
(Thales Defense Mission Systems)

E-mail: amal.elfallah@lip6.fr

DOI: 10.12762/2020.AL15-03

Multi-Agent Paradigm to Design the Next Generation of Airborne Platforms

Airborne platforms such as Remote Piloted Aircraft Systems (RPAS) operate in highly critical contexts. The next generation of RPAS will be endowed with multifunction sensors (*i.e.*, each sensor offers a large panel of functions to the platform's manager during the mission). As a platform, RPAS carry out a wide collection of complex tasks, thanks to the interleaving of the various services of sensors. The sensors are in charge of collecting data from the environment. Our main goal is to design a system as a software medium layer between the platform manager and the hardware resources on board the airborne platform (*i.e.*, multifunction sensors).
Today, the requirements of the platform in terms of autonomy, modularity, robustness and reactivity, as well as the industrial constraints, call for the design of a new multifunction system architecture. Such a design may rely on a multi-agent paradigm since it is modular by nature and the agents naturally bring autonomy and pro-activity to the system.
This paper presents new and original contributions: (1) an original agentification of the system, in the form of a multi-agent architecture that captures the dynamics of the environment by creating agents depending on objects that may appear in the mission theater; (2) agents that generate a task plan (a task is an action that will require a sensor to be achieved) according to the resources (e.g., the sensors) needed; (3) a scheduler that handles the task plans issued by the agents in order to provide efficient sensor scheduling.

The context

Nowadays, airborne platforms are used worldwide for air superiority, as a strategic asset during various kinds of operations, including conflicts, surveillance and rescue. These operations occur in highly dynamic environments with a low predictability under scenarios combining up to a thousand entities. The involved entities all have their own behaviors, speeds and trajectories. In this context, onboard instruments (*i.e.,* sensors) allow the platform, hence the mission manager, to collect knowledge from the field.

Throughout the years, sensors have become complex systems, multifunction, able to share data, communicate and, since recently, collaborate. Sensors are all specific to various physical dimensions (electromagnetic at different wavelengths, optics, infrared, etc.) and different ranges (few meters to hundreds of kilometers, shallow to wide angles, etc.). Due to this variety, collaboration between sensors allows new data to de deduced concerning the environment by overlapping outputs coming from many sensors.

The sensor scopes and ranges are not limitless, the function set is expanding, and with a maximum of about a thousand entities in the field, the global sensor capacity is the main limit for the enhancement of the MSS.

As a result of sensor limits in terms of range and scopes, the platform's localization is one of the main requisites for sensor efficiency. This requirement implies that the MSS has to be fully aware of the platform trajectory and speed.

Due to the criticality of the context and the mission's objectives, operators are expecting a certain determinism from the decisions proposed by the MSS. The MSS will be following clearly defined rules specifying sensor actions and which tasks will be accepted by the scheduler.

The evolution of battlefields due to many factors, including new technologies and conflict transformation, leads to emerging needs [4]. These needs directly affect the development of airborne platforms and thus of Multi-Sensor Systems (MSS). On the one hand, new operating conditions entail the use of autonomous platforms with advanced flexibility and multirole capabilities [9]. On the other hand, the rapid evolution of the technologies together with the cost reduction objective are leading industries to develop more reliable and durable systems [2]. Sensors carried by RPAS are now able to perform a large panel of functions, such as image acquisition, spectrum analysis, and object tracking [5]. All of these sensors play a major role in operation and their optimization has become essential.

In this article, we will study the management of resources onboard Remote Piloted Aircraft Systems (RPAS). Our approach is aimed at designing a suitable architecture to deal with resources; *i.e.,* various sensors in our target application. We adopt the multi-agent paradigm by using an agent-based architecture for the multi-sensor and multi-function system. This review presents new and original contributions: (1) a multi-agent architecture that captures the environment dynamics by creating agents watching the mission theater, which corresponds to an original agentification of the system; (2) each agent generates a task plan according to the resources that it needs, namely the sensors; (3) our scheduler, which handles the task plans issued by the agents in order to provide efficient sensor scheduling. The coordination of the sensors is then supported by a scheduling mechanism, in order to satisfy the requirements of the mission and the platform in a hardly-constrained environment.

Our paper goes on to present a realistic scenario and shows, through simulations, how the multi-agent system evolves and how our scheduler manages the agents' task plans in a realistic mission theater.

This paper is organized as follows: Section 2 briefly presents the multi-agent paradigm and related work, and emphasizes the originality of our contributions. Section 3 presents our framework, including the multi-agent architecture that we propose for the design of the next generation of airborne platforms; Section 4 details the scheduling mechanism; Section 5 provides our experimental results based on the scenario given by our industrial partner. Finally, Section 6 concludes this paper and presents our perspectives.

Related Work

In artificial intelligence, an intelligent agent (IA) refers to an autonomous and goal driven entity called agent that acts in order to achieve goals. An agent is usually embedded in an environment that can perceive through sensors and modify consequent actuators. It may be simple (such as a reactive agent) or complex (such as a BDI agent, or cognitive agent) depending on the modelling requirements. In all cases, intelligent agents may be endowed with skills to achieve their goals. Cognitive agents may use knowledge and intelligent skills, such as methodic, functional, procedural approaches, algorithmic search or reinforcement learning. A multi-agent system (MAS) is composed of an organization of multiple interacting intelligent agents. A multi-agent paradigm can solve problems that are difficult or impossible for an individual agent or a monolithic system to solve [11].

Agent-based online architectures are currently used within the Air Traffic Controllers (ATC) [7,1] of many Airports. These agent ATC architectures demonstrated the advantages brought by agents in terms of autonomy. The objectives of ATC are to control the traffic in geographical areas [10]. This task is usually done by a human operator, who can be potentially overburdened depending on area attendance [3]. In this context, agents can be used to follow the location of aircraft in a geographical area, and assist/alert the operator in various situations.

In ATC, agents are mainly used as secondary operators assisting the main system's user with automatic treatment, freeing the operator from some of the workload. ATCs have many constraints in common with a MSS, especially complex visualization of the field, data overloads, high criticality and low delays.

The fundamental difference between ATC and MSS lies in the presence of the sensors. Sensors in this kind of airborne platforms are highly complex instruments, continuously expecting precise requests to work (time, orientation, duration, power, movement tracking, etc.). Furthermore, all requests, treatments and products should be processed in a real-time manner, leading to highly responsive and predictive sensor behaviors.

Driving sensors through a multi-agent system has been studied previously in the context of sensor-mission assignment [6]. In this previous architecture, sensors were *agentified* and shared missions, which were given by a mission manager. In our system, the MSS also generates sensor plans by analyzing the data coming from the field and making sensor plans in consequence. This feature leads the MSS to support low-level sensor requirements, as well as high-level autonomy goals simultaneously.

From a scheduling point of view, our scheduler manages task plans (tasks are complex actions that require resources such as sensors to be achieved) that are feasible within a particular time window. Each task is specified by precedence and duration constraints. The plans are weighted by an operationally determined priority coefficient, and the industrial need requires mainly this coefficient to be taken as input. In our architecture, the objective is not to balance the use of resources, since each task is dedicated to one precise resource, but rather to have all priority plans scheduled at the end of the scheduling process. This approach is quite different from those described in the scheduling literature, which is mainly centered on sharing divisible tasks with dynamic priority, in order to distribute them among resources in an optimized way.

The MSS framework

At first sight, the MSS acts as an interface between the Mission Manager and the sensors' aperture set. The MSS helps to provide high-autonomy features, as well as an accurate control of sensors and efficient use of limited available resources (sensor apertures, power, cooling, computing power, etc.) [2]. To build this MSS, we will resort to a multi-agent architecture since the agents are suitable for bringing the flexibility and the autonomy required by the MSS. The following section will describe our proposed architecture, given in the figure, as well as the inputs and outputs of our MSS architecture.

MSS Architecture description

High-Level Orders and Policies

Policies and high-level orders are the only commands available to the operator for regulating the MSS behavior. They are sent dynamically (according to the changes that may occur during the flight) to the Mission Manager for effective control of the airborne platform.

High-level orders are defined as objectives to be achieved by the mission as a whole, while the policies are defined as a set of rules to be followed by the MSS. Policies impact the behavior of the multi-agent system that implements the MSS, and thus act at various levels. This may imply some restrictions on the sensors, or on the autonomy of the agents' behavior. Indeed, the policies are transmitted to every agent when updated by the mission manager.

Knowledge Base

The knowledge base gathers all knowledge needed by the MSS agents to plan for the use of required sensors. These data specify the generic characteristics of the field objects, the generic agents' behaviors and data acquisition procedures on the high-level decision side. On the low-level side, available knowledge is about sensor delays, scope specifications, speed requirements, running times and all kinds of data required for the evaluation of sensor operations. They are loaded offline to the MSS.

Platform Data

The platform's data communication provides all essential data for high and low level decisions. In fact, the air temperature, platform altitude, speed, position or even weather are required data to plan sensor operations, since they affect sensor operations. Since the current platform position is needed for real-time sensor requests, upcoming positions are also required in order to plan next sensor actions. For this reason, the platform flight plan is available to the MSS. This flight plan is dynamic and can be adapted by the operator during the mission; however, the actual position can differ from it; e.g.*,* in case of an unplanned maneuver.

Global Scheduler

The global scheduler receives all of the plans from the agents, to schedule them accurately on sensor timelines. Due to the number of objects present in the field (*i.e.,* a large number of agents in the architecture), the scheduling is an important process in our system.

Resource Managers

When one of the resource managers receives the plan produced by the scheduler, the resource accurately plays the content of its timeline as specified in the global plan. The manager reads all of the timeline tasks present and, in the case of material resources, sends the corresponding orders to the sensors.

The Track Merger

After the sensors have accomplished their tasks, the results are sent to the track merger. The track merger merges all data coming from sensors and delivering data to the agents. The merging is a complex operation, due to the scattered data recovered from the field. The partial observation of the field leads to a lack of object data continuity.

MSS Outputs

This architecture assists the Mission Manager during decision making, so the purpose of the MSS is to share all of the data gathered by the sensors. The MSS and the Mission Manager are both counterparts exchanging information about the tactical situation (*i.e.,* the theater's body of knowledge). In addition, the Mission Manager is able to control the MSS manually during operation and to bypass the MSS' decisions.

Closed Loop Sensor Control

This four-step process of planning, scheduling, sensing and merging operations constitutes a fast sensor closed control loop.

Between high-level decisions and sensor management, agents play an important role in the MSS architecture.

Agent Design

Agents have a unique objective: to collect as much data as possible about field objects through the use of sensors in order to fulfill high-level orders. To achieve its goal, an agent will try to select and execute one of the available functions on the NGAP. In practice, a function will rely on a pre-compiled task plan while the local scheduler is in charge of time instantiation (tasks durations, deadlines, etc.). In our framework, each task is associated with a resource and sensors are assimilated to material resources.

Agents are equipped with communication modules, memory and a core. They have a double role: creating high-level sensor objectives and generating sensors plans.

To increase the architecture's potential, we consider three classes of resource: a) sensors (e.g., an antenna); b) any type of equipment that can be reserved for the functioning of the sensors (e.g., an image processing unit); and c) any physical magnitude necessary for the proper functioning of the sensors (e.g., frequency).

This allows the agents to make task plans involving sensors, as well as the resources needed by sensors. This classification of resources has proved to be useful for the exclusive use of sensors when they need access to the same non-material or material resource. With this classification, the task dependencies are reduced, and the allocation process is faster due to fewer exchanges between the scheduler and the agents.

In our architecture, all of the resources are considered as artifacts [8]. In this context, an agent has a double role: creating high-level sensor objectives and generating, for a given function, a feasible task plan with an accurate allocation of resources.

Communication features are needed for exchanging data with an agent's environment while the memory feature provides necessary variables for the agent's operations and rational behavior. The core is hosting all running algorithms supported by the two previous features, in order to exchange and to store computed data.

An agent has in its memory a map of all variables standing for real objects, such as speed, altitude, position, attitude and vital signs. This map is empty at the creation of the agent, and is filled over time with collected data, to be aggregated in order to provide knowledge about objects.

Two main processes are at the core of the decision of the agent and determine which, how and when sensor functions should be activated.

The first main process determines which high-level functions (*i.e.,* operational functions) should be achieved in order to acquire data.

For this purpose, groups of algorithms need many inputs, such as platform data, operational knowledge (e.g., rules of actions depending on object characteristics), group orders, policies, agent orders, and specific field object variables.

The second main process considers the previous result and evaluates the precise timing of function execution. Firstly, it considers the sensor knowledge suitable for the function, such as sensor scopes, resource needs, timing constraints, etc.

Then, the global platform context, such as the flight plan, temperatures, weather and all platform variables likely to influence the sensors' work. Finally, all of these data modify the chosen function to a specific sensor plan placed along the platform's trajectory.

Feedback between these two processes is important to prevent the unfeasibility of the high-level function. In fact, the agent's main benefit is autonomy. This benefit is also brought by the adaptability of decisions taken by agents and the ability that it has to propose new solutions if a particular sensor is not available (sensor failure, weather incompatibility or other sensor conflict).

Agents roles

Agents are generic when created, meaning that they are all able to instantiate various available task plans at the system birth. Agents become specialized along the platform flight after receiving field data. It should be specified that the MSS can detect an object without knowing either what kind of object it is or the object's position.

Therefore, it should also be specified that not all sensors can be used with all kinds of object. If the agent is not specified because of a weak data feed, available functions for this agent would refer to a very large set of sensors. The connection between the agents and field objects brings many advantages:

- A natural virtual embedded vision of the field with a network of active objects.
- Easy access to behavior analysis and learning functions when faced with in-field unexpected events.
- Strong modularity of development.
- High autonomy of the MSS provided by the agents' proactiveness.
- Easy modeling of an open system, with objects that appear or disappear dynamically.
- A first step for a fully decentralized tactical situation architecture.

From the operational point of view, all field objects possess a specific degree of interest for sensors. For instance, a highly critical or dangerous object in the field would naturally lead to a proportional use of sensors to gather knowledge about this object. The closed control loop achieves this autonomy objective with the implication of agents.

Scheduling

MSS Efficiency

The efficiency of the MSS relies on the consistency of achieved tasks according to environment parameters:

- Events from the field (e.g., weather changes).
- Platform condition (e.g., platform speed and attitude).
- MSS state (e.g., sensor failure).
- Field object behaviors (e.g., an object's appearance or attitude changes).
- Operator instructions (e.g., specific operating policies given by different operators).

The highest efficiency is reached if the MSS has collected the maximum volume of significant information about the field with regard to all of the previous parameters.

The great number of objects present in the field implies a large quantity of sensor plans created by the agents. Many of these plans can be insignificant from an operational point of view. As an example, we can imagine a scenario in which the platform is tracking an important object in the field through the radar sensor; the importance of the object implies a high level of priority.

For instance, if the platform is approaching a highway used by 300 vehicles with low operational interest, agents will send sensor plans corresponding to an identification procedure.

After sorting by priority order, not all of the requests will be achievable by the same sensor. A part of these would be achieved by another one (e.g., a camera sensor) while the other part would be simply unachieved. Despite a partial realization of agent requests, the resulting efficiency is optimal in the given situation.

The determination of the agents' priority level is an important point of the scheduling consistency.

Task plan

Year after year, the number of functions (e.g., *take a picture* or *listen to signals on M-band*) achievable by an MSS has multiplied. Today, sensors allow many different functions to be carried out. Each function is achieved through a specific task plan.

A task is an indivisible action achieved by a resource. A task can be identified as T_k, of duration D_k and scheduled on the timeline of a resource r_j. The task starts at t_s and finishes at $t_s + D_k$.

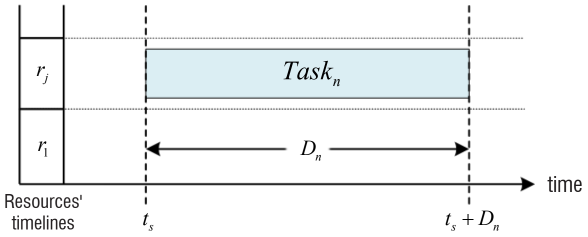


Figure 1 – A task and its parameters

A task plan is an ordered set of tasks to achieve a sensor function (e.g., *Take a picture* requires the use of two resources: An *optical camera* and an *optical image-processing unit*).

Figure 2 represents a plan composed of three tasks, each needing a distinct resource. This figure shows the asynchronous and indivisible features of resource occupations. In fact, Tasks 1 and 2 start and end at the same time, while Task 3 starts before the previous tasks end. The resources are fully allocated during the tasks. The plan weight is determined by agents and reflects the importance of executing the plan at an operational level.

Plan tasks and tasks directly inherit the agent's priority.

Plan P_k is defined such that $P_k = \{\alpha, T_r, T_d, C, T\}$, where α is the plan's priority, T_r is the release time of the plan, T_d is the plan's execution deadline and C is the set of constraints that specifies the order of the set of tasks $T_k = \{T_1, T_2, T_3\}$.

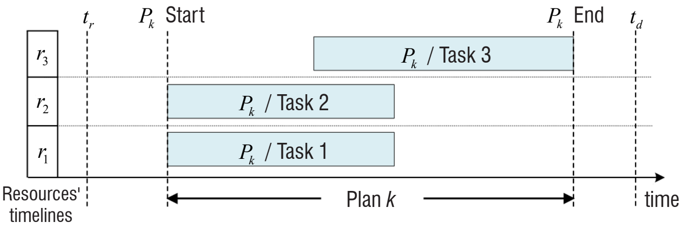


Figure 2 – A task plan involving three resources

Scheduler

The scheduler takes as input the plans issued by the agents and the plans already scheduled on the timelines, as well as their priorities, and defines a global schedule. After sorting all of the plans by priority, the scheduler's algorithm calculates the start time for each task contained in the plans.

The result is a global schedule constituted of interleaved tasks. This scheduling is achieved for a temporal horizon T_H .

The plans that were not accepted within the temporal horizon are not scheduled and will be processed later when the average priority of all of the plans will be lower. If a plan is not scheduled, the agent is advised about the failure and is able to submit a new plan on less busy resources.

Given that our algorithm schedules plans one by one, and the industrial context requires the plans with the highest priority to be scheduled despite the low-priority plans, the plans of highest priority are scheduled first.

The scheduling can be based on highest priority first plan insertion through starting date computation, with the possibility of ordering plans if needed.

Experimental Results

Scenario

Since testing in real situations is complex and very expensive to be achieved with this kind of platform and MSS, we implemented this architecture and its environment in simulation. Hence, we developed a special test scenario, able to show the main decisions that an operator takes during a mission. This scenario gathers up to 10 steps where the platform is deployed in various different contexts with different criticality. Thanks to this scenario, we can now evaluate the behavior and the decisions taken by our MSS architecture by simulation in realistic situations.

Figure 4 shows the visualization of the main window of the simulation engine.

The bottom frame represents functions and resources available in the MSS. Framed resources and functions are currently working and

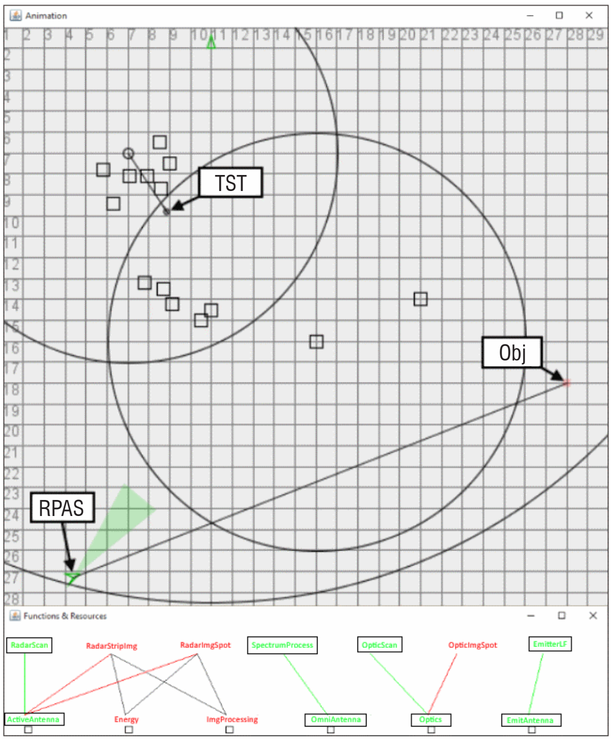


Figure 4 – Visualization of the simulator's main frames

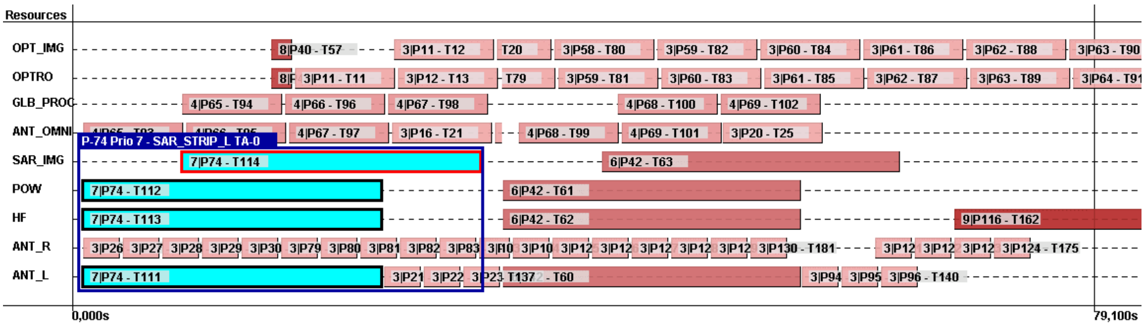


Figure 3 – Resulting global scheduling

unframed ones are not. The links between functions depict the functions' dependencies on the sensors.

At this step of the scenario, the *watch* mode of the RPAS, which was enabled at the power-on of the MSS, has planned and executed the use of an electromagnetic detector. It detected the presence of a radar ("Obj" in the figure), and is heading towards the emitting object to obtain more data about it. As in reality, the MSS is not managing the platform attitude (nor deciding the platform maneuvers or controlling RPAS surfaces), but the Mission Manager is deciding to go toward the object after the MSS shared data received and proposed an identification procedure (proposition emitted by the corresponding agent) at that point. The detection of this object led to the activation of various other sensors. The cone around the RPAS is the visualization of an optical sensor (*i.e.,* camera) turning around the platform. This sensor was also activated by the *watch* function.

After many tasks, an objective is given to the platform: "seek the object TST" (*i.e.,* Time Sensitive Target) in a particular area. After 2 minutes and many achieved tasks, the TST was found as expected without human control over the MSS' sensors.

Some functions were implemented to enhance the robustness of the MSS, including agent death and replication to avoid blocked agent issues by detecting and killing blocked agents and creating a new agent with the data backup from the previous one.

The MSS' global behavior matched our expectations during simulations and sensor tasks were scheduled in time and consistently with regard to the simulated field. The modularity of the MSS is improved by this architecture and the agent nature allows the architecture's characteristics to be specified block by block.

With regard to the system autonomy, the simulation showed the ability that the MSS has for managing high-level objectives depending on its own observations, without any intervention from the operator.

The previous states of the visible tactical situation were accomplished in a fully automated way with no human manipulation. All of the necessary data were embedded in the algorithms, plan descriptions and tactical rules, like for a real platform during flight preparation. The simulation started with the creation of the RPAS agent, which is a particular derivation of the agent class. The RPAS goal is to build plans able to collect data from the field in order to detect objects. Once the RPAS was created, many watching plans were built and sent to the scheduler. The sensors worked according to the plans and the produced data were sent to the track merger, responsible for agent creation.

After many tasks, an objective was given to the platform: "seek the object TST" (*i.e.,* Time Sensitive Target) in a particular area. After a while and many sensors tasks, the TST was found as expected without human control over the MSS' sensors.

Some functions were implemented to enhance the robustness of the MSS, including agent death and replication to avoid agent-blocking situations.

The MSS' global behavior matched our expectations during simulations and sensor tasks were correctly scheduled. Work should be done to refine choice models concerning agent plans, sensor behaviors, and object behaviors. However, the modularity of the MSS is improved by this architecture and the agent nature allows architecture characteristics to be specified block by block.

With regard to the system autonomy, the simulation showed the ability that the MSS has for managing high-level objectives depending on its own observations, without any interventions from the operator other than specifying policies.

Experimentation

Agents submitted 100 plans to the global scheduler (each agent has a local scheduler that generates pre-compiled plans). As done by our algorithm, the scheduling results are given in Figure 5 and Figure 6.

Figure 5 shows that the number of scheduled plans increases with the size of the temporal horizon. In our simulation, whatever the priority of the plan, its deadline coincides with the temporal horizon. If the plans are quite temporally constrained, then giving the scheduler more time is not useful to increase the number of scheduled plans. It also shows that the scheduling time depends on the temporal horizon. The larger the horizon is, the less reactive the scheduler is. From an operational point of view, a scheduling time above 660ms is not acceptable: the temporal horizon should be under 120s to keep the scheduling time under 100ms, depending on the mission.

Figure 6 shows that the plans with the highest priority are scheduled as soon as possible, even in a narrow window (temporal horizon).

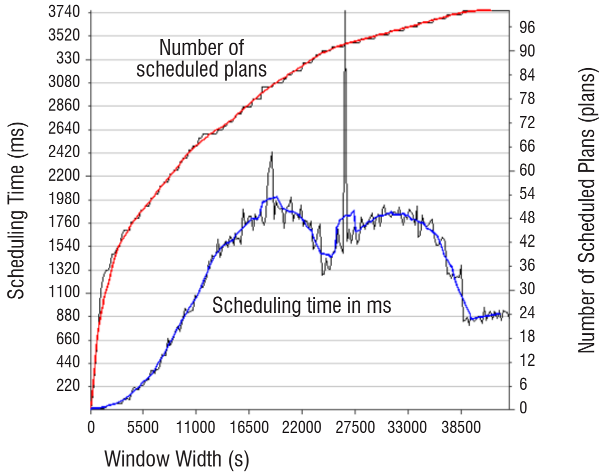


Figure 5 – Number of scheduled plans and scheduling time depending on the temporal horizon.

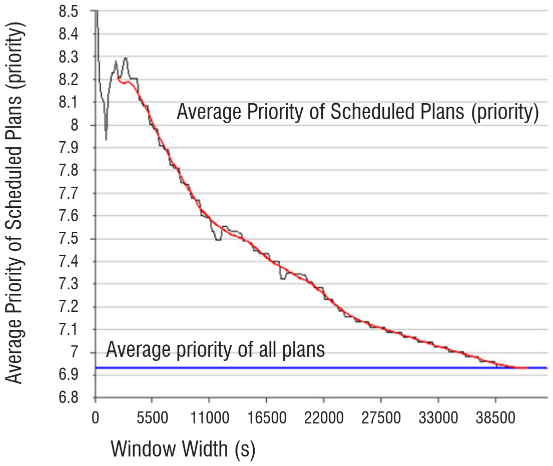


Figure 6 – Average priority of scheduled plans depending on temporal horizon.

In addition, the average priority of the scheduled plans converges to the average priority of all of the plans.

The operational requirements are met by the scheduling that we propose, since most of the time the MSS faces situations that require short-time scheduling with few plans having high priorities.

In this dynamic instantiation of the scheduler, the global schedule is redefined each time a plan with a priority higher than the lowest priority of the scheduled plans is received from agents. To avoid started plans being stopped before they are finished, they are isolated from the schedule queue. Started plans are stopped only if higher priority plans cannot successfully be scheduled because of their time window constraints (*i.e.,* release/deadline times).

Conclusion

Our study is aimed at dealing with new scheduling problems in the context of RPAS. We are interested in the scheduling of task plans instead of the classical scheduling of tasks. This implies several differences with existing algorithms. For instance, removing an unfeasible task plan releases a set of resources, which strongly affects the ongoing scheduling.

We also have to deal with a flow of requests from the agents. This can be roughly viewed as online scheduling, but at this stage we have no information about the probabilities of agent requests.

From the architecture point of view, our design of a multi-agent system allows dynamic and open theaters to be considered. The dynamics of the architecture, its flexibility and the first results of our scheduling mechanism provide a promising solution for the next generation of airborne platforms. Indeed, the multifunction and multi-sensor features of this platform are fully exploited by the multi-agent system.

The closed loop sensor functions and resource allocation control by agents is a first breakthrough concerning agent-based online architectures for strongly constrained systems like MSS.

The results provided by the simulation gave us a first proof of concept concerning the architecture. The general behavior of the simulated MSS, the agents' planning abilities, and the general flexibility of implementation met our expectations.

The various different software blocks are henceforth welcoming innovative algorithms concerning tactical situation forecasting, improved track merging, collaborative agents, refined scheduling, and enhanced communication protocols.

These future algorithm developments will have an important role in the architecture efficiency and sustainability. Scheduling is one of the main algorithms affecting the MSS overall consistency and will be the topic of upcoming research.

A modular software architecture for autonomous and optimized sensor driving has been presented. However, this approach does not propose any hardware architecture. Since the final MSS capabilities and modularity also depend on the supporting hardware and its distribution within the platform, special attention should be paid during its designing task to appreciate all of the features provided by the architecture.

Finally, the architecture may be potentially adapted to less constraining platforms like underwater vehicles, piloted aircraft, or land vehicles.

Perspectives are various, ranging from the improvement of the scheduling by using a learning approach in order to anticipate the law of arrival of demands from the agents, to the decentralization of the architecture and multi-platform cooperation ■

References

[1] T. J. CALLANTINE - *CATS-based Air Traffic Controller Agents.* San Jose State University, 2002.

[2] L. CHABOD, P. GALAUP - *Shared Resources for Airborne Multifunction Sensor Systems*. IET International Conference on Radar Systems, 2014.

[3] Y. IBRAHIM, P. HIGGINS, P. BRUCE - *Evaluation of a Collision Avoidance Display to Support Pilots' Mental Workload in a Free Flight Environment.* IEEE International Conference on Industrial Engineering and Engineering Management, 2013.

[4] S. KEMKEMIAN, M. NOUVEL, P. CORNIC, P. LE BIHAN, P. GARREC - *Radar Systems for Sense and Avoid on UAV.* International Radar Conference, October 2009.

[5] S. KEMKEMIAN, M. NOUVEL-FIANI - *Toward Common Radar & EW Multifunction Active Arrays.* IEEE International Symposium on Phased Array Systems and Technology. 77784, 2010.

[6] T. LE, T. J. NORMAN, W. VASCONCELOS - *Agent-Based Sensor-Mission Assignment for Tasks Sharing Assets.* IFAAMA, 2009.

[7] M. NGUYEN-DUC, Z. GUESSOUM, O. MARIN, J.-F. PERROT, J.-P. BRIOT - *A Multi-Agent Approach to Reliable Air Traffic Control*. International Symposium on Agent Based Modeling and Simulation, Vienna, Austria, 2008.

[8] A. OMICINI, A. RICCI, M. VIROLI - *Agens Faber: Toward a Theory of Artefacts for MAS.* Electronic Notes in Theoretical Computer Science 150 (3): 21–36, May 2006.

[9] A. SCHULTE, D. DONATH, F. HONECKER - *Human-System Interaction Analysis for Military Pilot Activity and Mental Workload Determination.* IEEE International Conference on Systems, Man, and Cybernetics, 2015.

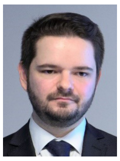
[10] K. TUMER, A. AGOGINO - *Distributed Agent-Based Air Traffic Flow Management.* The Sixth Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems, AAMAS, 2007.

[11] R. H. BORDINI, A. EL FALLAH SEGHROUCHNI, K. HINDRIKS, B. LOGAN, A. RICCI - *Agent Programming in the Cognitive Era.* JAAMAS (Autonomous Agents and Multi-Agent Systems) volume 34, Article number: 37 (October 2020). https://doi.org/10.1007/s10458-020-09453-y.

Amal El Fallah Seghrouchni (is a Full Professor at Sorbonne University – Faculty of Science and Engineering, and is currently on leave from the CNRS. She is a researcher assigned to the LIP6, and she leads the Multi-Agent Systems Group and co-leads the Artificial Intelligence and Data Science line of research (covering human and machine learning, deep learning, automatic decision-making, intelligent agents and multi-agent systems, etc.). Amal El Fallah Seghrouchni is also a member of the COMEST (World Commission on the Ethics of Scientific Knowledge and Technology) of UNESCO (2020 -2023) and Holder of the Chair of Excellence "Design of hybrid, cognitive and collaborative AI systems" – Sorbonne University / Thales (2020 – 2024).

Her research focuses on Artificial Intelligence, including the Design of Autonomous Systems and Ambient Intelligent Applications. She is developing an approach based on cognitive agents and multi-agent systems. This includes coordination models (negotiation, distributed planning, interaction protocols), and the spatial-temporal design of contextual systems based on adaptation, individual and collective learning, and human-artefact interaction. Her research themes find a large number of applications in the fields of complex system design (e.g., drones of the future), simulation (e.g., urban), contextual applications and smart cities (e.g., intelligent assistants). These applications and topics are supported by a large number of industrial partnerships and international collaborations.

Amal El Fallah Seghrouchni has also held a number of scientific posts in leading international organizations (e.g., General Chair at AAMAS 2020, Track Chair at AAMAS 2019, Track Chair at IJCAI 2019, Chair of the APIA 2019 Program Committee and of the APIA 2020 Program Committee, member elected to the IFAAMAS and EURAMAS boards, etc.). She has also led the research work of more than 30 doctoral students and has published numerous books and articles in the best conferences on Artificial Intelligence and Multi-Agent Systems (for more details, see http: //www-poleia.lip6.fr/elfallah~/).



Ludovic Grivault is a Doctor Engineer working for Thales Defense Mission Systems – He obtained his PhD in December 2018 under the supervision of Professor Amal El Fallah Seghrouchini at Sorbonne University – Faculty of Science and Engineering. His thesis topic was "Architecture multi-agent pour la conception et l'ordonnancement de systèmes multi-senseur embarqués sur plateformes aéroportées". His research focuses on embedded systems and artificial intelligence, more specifically on complex system architectures, multi-agent technologies, scheduling and airborne sensors.

F. Teichteil-Koenigsbuch,
G. Poveda
(Airbus AI Research)

E-mail: florent.teichteil-koenigsbuch@airbus.com

DOI: 10.12762/2020.AL15-04

Collaborative Common Path Planning in Large Graphs

This paper studies two-agent path planning algorithms in graphs, where the two agents are assigned independent initial and goal states but are incentivized to share some parts of their travel glued together by scaling down the duet cost function when they move in formation. Applications range from ride sharing to formation flights. After presenting an optimal but unscalable algorithm, we propose a decoupled approach that separates spatial and temporal reasoning by first geometrically finding formation and breaking nodes in the graph then temporally synchronizing the agents on the formation node by adapting their speeds along their paths in the graph. We also introduce an original heuristic function, which accounts for the potential formation paths in the graph and that is used to guide A* search on a cross-product graph representing the coordinated moves of the agents. We finally experiment our framework and compare its variants on grid-like and aircraft formation flight problems.

Introduction

Path planning has a long history of research dating back to the early days of Artificial Intelligence. Many variants have been studied, from continuous motion planning [2] to discrete optimization in graphs [5] including sampling approaches [16], investigating both single-agent [9] and multi-agent [23] settings. Multi-agent frameworks have mainly looked at optimizing trajectories for a set of agents while avoiding collisions [13][11], or at coordinating the trajectories of a set of agents to make them accomplish a common group objective; e.g., building formation patterns [7][4][1]. A specific case consists in incentivizing two or more agents to execute paths that share common moves; i.e., moves with the same current and next positions simultaneously, by reasoning about a cost function, which is lower when the agents move in formation side by side than when they follow different routes. In this paper we refer to this problem as "Collaborative Common Path Planning" (CCPP), depicted in Figure 1.

Applications of CCPP range from ride sharing planning, where two or more traveling people save money if they share the same vehicle along a common portion of their routes, to freight dispatch and routing where operational costs can be significantly reduced by transiting goods via intermediate common hubs, including comprehension of behaviors observed in nature, such as formation flights of migratory birds. Actual research works on CCPP, which have mainly originated

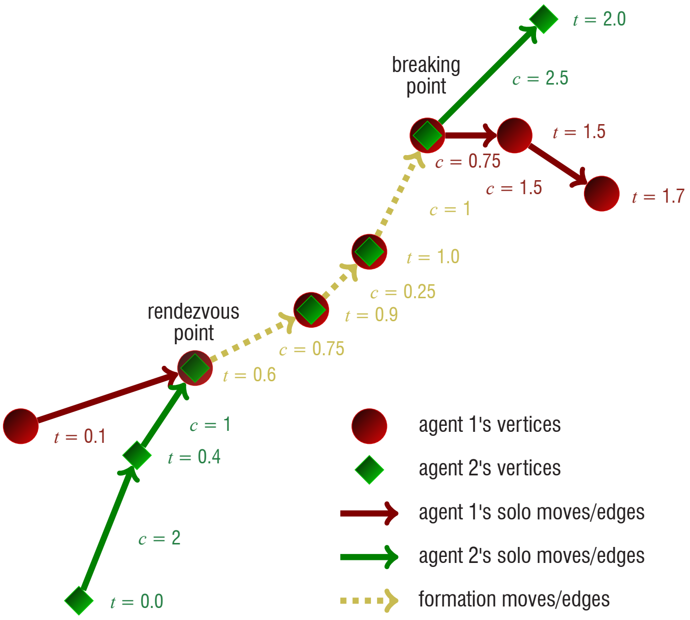


Figure 1 – Collaborative common paths example. Agent 1 (resp. 2)'s path duration is 1.7 (resp. 2.0). The agents move together in formation from t = 0.6 where they meet at the same navigation point until t = 1.4 where they break the formation. The global cost of their paths is 12.25.

from this latter example to the best of our knowledge [17][22][21], have recently studied how aircraft routes can be planned so that two given aircraft whose geodesic routes from their departure airports to their destination airports are spatially close can fly one behind the other along a common portion of their flight routes to save global fuel burn. Indeed, the lift of the follower aircraft can be increased if it flies in the aerodynamic vortex created by the leader aircraft wings, allowing it to reduce the thrust and thus reduce fuel burn. Some works also studied leader-follower path planning for mobile robots [4], but as CCPP for aircraft; all of these works assume the vehicles to be moving in continuous spaces using geometrical equations.

However, in reality, aircraft routes are rather defined over worldwide navigation graphs. The other aforementioned applications of CCPP also involve moving in discrete graphs rather than in continuous spaces. In this paper we study what we believe to be the first attempt to solve the CCPP problem in graphs, also paying special attention to making our approach scalable. We assume two agents to be moving in their own graphs – which can potentially be the same – in order to minimize the overall cost of reaching their goal vertices for each of them, while having the opportunity to significantly cut down atomic moving costs when they travel along specific pairs of (Agent 1's edge, Agent 2's edge) at the same time. Figure 1 represents the specific case where the graphs of each agent are identical. Importantly, our model assumes that the agents can control their speeds in order to synchronize at potential joining points from where they can move in formation, and that these speeds (or similarly, edge durations) can only be chosen from a discrete set – as is the case in aircraft formation flight, where edge cost evaluation is based on very time-consuming aerodynamic models that prevent a continuum of edge duration values from being explored. We present an optimal but non-scalable algorithm based on a transformation of the CCPP problem to a single agent path planning problem in the cross product of the agents' graphs, which can then be solved by heuristic search methods like A*, as well as an efficient but suboptimal spatio-temporal decoupling algorithm. We also discuss generic but non informative heuristics and propose efficient geometric heuristics specific to graphs defined on Euclidean spaces. We finally experiment with our framework and compare its variants on grid-like problems, as well as world-scale formation-flight experiments using real aircraft models and world flight networks.

Related work

There is an abundant literature on multi-agent path planning that may take a close look at the research that we investigate in this paper. The graph search community has investigated for many years multi-agent collision-free path planning [6]. Even though the objective of this research consists in minimizing the team cost-to-go summed over all of the agents, it sensibly differs from our work in the sense that avoiding collisions is the opposite objective to incentivizing the agents to meet and share common paths. Another line of research is aimed at planning dynamic formations for many moving agents [8], [19]. However, formations have a different meaning to ours: while we want the agents to move along the same graph edges to reduce their moving costs, like bird formation flights, those works dynamically assign predefined geometrical formation patterns to a group of agents that do not necessarily travel along the same edges. Ridesharing trip planning [18] is closer to our research, since the objective is to share vehicles among different passengers starting and ending at different locations. Since they share the same car, those passengers necessarily travel along the same edges on a portion of their trip. However, most works on ridesharing that we are aware of consider predefined meeting points, whereas we notably consider meeting points as part of the optimization problem itself. In ridesharing with passenger transfer optimization [3], the path from the passengers' starting location to the meeting point and from the breaking point to their ending location is additionally optimized, but the meeting and breaking points are still predefined and fixed. Along the same line of research, multi-modal path planning [15][12] look at optimizing the global flow of passengers between different sources and targets, while partly travelling by using common transportation means. However, multi-modal stations that serve as meeting and breaking points are also fixed, as for the meeting and breaking time tables. To the best of our knowledge, aircraft formation flight optimization in continuous airspace [22][17][21] is the closest problem to ours. Indeed, the specificity of the cost function for this problem, which reduces the sum of agent costs when they travel along the same edges, makes this line of research quite singular in the multi-agent landscape. We believe that our work is one of the first to investigate formation flight planning in discrete structured airspaces.

Background

There has been a long history of research on path planning in graphs. Many single-agent algorithms are based on the famous A* algorithm [9][10] which allows for lazy and partial exploration of the agent's navigation graph by guiding the search with heuristic estimates of the distance from the current node to the goal. Since we will use single-agent path planning as a generic tool to solve CCPP via a transformation to single-agent path planning, we only present here the plain A* algorithm. The interested reader is invited to look at [20] for state-of-the-art alternatives to A* for solving the single-agent transformation of our problem.

A* reasons about graph $\mathcal{G}=(V,E)$, which represents the feasible moves of the agent, with the edges being labelled by a cost function $c:E\to\mathbb{R}_+$. This iteratively expands the vertices of the graph from the starting vertex $v_s\in V$ up to reaching the goal vertex $v_g\in V$ via a minimum-cost path. Vertex expansion is guided by a heuristic function $h:V^2\to\mathbb{R}_+$ which gives numerical under-estimates of the distance from the current vertex to the goal. In order for A* to be optimal, the heuristic function needs to be both *admissible*, i.e., by noting as $C^*(v_1,v_2)$ the optimal path between any distant vertices $v_1\in V$ and $v_2\in V$, we have $h(v_1,v_2)\leqslant C^*(v_1,v_2)$, and *monotonic*, i.e., for any edge $e\in E$ and vertex $v\in V$ we have $h(e.\mathrm{in}(),v)\leqslant c(e)+h(e.\mathrm{out}(),v)$ by noting as $e.\mathrm{in}()$ (resp. $e.\mathrm{out}()$) the incoming (resp., outgoing) vertex of e.

CCPP problem formulation

Let $\mathcal{G}^{(i)}=\left(V^{(i)},E^{(i)}\right)$ be the labeled graph of Agent $i\in\{1,2\}$, which represents its possible moves in its own navigation network. Both agents' navigation networks do not need to be identical, nor semantically equivalent. Graph vertices (resp., edge ends) stand for positions (resp., moves). Edges are equipped with label pairs (c,d) representing the cost and the duration of each move. Whereas standard single-agent path planning algorithms do not reason about moving duration, collaborative common path planning needs move duration information in order to temporally synchronize the two agents on vertices where they can initiate a formation. We add exponent notations

(i) to labels c and d, and edge flows *in* and *out* to indicate to which agent the concept is related.

We note as $\mathcal{F}^{\otimes} \subset E^{(1)} \times E^{(2)}$ the set of edge pairs for both agents on which formations are possible. Elements of $\mathcal{F}^{\otimes}$ must be temporally coherent, meaning that both agents' moves must have the same duration when in formation; i.e., $\forall\left(e^{(1)}, e^{(2)}\right) \in \mathcal{F}^{\otimes}, d^{(1)}\left(e^{(1)}\right) = d^{(2)}\left(e^{(2)}\right)$. In Figure 1, the dashed yellow lines are elements of $\mathcal{F}^{\otimes}$. Depending on the application, other properties might be required typically that the start and end positions of the agents always be the same in the case where their navigation graphs are identical (e.g., vehicles moving in a same route network, as in Figure 1). It is also required that the two agents visit candidate formation vertices at the same time; i.e., $\delta^{(1)}\left(e^{(1)}\right) = \delta^{(2)}\left(e^{(2)}\right)$ for $\left(e^{(1)}, e^{(2)}\right) \in \mathcal{F}^{\otimes}$, where $\delta(i) : E^{(i)} \rightarrow \mathbb{R}_+$ represents the time when the incoming vertex of an edge is visited by Agent i in its own graph (implicitly dependent on the previously visited vertices from the start up to the edge's incoming vertex).

Let $\Pi_{v_s^{(i)}}^{v_g^{(i)}}$ be the set of all possible *timed* sequences of Agenti i's adjacent edges, which represent all of its possible paths in its own graph from its start position to its end position, all such edges being labelled by the duration (and cost) of the move along the edge – which controls the agents' speeds so that they can synchronize their formation rendezvous point. The set of all possible collaborative paths is $\Pi_{v_s^{(1)}}^{v_g^{(1)}} \times \Pi_{v_s^{(2)}}^{v_g^{(2)}}$, where sections of each agent's path can correspond to common formation moves. Note that both agents' paths separately extracted from a given collaborative path do not need to have the same length, but they can contain common edges where the agents move in formation. The objective of the collaborative common formation path planning problem is to find a path for each agent with common formation sections where they can move side by side. The global cost of the pair of paths for the two agents is split into two parts: (1) cost of individual sections where no formation moves are possible, corresponding to the sum of each agent's individual moves; (2) cost of common sections where both agents are moving together in formation, corresponding to the formation cost $c^{\otimes}$. Thus, the cost function of a pair of collaborative formation paths $\left(\sigma^{(1)}, \sigma^{(2)}\right) \in \Pi_{v_s^{(1)}}^{v_g^{(1)}} \times \Pi_{v_s^{(2)}}^{v_g^{(2)}}$ is defined as $C\left(\sigma^{(1)}, \sigma^{(2)}\right) =$

$$\underbrace{\sum_{\substack{e^{(1)} \in \sigma^{(1)} \\ \nexists e^{(2)} \in \sigma^{(2)} : \left(e^{(1)}, e^{(2)}\right) \in \mathcal{F}^{\otimes}, \delta^{(1)}\left(e^{(1)}\right) = \delta^{(2)}\left(e^{(2)}\right)}} c^{(1)}\left(e^{(1)}\right)}_{\text{agent (1)'s edges not in formation with agent (2)' edges (red in Figure 1)}} +$$

$$\underbrace{\sum_{\substack{e^{(2)} \in \sigma^{(2)} \\ \nexists e^{(1)} \in \sigma^{(1)} : \left(e^{(1)}, e^{(2)}\right) \in \mathcal{F}^{\otimes}, \delta^{(1)}\left(e^{(1)}\right) = \delta^{(2)}\left(e^{(2)}\right)}} c^{(2)}\left(e^{(2)}\right)}_{\text{agent (2)'s edges not in formation with agent (1)' edges (green in Figure 1)}} +$$

$$\underbrace{\sum_{\substack{e^{(1)} \in \sigma^{(1)}, e^{(2)} \in \sigma^{(2)} \\ \left(e^{(1)}, e^{(2)}\right) \in \mathcal{F}^{\otimes}, \delta^{(1)}\left(e^{(1)}\right) = \delta^{(2)}\left(e^{(2)}\right)}} c^{\otimes}\left(e^{(1)}, e^{(2)}\right)}_{\text{agent (1)'s edges and agent (2)' edges in formation (yellow in Figure 1)}}$$

The collaborative formation path planning problem consists in finding collaborative paths that minimize the global cost of the paths by taking account for common formation moves:

$$(\mathcal{P}) \quad \min_{\left(\sigma^{(1)}, \sigma^{(2)}\right) \in \Pi_{v_s^{(1)}}^{v_g^{(1)}} \times \Pi_{v_s^{(2)}}^{v_g^{(2)}}} C\left(\sigma^{(1)}, \sigma^{(2)}\right)$$

Since graph edges are weighted by duration (and cost), the variables of the minimization problem above are not only vertex positions but also edge durations, which is required to synchronize the agents at their formation rendezvous point.

Algorithms

Problem $(\mathcal{P})$ consists in finding the agents' paths in their graphs and synchronizing them both temporally and spatially, which can be viewed as a single path finding problem in a cross-product graph that aggregates the positions of each agent. Figure 2 represents how agents' positions are aggregated and tracked together within a single cross-position vertex (depicted in yellow in the figure) that can lie at intermediate positions respective to each agent's graph. In fact, the figure shows that it is not sufficient to synchronize each agent's atomic move on the vertices of their respective graphs: in order to make a formation at $t = 0.6$, Agent 1 has to perform 1 atomic move whereas Agent 2 must execute 2 atomic moves. Therefore, we need to represent an intermediate position for Agent 1 between its first and second vertices, which coincides with Agent 2's move through its second vertex. Thus, Agent 1 has to control its speed in order to meet Agent 2 at the same time point ($t = 0.6$ in the figure) at the rendezvous point. This example shows that edges must contain duration information about which we have to reason in order to temporally and spatially synchronize agents.

By casting the collaborative common path planning problem into a single path planning problem, we can run an A* algorithm on the cross-product graph. To this end, additional information must be embedded in the nodes of the product graph in order for A* to expand the product graph properly. Said differently, rules to expand the product graph on nodes closed by A* need specific information explained below.

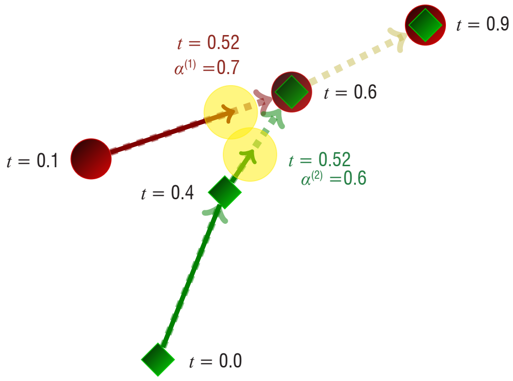


Figure 2 – Cross-product graph: at $t = 0.52$, Agent 1 is at the $\alpha^{(1)}$ portion of the edge linking the first and second vertices of its graph, while Agent 2 is at the $\alpha^{(2)}$ portion of the edge linking the second and third vertices of its graph.

Optimal algorithm

The optimal algorithm reasons in the combined spatial and temporal spaces without decoupling spatial and temporal synchronization logics of the agents. To this end, we define the following cross-product graph.

Cross-product labelled graph

The coupled cross-product labelled graph is defined as $\mathcal{G}^{\otimes} = \left(V^{\otimes}, E^{\otimes}\right)$ with:

- $V^{\otimes} = E^{(1)} \times [0;1] \times E^{(2)} \times [0;1]$, such that $v^{\otimes} = \left(e^{(1)}, \alpha^{(1)}, e^{(2)}, \alpha^{(2)}\right) \in V^{\otimes}$ represents the $\alpha^{(i)}$ percentage of completion of each agent i's move along edge $e^{(i)}$ in its own graph. We recall that $e^{(1)}$ and $e^{(2)}$ are each labelled with the cost and duration of the move of the agent in its own graph, meaning that they each take the form of a tuple in $V^{(i)} \times V^{(i)} \times \mathbb{R}_+ \times \mathbb{R}_+$. It is especially important for the understanding of the next paragraphs to note that *there can exist in each agent's graph many different edges with the same incoming vertex and outgoing vertex but with different durations (and costs)*.

- $E^{\otimes} \subseteq V^{\otimes} \times V^{\otimes} \times \mathbb{R}_+^4$, such that $e^{\otimes} = \left(v_1^{\otimes}, v_2^{\otimes}, c^{(1)}, d^{(1)}, c^{(2)}, d^{(2)}\right)$ represents a labelled cross-product transition of both agents from a cross-product position $v_1^{\otimes}$ to another cross-product position $v_2^{\otimes}$, with labels $\left(c^{(1)}, d^{(1)}, c^{(2)}, d^{(2)}\right)$, representing the cost and the duration of the original individual graph edges that are being travelled along by each agent. These labels are needed when constructing the edges outgoing from a given cross-product position, in order to know how much time and cost remain to be consumed and paid along the current edge of each agent's original graph, in case the cross-product position may correspond to an intermediate position of one of the agents in its own graph (meaning that the agent has not yet finished travelling along its current edge in its own graph when transitioning to another cross-product vertex).

For instance, consider Figure 2 and assume that Agent 1 (resp., Agent 2)'s path as it appears from left to right in this figure is noted as $\sigma^{(1)}$ (resp. $\sigma^{(2)}$). The current positions of both agents depicted in Figure 2 is encoded in a product graph vertex $v_1^{\otimes} = \left(\sigma^{(1)}(1), 0.7, \sigma^{(2)}(2), 0.6\right)$, which means that Agent 1 (resp. Agent 2) is at 70% (resp., 60%) of completion in transitioning to the in-vertex to the out-vertex of its first (resp., second) transition along its path. Now, assume that the next event corresponds to Agents 1 and 2 reaching the out-vertices of their current respective edges (resp., $\sigma^{(1)}(1)$ and $\sigma^{(2)}(2)$) simultaneously at t = 0.6, which happens to be a rendezvous point, the next cross-product vertex will be $v_2^{\otimes} = \left(\sigma^{(1)}(2), 0, \sigma^{(2)}(3), 0\right)$[1] and the corresponding transition will be $e^{\otimes} = \left(v_1^{\otimes}, v_2^{\otimes}, c^{(1)}, 0.5, c^{(2)}, 0.2\right)$, since Agent 1 (resp., Agent 2) was traveling along its current edge for $d = 0.6 - 0.1 = 0.5$ (resp., $d = 0.6 - 0.4 = 0.2$) time units.

Graph expansion

Once the cross-product graph is constructed, we can run a single-agent path planning algorithm like A* on it, then retrieve the individual moves from the cross-product solution plan. Thus, we first need to instantiate the cross-product graph: we generate it lazily from the starting cross-product vertex – which is simply the pair of starting vertices of each agent – and expand it on-demand (e.g., when A* looks at successor vertices of the current closed vertex) by constructing outgoing cross-product edges from a given current cross-product vertex. Algorithm 1 formalizes this lazy cross-product graph expansion.

The algorithm tests different cases corresponding to whether each agent i is located at a vertex of its own graph (i.e., $\alpha^{(i)} = 1$), where a decision regarding the next successor vertex to target has to be taken, or if it is located between two successive vertices of its own graph (i.e., $0 < \alpha^{(i)} < 1$), where it can only continue to execute the current move to the next targeted successor vertex. When an agent is at a decision point, we iterate over the edges of its own graph whose incoming vertex is equal to the outgoing vertex of the previously executed edge (e.g., Lines 3 and 19). Then, we compare the remaining durations of the edges of each agent in their own graphs (e.g., Lines 4 and 20), in order to know which agent will reach its next targeted successor vertex first and update the completion ratio of the other agent – which may not reach its next targeted successor vertex in the same move – accordingly (e.g., Lines 7, 23 and 27). Note that the case $\left(1-\alpha^{(1)}\right) \cdot d^{(1)}\left(e^{(1)}\right) < \left(1-\alpha^{(2)}\right) \cdot d^{(2)}\left(e^{(2)}\right)$ is impossible, since at least one agent reaches its next targeted successor vertex at each graph expansion. When both agents are at a decision point (i.e., Line 2) we check whether a formation is feasible, in which case we add cross-product edges corresponding to the two agents traveling along the same common path with appropriate formation costs and durations (Lines 13 to 17).

Conditions on initial and goal vertices

Special attention specific to the CCPP problem must be paid to the initial and goal vertices. If we do not allow an agent that has reached its goal vertex to wait for the other agent to reach its own goal, it significantly reduces the chance of finding a solution because it means that we would force the two agents to reach their goal vertices at the same time. However, such a condition is not desired in practice, so we allow each agent to wait for the other agent at its goal vertices by adding in its own graph self-transitions from its goal vertices labelled with the durations of all of the edges of the other agent. Doing so, whatever the location of the other agent in its own graph, the agent at the goal will always transit again to its goal for each atomic move of the other agent. Since this "goal holding" property might be undesirable in very specific applications, we activate or deactivate it according to a test function named `can_hold`. This logic, which is implemented in Lines 4 to 6 of Algorithm 2, is also valid for initial vertices since holding the initial vertex of an agent while the other has begun to move increases the chance of finding a (spatial and temporal) synchronization point where both agents can begin to move in formation. Indeed, meeting at a given rendezvous point at a given time can be achieved by controlling each agent's edge speed or initial vertex starting time. Even if better formation plans can be found by allowing one agent to hold its initial vertex for an amount of time dictated by the other agent's initial transition duration, it is of course less optimal than an approach that would allow a free continuous holding duration irrespective of the other agent's initial transition duration. However, such an approach would be much harder computationally, since it would require continuous optimization variables to be considered, whereas our approach allows us to keep reasoning about only discrete optimization variables.

Moreover, some applications might require the two agents to start at predefined time points, or, in other words, that the difference of their starting times be constrained to a predefined value. This is, for instance, the case of two flights that might want to fly somewhere in formation in order to save fuel burn given the current day's weather conditions but whose takeoff times are set several months in advance and cannot be significantly changed at the last minute. To do this, we test for a condition `must_shift` that, if true, adds a predefined amount of time to the duration of the first transition of each agent from their initial vertices (see Lines 7 to 9 of Algorithm 2). Note that this option is incompatible with the "holding" one, since an agent cannot start within a predefined time interval from the other agent's start while holding its starting vertex.

[1] In this example $\sigma^{(1)}(2) = \sigma^{(2)}(3)$ because the agents travel in formation from their previous vertices, and their graphs are the same. However, note that in general their graphs do not need to be the same, so their edges can be different even in formation.

Algorithm 1 – Cross-product graph expansion

```
    Data: $v^{\otimes} = (e^{(1)}, \alpha^{(1)}, e^{(2)}, \alpha^{(2)}) \in V^{\otimes}$: cross product vertex
    Result: $[e^{\otimes}]_{e^{\otimes} \in E^{\otimes}}$ : list of cross product edges outgoing from $(e^{(1)}, \alpha^{(1)}, e^{(2)}, \alpha^{(2)})$
 1  successors ← list()
 2  if $\alpha^{(1)} = 1$ and $\alpha^{(2)} = 1$ then
 3      for $(\overline{e}^{(1)}, \overline{e}^{(2)}) \in E^{(1)} \times E^{(2)} : e^{(1)}$.out() $= \overline{e}^{(1)}$.in() and $e^{(2)}$.out() $= \overline{e}^{(2)}$.in() do
 4          if $d^{(1)}(\overline{e}^{(1)}) < d^{(2)}(\overline{e}^{(2)})$ then
 5              $\overline{c} \leftarrow c^{(1)}(\overline{e}^{(1)}) + c^{(2)}(\overline{e}^{(2)}) \cdot \frac{d^{(1)}(\overline{e}^{(1)})}{d^{(2)}(\overline{e}^{(2)})}$;
 6              $\overline{d} \leftarrow d^{(1)}(\overline{e}^{(1)})$;
 7              $\overline{\alpha}^{(1)} \leftarrow 1$; $\overline{\alpha}^{(2)} \leftarrow \frac{d^{(1)}(\overline{e}^{(1)})}{d^{(2)}(\overline{e}^{(2)})}$;
 8          else
 9              Symmetry of Lines 5 to 7
10          $\overline{v}^{\otimes} \leftarrow V^{\otimes}$.add($\overline{e}^{(1)}, \overline{\alpha}^{(1)}, \overline{e}^{(2)}, \overline{\alpha}^{(2)}$);
11          $\overline{e}^{\otimes} \leftarrow E^{\otimes}$.add($v^{\otimes}, \overline{v}^{\otimes}, \overline{c}, \overline{d}$);
12          successors.add($\overline{e}^{\otimes}$);
13          if can_make_formation($\overline{e}^{(1)}, \overline{e}^{(2)}$) then
14              $\overline{c} \leftarrow c^{\otimes}(\overline{e^{(1)}}, \overline{e}^{(2)})$; $\overline{d} \leftarrow d^{\otimes}(\overline{e}^{(1)}, \overline{e}^{(2)})$;
15              $\overline{v}^{\otimes} \leftarrow V^{\otimes}$.add($\overline{e}^{(1)}, 1, \overline{e}^{(2)}, 1$);
16              $\overline{e}^{\otimes} \leftarrow E^{\otimes}$.add($v^{\otimes}, \overline{v}^{\otimes}, \overline{c}, \overline{d}$);
17              successors.add($\overline{e}^{\otimes}$);
18  else if $\alpha^{(1)} = 1$ and $\alpha^{(2)} < 1$ then
19      for $\overline{e}^{(1)} \in E^{(1)} : e^{(1)}$.out() $= \overline{e}^{(1)}$.in() do
20          if $(1 - \alpha^{(2)}) \cdot d^{(2)}(e^{(2)}) < d^{(1)}(\overline{e}^{(1)})$
21              $\overline{c} \leftarrow (1 - \alpha^{(2)}) \cdot \left( c^{(1)}(\overline{e}^{(1)}) \cdot \frac{d^{(2)}(e^{(2)})}{d^{(1)}(\overline{e}^{(1)})} + c^{(2)}(e^{(2)}) \right)$
22              $\overline{d} \leftarrow (1 - \alpha^{(2)}) \cdot d^{(2)}(e^{(2)})$
23              $\overline{\alpha}^{(1)} \leftarrow \frac{(1 - \alpha^{(2)}) \cdot d^{(2)}(e^{(2)})}{d^{(1)}(\overline{e}^{(1)})}$; $\overline{\alpha}^{(2)} \leftarrow 1$
24          else
25              $\overline{c} \leftarrow c^{(1)}(\overline{e}^{(1)}) + c^{(2)}(e^{(2)}) \cdot \frac{d^{(1)}(\overline{e}^{(1)})}{d^{(2)}(e^{(2)})}$
26              $\overline{d} \leftarrow d^{(1)}(\overline{e}^{(1)})$
27              $\overline{\alpha}^{(1)} \leftarrow 1$; $\overline{\alpha}^{(2)} \leftarrow \overline{\alpha}^{(2)} + \frac{d^{(1)}(\overline{e}^{(1)})}{d^{(2)}(e^{(2)})}$
28          $\overline{v}^{\otimes} \leftarrow V^{\otimes}$.add($\overline{e}^{(1)}, \overline{\alpha}^{(1)}, e^{(2)}, \overline{\alpha}^{(2)}$);
29          $\overline{e}^{\otimes} \leftarrow E^{\otimes}$.add($v^{\otimes}, \overline{v}^{\otimes}, \overline{c}, \overline{d}$);
30          successors.add($\overline{e}^{\otimes}$);
31  else if $\alpha^{(1)} < 1$ and $\alpha^{(2)} = 1$ then
32      Symmetry of Lines 19 to 27
33  return successors;
```

Algorithm 2 – Make initial and goal conditions

```
Data: $v_s^{(i)}, i \in \{1,2\}$ : Agent $i$'s starting vertex; $v_g^{(i)}, i \in \{1,2\}$ : Agent $i$'s goal vertex
for $i \in \{1,2\}$ do
    $j \leftarrow i$ mod $2+1$;
    for $v^{(i)} \in \{v_s^{(i)}, v_g^{(i)}\}$ do
        if can_hold( $v^{(i)}$ ) then
            for $e^{(j)} \in E^{(j)}$ do
                $E^{(i)}$ .add( $v^{(i)}, v^{(i)}, 0, d^{(j)}(e^{(j)})$ );
    if must_shift( $v_s^{(i)}$ ) then
        for $e^{(i)} \in E^{(i)} : e^{(i)}$.in() = $v_s^{(i)}$
            $d^{(i)}[e^{(i)}] \leftarrow d^{(i)}[e^{(i)}]$ + shift ($v_s^{(i)}$);
```

Global algorithm

The optimal CCPP solving procedure is presented in Algorithm 3. It proceeds in three logical steps:

1. Establish initial and goal conditions using Algorithm 3 (Line 1);
2. Construct the cross-product initial and goal vertices and running A* by lazily expanding the cross-product graph on closed vertices using Algorithm 2 (Lines 2 to 4);
3. Extract the individual plans of each agent from the cross-product plan by looking at cross-product vertices whose completion ratios are equal to 1, meaning that the corresponding agent is located at a decision vertex but not in-between two successive vertices of its own graph (Lines 5 to 9).

Decoupled spatial/temporal algorithm

The complexity of the optimal algorithm lies in the coupled spatio-temporal space: in order to synchronize the agents at potential formation points, both spatially and temporally, it has to memorize cross-product vertices containing the agents' original graph vertices and completion ratios of traveling along these edges. By noting as $D^{(i)}$ an upper bound on the number of different durations of each travel between two successive vertices of Agent i's own graph (i.e., the maximum number of edges with the same given incoming and outgoing vertices) the potential number of cross-product vertices explored by A* is $|E^{(1)}| \cdot D^{(2)} \cdot |E^{(2)}| \cdot D^{(1)} = |V^{(1)}|^2 \cdot D^{(2)} \cdot |V^{(2)}|^2 \cdot D^{(1)}$. In the case where both agents share the same graph and are identical (implying same possible edge durations) it amounts to visiting at most $|V|^4 \cdot D^2$ vertices, which is intractable even for very small instances.

A simple idea consists in decoupling the problems of synchronizing the agents spatially and temporally. First, we aim at finding a pair of vertices for each agent where they can begin a formation by discarding all temporal information on edge durations. It is achieved by running lazy A* with a vertex expansion procedure presented in Algorithm 4, which is composed of two phases: (1) filter out durations from each agent's graph edges by replacing each set of edges with the same incoming and outgoing vertices by a single edge without duration and whose cost is averaged on-the-fly over all of the costs of the replaced edges (Lines 1 to 10); (2) generate successor cross-product vertices from the filtered edges while checking for the possibility of beginning a formation, in which case the formation cost is appropriately set (Lines 11 to 15). Crucially, a self-transition from each agent's current vertex is added, in order to simulate the fact that, in reality, an agent might not have reached its next successor vertex when the other has indeed reached its own next vertex (Line 10). This trick actually allows the agents to reach a common formation point even if their individual numbers of atomic moves to reach it are different (see Figure 2).

Algorithm 3 – Optimal CCPP algorithm

```
   Data: $v_s^{(i)}$ : Agent $i$'s start node; $v_g^{(i)}$ : Agent $i$'s goal node; $\mathcal{G}^{(i)} = (V^{(i)}, E^{(i)})$ : Agent $i$'s move graph; $c^{\otimes} : E^{\otimes} \rightarrow \mathbb{R}_+$: cross-product cost function; $h^{\otimes} : V^{\otimes} \times V^{\otimes} \rightarrow \mathbb{R}_+$ : cross-product heuristic function
   Result: $\pi^{(i)}, i \in \{1,2\}$: optimal path from $v_s^{(i)}$ to $v_g^{(i)}$
 1 make_initial_and_goal_conditions();
 2 $v_s^{\otimes} \leftarrow ((v_s^{(1)}, v_s^{(1)}), 1, (v_s^{(2)}, v_s^{(2)}), 1)$;
 3 $v_g^{\otimes} \leftarrow ((v_g^{(1)}, v_g^{(1)}), 1, (v_g^{(2)}, v_g^{(2)}), 1)$;
 4 $\pi^{\otimes} \leftarrow$ lazy_astar($v_s^{\otimes}, v_g^{\otimes}, h^{\otimes}$) using cross_product_graph_expansion;
 5 $\pi^{(1)} \leftarrow$; list(); $\pi^{(2)} \leftarrow$ list();
 6 for $e^{\otimes} \in \pi^{\otimes}$ do
 7     $(e_{out}^{(1)}, \alpha_{out}^{(1)}, e_{out}^{(2)}, \alpha_{out}^{(2)}) \leftarrow e^{\otimes}$ .out();
 8     if $\alpha_{out}^{(1)} = 1$ then $\pi^{(1)}$ .add( $e_{out}^{(1)}$ );
 9     if $\alpha_{out}^{(2)} = 1$ then $\pi^{(2)}$ .add( $e_{out}^{(2)}$ );
10 return $(\pi^{(1)}, \pi^{(2)})$;
```

Algorithm 4 – Spatial graph expansion

```
Data: $v^{\otimes} = (v^{(1)}, v^{(2)}) \in V^{(1)} \times V^{(2)}$ : cross-product vertex without time info
Result: $[e^{\otimes}]_{e^{\otimes} \in \left(V^{(1)} \times V^{(2)}\right)^2 \times \mathbb{R}_+}$ : list of cross-product edges outgoing from $v^{\otimes}$

 1  for $i \in \{1,2\}$ do
 2      $\tilde{E}^{(i)} \leftarrow$ list(); $n \leftarrow$ map();
 3      for $e^{(i)} \in E^{(i)} : e^{(i)}$.in() $= v^{(i)}$ do
 4          if $\forall e \in \tilde{E}^{(i)}, e^{(i)}$.out() $\neq e$.out() then
 5              $\tilde{E}^{(i)}$.add($e^{(i)}$.in(), $e^{(i)}$.out(), $c^{(i)}(e^{(i)})$);
 6              $n\left[e^{(i)}\text{.out()}\right] \leftarrow 1$;
 7          else
 8              $\tilde{c}^{(i)}\left(e^{(i)}\right) \leftarrow \frac{1}{n\left[e^{(i)}\text{.out()}\right]+1}\left(n\left[e^{(i)}\text{.out()}\right] \cdot \tilde{c}^{(i)}\left(e^{(i)}\right) + c^{(i)}\left(e^{(i)}\right)\right)$;
 9              $n\left[e^{(i)}\text{.out()}\right] \leftarrow n\left[e^{(i)}\text{.out()}\right]+1$;
10      $\tilde{E}^{(i)}$.add($v^{(i)}, v^{(i)}, 0$);
11  $successors \leftarrow$ list();
12  for $\tilde{e}^{(1)} \in \tilde{E}^{(1)}$ and $\tilde{e}^{(2)} \in \tilde{E}^{(2)}$ do
13      $successors$.add( $\tilde{e}^{(1)}, \tilde{e}^{(2)}, \tilde{c}^{(1)}\left(\tilde{e}^{(1)}\right) + \tilde{c}^{(2)}\left(\tilde{e}^{(2)}\right)$ );
14      if can_make_formation( $\tilde{e}^{(1)}, \tilde{e}^{(2)}$ ) then
15          $successors$.add( $\tilde{e}^{(1)}, \tilde{e}^{(2)}, c^{\otimes}\left(\tilde{e}^{(1)}, \tilde{e}^{(2)}\right)$ );
16  return $successors$
```

The second phase consists in planning for the agents' coordination in the temporal space only by running lazy A* with a temporal graph expansion schema, which constrains the spatial vertex transitions to the plans found during the first spatial planning phase, as formalized in Algorithm 5. Thus, the cross-product vertex is composed of the pair of indices of the moves in the spatial plans currently executed by the agents, which is sufficient to track each agent's plan execution and to obtain the spatial edge corresponding to the current plan indices (Lines 2 to 6). Once this edge has been extracted, we obtain all of the original time-labelled edges with the same incoming and outgoing vertices, in order to retrieve timing information that was lost during the first spatial synchronization phase (Lines 7 to 9). Finally, we generate successor cross-product vertices from the reconstructed time-labelled edges while checking for the possibility of beginning a formation, in which case the formation cost is appropriately set (Lines 10 to 14). Note that, at this stage, time labels are part of the reconstructed vertices so we can check whether it is possible for the two agents to simultaneously reach a potential formation vertex.

Algorithm 5 – Temporal graph expansion

```
Data: $\{\tilde{\pi}^{(i)}\}_{i \in \{1,2\}}$: Agent $i$'s spatial plan w/o timing information; $v^{\otimes} = \left(k_1, k_2\right)$: cross-product vertex containing indexes of current
      spatial vertices in Agents 1 and 2's plans
 1  for $i \in \{1,2\}$ do
 2      $v^{(i)} \leftarrow \tilde{\pi}^{(i)}\left[k_i\right]$;
 3      if $k_i \neq \tilde{\pi}^{(i)}$.length() $-1$ then
 4          $\overline{v}^{(i)} \leftarrow \tilde{\pi}^{(i)}\left[k_i + 1\right]$;
 5      else
 6          $\overline{v}^{(i)} \leftarrow v^{(i)}$;
 7      $\tilde{E}^{(i)} \leftarrow$ list();
 8      for $e^{(i)} \in E^{(i)} : e^{(i)}$.in() $= v^{(i)}$ and $e^{(i)}$.out() $= \overline{v}^{(i)}$ do
 9          $\tilde{E}^{(i)}$.add($e^{(i)}$);
10  $successors \leftarrow$ list();
11  for $\tilde{e}^{(1)} \in \tilde{E}^{(1)}$ and $\tilde{e}^{(2)} \in \tilde{E}^{(2)}$ do
12      $successors$.add( $\tilde{e}^{(1)}, \tilde{e}^{(2)}, \tilde{c}^{(1)}\left(\tilde{e}^{(1)}\right) + \tilde{c}^{(2)}\left(\tilde{e}^{(2)}\right)$ );
13      if can_make_formation( $\tilde{e}^{(1)}, \tilde{e}^{(2)}$ ) then
14          $successors$.add( $\tilde{e}^{(1)}, \tilde{e}^{(2)}, c^{\otimes}\left(\tilde{e}^{(1)}, \tilde{e}^{(2)}\right)$ );
15  return $successors$
```

Algorithm 6 – Spatio-temporal decoupled CCPP

```
Data: $v_s^{(i)}$ : Agent $i$'s start node; $v_g^{(i)}$ : Agent $i$'s goal node; $\mathcal{G}^{(i)} = \left(V^{(i)}, E^{(i)}\right)$: Agent $i$'s move graph; $c^{\otimes} : E^{\otimes} \to \mathbb{R}_+$ : cross-product cost
      function; $h^{\otimes} : V^{\otimes} \times V^{\otimes} \to \mathbb{R}_+$: cross-product heuristic function
Result: $\pi^{(i)}, i \in \{1,2\}$ : optimal path from $v_s^{(i)}$ to $v_g^{(i)}$
make_initial_and_goal_conditions();
$v_s^{\otimes} \leftarrow \left(v_s^{(1)}, v_s^{(2)}\right)$
$v_g^{\otimes} \leftarrow \left(v_g^{(1)}, v_g^{(2)}\right)$
$\pi_{spatial}^{\otimes}$ ← lazy_astar($v_s^{\otimes}, v_g^{\otimes}, h^{\otimes}$) using spatial_graph_expansion;
$\{\tilde{\pi}^{(i)}\}_{i \in \{1,2\}}$ ← extract_individual_plans($\pi_{spatial}^{\otimes}$);
$\tilde{v}_s^{\otimes} \leftarrow (0,0)$;
$\tilde{v}_g^{\otimes} \leftarrow \big(\tilde{\pi}^{(1)}$.length()$-1, \tilde{\pi}^{(2)}$.length()$-1\big)$;
$\pi_{temporal}^{\otimes}$ ← lazy_astar($\tilde{v}_s^{\otimes}, \tilde{v}_g^{\otimes}, h^{\otimes}$) using temporal_graph_expansion($\{\tilde{\pi}^{(i)}\}_{i \in \{1,2\}}$);
return extract_individual_plans($\pi_{temporal}^{\otimes}$);
```

The global decoupled spatio-temporal procedure is presented in Algorithm 6. The pseudo-code is quite obvious given the previous explanations of the successive spatial and temporal synchronization phases. Whereas this decoupled schema certainly does not bring optimal solutions to the CCPP problem, it allows us to solve large problems with satisfactory quality in a very small fraction of the time spent by the optimal coupled algorithm.

Heuristics

Given that our algorithms rely on A*, we need a heuristic function to guide the search in the cross-product graph. Designing a generic admissible and informative heuristic is especially challenging because the cost function changes when the agents move in formation and we do not know in advance when and where they will begin or end a formation pattern. A brute-force method would consist in iterating over all possible pairs of each agent's vertices representing joining and breaking formation points, in order to ease heuristic estimate computations. However, we would spend too much time iterating over too numerous pairs, which would ruin the benefit of using heuristic functions.

Admissible heuristic

A simple admissible but non-informative heuristic estimate consists in summing individual heuristic estimates for each agent in their own graphs using formation costs $c^{\otimes}$, even if they do not travel in formation. Any admissible heuristic for single-agent path planning (e.g., Euclidean distance, Manhattan distance, etc.) can be used to calculate the individual heuristic estimates but considering formation costs instead of standard single-agent costs (i.e., like in the case of agents travelling in formation from their starting to ending locations). This allows us to obviate the difficult question of when and where they join together or break the formation, but we lose the crucial information that their individual moving costs are actually generally much higher when they do not travel in formation. Formally, we note as $h^{(i)} : V^{(i)} \times V^{(i)} \to \mathbb{R}_+$ a heuristic function defined on Agent i's vertices in its own graph, such that $h^{(i)}\left(v_A^{(i)}, v_B^{(i)}\right) \leqslant C^{\otimes,i}\left(v_A^{(i)}, v_B^{(i)}\right)$ where $C^{\otimes,i}\left(v_A^{(i)}, v_B^{(i)}\right)$ is Agent i's contribution to the duet cost-to-go function if it travels in formation with the other agent from the starting vertex $v_A^{(i)}$ to the target vertex $v_B^{(i)}$. Said differently, $h^{(i)}$ is computed with edge costs assumed to be all equal to Agent i's contribution to $c^{\otimes}$, even if it does not travel in formation. Therefore, an admissible heuristic is $h_{adm}^{\otimes} = h^{(1)} + h^{(2)}$.

Informative heuristic

In order to use the important information that the cost function is higher when the agents do not move in formation, we try to approximately guess with simple geometric reasoning where the agents will make a formation and further break it. This simple geometric method, depicted in Figure 3 and formalized in Algorithm 7, assumes the vertices of the agents' own graphs to be elements of a Euclidean space. Given a pair of starting vertices $(v_A^{(1)}, v_A^{(2)})$ and target vertices $(v_B^{(1)}, v_B^{(2)})$ we compute their respective barycenters $\tilde{v}_A$

Algorithm 7 – CCPP heuristic on Euclidean space

```
Data: $\{\mathrm{v}_A^{(i)}\}_{i \in \{1,2\}}$ : Agent $i$'s starting point (vector); $\{\mathrm{v}_B^{(i)}\}_{i \in \{1,2\}}$ : Agent $i$'s target point (vector); $\varepsilon > 0$ : spatial precision between possible joining or breaking points
Result: Heuristic estimate of the CCPP problem from joint starting point to joint target point
$\tilde{\mathrm{v}}_A \leftarrow \frac{1}{2}\left(\mathrm{v}_A^{(1)} + \mathrm{v}_A^{(2)}\right)$ ; $\tilde{\mathrm{v}}_B \leftarrow \frac{1}{2}\left(\mathrm{v}_B^{(1)} + \mathrm{v}_B^{(2)}\right)$ ;
$N \leftarrow argmin\left\{n \in \mathbb{N}_+ : n \geqslant \frac{\|\tilde{\mathrm{v}}_B - \tilde{\mathrm{v}}_A\|}{\varepsilon}\right\}$ ;
for $i \leftarrow 0$ to $N$ do
    $\tilde{\mathrm{v}}_i \leftarrow \tilde{\mathrm{v}}_s + \frac{i}{N}\left(\tilde{\mathrm{v}}_B - \tilde{\mathrm{v}}_A\right)$ ;
$h^{\otimes}\left((\mathrm{v}_A^{(1)}, \mathrm{v}_A^{(2)}), (\mathrm{v}_B^{(1)}, \mathrm{v}_B^{(2)})\right) \leftarrow min_{0 \leqslant i \leqslant N/2}\left\{\omega^{\otimes}\left\|\tilde{\mathrm{v}}_{N-i} - \tilde{\mathrm{v}}_i\right\| + \omega^{(1)}\left(\left\|\tilde{\mathrm{v}}_i - \mathrm{v}_A^{(1)}\right\| + \left\|\tilde{\mathrm{v}}_{N-i} - \mathrm{v}_B^{(1)}\right\|\right) + \omega^{(2)}\left(\left\|\tilde{\mathrm{v}}_i - \mathrm{v}_A^{(2)}\right\| + \left\|\tilde{\mathrm{v}}_{N-i} - \mathrm{v}_B^{(2)}\right\|\right)\right\}$ ;
return $h^{\otimes}\left((\mathrm{v}_A^{(1)}, \mathrm{v}_A^{(2)}), (\mathrm{v}_B^{(1)}, \mathrm{v}_B^{(2)})\right)$ ;
```

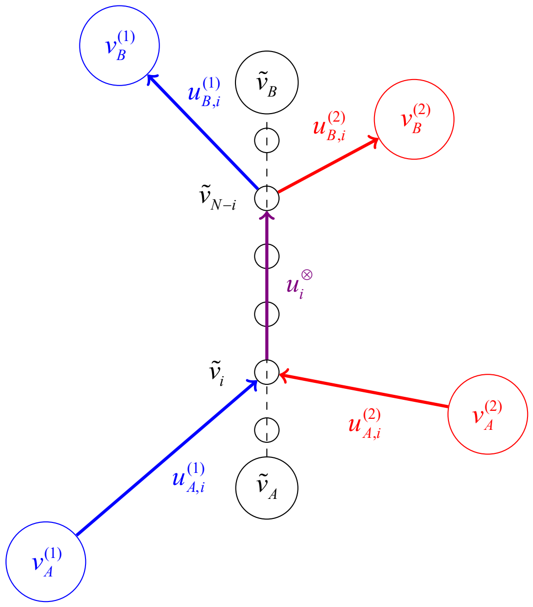


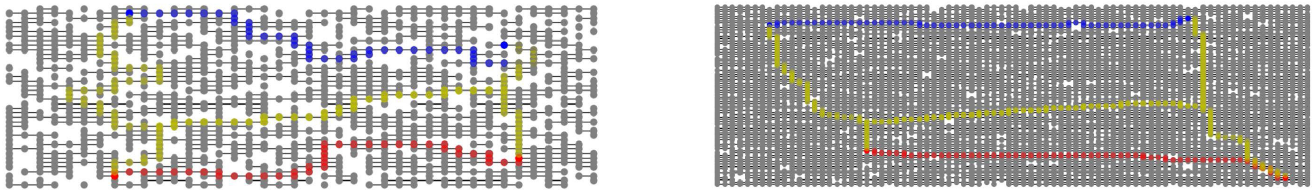
Figure 3 – Heuristic estimate of the CCPP problem in Euclidean spaces from a joint position $v_A^{\otimes}=(v_A^{(1)},v_A^{(2)})$ to $v_B^{\otimes}=(v_B^{(1)},v_B^{(2)})$: $h^{\otimes}\left(v_A^{\otimes},v_B^{\otimes}\right)= min_{0\leqslant i\leqslant N/2}\left\{\omega^{\otimes}\left\|u_i^{\otimes}\right\|+\omega^{(1)}\left(\left\|u_{A,i}^{(1)}\right\|+\left\|u_{B,i}^{(1)}\right\|\right)+\omega^{(2)}\left(\left\|u_{A,i}^{(2)}\right\|+\left\|u_{B,i}^{(2)}\right\|\right)\right\}$, where ω's represent speed-dependent moving costs per length unit as functions of duration-labelled edges

and $\tilde{v}_B$ (Line 1) and assume that the agents will travel in formation somewhere along the segment joining these barycenters – which is obviously not guaranteed, but hopefully close to the optimal formation segment, preventing us from proving the admissibility of this heuristic estimate. Then we iterate over possible joining and breaking points along the median segment $[\tilde{v}_A;\tilde{v}_B]$, assumed to be symmetric for simplicity (Lines 2 to 4). Finally (Line 5), we search for the minimum estimate over these possible joining/breaking patterns of the sum of the individual contributions of the agents when not in formation from their initial starting vertices to the joining point and from the breaking point to their target vertices (blue and red segments in Figure 3), and of the heuristic formation cost-to-go from the joining point to the breaking point (violet segment in Figure 3).

Experiments

Navigation grids

In this section we propose to experimentally compare the different variants of our CCPP solving algorithms on navigation grids that are Euclidean spaces, so that we can test our informative geometric heuristic. As shown in Figure 1, we experimented with various random grids having various sizes, obstacle densities and formation moving costs. Table 2, whose caption defines shortcut names for the tested variants, summarizes the results. We implemented the algorithms in pure Python and run the tests on Intel's core i7 with 2.80 GHz CPUs and 16 Go of RAM (the tests used at most 1.5 Go of RAM). For each



(a) Small sparse grid

(b) Large dense grid

Figure 4 – Formation path planning experimented on randomly generated grids with various sizes and obstacle densities. Individual optimal agents' paths regardless of the CCPP problem are in red and blue, while CCPP agent paths are in yellow.

	PATHS TEAM COST				NUMBER OF EXPLORED A* NODES				CPU TIME (seconds)			
Problem	CA	CI	DA	DI	CA	CI	DA	DI	CA	CI	DA	DI
NG-5-5-75	**1.8**	2.18	**1.8**	**1.8**	22525	11577	422	**296**	2.67	0.888	0.0377	**0.0328**
NG-5-5-50	**1.1**	**1.1**	**1.1**	**1.1**	6819	2456	164	**86**	0.319	0.0549	0.0169	**0.013**
NG-10-10-75	**2.94**	2.99	4.4	2.99	582997	33464	2669	**784**	85.5	1.92	0.658	**0.263**
NG-10-10-50	**3.1**	3.2	3.43	3.5	512555	159521	2107	**1620**	77	16	0.6	**0.47**
NG-20-20-75	—	**7.03**	9.7	8.23	—	757991	17775	**8004**	—	99.8	6.35	**3.59**
NG-20-20-50	—	**7.13**	8.05	7.58	—	1139752	11859	**4066**	—	151	11.2	**5.83**
NG-40-40-75	—	—	17	**16.7**	—	—	47172	**14167**	—	—	153	**49.8**
NG-40-40-50	—	—	16.9	**14.3**	—	—	21410	**18504**	—	—	184	**98.7**
NG-80-80-75	—	—	—	**34.1**	—	—	—	**97244**	—	—	—	**732**
NG-80-80-50	—	—	—	**40.6**	—	—	—	**110855**	—	—	—	**1140**

Table 1 – Navigation grid experiments: comparison of solution path team costs, numbers of nodes explored by A* in the cross-product graph and CPU times, for the coupling algorithm using the admissible generic heuristic (CA) or the informative geometric one (CI), and similarly for the decoupling algorithm (DA and DI). Problems are noted as NG-X-Y-P, where X and Y stand for the x and y dimensions of the grid, and P represents the cost reduction percentage of edges where the two agents move in formation side by side, in comparison with non-formation edges (i.e., $c^{\otimes}=(1-P)/100\times(c^{(1)}+c^{(2)})$).

algorithm we time out the search at 3 minutes of computation. Note that the only optimal algorithm is CA, i.e., the coupling algorithm equipped with the admissible heuristic.

Coupling algorithm vs. decoupling algorithm

The coupling algorithms can only solve the first problems, since they explore significantly more nodes during A* graph search than the decoupling algorithms do: up to 280 times more for the same heuristics. The best version of the coupling algorithm times out after the 7th problem. The decoupling algorithms' solution quality is at most at 12% of the optimal solution, even finding optimal ones on small problems.

Admissible heuristic vs. informative heuristic

As expected, the informative heuristic allows the coupling or decoupling algorithms to solve more problems, since it provides more accurate estimates of the team cost-to-go. However, it is not admissible, which degrades the solution quality of the coupling algorithm in comparison with the admissible heuristic. However, interestingly, we observe the inverse behavior for the decoupling algorithm, which seems to indicate that the negative impact on the solution quality of decoupling is higher than when using a non-admissible heuristic.

Impact of formation edge cost reduction

Since higher cost reductions tend to incentivize agents more to make formations, we expect the CCPP problem to be more difficult to solve with lower cost reductions, since more cross-product nodes should be explored before finding potential joining points that can provide some benefits. For reasons that we do not fully understand yet, this is partially observed with the informative heuristic only. More nodes are systematically explored by A* with 75% formation cost reduction using the admissible heuristic for both algorithms.

Formation Flight

We ran the DI algorithm, i.e., the only scalable one, on formation flight problems for hundreds of flights constrained to fly over official airway graphs published by aviation authorities, which contain more than 30000 of vertices at different altitudes. Each aircraft is assigned a route, which is defined by a pair of departure and destination airports, but the flight paths have to be optimized in the 4D space composed of the aircraft's 3D waypoint positions and the times when it flies through each waypoint. In addition to optimizing the flight routes for a given pair of leader-follower aircraft, we must choose the best leader-follower pairs among all of the possible pairs. To this end, we note as $N \in \mathbb{N}_+$ the number of aircraft to pair and as $\mathcal{F} \subseteq [1;N]^2$ the possible leader-follower pairs for which the CCPP problem returns a feasible solution. For $i \in [1;N]$, we note as $\mathcal{F}^F(i) = \{ j \in [1;N], (i,j) \in \mathcal{F} \}$ the set of possible followers that can pair with i. Symmetrically, for all $j \in [1;N]$, $\mathcal{F}^L(j) = \{ i \in [1;N], (i,j) \in \mathcal{F} \}$ is the set of possible leaders that can pair with j. Finally, we note c_{ij} the global cost of the flights of a leader i and follower j flying in formation on a subset of their routes (i.e., solution to the CCPP problem), and c_i and c_j the costs of their solo flights if they had flown without pairing at some point.

The solution is computed in two successive phases. First, we compute all of the possible leader-follower formation routes by running the CCPP algorithm on all possible pairs of aircraft, as well as all of the solo flight routes. This phase allows us to compute the set $\mathcal{F}$ of feasible formation flights, to fill in the formation costs c_{ij} for all $(i,j) \in \mathcal{F}$, and the solo flight costs c_i for all $i \in [1;N]$. Second, we compute the best possible pair assignments by running the following integer linear program whose optimization variables are $f_{ij} \in \{0,1\}$:

$$maximize : \sum_{(i,j)\in\mathcal{F}} f_{ij}\left(c_i + c_j - c_{ij}\right) \tag{1}$$

$$subject\ to : \forall i \in [1;N], \sum_{j \in \mathcal{F}^F(i)} f_{ij} \leqslant 1 \tag{2}$$

$$\forall j \in [1;N], \sum_{i \in \mathcal{F}^L(j)} f_{ij} \leqslant 1 \tag{3}$$

$$\forall (i,j) \in \mathcal{F}, f_{ij} + \sum_{k \in \mathcal{F}^L(i)} f_{ki} \leqslant 1 \tag{4}$$

$$\forall (i,j) \in \mathcal{F}, f_{ij} + \sum_{k \in \mathcal{F}^F(j)} f_{jk} \leqslant 1 \tag{5}$$

Decision variables f_{ij} are equal to 1 whenever Leader i is paired with Follower j. Constraint 2 (resp. 3) means that each possible leader i (resp. follower j) can be paired with a single follower j (resp. leader i). Constraint 4 (resp. 5) means that a given leader (resp. follower) cannot be the follower (resp. leader) of another aircraft. The objective function is to maximize the gain of flying in formations, i.e., the difference between the sum of individual solo flight costs $c_i + c_j$ and the formation cost c_{ij} whenever a formation f_{ij} is selected. An excerpt of the resulting formation flights over the Atlantic Ocean is shown in Figure 5. Note that aircraft have to fly though so-called *North Atlantic Tracks,* which are very beneficial to formation flights since solo flights take those tracks anyway.

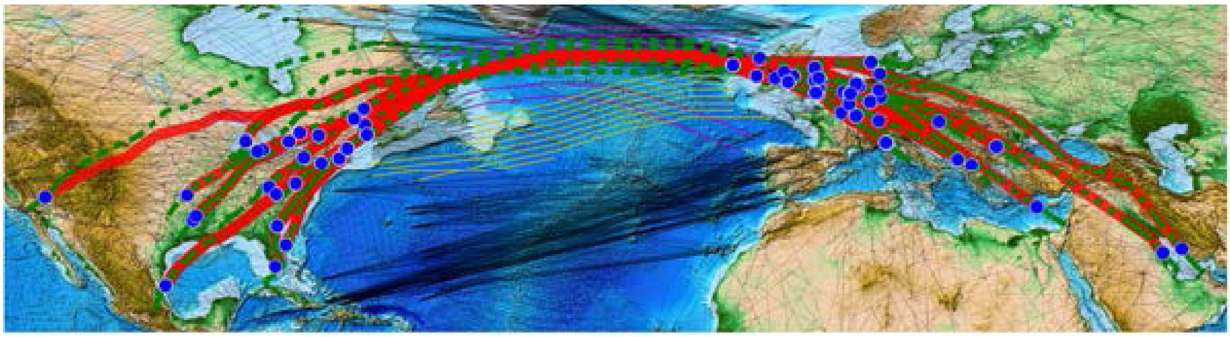


Figure 5 – Formation Flight test bench: common formation paths are shown in solid red, solo (non-cooperative) paths appear in dashed green, the airway graph is shown in solid gray, and airports appear as blue dots.

Discussion

Our experimental results show that only the decoupling algorithm equipped with informative heuristics like the geometric one that we proposed can reasonably scale. However, we also observed a loss in quality, which can be of up to 12 % of the optimal team cost on some benchmarks even if the loss is in general much lower. These results encourage us to conduct further research on improving the decoupling algorithm by trading a little computation time off for quality regaining. Possible research directions would consist in alternatively iterating between the spatial and temporal synchronization phases instead of a single iteration that freezes the spatial phase independently from the temporal phase's solution constraints as we currently do. Another important line of research would look at improving heuristics: making the informative geometric heuristic admissible, as well as designing a generic informative heuristic that differentiates the costs of the pre- and post-formation paths of the agent from the formation common path – like our geometric heuristic, but in general state spaces that are not necessarily Euclidean. Finally, extension of the CCPP problem to continuous edge duration spaces would require the mixing of A* search in the spatial space, followed by scheduling techniques in the temporal space to control the speed of frozen spatial moves as a true continuum of values instead of continuous values chosen in a discrete set [14]. However, note that such settings are not always desirable, especially not in formation flight, as explained in the introduction ■

References

[1] Y. CHEN, J. YU, X. SU, G. LUO - *Path Planning for Multi-UAV Formation.* Journal of Intelligent & Robotic Systems, 77(1):229-246, 2015.

[2] H. CHOSET, K. M. LYNCH, S. HUTCHINSON, G. A. KANTOR, W. BURGARD, L. E. KAVRAKI, S. THRUN - *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, Pittsburgh, PA, 2005.

[3] B. COLTIN, M. M. VELOSO - *Ridesharing with Passenger Transfers.* 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014, pages 3278-3283. IEEE.

[4] L. DENG, X. MA, J. J. GU, Y. LI - *Multi-Robot Dynamic Formation Path Planning with Improved Polyclonal Artificial Immune Algorithm*. Control and Intelligent Systems, 42, 2014.

[5] D. FERGUSON, M. LIKHACHEV, A. T. STENTZ - *A Guide to Heuristic-Based Path Planning.* Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS), Pittsburgh, PA, 2005.

[6] M. GOLDENBERG, A. FELNER, R. STERN, J. SCHAEFFER - *A* Variants for Optimal Multi-Agent Pathfinding.* D. Borrajo, A. Felner, R. E. Korf, M. Likhachev, C. L. López, W. Ruml, N. R. Sturtevant, editors, Proceedings of the Fifth Annual Symposium on Combinatorial Search, SOCS 2012, Niagara Falls, Ontario, Canada, July 19-21, 2012. AAAI Press.

[7] W. GUANGHUA, L. DEYI, G.WENYAN, J. PENG - *Study on Formation Control of Multi-Robot Systems.* 2013 Third International Conference on Intelligent System Design and Engineering Applications, pages 1335-1339, 2013.

[8] Y. HAO, S. K. AGRAWAL - *Planning and Control of UGV Formations in a Dynamic Environment: A Practical Framework with Experiments.* Robotics Auton. Syst., 51(2-3):101-110, 2005.

[9] P. E. HART, N. J. NILSSON, B. RAPHAEL - *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* IEEE Trans. Systems Science and Cybernetics, 4(2):100-107, 1968.

[10] P. E. HART, N. J. NILSSON, B. RAPHAEL - *Correction to "a Formal Basis for the Heuristic Determination of Minimum Cost Paths".* SIGART Newsletter, 37:28-29, 1972.

[11] W. HÖNIG, T. K. S. KUMAR, L. COHEN, H. MA, H. XU, N. AYANIAN, S. KOENIG - *Multi-Agent Path Finding with Kinematic Constraints.* Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016., pages 477-485, 2016.

[12] J. HRNCIR, M. JAKOB - *Generalised Time-Dependent Graphs for Fully Multimodal Journey Planning.* 16th International IEEE Conference on Intelligent Transportation Systems, ITSC 2013, The Hague, The Netherlands, October 6-9, 2013, pages 2138-2145. IEEE.

[13] M. KATSEV, J. YU, S. M. LAVALLE - *Efficient Formation Path Planning on Large Graphs.*2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013, pages 3606-3611.

[14] E. KELAREVA, K. TIERNEY, P. KILBY - *CP Methods for Scheduling and Routing with Time-Dependent Task Costs*.Springer, Berlin, Heidelberg, pp. 111-127 2013.

[15] D. KIRCHLER - *Efficient Routing on Multi-Modal Transportation Networks.* (Routage efficace sur réseaux de transport multimodaux). PhD thesis, École Polytechnique, Palaiseau, France, 2013.

[16] S. M. LAVALLE, J. J. KUFFNER - *Rapidly-Exploring Random Trees: Progress and Prospects.* Algorithmic and Computational Robotics: New Directions, pp. 293-308, 2000.

[17] R. L. RAFFARD, C. J. TOMLIN, S. P. BOYD - *Distributed Optimization for Cooperative Agents: Application to Formation Flight.* 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), volume 3, pages 2453-2459 Vol.3, 2004.

[18] M. SCHREIECK, H. SAFETLI, S. A. SIDDIQUI, C. PFLÜGLER, M. WIESCHE, H. KRCMAR - *A Matching Algorithm for Dynamic Ridesharing.* Transportation Research Procedia, 19:272-285, 2016.

[19] R. SILVEIRA, E. P. E SILVA JR., L. P. NEDEL - *Managing Coherent Groups.* Comp. Anim. Virt. Worlds, 19(3-4):295-305, 2008.

[20] R. TIWARI, A. SHUKLA, R. KALA - *Intelligent Planning for Mobile Robotics: Algorithmic Approaches.* Information Science Reference, 2012.

[21] M. VAN HELLENBERG HUBAR - *Multiple-Phase Trajectory Optimization for Formation Flight in Civil Aviation*. 2016.

[22] J. XU, S. A. NING, G. BOWER, I. KROO - *Aircraft Route Optimization for Formation Flight.* Journal of Aircraft, 51, 2014.

[23] J. YU, S. M. LAVALLE - *Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics.* IEEE Transactions on Robotics, 32(5):1163-1177, 2016.

Florent Teichteil-Koenigsbuch graduated in 2002 from the French aerospace engineering school SUPAERO. He then obtained a PhD in Artificial Intelligence from the University of Toulouse in 2005. After having worked at ONERA as a research scientist in Robotics and Artificial Intelligence from 2005 to 2015, he moved to the Airbus Central Technology Office where has been working as a data scientist and research project leader. He has published several conference and journal papers on AI decision making and autonomous robotics.

Guillaume Poveda has been an AI research engineer in Airbus R&T since 2018. He graduated from ISAE-SUPAERO in 2016 with a Master's degree in engineering along with a Master's degree in operational research. Since then, Guillaume has been working on optimization topics for aerospace. When he's not in front of his computer, Guillaume is on his bike seat or somewhere in the Pyrénées.

C. Pralet, S. Roussel
(ONERA)

J. Jaubert
(CNES)

J. Queyrel
(ONERA)

E-mail: cedric.pralet@onera.fr

DOI: 10.12762/2020.AL15-05

Planning for Space Telescopes: Survey, Case Studies, and Lessons Learned

In this article, we present planning and scheduling techniques that we developed for optimizing the operations of space telescopes. The latter are satellites whose mission is to observe celestial objects such as planets, exoplanets, stars, or galaxies. After a survey of some existing mission planning tools, we present three case studies that we tackled using a constraint-based optimization and operations research approach, with for each case study the lessons that we learned. Based on these, we propose future work directions related to the development of generic mission planning tools, to the management of uncertain events, and to the definition of a global mission planning concept for several telescopes.

Introduction

Space telescopes (or space observatories) are satellites whose mission is to observe celestial objects such as planets, exoplanets, stars, or galaxies. As expressed in the European Space Agency (ESA) Cosmic Vision program,[1] they are used by the scientists to answer questions concerning the formation of planets, the emergence of life, the way in which the Solar system works, and to understand the physical laws of the Universe and its origins.

To achieve this goal, space telescopes usually embed several instruments such as gamma-ray imagers, gamma-ray detectors, X-ray imagers, optical cameras, photometers, photodetectors, infrared cameras, infrared spectrometers, ultraviolet sensors, etc. All instruments are body-mounted on the satellite for acquisition quality reasons, and to observe a specific target the whole satellite must be pointed at a specific direction, with a need for a high angular precision and for a very stable pointing.

For space telescopes, Artificial Intelligence (AI) can come into play to analyze the set of data produced by the instruments. It also comes into play for the construction of the telescope activity plans. The latter include tasks such as *observation tasks* requested by the scientists, *calibration tasks* used for setting up the instruments, *maneuvers* used for pointing the telescope toward particular directions, and *communication tasks* used to receive telecommands and to send observation data to ground stations. In this context, the plans constructed must be valid according to various constraints related to physical limitations or user requirements. They must also optimize the exploitation of the telescope over a given time period, for various objective functions related to the priority of some targets, the temporal dispersion of groups of observations, or the amount of resources consumed. One reason that motivates the use of automated AI planning and scheduling is that there are usually a lot of candidate tasks (thousands or tens of thousands), and it is not straightforward even for a system expert to manually define valid and efficient activity plans.

Globally, AI planning and scheduling is used both for the *long-term planning* phase, where the activity plans are constructed over long time periods (e.g., one year), and for the *short-term planning* phase, where the plans must be refined and where last-minute observation requests may be received. The latter include Targets of Opportunity (ToOs), corresponding to high-priority targets for which events are detected by other ground or space telescopes. The short-term plans sent to the telescope can also be updated directly on-board if highly relevant events such as Gamma-Ray Bursts (GRBs) are detected by the embedded instruments. It is then useful to be able to reschedule the parts of the observation plan canceled due to the arrival of ToOs and GRBs. It can also be noted that for the short-term planning phase all activities of the telescope are taken into account, but for the long-term planning phase some of them are sometimes not explicitly modeled. This can occur for maneuvers, when their duration is not significant compared to the duration of observations, or for operations that are carried out during specific parts of the orbit where no observation is possible.

In this article, we first give an overview of some mission planning systems available in the literature to manage space telescopes (Section "A survey of some mission planning systems"). After that, we provide feedback on our experience in the field, based on three

[1] https://en.wikipedia.org/wiki/Cosmic_Vision

case studies that we tackled. On the latter point, Section "Planning for INTEGRAL" deals with long-term mission planning for the active ESA INTErnational Gamma-Ray Astrophysics Laboratory (INTEGRAL), Section "Planning for SVOM" describes mission planning for the future French-Chinese (CNES-CNSA) Space Variable Objects Monitor (SVOM), and Section "Planning for ARIEL" presents long-term mission planning for the future ESA Atmospheric Remote-Sensing Infrared Exoplanet Large-survey (ARIEL) mission. We describe a constrained-based scheduling and operations research approach used for these missions, as well as the results obtained. The descriptions provided for these three missions correspond to a global view of the full descriptions available in dedicated papers [35, 34, 38]. Last, Section "Conclusion and future work directions" provides future work directions concerning the development of generic constraint-based optimization tools, the production of robust plans that better anticipate the arrival of ToOs and GRBs, and the definition of a planning system for managing several telescopes.

A survey of some mission planning systems

Nowadays, a little over 100 space telescopes are referenced.[2] This section gives a synthetic view of the mission planning systems developed for some of them, based on the restricted list given in Table 1 composed of the Hubble Space Telescope (HST), X-ray Multi-Mirror (XMM)-Newton, INTEGRAL, Spitzer, Swift, Herschel, the James Webb Space Telescope (JWST), SVOM, the Exoplanet Characterization Observations (EChO) telescope, ARIEL, and the Advanced Telescope for High ENergy Astrophysics (ATHENA).

Telescope	Space agency	Orbit	Launch date	Termination	References
HST	NASA	Low-Earth Orbit (590km)	1990	-	21, 20, 22, 26, 13, 14
XMM-Newton	ESA	highly elliptical orbit around the Earth	1999	-	5
INTEGRAL	ESA	highly elliptical orbit around the Earth	2002	-	35, 24
Spitzer	NASA	heliocentric orbit (372-day period)	2003	-	29
Swift	NASA	Low-Earth Orbit (600km)	2004	-	31
Herschel	ESA	around the L2 Lagrange point	2009	2013	16, 4, 7
JWST	NASA	around the L2 Lagrange point	2021 (planned)	-	37, 15
SVOM	CNES-CNSA	Low-Earth Orbit (625km)	2021 (planned)	-	34, 19
EChO	ESA	around the L2 Lagrange point	canceled	canceled	32, 30, 12
ARIEL	ESA	around the L2 Lagrange point	2028 (planned)	-	38
ATHENA	ESA-JAXA	around the L2 Lagrange point	2031 (planned)	-	18

Table 1 - Analyzed space telescopes and references to their mission planning systems

A previous generic analysis [8] is based on a list of telescopes different from the one considered here. Globally, similar conclusions are derived as regards the mission needs, but we try to give a few more details on the problem components with regard to standard constraint-based optimization concepts. Note also that, for some telescopes such as Planck [11], there is no need for AI planning and scheduling since the successive pointings used by the satellite are given by a predefined pointing law.

Problem features

High-level observation requests

Many telescope missions involve high-level observation requests that cover multiple elementary observation tasks. These tasks can be linked by various kinds of constraints. For instance, some elementary tasks might have to be performed sufficiently close to each other or, more generally, the elementary tasks can be linked by minimum and maximum distance constraints. The latter are temporal constraints of the form $y - x \in [a,b]$, where x and y are two variables representing the start or end times of two tasks, and where a and b are two constants. Moreover, it might be forbidden to interleave the elementary observations of a given request with other elementary observations.

The elementary observations can also be linked by various kinds of preference functions. For instance, a preference for grouping the elementary tasks as much as possible, or a preference for performing them as periodically as possible, might exist.

Task selection

Almost all telescope mission planning problems are over-constrained, meaning that it is not possible to perform all of the candidate tasks, one exception being the long term mission planning for EChO, where the goal is to perform all tasks within a minimum amount of time [32, 30, 12]. Several constraints can be imposed on the selection of tasks, including (1) constraints forcing some mandatory tasks to be performed, (2) constraints linking the performance of several tasks, such as constraints imposing that a specific calibration task should be selected only if one of the observation task requiring this calibration is selected, or (3) constraints specifying that a high-level observation request is selected if and only if a sufficient number of its elementary observation tasks is scheduled.

In many missions, the observation tasks to be performed are fully specified by the users, but in some cases there is a freedom on some parameters. This occurs for HST and JWST, where an observation orientation must be chosen within an orientation range specified by the users, and where relative orientation constraints can impose that the angular distance between the orientations chosen for two observation o_1, o_2 must be within a given range.

Choice of realization windows

Usually, the target associated with an observation task is visible only during certain *visibility windows*, computed by taking into account elements like the positions of the Earth, the Sun, the Moon, and other planets, or the user requirements concerning the earliest and latest times at which observation data must be collected. One issue is then to choose, for each selected observation task, the visibility window(s) within which it is carried out, as in works on scheduling with multiple time windows. The time granularity used for representing the visibility

[2] https://en.wikipedia.org/wiki/List_of_space_telescopes

windows can differ from one mission to another: windows expressed in days, revolutions, hours, etc.

Disjunctive or cumulative telescope resource

For the short-term planning phase, the telescope is viewed as a *disjunctive resource*, that is, as a resource that cannot carry out two observation tasks in parallel. Indeed, given that all instruments are body-mounted on the satellite, two observations associated with distinct pointing directions cannot be performed simultaneously.

For the long-term planning phase, the telescope is also modeled as a disjunctive resource most of the time. In some missions, however, it is viewed as a *cumulative resource* that can carry out activities in parallel up to a certain limit. This is the case for HST and JWST, where the long-term planner computes, for each selected observation task, a *soft realization window* that is larger than the actual window required to perform the observation. Then, soft realization windows can overlap up to a certain limit, which leads to a model based on a cumulative resource instead of a disjunctive one. The rationale is that instead of choosing fixed windows in long-term plans, some level of temporal flexibility is kept in prevision of last-minute urgent observation requests.

Knapsack and cardinality constraints

For most of the missions a long-term plan is built, but it is necessary to take into account the fact that the occurrence of events such as GRBs or ToOs leads to new mandatory observation tasks, which induce a temporary interruption of the nominal mission plan. To build long-term plans that are robust to such random events, the usage rate of some time periods is sometimes limited. For instance, for INTEGRAL, a maximum usage rate is imposed over every revolution [35], and for XMM-Newton a maximum usage rate is imposed over groups of 4-5 successive revolutions [5]. In terms of optimization problems, the revolutions or groups of revolutions are kinds of knapsacks whose capacity corresponds to the maximum usage duration, and the observations carried out are items whose size corresponds to their duration. In some cases, cardinality constraints must be satisfied, such as when there is a limit on the number of observations per revolution, to indirectly minimize maneuvers.

State constraints

As mentioned previously, all telescopes considered embed several instruments. The latter can sometimes operate in different states, and setup operations are needed to reach a given state. For instance, in the case of Herschel [4], cool down operations are required to set up the instrument(s) before an observation, and these operations are quite long (several hours) and consume liquid Helium. In this case, to avoid spending too much time and energy for state changes, a (unique) state of the instrument(s) is chosen for each day of operation. Moreover, there exists a minimum number of successive days during which a given state must be maintained. From a scheduling point of view, if setup operations are explicitly modeled, the problem reached shares similarities with Job Shop Scheduling Problems with Sequence-Dependent Setup Times [1]. Another example where cool down operations are explicitly considered is the ATHENA mission [18], where one issue is to plan cryocooler regeneration cycles that allow one of the instruments to be cooled down and observations to be performed for a given duration. One issue is then to define cooling strategies that allow the response time to ToOs to be optimized, in addition to being efficient for carrying out the long-term program.

Resources with consumption and production

In terms of mission constraints, the more complex specifications are probably those of the JWST. Basically, due to its large surface, the JWST is subjected to a significant solar radiation pressure. The perturbation induced on the kinetic momentum is countered using reaction wheels available on board. However, the accumulation of the perturbations can lead to a saturation of the reaction wheels, when their maximum speed is reached. It is then necessary to desaturate these wheels by using some of the fuel available (so-called *momentum dumping* operations). It is also possible to "produce" momentum by performing observations in a direction that allows the wheels to be slowed down. As a result, the mission planner for the JWST must handle a *momentum resource* [37] that is subject to both momentum consumptions and momentum productions.

Optimization criteria

One common point between the mission planning problems associated with the telescopes considered is that there is never a single optimization criterion. Examples of objective functions encountered are:

- maximization of the number of observation requests that are carried out, while potentially taking into account the priority associated with each request;
- maximization of the total usage duration of the telescope for scientific purposes (useful mission time);
- minimization of the cumulated slew required to carry out maneuvers between the successive tasks of the plan;
- minimization of the use of consumable resources, like the fuel, to increase the long-term lifetime of the mission;
- maximization of preference degrees over the realization times of some observations: preference for earliest performance times, preference for grouping as much as possible the elementary observations associated with a single request, preference over the regularity of the performance of periodic observations, user-preference for performing observations during specific parts of the orbits, etc.;
- fair sharing of the satellite among the mission contributors or among the pointing directions;
- minimization of the degree of violation of some *soft* constraints, that should ideally be satisfied but for which partial satisfaction is allowed if needed;
- maximization of the robustness of the plans produced, with regard to the arrival of GRBs and ToOs; one goal here is to provide the users with a good estimation of the time at which the data they requested will be available.

Planning techniques

The mission planning systems developed for the space telescopes analyzed all use incomplete search algorithms, which are able to quickly find good quality solutions but that offer no guarantee with regard to the optimality of the solution produced. The reasons for this are twofold. First, the size of the instances that must be solved precludes the use of systematic techniques guaranteeing that the whole search space is explored. Second, the notion of optimal solution is often hard to define due to the presence of multiple objective functions. As detailed below, two kinds of mission planning algorithms are used in practice, namely (1) greedy search, and (2) local search coupled with metaheuristics (simulated annealing, tabu search, genetic algorithms, etc.).

Greedy search

For space telescope planning, a greedy search consists in (1) choosing at each step an elementary observation task or an observation request, (2) inserting exposure time (*i.e.,* observation time) within the current plan so as to fulfill the observation task or the request selected, (3) iterating this process so as to obtain a plan that is filled as much as possible, given that an activity inserted into the plan at some step can never be removed.

The main parameters of such a greedy search scheme are the *selection heuristic*, used for selecting an observation at each step, and the *insertion heuristic*, used for choosing an insertion position in the current plan. As an illustration, for the Swift telescope, the greedy search scheme implemented in the TAKO planning tool [31] is based on a *static* selection heuristic that successively selects tasks based on fixed priorities, and on an insertion heuristic that inserts each task at the earliest position in which there is a sufficient idle period. For the SVOM mission [34] detailed later in this article, the selection heuristic is dynamic (depending on the content of the current plan), and the insertion heuristic also inserts tasks at the earliest position in which there is a sufficiently idle period, but in this case with the possibility of moving back the tasks of the current plan (manipulation of temporally flexible solutions). In the HST [22] or XMM-Newton [5], the selection strategy considers the observations that are more constrained first, since these observations become increasingly difficult to insert as the search progresses, especially the observations composed of several elementary observations that must be carried out following a specific pattern. It is also possible to consider first the observations that are the least constrained, the intuition being that these observations will be easier to rearrange in case GRBs or ToOs occur.

Greedy decision rules are also used for short-term replanning following the arrival of high-priority requests. In this case, the conflicts between the new mandatory observations and the observations of the current plan are analyzed and resolved based on fast priority-based decision rules.

Local search and metaheuristics

In another direction, several mission planners developed for space telescopes use local search and global optimization methods (or metaheuristics) that allow the exploration of the search space to be diversified and local optima to be escaped from.

On the local search side, the mission planners for the HST and JWST use the *minconflicts* algorithm [28], which starts from an inconsistent plan and reduces step-by-step the number of conflicts between the observations of the plan (iterative conflict resolution approach [22]). For other telescopes like INTEGRAL or XMM-Newton, the local search techniques proposed handle only consistent plans at each step of the search, where tasks can be successively added or removed.

Concerning metaheuristics, various global optimization methods were tested for space telescopes: multi-objective evolutionary algorithms for the JWST [15], stochastic hill-climbing or tabu search for INTEGRAL [35, 24], iterated local search for SVOM [34], and simulated annealing, or restart techniques that allow to diversify the exploration of the search space thanks to the stochastic nature of some decision rules. Some works also use optimization strategies that first build a plan based on a coarse-grain model and then refine this plan based on a detailed model [24].

Generic techniques

For several missions, the low-level constraints of the problem are handled by core planning and scheduling frameworks. One example is the APSI framework [6] initially developed by ISTC-CNR (Rome, Italy) for ESA, and which was used for INTEGRAL [35] and XMM-Newton [5] to determine whether scheduling a small set of observation tasks within a given revolution is feasible. It was also tested for planning the mission of Herschel [7].

A second example is the constraint-based Spike tool [22, 26], developed by the US *Space Telescope Science Institute* (STScI). This tool is used for planning the activities of many ground and space telescopes, including the HST and JWST. Basically, Spike helps to prune the task insertion positions that would lead to dead-ends given the set of temporal constraints of the problem.

A last example is the InCELL library [36, 33], which was used for two of the three telescope mission planning problems detailed in the next sections. Basically, the scheduling layer of InCELL handles an ordering over the tasks of the plan and maintains the earliest and latest start times of these tasks as a consequence of the current ordering. InCELL also allows various optimization criteria to be modeled, whereas for instance on the HST specific work was necessary to formalize the objectives truly optimized by the Spike-based planner after several years of operations [14]. As for common points, both Spike and InCELL are constraint-based, and both can manage at some step temporally inconsistent plans. Compared to APSI, InCELL is dedicated to the implementation of local searches and metaheuristics.

Interactive scheduling

A last point that is common to many space telescope planning systems is the availability of manual and interactive scheduling modes in addition to fully automated search. This aspect is highly relevant for missions where it is hard to aggregate several optimization criteria into a single one, and where it is useful to propose several mission plans to the end-users, with some statistics that help in evaluating the quality of each of the plans proposed.

Uncertainty management

Almost all space telescopes must manage *uncertain events* such as ToOs or GRBs. As seen before, the precise management of uncertainty varies depending on the mission. For example, in INTEGRAL [35], a maximum filling percentage is specified for revolutions located in periods of the year where GRBs are more likely to be observed. In XMM-Newton [5], a maximum filling percentage is imposed over groups of successive revolutions. In Spitzer [29], uncertainty about data volume is handled on-board to avoid the over-filling of the spacecraft mass memory. In the long-term planning phase of the HST, soft realization windows are chosen for observations instead of fixed non-flexible windows [26]. For SVOM, a task reschedulability measure is optimized [34]. On the execution side, in SVOM, GRB events directly erase parts of observation plans, whereas for telescopes such as the HST and JWST, the plans uploaded to the satellites are ordered lists of observations that can be postponed at execution time instead of being erased.

Planning for INTEGRAL

In the following, we describe three case studies that we tackled in the past and show how the combination of incomplete search techniques and constrained-based approaches allowed us to produce good-quality plans. The first realization presented was developed within the context of a study initiated in 2007 by ESA, involving ISTC-CNR (Rome, Italy), VEGA (Darmstadt, Germany), the *Politecnico di Milano* (Milan, Italy), and ONERA. The objective of this study was to explore the use of AI Planning and Scheduling techniques for *ESA* missions [41]. ONERA was in charge of one of the test cases, namely the long-term planning (over one year) of the observation activities of the INTEGRAL space telescope. In the following, we give an overview of the specifications of the planning problem, the constraint-based model defined, the planning algorithm developed, the results obtained, and the lessons drawn from this work.

Mission description

INTEGRAL is an *ESA* mission, managed in cooperation with Russia and the USA, whose goal is to observe gamma-ray emissions from the universe. Starting in 2002 for at least two years, it has now been extended until 2020. As shown in Figure 1, the INTEGRAL satellite is moving on a highly elliptical orbit around the Earth. One orbit corresponds to 72 hours and only about 58 hours of these, out of the Earth radiation belts, are available for observation. Due to the presence of the Sun, the Earth, the Moon, and other planets, a given target is not permanently observable during these 58 hours. The satellite embeds four instruments: a gamma-ray spectrometer named *SPI*, a gamma-ray imager named *IBIS*, an X-ray monitor named *Jem-X*, and an optical monitor camera named *OMC*. These four instruments are fixed on the platform and point in the same direction. The Attitude and Orbit Control System (AOCS) allows the platform (and thus the instruments) to remain pointed in a given direction during an observation, and to move from one direction to another between two successive observations. For the long-term planning phase, the slewing time between two successive observations is indirectly considered by limiting the number of different observations within each orbit. Moreover, in order to keep some time available for opportunistic observations of unexpected events, such as the appearance of new X-ray/gamma-ray sources, only a given percentage of the observation time within each orbit is considered to be available. Constraints related to energy, data recording and downloading are not taken into account at this step.

From the long-term planning point of view, an Announcement of Opportunity (*AO*) is emitted each year, to which scientists answer by posting observation requests over targets of interest. Then, a target allocation committee selects observation requests and assigns to each of them a priority and a realization percentage above which the request is considered to be achieved. In general, a percentage of 100% is not mandatory. The AO used here covers a period from August 2007 to August 2008 and involves 123 orbits and 35 observation requests.

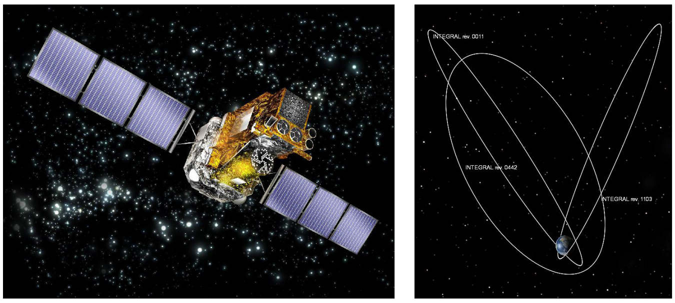


Figure 1 - The INTEGRAL satellite and its highly elliptical orbit (image credit: ESA)

Each high-level observation request r is decomposed into $NEO(r)$ elementary observations corresponding to precise pointings defined by an observation mask. Observations of a given request all have the same duration, and $NEO(r)$ is within the interval [1,1023] in practice, most of the observation requests requiring several hundreds of elementary observations. It is not mandatory and it is often impossible to perform all of these $NEO(r)$ elementary observations within a single orbit of the satellite.

Figure 2 shows an example of a solution plan for an instance involving 5 orbits and 6 observation requests. For request r_1, we have 6 observation windows and 4 observation activities: the first one involving 3 elementary observations in the first window and the other three involving each 2 elementary observations in the last three windows. For request r_2, we have 4 observation windows and 4 observation activities, each involving only one elementary observation in each window. We can observe 2 observation activities in the same window for request r_3 in the last window. We can also observe that no observation activity is associated with requests r_5 and r_6. Finally, we can observe that the duration of an elementary observation depends on the request considered: for example, greater for r_2 than for r_1.

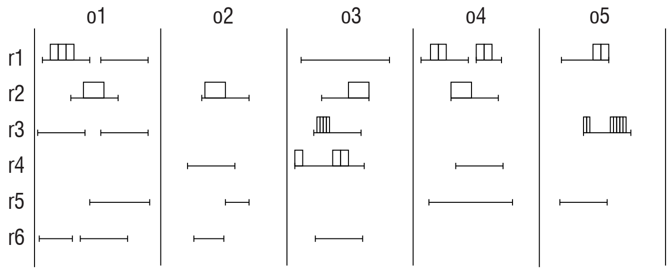


Figure 2 - Example of a solution plan for the long-term planning phase of INTEGRAL

In INTEGRAL, a type is associated with each observation request of a given target. This type specifies the way in which the observations must be performed, and there are four types of requests:

- normal observation requests (NO) which must be split as little as possible and ended as early as possible after they have started;
- no-splitting observation requests (NS), which must not be interleaved with other observation requests;
- periodic observation requests $PE(p,t)$, which must be decomposed into elementary observations performed every p orbits with a tolerance t on the deviation from the period;
- spread observation requests ($SP(n)$), which must be decomposed into n sub-observations to be spread as much as possible over the year.

The long-term planning problem consists in selecting and scheduling over the next observation period, generally of one year duration, the observations associated with the current *AO*, plus the observations selected at the previous *AO* but which were only partially carried out.

The resulting problem is a kind of over-constrained scheduling problem [40, 25] where the objective is to fulfill requests as much and as well as possible, taking request priorities into account and knowing that observations cannot overlap. The long-term plan also serves as an input for regular short-term planning which decides on the detailed activities to be performed by the satellite, taking into account the arrival of new urgent observation requests.

Constraint-based modeling

In this study, following a constraint-based optimization approach, the first step was to formalize the problem by providing a clear definition of the input data, the decision variables, the constraints, and the objective function. The full model is available in [35], and we provide thereafter only its main features.

Input data

We consider a set O of orbits over the planning horizon, with a maximum observation duration allowed for each orbit. We also consider a set R of observation requests, with for each request $r \in R$:

- a type $TY(r)$ among the four possible ones (NO, NS, $PE(p,t)$, $SP(n)$);
- a set $W(r)$ of windows available for carrying out r over the year; each window is included in a single orbit and corresponds to a period where the telescope is outside the Earth radiation belts;
- a weight $WE(r)$ reflecting the priority of the request;
- a number $NEO(r)$ of elementary observations that all have the same duration;
- a percentage above which r is considered to be achieved.

To model the problem, the main difficulty is that the number of *observation activities* used for one request r is not known initially, where an observation activity corresponds to a set of contiguous elementary observations performed for r within a given visibility window $w \in W(r)$. For instance, in Figure 2, there is one observation activity composed of 3 elementary observations in the first visibility window of r_1. In practice, it is not feasible to introduce as many observation activities as the number of elementary observations, which is why the maximum number of observation activities per request is restricted. With each normal observation request $r(TY(r)=NO)$, we systematically associate two possible observation activities per window $w \in W(r)$, and with each special observation request $r(TY(r) \neq NO)$, we associate only one observation activity per window $w \in W(r)$. For the instance we worked on, we ended up with 2731 candidate observation activities. In the sequel, OA denotes the set of candidate observation activities and $OA(r)$ (respectively, $OA(o)$) denotes the set of candidate observation activities associated with request r (respectively, orbit o). We also consider a maximum number of non-empty observation activities per orbit, to limit the overall slewing time.

Variables

The problem is then to choose for each candidate observation activity $oa \in OA$, its starting time $s(oa)$ and the number $neo(oa)$ of elementary observations that it involves.

From this, it is possible to compute the values of other variables that are functionally dependent on the $s(oa)$ and $neo(oa)$ variables. First, the ending time $e(oa)$ of oa can be directly deduced from the fixed duration of each elementary observation. Then, for each request $r \in R$, the total number $neo(r)$ of elementary observations performed for r is defined as $neo(r) = \sum_{oa \in OA(r)} neo(oa)$. This number must not exceed $NEO(r)$, the total number of elementary observations required for r. Last, for each request r, it is possible to compute the sequence of non-empty observation activities associated with r, ordered according to their starting times, and to get a start time $s(r)$ (start time of the first observation activity carried out for r) and an end time $e(r)$ (end time of the last observation activity carried out for r).

Constraints

Several constraints must be satisfied by the variables of the model. Some of them are quite standard in terms of optimization:

- *time windows*: each observation activity must be included in the visibility window w with which it is associated;
- *disjunctive resource constraints*: given that there is a unique telescope resource over which the instruments are body-mounted, observation activities cannot overlap; non overlapping constraints can be expressed separately for each orbit, because windows associated with two distinct orbits are disjoint;
- *knapsack constraints*: for each orbit o, the maximum observation duration within o must not be exceeded;
- *cardinality constraint*: for each orbit o, the maximum number of non-empty observation activities must not be exceeded.

Beside these standard constraints, each specific request type induces side constraints that are less standard for generic optimization methods:

- for each *no-splitting observation request* r, there must be no interleaving between the observation activities associated with r and the observation activities associated with other requests;
- for each *periodic observation request* r, its period must be respected (up to the tolerance allowed) and only elementary observation activities must be used ($neo(oa)=1$);
- for each *spread observation request* r, there is a maximum number of elementary observations for each observation activity associated with r.

Objective functions

In INTEGRAL, the definition of the optimization criteria is not as easy as the definition of the constraints is. After discussion with the end-users, we adopted the following approach. With each request r are associated:

- a *quality of completion* $qc(r) \in [0,1]$, which measures the percentage of completion of r;
- a *quality of realization* $qr(r) \in [0,1]$, which measures to what extent the set of observation activities used for r are consistent with the type of r; the definition of this quality of realization depends on the request type; for normal and no-splitting requests, the quality is higher when r finishes as early as possible after it started (maximum grouping objective); for periodic observation requests, the realization quality is higher when the deviation from the ideal period is lower; for spread observation requests, the realization quality is the mean realization quality of its observation activities;

- an *overall quality* $q(r)$ obtained as a linear combination of the two previous qualities: $q(r)=\alpha\cdot qc(r)+(1-\alpha)\cdot qr(r)$ with $\alpha\in[0,1]$ a parameter that can be adapted by the end-users.

The global criterion q to be maximized is defined as the normalized weighted sum of request qualities:

$$q=\frac{\sum_{r\in R}WE(r)\cdot q(r)}{\sum_{r\in R}WE(r)} \qquad (1)$$

Stochastic hill-climbing with restarts

Considering only variables $neo(oa)$ for the real instance to be solved leads to 2731 variables whose domain size is between 2 and 1024. First experiments were made with generic constraint-based optimization tools, which have, in theory, the capacity to find the optimal solution, but the results were not satisfying essentially due to the large size of the search space and to the complexity of the non-standard constraints and criteria. Instead, local search algorithms and meta-heuristics were used in order to produce good quality solutions within limited computation times. The main features of the algorithm developed are the following:

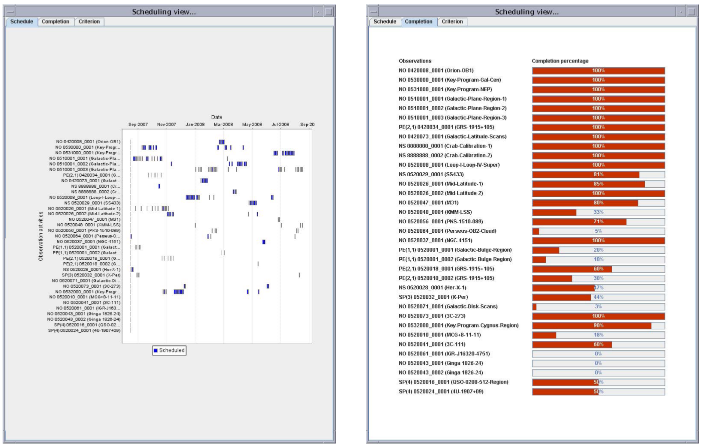
- *local search moves:* The algorithm starts from an empty plan; it maintains a current plan and it modifies it iteratively using two kinds of local moves: either the enlargement of an observation activity oa (by adding to oa as many elementary observations as possible), or the enlargement of an observation activity oa after the removal of another observation activity oa' located in the same orbit; after each step of the algorithm, the consistency of the current plan is maintained, meaning that all of the model constraints are satisfied;
- *restricted neighborhood:* at each step of the algorithm, a small subset of the set of possible local moves is pre-selected, by taking into account the weights associated with the requests; pre-selection is necessary because of the huge number of observation activities and thus of possible local moves; to avoid cycles around local optima, a tabu list of the T previous local moves is maintained, and the local moves included in the tabu list cannot be pre-selected;
- *stochastic hill-climbing*: all of the pre-selected local moves are evaluated by estimating their positive or negative impact on the optimization criterion; one candidate move is then randomly selected among the best ones; the selected local move is effectively applied if the estimated impact is strictly positive and applied with a certain probability if the estimated impact is negative or null, as in simulated annealing [23];
- *restarts*: the algorithm restarts from an empty plan each time a maximum number of local moves without improvement is reached.

Inside the algorithm, the basic scheduling constraints associated with each individual orbit are managed by the core APSI toolbox [6]. Basically, the latter models dynamic systems based on a set of *timelines*, each of which represents the evolution of a component of the system. In APSI, different types of timelines can be used, including timelines modeling state variables or resources, and different kinds of constraints over timelines can be specified, including state and temporal constraints. For INTEGRAL, the specific constraints associated with no-splitting, periodic, and spread requests, as well as the optimization of the plan quality, are managed outside of APSI.

Experimental results

The planning tool implemented can be used to visualize the evolution of the current plan during a search, the best plans found, and statistics on the quality of completion and realization of each observation request. Figure 3 shows some visualizations available. For advanced users, it is also possible to set parameters such as the probability of acceptance of a local move that does not increase the plan quality.

In the one-year instance described before, the algorithm takes in general only some minutes to achieve plans whose quality is close to 0.97 or 0.98, *i.e.*, very close to 1, which is an upper bound on the plan quality. This means that, in the worst case, the best quality obtained is only 2 or 3% below the optimal one.


Scheduling view...
Schedule
Completion
Criterion
Date
Sep-2007
Nov-2007
Jan-2008
Mar-2008
May-2008
Jul-2008
Sep-20
Observation activities
NO 0420008_0001 (Orion-OB1)
NO 0530000_0001 (Key-Progr...
NO 0531000_0001 (Key-Progr...
NO 0510001_0001 (Galactic-Pla...
NO 0510001_0002 (Galactic-Pla...
NO 0510001_0003 (Galactic-Pla...
PE(2,1) 0420034_0001 (G...
NO 0420073_0001 (Galact...
NS 8888888_0001 (Cr...
NS 8888888_0002 (Cr...
NO 0520008_0001 (Loop-I-Loop-...
NS 0520029_0001 (SS433)
NO 0520026_0001 (Mid-Latitude-1)
NO 0520026_0002 (Mid-Latitude-2)
NO 0520047_0001 (M31)
NO 0520048_0001 (XMM-LSS)
NO 0520056_0001 (PKS-1510-089)
NO 0520064_0001 (Perseus-O...
NO 0520037_0001 (NGC-4151)
PE(1,1) 0520001_0001 (Galact...
PE(1,1) 0520001_0002 (Galact...
PE(2,1) 0520018_0001 (G...
PE(2,1) 0520018_0002 (G...
NS 0520028_0001 (Her-X-1)
SP(3) 0520032_0001 (X-Per)
NO 0520071_0001 (Galactic-Di...
NO 0520073_0001 (3C-273)
NO 0532000_0001 (Key-Progr...
NO 0520010_0001 (MCG+8-11-11)
NO 0520041_0001 (3C-111)
NO 0520061_0001 (IGR-J163...
NO 0520043_0001 (Ginga 1826-24)
NO 0520043_0002 (Ginga 1826-24)
SP(4) 0520016_0001 (QSO-02...
SP(4) 0520024_0001 (4U-1907+09)
Scheduled
Scheduling view...
Schedule
Completion
Criterion

Observations	Completion percentage
NO 0420008_0001 (Orion-OB1)	100%
NO 0530000_0001 (Key-Program-Gal-Cen)	100%
NO 0531000_0001 (Key-Program-NEP)	100%
NO 0510001_0001 (Galactic-Plane-Region-1)	100%
NO 0510001_0002 (Galactic-Plane-Region-2)	100%
NO 0510001_0003 (Galactic-Plane-Region-3)	100%
PE(2,1) 0420034_0001 (GRS-1915+105)	100%
NO 0420073_0001 (Galactic-Latitude-Scans)	100%
NS 8888888_0001 (Crab-Calibration-1)	100%
NS 8888888_0002 (Crab-Calibration-2)	100%
NO 0520008_0001 (Loop-I-Loop-IV-Super)	100%
NS 0520029_0001 (SS433)	81%
NO 0520026_0001 (Mid-Latitude-1)	85%
NO 0520026_0002 (Mid-Latitude-2)	100%
NO 0520047_0001 (M31)	80%
NO 0520048_0001 (XMM-LSS)	33%
NO 0520056_0001 (PKS-1510-089)	71%
NO 0520064_0001 (Perseus-OB2-Cloud)	5%
NO 0520037_0001 (NGC-4151)	100%
PE(1,1) 0520001_0001 (Galactic-Bulge-Region)	20%
PE(1,1) 0520001_0002 (Galactic-Bulge-Region)	10%
PE(2,1) 0520018_0001 (GRS-1915+105)	60%
PE(2,1) 0520018_0002 (GRS-1915+105)	30%
NS 0520028_0001 (Her-X-1)	47%
SP(3) 0520032_0001 (X-Per)	44%
NO 0520071_0001 (Galactic-Disk-Scans)	3%
NO 0520073_0001 (3C-273)	100%
NO 0532000_0001 (Key-Program-Cygnus-Region)	90%
NO 0520010_0001 (MCG+8-11-11)	18%
NO 0520041_0001 (3C-111)	60%
NO 0520061_0001 (IGR-J16320-4751)	0%
NO 0520043_0001 (Ginga 1826-24)	0%
NO 0520043_0002 (Ginga 1826-24)	0%
SP(4) 0520016_0001 (QSO-0208-512-Region)	50%
SP(4) 0520024_0001 (4U-1907+09)	50%

Figure 3 - Left: current plan over the year (one line per request, one blue rectangle per observation activity). Right: current completion percentage of each request

Lessons learned

In this study, it was possible to build an unambiguous constraint-based model for the long-term planning phase of INTEGRAL. The main effort was made on the formal aspects related to the specific request types, on the partition between features modeled as constraints and features modeled as optimization criteria, and on the decomposition of observation requests into observation activities potentially spread over several visibility windows. On the solving side, the core APSI framework was able to manage the basic scheduling constraints, such as the constraints of no overlapping between observation activities within each orbit. Several aspects related to specific request types were, however, not manageable by APSI in its 2009 version, and specific developments were required for managing these specifications outside APSI. Also, a stochastic hill-climbing algorithm with restarts was able to produce very good results in terms of plan quality and computing time, and in the end what previously required some days of manual work now requires only a few minutes of computing. Last, APSI was used as a non-incremental planning and scheduling engine in the sense that the scheduling problem associated with one orbit o is solved from scratch whenever one observation activity is added to o. The number of local moves carried out per second might have been higher by using an incremental solving strategy or a core planner tuned for local search.

Planning for SVOM

We now consider a second mission and show the results obtained by following a similar methodology: definition of a formal constrained-based model, definition of local search and metaheuristics, and development of visualization tools. One difference between SVOM and INTEGRAL is that between the two missions, we developed a core constraint-based optimization library dedicated to local search [36, 33].

Mission description

SVOM is a future Chinese-French space mission, which should be launched in 2021. It is dedicated to the study of the transient universe, which includes the observation of GRBs. The main partners involved are the Chinese Academy of Science, the Chinese and French space agencies (CNSA, CNES), the Shanghai Engineering Center for Microsatellite (SECM), and several Chinese and French science labs. The SVOM mission has a nominal duration of 3 years and an extension phase of 2 years.

As shown in Figure 4, the SVOM telescope carries four instruments: a coded-mask gamma ray imager named ECLAIRs, a gamma-ray spectrometer named GRM, a Micro-channel X-ray Telescope named MXT, and a Visible-band Telescope named VT. It will operate around the Earth at an altitude of 650km and follow a default attitude law called "B1 pointing law" which is roughly anti-solar. It provides an effective observation during the night hemisphere to enhance the follow-up possibilities and also avoids the Galactic plane and some bright X-sources to foster GRBs detection [9].

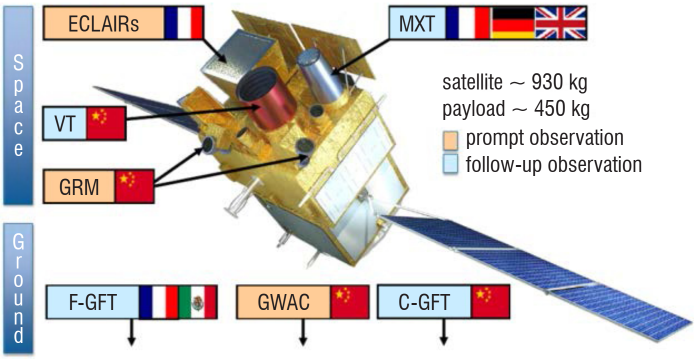


Figure 4 - Instruments of the SVOM space telescope (http://www.svom.fr/en/)

The SVOM scientific program is divided into three categories:

- the *Core Program* (CP), dedicated to GRBs; the latter are detected and managed autonomously on board; with an expected GRB rate of around 60-70 per year; GRBs are unpredictable by nature, and when a GRB is detected on board the current observation plan is interrupted and the satellite remains pointed towards the GRB source for 14 orbits ($\simeq$ 1 day);
- the *Target of Opportunity* (ToO) program, which allows new and urgent observations to be triggered from the ground; ToO observations must be performed by the satellite within 48h for a standard ToO and within 12h for an exceptional ToO (e.g., galactic supernova or gravitational wave alert); the typical observation duration is between 1 orbit and 14 orbits;
- the *General Program* (GP), consisting in observations requested by the scientific community and which are pre-planned one year in advance; the GP targets are classified into three categories: *A-targets* (high priority), *B-targets* (low priority), and *Fill-In Targets* used to provide a default target to the satellite if needed; in the following, we do not consider the fill-in-targets; the duration of a GP request is from 1 orbit up to 5 days; the observation time is allocated to scientific users' requests with a ratio of 60% for the Chinese users' group and 40% for the French users' group; also, the GP allows pointing at sources close (within 10°) to the default B1 attitude pointing law.

Figure 5 gives an idea of the distribution of the mission time among these three programs, both for the nominal mission and for the extended mission. One challenge is then to maximize the GP completion at the end of the year, especially for the A-targets, despite the occurrence of GRBs and ToOs. This challenge must be tackled given that in SVOM the mission planning process works as follows:

- a GP pool of proposals is established once a year after a Call for Obs and a selection process, and a first schedule is computed over the one year span;
- every week the pool of proposals is updated to take into account the GP observations canceled or partially carried out the week before due to the occurrence of GRBs and ToOs; the schedule is fully recomputed up to the end of the year to plan again these missing parts if possible;
- if an observation is not finished by the end of the year, the missing part might be selected to be included in the set of candidate requests for the next year; for each GP observation, it is assumed that at least 95% of the requested exposure time must be fulfilled to obtain usable data.

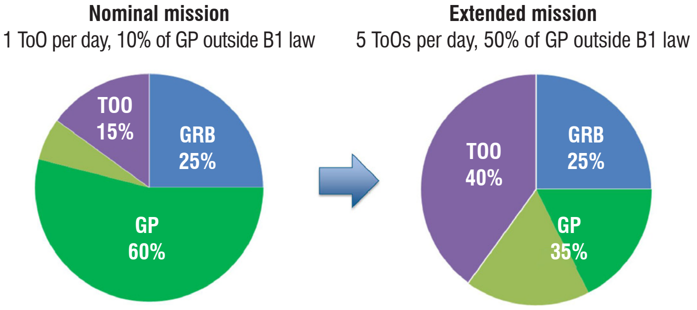


Figure 5 - Distribution of the useful mission time over one year; GP observations are split between those performed outside the B1 pointing law (light green) and those performed around this default law (dark green)

Constraint-based modeling

As for INTEGRAL, we give an overview of the constraint-based model developed for SVOM. See [34] for a full description.

Input data

We consider a set R of observation requests that are candidates for being scheduled over the remaining part of the year, and a set of coarse-grain directions containing B1 (around the default B1 pointing law), HL (High Latitude), and MWC / MWEP / MWWP (Milky Way Center / East Part / West Part). Each request $r \in R$ is defined by several elements:

- a request category among ToO (Target of Opportunity), CAL (Calibration), and GP (General Program), with a priority level

(A or B) and the user requiring the observation ("CH" for Chinese, "FR" for French) for GP requests;

- a direction category in $\{B1, HL, MWC, MWEP, MWWP\}$ and the precise coordinates of the target associated with the request;
- a list $W(r)$ of time windows during which the target associated with r is visible;
- the observation duration associated with r, split between the remaining observation duration for r and the observation duration already elapsed (not null for observations truncated due to GRBs and ToOs).

As an input, we also have, for each user $u \in \{CH, FR\}$ (respectively, for each direction $d \in \{B1, HL, MWC, MWEP, MWWP\}$), the desired satellite usage ratio for user u (respectively, for direction d). Considering usage ratios for directions is useful to make sure that the telescope does not point outside the B1 pointing law too much, which favors the observation of GRBs.

Decision variables

Following the specifications of the mission, we impose that observation requests must be planned in a single block. This means that the observations are non-preemptable at planning time, even if GRBs and ToOs might cause interruptions at execution time. An observation plan is then defined as a sequence $seq = [r_1, \ldots, r_k]$ of successive observation requests planned for the satellite, with for each request $r_i \in seq$ a time window $win(r_i) \in W(r_i)$ chosen for carrying out r_i.

Constraints

Two basic scheduling constraints must be satisfied:

- *time windows*: each observation for a request r must be carried out within the window $win(r)$ chosen for r;
- *no overlap*: the observations successively carried out must not overlap, since all instruments are body-mounted on the telescope.

Such constraints lead to a standard scheduling problem with a unique machine (the telescope) and time windows.

Objective functions

From a scheduling perspective, the main challenge in SVOM is actually to be able to manage multiple objectives for the construction of good quality plans. The model considers six sets of objective functions listed below:

- $nObs(p)$, which measures the number of GP observations carried out for priority p; this number must be maximized, with a strict preference for priority A;
- $duPrio(p)$, which measures the total observation duration for priority p; this duration must be maximized, with a strict preference for priority A;
- $duUser(u)$, which measures the cumulated observation duration for user u; this duration must respect the desired usage ratio as much as possible;
- $duDir(d)$, which measures the cumulated observation duration for direction d; this duration must respect the desired usage ratio as much as possible;
- $slew$, which measures the cumulated slew induced by the chosen observation plan, given the coordinates of the successive targets to which the telescope must be pointed; this cumulated slew must be minimized;
- $reschedulability(r)$, which measures the *reschedulability* of each request r; this metric corresponds to the amount of time available for rescheduling r (or parts of r) in case of interruptions due to the occurrence of GRBs and ToOs; for a request r in the observation sequence, this reschedulability index is maximum when the realization of r starts at the beginning of the earliest realization window of r (potentially many possibilities to reschedule r in this case), and it is equal to 0 when the realization of r ends at the end of its latest realization window (no opportunity to reschedule r in this case); ideally, the mean reschedulability must be maximized and its standard deviation must be minimized, to achieve a fair distribution of reschedulability between observation requests.

As shown in the next section, some choices were made at the level of the planning algorithms to establish an order between these objective functions.

Greedy search and iterated local search

We now describe the components of the planner developed for constructing plans every week over the rest of the year.

Core constraint-based reasoning engine

As mentioned before, the generic InCELL library is used to handle the constraint-based model of the mission. Basically, InCELL implements the Constraint-Based Local Search paradigm [17]. It allows us to incrementally evaluate the impact of additions or removals of observations on all constraints and objectives of the model, and to use predefined primitives for implementing local searches and metaheuristics.

Greedy search

The SVOM planner starts from a plan that contains only the set of regular calibrations placed in fixed windows and the set of mandatory ToOs known at planning time. The algorithm then tries to insert GP-observations one by one into this plan. This search phase is greedy in the sense that once an observation is inserted into the plan, it is never removed. As discussed previously, in order to fully instantiate such a search procedure, two main parameters must be set, namely a *selection heuristic* to select a candidate observation at each step and an *insertion heuristic* to determine an insertion position into the current plan.

The selection heuristic defined for SVOM promotes: (1) the fair sharing of the telescope among the users (relatively to the ideal usage ratios defined in the input data), (2) the fair sharing of the telescope among the categories of directions (again, relatively to the ideal usage ratios defined in the input data), and (3) the realization of the highest priority requests. A fixed lexicographic ordering is used to combine these three aspects and obtain at each step a set of candidate requests. One candidate in this set is chosen based on a portfolio of possible decision rules, such as (1) the selection of one request that is the most constrained in terms of available time windows, (2) the selection of one request whose duration is minimum, (3) the random selection of one request, or (4) the selection of one request whose observation has already been started in the

Algorithm	CPU time (sec)	Priority A		Priority B		user ratios	slew (deg)
		nObs	reschedulability in days	nObs	reschedulability in days		
greedy	21.4	190/193	**115.5(70.2)**	250/408	40.0(38.3)	0.41/0.59	23483
ILS (perturb 0.1)	300	**192**/193	99.1(74.7)	**261**/408	41.2(43.0)	**0.40/0.60**	24338
slew optimization	300	**192**/193	98.3(73.1)	**261**/408	**42.9(44.8)**	**0.40/0.60**	**13219**

Table 2 - Results obtained in a one-year scenario including more than 600 observation requests (Intel i5-520 1.2GHz 4GBRAM processor, time limit of 5 minutes); for the reschedulability, we give the mean value and the standard deviation in parentheses

past, to favor the completion of observations interrupted by GRBs or ToOs. For the insertion heuristic, several strategies were tested, including (1) insertion at the earliest feasible position that exploits an idle period of the telescope; (2) insertion at a position that maximizes the mean reschedulability of GP requests of priority A; and (3) insertion at a position that minimizes the cumulated slew. For these insertion heuristics, moving back observations of the current plan is allowed, and only insertion positions leading to a feasible plan are considered.

Iterated local search

Given that the heuristics used in greedy search are imperfect, a more efficient planning strategy was developed. The latter uses the *Iterated Local Search* (ILS) metaheuristic [27]. It starts from the solution produced by the greedy search process, and then iterates two search phases until a maximum CPU time is reached:

- a *perturbation phase*, during which x% of the observations are removed from the current plan with x a parameter to be set; the observations removed are chosen using a uniform random distribution;
- a *reoptimization phase*, during which the solution obtained after the perturbation phase is reoptimized; this phase reuses the greedy search scheme to fill the plan again; to diversify search, the heuristic used for filling the plan is chosen randomly among the portfolio of decision rules.

Throughout the iterations, the best plan found is systematically recorded, based on a lexicographic ordering of the optimization criteria.

Post-processing: slew optimization

For SVOM, the cumulated slew can actually be considered as less important than the other objectives because the pointing of the telescope will be perturbed anyway due to the commitment to carry out GRBs and ToOs. This is why the slew objective is considered only at the last optimization step.

To achieve an observation sequence $seq=[r_1,\ldots,r_n]$ that optimizes the cumulated slew, local search techniques developed for routing problems with time windows are used [39]. The corresponding techniques are *or-opt moves* [2], which try to better position a block of k successive observations inside the observation sequence, and *2-opt moves* [10], which consider k successive observations and try to perform them in the reverse order. Local moves are carried out while improvements are made, i.e., until a locally optimal cumulated slew is reached. Then, an ILS search scheme is used, with a perturbation phase that randomly updates the ordering of some observations and a slew reoptimization phase based on or-opt and 2-opt moves again. This mechanism is applied until the maximum CPU time allowed is reached.

Experimental results

Experiments were performed on several data sets to evaluate the performance of the algorithms for the initial planning phase, when the full-year schedule must be synthesized. Table 2 gives the results obtained on one data set involving approximately 600 GP requests. In this scenario, greedy search delivers good quality solutions in a few seconds. Experiments on the different selection and insertion heuristics indicate that the heuristic that fills the earliest idle periods is a good compromise between the computation times and the plan quality. Also, ILS leads to better results in terms of number of observations performed and in terms of observation duration, especially for GP requests of priority B. Using ILS can however penalize the reschedulability criterion a bit. In 5 minutes, ILS performs up to 100 plan perturbation-reoptimization steps. Last, the slew optimization phase has a strong impact on the cumulated slew (50% reduction), while having little impact on the reschedulability objective.

We also simulated the dynamic behavior of the planning system by calling the planner every week to schedule the remaining part of the year. The objective was to determine whether GP requests were completed despite the random arrival of ToOs and GRBs. We conducted the experiments on a set R composed of 426 GP-real-life requests. For each set of scheduling parameters, 20 one-year simulations were completed, each time with random occurrence dates for ToOs and GRBs. For these 20 simulations, Table 3 shows the mean number n_{GP} (respectively, n_{GP-A}) of GP requests (respectively, GP requests of priority A) that are completed up to 95% at least at the end of the year, and the mean observation time dedicated to these requests (columns t_{GP} and t_{GP-A}).

The results show that the ILS phase, which starts here from the solution found by the greedy search, improves the performance obtained after one year of mission time.

	n_{GP}	t_{GP} (days)	n_{GP-A}	t_{GP-A} (days)
greedy	206	142	177	116
ILS - perturb 10%	**218**	**160**	**186**	**132**

Table 3 - Number of observations and observation duration after a one-year simulation

Short-term planning

The techniques presented so far concern the long-term planning phase of SVOM. For the short-term planning phase, another approach is defined to deal with the arrival of ToOs [19]. The main idea is that when a ToO request is received, there is a need to construct a ToO observation plan over a one-day span. For this, the relevant sky areas are decomposed into tiles, and the main task of the short-term planner is to select a subset of the tiles and define the order in which the selected tiles are observed, given constraints on stabilization times, visibility windows, and maximum number of tiles per orbit, and given the likelihood that a given tile contains the source sought. Basically, this tile-sequencing process is a chronological greedy algorithm that iteratively inserts tile observation activities at the end of the current plan. At each step, the algorithm selects either a tile that has the maximum likelihood of containing the source (static selection heuristic), or a tile that maximizes a value depending both on the source presence likelihood and on the last tile of the plan (dynamic selection heuristic).

Lessons learned

On the modeling side, the mission constraints for SVOM are very simple but the optimization criteria required a bit more effort to be derived. On the implementation side, one lesson is that it was very useful to have a generic tool for managing the core constraint-based model. It allowed the lexicographic ordering between the different objective functions to be easily updated during the project. It also allowed various search parameters to be tested. Also, to optimize the slew, the use of standard Operations Research techniques was very beneficial.

Last, even if there is no theoretical guarantee on the stability of the plans produced for GP observations, some settings used favor replanning observations that have already been started or that are more time-constrained. This is why the approach manages to complete observations up to 95% at the end of the year. More precisely, the uncertainty about GRBs and ToOs is handled through two main mechanisms: (1) regular replanning each week during the year, which allows parts of GP observations aborted because of high priority events to be programmed again, and (2) optimization of the reschedulability metric. Given that GRB events will cover around 25% of the useful mission time, and given that ToOs will cover between 15% and 40% of the time, more proactive planning strategies have been sought. In particular, we started to develop a planning process in which, in a first phase, all observation durations are scaled proportionally to the percentage of useful mission time covered by random events. Returning to nominal durations, opportunistic observations can then be added to avoid under-using the telescope. This process has the potential to produce more stable plans, since the plan in which all observation durations are scaled could serve as a reference to be followed, as much as possible, over the year.

Planning for ARIEL

Mission description

We present in this section the ARIEL mission, which is the fourth medium-class mission within the ESA Cosmic Vision science program, with a launch planned in 2028. The total lifetime duration is four years and could be extended for two additional years.

In astronomy, a *transit*, or *occultation*, is the phenomenon when a planet passes directly in front of or behind its host star from the satellite point of view. ARIEL will analyze the atmospheres of around 1000 planets (warm and hot transiting gas giants, Neptunes and super-Earths) orbiting around a range of host star types, using transit and occultation spectroscopy in the $\sim 2-8\ \mu$m spectral range and broadband photometry in the optical to determine their chemical composition and physical conditions. The results will help scientists better understand planet formation, putting our own Solar System in context.

Transit and occultation spectroscopy methods, whereby the signal from the star and planet are differentiated using precise knowledge of the planetary ephemerides, allow atmospheric signals from the planet to be measured. Figure 6 illustrates the orbital lightcurve of the transiting exoplanet HAT-P-7b as observed by Kepler [3], which the methods adopted by ARIEL are based on.

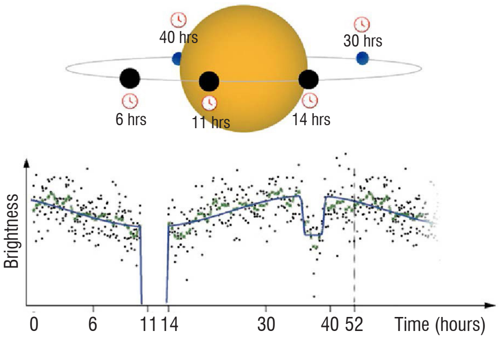


Figure 6 - Orbital lightcurve of exoplanet HAT-P-7b [3]

Observation objectives

ARIEL will visit a large and well-defined set of a few hundred targets. Repeated visits are required to build up the Signal to Noise Ratio (SNR) of individual target spectra. Most of the targets will require between one and a few tens of transit/occultation observations, depending on the brightness and spectral type of the host star and planetary radius and temperature. The maximum duration of a visit to a target system will be less than 10 hours. The time between successive transit/occultation observations will depend on the orbital period and could be as little as a fraction of a day to as long as a few days, with the exception of highly eccentric orbit planets.

Every targeted planet is associated with one or more of the following scientific objectives:

- Basic survey objective (denoted by *Survey*): it requires a minimal number of transit or occultation observations in order to assess the scientific interest of the planet and its main coarse characteristics. It basically addresses all of the targeted planets.
- Deep survey objective: it requires additional transit or occultation observations in order to obtain a better SNR and to achieve a detailed characterization of the planet and its atmosphere. It addresses a large sub-sample of the set of targeted planets.
- Benchmark objective: it requires even more transit or occultation observations to reach the best possible SNR allowing a very detailed knowledge of the chemistry and dynamics of the planet. It addresses a few tens of planets orbiting very bright stars.

According to the definition of these objectives, the global observing strategy that is promoted is to prioritize, at the beginning of the lifetime, the observations for planets with either a benchmark objective first or a basic survey objective, and to postpone the observations accounting only for a *Deep* objective after a period of about 1 or 1.5 years from the beginning of the mission.

Payload calibration needs and operational constraints

During the mission, it is assumed that the operational needs are as follows:

- one housekeeping sequence during 4 hours every 28 days, with a tolerance of +/- 2 days;
- one long calibration sequence during 6 hours every 30 days, with a tolerance of +/- 10 days;
- one short calibration sequence during 1 hour every 36h, with a tolerance of +/- 12h;
- some specific targets may require a short calibration sequence to be performed just before/after the transit observation.

All of the calibration sequences use dedicated targets to point to numerous wide spread predefined G-stars. The mission data downlinks are assumed to not conflict with the observations.

Mission planning challenges and constraints

A mission planning process is required to establish the observation schedule from a set of requests R initially selected and classified by a scientific board. It provides, in particular, the allocations for the three objectives and the types and numbers of observations needed. The mission planning process must take into account the observing strategy promoted for ARIEL and all of the relevant system constraints.

Note that at the time when the study on ARIEL was performed, it was still a mission candidate. Consequently, the goal of the study presented was the evaluation of the global scientific impact of the mission in the following way:

- the various objectives could be achieved for the highest number of targeted planets. In particular, the *Benchmark* objective could be met for all of the planets concerned and a high number of *Survey* objectives could be met at the beginning of the lifetime (500 surveys done the first year);
- a sufficient part of the useful mission time was devoted to scientific observations (the goal is of 80%) avoiding long time periods without observation activities;
- the regular calibration and housekeeping tasks could be fulfilled most of the time.

We describe here the main outcomes of this study [38]: a constraint-based model, two algorithmic approaches, experimental results, and lessons learned.

Constraint-based modeling

Input data

We consider the following input data:

- *mission dates* – start and end dates of the mission, and desired dates before which the *Benchmark*, *Survey* and *Deep* objectives should be performed;
- *satellite features* – slew speed and duration to wait after each slew in order to be thermally and mechanically stable;
- *operational tasks* – each operational task can either be a short calibration, a long calibration or a housekeeping task. Each type has an associated duration, period and flexibility;
- *calibration stars* – stars (name, right ascension and declination) that are the targets to point to for calibration tasks;
- *exoplanets* – each planet is characterized by a name, its orbiting star, its period and its coordinates (right ascension and declination). Candidate transit and occultation events for each planet can be pre-computed along with their start and end dates. The duration of each event is also given as an input;
- *scientific requests* – description of all of the planet observations that can be carried out during the mission. Each scientific request targets one planet and can focus either on its transit events or occultation events.

The numbers of observations required to achieve each objective (among *Survey*, *Deep*, or *Benchmark*) are also given as an input. A request is called a *Survey* request if and only if its number of *Deep* and *Benchmark* observations is equal to 0. We define *Deep* and *Benchmarks* requests along the same lines.

Orthogonally to the objective dimension, requests can also be partitioned into:

- a set of *simple requests*, for which there is no required task before and after observations of events;
- a set of *requests with calibration*, for which each observation must be preceded and/or followed by a short calibration that is performed on the closest G-star to the planet; such requests can be considered as simple requests with a longer duration, for which the pointing target can be either the planet or the G-star depending on the calibration requirement;
- a set of *requests with additional observations*, for which a specific task must be performed before and/or after each observation of an event.

Variables

There are two classes of discrete decision variables:

1. Boolean variables for deciding which observation candidates are in the final plan; the start and end dates of these observations are fixed (dates associated with transit and occultation events) and therefore do not require any decision;
2. Boolean variables for deciding which operational tasks are in the final plan and integer variables for their start and end dates.

Constraints

We consider the following constraints:

- *no overlap*: the tasks performed by the telescope should not overlap. This takes into account the slewing duration between the pointing targets of tasks and the stabilization duration;
- *requests with additional observations*: for a request with additional observations, a candidate observation is part of the final plan if and only if the observation just before and/or just after is also part of the final plan;
- *operational tasks periodicity*: operational tasks must be performed periodically, with a given flexibility as specified by their types;
- *deep observation release date*: observations that allow the objective of *Deep* requests to be achieved cannot be made before a fixed date.

Objective functions

We first list the different elementary criteria taken into account for the mission and then describe two combinations considered.

- $crit\_nReq$ - maximize the number of completed requests;
- $crit\_nB$ (respectively, $crit\_nD$ and $crit\_nS$) - maximize the number of completed *Benchmark* (respectively, *Deep* and *Survey*) requests;
- $crit\_nB\_d$ (respectively, $crit\_nS\_d$) - maximize the number of *Benchmark* requests (respectively, *Survey* objectives) completed before the desired date;
- $crit\_nOp$ - maximize the number of operational tasks carried out;
- $crit\_nHk$ (respectively, $crit\_nLc$ and $crit\_nSc$) - maximize the number of housekeeping (respectively, long and short calibrations) carried out during the mission.

All of the criteria above do not have the same weight, because of the mission observation objectives. Following these mission priorities, we considered two main criteria:

- *Upper bound criterion*. This criterion does not take into account the request types, but rather focuses on the maximizations of the number of completed requests and the number of operational tasks in the plan, thus giving a clue of what could be an upper bound for the planning of ARIEL regarding the number of completed requests. Formally, the upper bound criteria $crit_{UB}$ is defined by a vector $[crit\_nReq, crit\_nOp]$ that is optimized lexicographically.
- *Criterion with request types*. This criterion, denoted as $crit_{Type}$, takes into account the type of requests and the date before which *Benchmark* requests should be completed. Formally, $crit_{Type}$ is vector $[crit\_nB\_d, crit\_nB, crit\_nD, crit\_nS, crit\_nHk, crit\_nLc, crit\_nSc]$ and is optimized lexicographically.

Greedy search and local search

We developed two different approaches. The first one is greedy-based. The second one uses a Constraint Based Local Search paradigm.

Note that a third approach could be adapted from the one used for EChO [30], which is an earlier version of the ARIEL problem in which there was no distinction between *Survey*, *Deep* and *Benchmark* requests. EChO used a two-phase strategy, where first scientific requests are planned using genetic algorithms, and then as many operational tasks as possible are inserted to fill in the gaps in the plan. We did not experiment with this approach on the ARIEL benchmarks.

Greedy approach

Many greedy-based algorithms were developed for solving the ARIEL planning problem. We describe here a hierarchical greedy algorithm that gives the best results, as described in Section "Experimental results".

Starting from an empty plan, the algorithm selects a candidate task and tries to insert it into the plan. More precisely, all requests and candidate observations are first labeled as *unprocessed*. While there is an unprocessed request, one request r is selected by considering first *Benchmark* requests, then *Deep* and *Survey* ones and with a tie-break favoring those with the least flexibility in further observations. Then, the first unprocessed observation for this request is inserted into the final plan if and only if its insertion does not violate any constraint. The observation is marked as *processed*. If the objective is achieved or if all candidate observations of r have been processed, then r is also labeled as processed. If the *Survey* objective of r is not achieved, then all of the observations inserted for r are removed. When all requests have been considered, a similar procedure is followed for operational tasks.

This algorithm tends to maximize the number of completed requests, while taking into account the priority of the tasks. The dates before which benchmark requests and survey objectives should be completed are also considered, since the inserted task is always the first one chronologically.

In order to increase the cumulated duration of activity of the satellite, two procedures have been defined. First, we try to insert into the plan as many observations as possible but only for requests that have at least achieved their *Survey* objective. The second procedure extends the time during which calibration associated with scientific events are carried out.

Min-conflicts approach

The second approach is a Constraint Based Local Search (CBLS) approach [17]. It is particularly suited for handling constraint programming problems with large benchmarks, especially because various parts of the search space can be explored in a short time. The algorithm built on top of that approach is based on a min-conflicts algorithm [28]. It starts with an initial plan in which all requests are randomly fulfilled and constraints are all satisfied except for the no-overlap ones. Then, the objective of the algorithm is to remove all conflicts due to overlaps and then optimize the criteria $crit$ that can be equal to $crit_{UB}$ or $crit_{Type}$. The steps are as follows:

1. while there exists an overlap conflict or the criteria *can theoretically be improved*, we select an observation o_{old} in the plan;
2. we select a candidate observation o_{new} of the same request that, if inserted into the plan in place of o_{old}, either decreases the number of overlaps, or improves the value of $crit$ without deteriorating the number of overlaps. The insertion of o_{new} must also satisfy all constraints except for the no-overlap ones. A random tie-breaking is used to choose between candidate observations that improve the criteria in the same way;
3. we remove o_{old} from the plan and insert o_{new} instead;
4. we insert all housekeeping tasks in their corresponding temporal interval as early as possible if and only if they do not overlap with already inserted observations. We then proceed the same way for long and short calibrations.

These steps are repeated until the criteria reach the maximum theoretical value or until a maximum number of iterations is reached. If overlap conflicts still exist, an observation from the plan is chosen and removed. The algorithm starts again from Step 1 and the procedure is repeated.

Note that $crit\_nReq$ cannot be improved if its value is the number of requests. Likewise, $crit\_nOp$ cannot be improved when its value is the number of operational tasks. These are the maximum theoretical values of criteria.

Experimental results

In this section, we present experiments carried out on a set of data provided by scientists working on the design of ARIEL.

Scenario

The benchmark that we worked on includes 710 planets and 728 corresponding requests. 61 are *Benchmark*, 240 are *Deep* and 427 are *Survey* requests. 709 requests have a *Survey* objective. A short calibration is required before and after each observation of an event for 34 requests. The benchmark does not contain requests with additional observations.

There are 128439 candidate observations and 4345 observations are required to complete all of the requests. The corresponding duration is equal to 3.52 years, which is longer than the mission duration.

Implementation

The greedy approach was implemented with Scilab on a Intel Core i3 processor with 2GB of RAM. The CBLS experiments were run on a four-Xeon 2.80GHz processors with 8GB of RAM. We implemented the algorithm in Java on top of library InCELL [36].

Results

Results of the experiments are illustrated in Figure 7 and Table 4. For analyzing these results, we consider the various different objectives of this study.

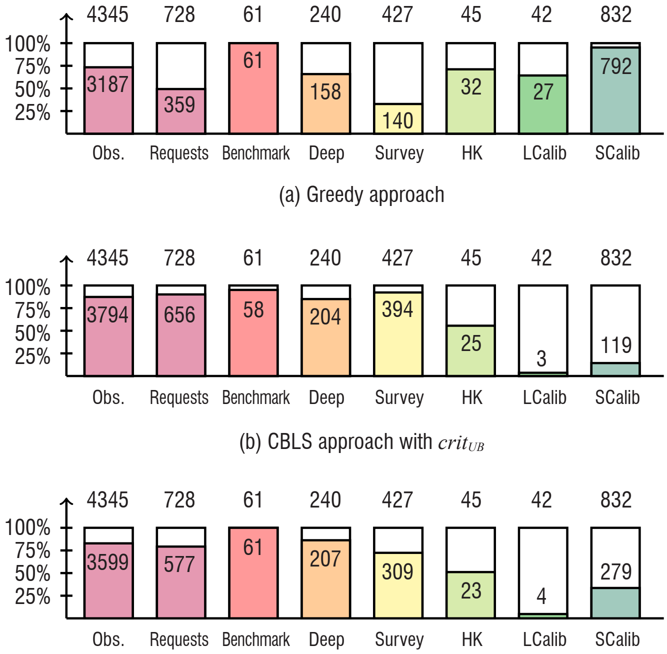
- *Number of completed requests* - As expected, $crit\_nReq$ is maximized with the CBLS approach along with the upper-bound criteria. When the types are taken into account, the best approach is CBLS with the criterion $crit_{Type}$.
- *Cumulated duration of activity* - All approaches generate a plan in which the satellite is active for more than 80% of the mission duration.
- *500 Survey objectives during the first year* - The best result is obtained by the CBLS approach with the criterion $crit_{Type}$. Better results might be hard to obtain, since the plan is saturated during the first year because of *Benchmark* requests.
- *Operational tasks* - All approaches have rather low results for that objective. The best approach is the greedy one: given that there is less time dedicated to observations, there is more time for operational tasks. If the insertion of operational tasks were to be a constraint instead of a preference, the overall results would really be lower. For instance, in this case the number of completed requests falls to 502 for CBLS with $crit_{Type}$ (44 *Benchmark*, 178 *Deep* and 280 *Survey*), and the cumulated activity duration of the satellite represents 74% of the mission duration. Given that this objective has a lower priority, it is not really achieved by the proposed approaches.

Lessons learned

The planning problem associated with the ARIEL mission stresses several challenges. First, it combines both an allocation problem for scientific observations and a scheduling problem for operational tasks, which implies the use of different types of decision variables. Second, there are several non-classical criteria, such as deadlines. Then, it requires requests covering multiple observations to be dealt with, also known as *linked observations*. Fourth, the number of requests can also be quite challenging. For the studied benchmark, there are more than 3 million potential conflicts between candidate observations. Finally, as the design of the mission was still on-going, there was a real need for a generic model and implementation.

Given that the ARIEL mission has been selected, the next step is to define more precisely the operational constraints on the system and make them high-priority. Moreover, the scheduler should be modified in order to dynamically integrate new sets of requests.

From a technical point of view, the initial plan produced maximizes the criteria but violates some constraints. While this outperforms the greedy approach along with several heuristics, some other approaches should also be investigated, such as CBLS starting from an empty plan or Iterated Local Search as mentioned earlier.



(a) Greedy approach

(b) CBLS approach with $crit_{UB}$

(c) CBLS approach with $crit_{Type}$

Figure 7 - Features of the plan generated by each approach. The first column contains the number of observations inserted into the plan, the second one contains the number of completed requests, and the three next columns detail how requests are satisfied per type. The last three columns contain the number of operational tasks that are part of the plan

Approach	*Activity dur.*	*Scientific activity dur.*	*Survey obj. 1st year*
Greedy	80%	72%	411
CBLS - $crit_{UB}$	89%	83%	295
CBLS - $crit_{Type}$	83%	77%	467

Table 4 - For each approach, the percentage of the mission duration during which the telescope is active, the percentage of the mission duration dedicated to scientific activities and the number of requests with *Survey* objective completed during the first year of the mission among the 709

Conclusion and future work directions

This article presented three telescope mission planning systems developed in the past, giving for each a description of the model, the algorithms, the results, and some lessons learned. Based on this experience, we believe that it may be relevant to address the three following challenges in the future.

A generic tool for space telescope planning

Given that many space telescopes share similarities in terms of planning, it would be useful to develop a generic mission manager. At the moment, we have a generic constraint-based optimization tool (InCELL) that allows us to quickly define new mission specific planners. To gain in genericity, we could try to define a generic telescope mission planning tool on top of InCELL, as done by the *Space Telescope Science Institute* (STScI) with Spike [22, 26]. In such a generic tool, we could have a domain specific language, a set of predefined decision rules (e.g., selection heuristics and insertion heuristics for greedy search), and a set of predefined local search and metaheuristics. On this basis, it would be possible to more easily compare several search algorithms (stochastic hill-climbing, iterated local search, min-conflicts, etc.). Also, the precise settings of the search parameters could be optimized by machine learning techniques.

Uncertainty management

Another challenge concerns the way in which the uncertainty is managed for space telescope missions. As seen previously, the degree of uncertainty is quite high for space telescopes, due to random events like ToOs or GRBs. The current practice is that in each mission specification, there is an indirect way of dealing with such events: limitation of the planned observation time during some periods, optimization of reschedulability measures, computation of flexible realization windows, etc. On this point, there is a need to compare the different approaches proposed in the literature. It is likely that it could be relevant to exploit a coarse-grain model of these random events and let the mission planner optimize the plans by using explicit stability measures in addition to the other performance measures. By doing so, the behavior of the telescope would be more predictable for the end-users. From an algorithmic point of view, the approach could be to search for an easily reschedulable backbone plan, and to add opportunistic observations when the occurrence rate of random events is lower than expected. Moreover, to manage uncertainty, it could be useful to develop on-board autonomy concepts, not only to automatically trigger follow-up observations when relevant events are detected, but also to abort observation activities when some conditions are not met.

Planning for several telescopes

Lastly, nowadays there is a significant number of space telescopes. Even if they do not embed the exact same instruments, it could be interesting to globally optimize their activities given a set of candidate targets, that is, to have a kind of centralized telescope planning tool or a kind of distributed planning engine with automated negotiation steps, at least at the level of each space agency to manage its "constellation" of space telescopes. Several reasons could motivate this choice. First, scientists might want to post an observation request r requiring several telescopes embedding complementary instruments. In this case, to maximize the scientific return, there would be a need to coordinate the decisions of the telescope mission planning centers, so that the elementary observations associated with r are either all selected or not selected, and so that these observations are carried out during similar periods of the year if needed. Another reason would be to share the use of the telescopes among the possible GRBs and ToOs. As an example, if a single random event is detected, it is not an issue to point all telescopes to the corresponding target. However, when several random events occur simultaneously, there could be some level of coordination between the mission centers to share the usage of the ground and space observatories because, in the end, all telescopes share common long-term goals ■

References

[1] A. ALLAHVERDI, C. T. NG, T. C. E. CHENG, M.Y. KOVALYOV - *A Survey of Scheduling Problems with Setup Times or Costs.* European Journal of Operational Research, 187(3):985-1032, 2008.

[2] G. BABIN, S. DENEAULT, G. LAPORTE - *Improvements to the or-opt Heuristic for the Symmetric Traveling Salesman Problem.* Journal of the Operational Research Society, 58:402-407, 2007.

[3] W. J. BORUCKI, D. KOCH, J. JENKINS, D. SASSELOV, R. GILLILAND, N. BATALHA, D.W. LATHAM, D. CALDWELL, G. BASRI, T. BROWN, *et al.* - *Kepler's Optical Phase Curve of the Exoplanet HAT-P-7b.* Science, 325(5941):709-709, 2009.

[4] J. BRUMFITT, P. GÓMEZ-ALVAREZ, A. VILLACORTA, P. GARCÍA-LARIO, R. LORENTE, L. O'ROURKE - *Herschel Mission Planning Software*. Proc. of the 7th International Workshop on Planning and Scheduling for Space (IWPSS'11), 2011.

[5] F. CASTELLINI, M. R. LAVAGNA - *Advanced Planning and Scheduling Initiative XMAS Tool: AI for Automatic Scheduling of XMM-Newton Long Term Plan.* Proc. of the 6th International Workshop on Planning and Scheduling for Space (IWPSS'09), 2009.

[6] A. CESTA, S. FRATINI - *The Timeline Representation Framework as a Planning and Scheduling Software Development Environment.* Proc. of the 27th Workshop of the UK Planning and Scheduling Special Interest Group (PlanSIG'08), 2008.

[7] A. CESTA, S. FRATINI, A. DONATI, H. OLIVEIRA, N. POLICELLA - *Rapid Prototyping of Planning and Scheduling Tools.* Proc. of the 3rd IEEE International Conference on Space Mission Challenges for Information Technology, 2009.

[8] J. COLOMÉ, P. COLOMER, J. GUARDIA, I. RIBAS, J. CAMPRECIÓS, T. COIFFARD, L. GESA, F. MARTINEZ, F. RODLER - *Research on Schedulers for Astronomical Observatories.* Proc. of SPIE, 2012.

[9] B. CORDIER, F. DESCLAUX, J. FOLIARD, S. SCHANNE - *SVOM Pointing Strategy: How to Optimize the Redshift Measurements?* AIP conference proceedings vol 1000, 2008.

[10] G. A. CROES - *A Method for Solving Traveling Salesman Problems.* Operations Research, 6:791-812, 1958.

[11] X. DUPAC, J. TAUBER - *Scanning Strategy for Mapping the Cosmic Microwave Background Anisotropies with Planck.* Astronomy and Astrophysics, 2005.

[12] A. GARCIA-PIQUER, I. RIBAS, AND J. COLOMÉ - *Artificial Intelligence for the EChO Mission Planning Tool.* Experimental Astronomy, 40(2-3):671-694, 2015.

[13] M. GIULIANO - *Achieving Stable Observing Schedules in an Unstable World.* Proc. of the 7th Conference on Astronomical Data Analysis Software and Systems (ADASS'97), 1997.

[14] M. GIULIANO - *The Mystery of the Missing Requirements: Optimization Metrics for Spike Long Range Planning.* Proc. of the 8th International Workshop on Planning and Scheduling for Space (IWPSS'13), 2013.

[15] M. E. GIULIANO, M. D. JOHNSTON - *Multi-Objective Evolutionary Algorithms for Scheduling the James Webb Space Telescope.* Proc. of the 18th International Conference on Automated Planning and Scheduling (ICAPS'08), 2008.

[16] P. GÓMEZ-ALVAREZ, J. BRUMFITT, R. LORENTE, P. GARCÍA-LARIO - *HILTS: The Herschel Inspector and Long-Term Scheduler.* Proc. of the 20th annual conference on Astronomical Data Analysis Software and Systems, 2011.

[17] P. VAN HENTENRYCK, L. MICHEL - *Constraint-Based Local Search.* The MIT Press, 2005.

[18] J. JAUBERT, V. DEBOUT - *Mission Planning and Scheduling Activities Conducted at French Space Agency (CNES) for the ATHENA Mission.* Proc. of the 11th International Workshop on Planning and Scheduling for Space (IWPSS'19), 2019.

[19] J. JAUBERT, V. DEBOUT - *Mission Planning and Scheduling Activities Conducted at French Space Agency (CNES) for the SVOM Mission.* Proc. of the 11th International Workshop on Planning and Scheduling for Space (IWPSS'19), 2019.

[20] M. D. JOHNSTON - *Spike: AI Scheduling for Hubble Space Telescope after 18 Months of Orbital Operations.* Proc. of the 6th Conference on Artificial Intelligence Applications, 1992.

[21] M. D. JOHNSTON, G.E. MILLER - *Artificial Intelligence Scheduling for NASA's Hubble Space Telescope.* Proc. of the 5th Conference on AI Systems, 1990.

[22] M. D. JOHNSTON, G.E. MILLER - *Spike: Intelligent Scheduling of Hubble Space Telescope Observations.* Intelligent Scheduling, pages 391-422. Morgan Kaufmann Publishers, 1994.

[23] S. KIRKPATRICK, C. GELATT, M. VECCHI - *Optimization by Simulated Annealing.* Science, 220, 1983.

[24] M. KITCHING, N. POLICELLA - *A Local Search Solution for the INTEGRAL Long Term Planning.* Proc. of the 7th International Workshop on Planning and Scheduling for Space (IWPSS'11), 2011.

[25] L. KRAMER, L. BARBULESCU, S. SMITH - *Understanding Performance Tradeoffs in Algorithms for Solving Oversubscribed Scheduling.* Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07), pages 1019-1024, 2007.

[26] L. A. KRAMER, M. E. GIULIANO - *Reasoning about and Scheduling Linked HST Observations with Spike.* Proc. of the 1st International Workshop on Planning and Scheduling for Space (IWPSS'97), 1997.

[27] H. R. LOURENÇO, O. C. MARTIN, T. STÜTZLE - *Handbook of Metaheuristics.* Chapter Iterated Local Search: Framework and Applications, pages 363-397. Springer, 2010.

[28] S. MINTON, M. D. JOHNSTON, A. B. PHILIPS, P. LAIRD - *Minimizing Conflicts: a Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems.* Artificial Intelligence, 58(1-3):161-205, 1992.

[29] D. S. MITTMAN, R. HAWKINS - *Scheduling Spitzer: The SIRPASS Story.* Proc. of the 8th International Workshop on Planning and Scheduling for Space (IWPSS'13), 2013.

[30] J. C. MORALES, J.-P. BEAULIEU, V. COUDÉ DU FORESTO, M. OLLIVIER, I. ORTEGA CASTELLO, R. CLÉDASSOU, J. JAUBERT, P. VAN-TROOSTENBERGHE, R. VARLEY, I. P. WALDMANN, *et al.* - *Scheduling the EChO Survey with Known Exoplanets.* Experimental Astronomy, 40(2-3):655-670, 2015.

[31] OMITRON. Swift TAKO user guide, 2003.

[32] I. ORTEGA CASTELLÓ - *Means for Analysing the Operational Capabilities of Mission Programming.* Master's thesis, ISAE, September 2013.

[33] C. PRALET - *An Incomplete Constraint-Based System for Scheduling with Renewable Resources.* Proc. of the 23rd Conference on Principles and Practice of Constraint Programming (CP'17), 2017.

[34] C. PRALET, S. ROUSSEL, J. JAUBERT, J. QUEYREL - *Observation Planning for the SVOM Space Telescope.* Proc. of the 10th International Workshop on Planning and Scheduling for Space (IWPSS'17), 2017.

[35] C. PRALET, G. VERFAILLIE - *AIMS: A Tool for Long-Term Planning of the ESA INTEGRAL Mission.* Proc. of the 6th International Workshop on Planning and Scheduling for Space (IWPSS'09), 2009.

[36] C. PRALET, G. VERFAILLIE - *Dynamic Online Planning and Scheduling using a Static Invariant-Based Evaluation Model.* Proc. of the 23rd International Conference on Automated Planning and Scheduling (ICAPS'13), 2013.

[37] R. RAGER, M. GIULIANO - *Evaluating Scheduling Strategies for JWST Momentum Management.* Proc. of the 5th International Workshop on Planning and Scheduling for Space (IWPSS'06), 2006.

[38] S. ROUSSEL, C. PRALET, J. JAUBERT, J. QUEYREL, B. DUONG - *Planning the Observation of Exoplanets: the ARIEL Mission.* Proc. of the 10th International Workshop on Planning and Scheduling for Space (IWPSS'17), 2017.

[39] M. W. P. SAVELSBERGH - *Local Search in Routing Problems with Time Windows.* Annals of Operations Research, 4(1):285-305, 1985.

[40] D. SMITH - *Choosing Objectives in Over-Subscription Planning.* Proc. of the 14th International Conference on Automated Planning and Scheduling (ICAPS'04), pages 393-401, 2004.

[41] R. STEEL, M. NIEZETTE, A. CESTA, S. FRATINI, A. ODDI, G. CORTELLESSA, R. RASCONI, G. VERFAILLIE, C. PRALET, M. LAVAGNA, A. BRAMBILLA, F. CASTELLINI, A. DONATI, N. POLICELLA - *Advanced Planning and Scheduling Initiative: MrSpock AIMS for XMAS in the Space Domain.* Proc. of the IJCAI-09 Workshop on Artificial Intelligence in Space, 2009.

AUTHORS

Cédric Pralet graduated from an engineering school specializing in the field of aeronautics and space in 2003 and achieved his PhD in Computer Science in 2006. Since then, he has been working at ONERA and obtained his authorization to supervise research ("Habilitation à Diriger les Recherches", HDR) in 2015. His research interests concern constraint-based optimization and automated planning and scheduling, applied to problems in the aerospace field.

Stéphanie Roussel graduated from an engineering school specializing in the field of aeronautics and space in 2007 and achieved her PhD in Computer Science in 2010. From 2010 to 2014, she worked as a research engineer at the CRIL (Centre de Recherche en Informatique de Lens). Since 2015, she has been working at ONERA, mainly on constraint-based optimization, knowledge representation and reasoning, and their application in the fields of aeronautics and aerospace.

Jean Jaubert graduated from engineering schools specializing in the fields of mechanics, aeronautics and space in 1989. Later, he joined the Centre National d'Etudes Spatiales (CNES), the French Space Agency, to work on flight dynamics and satellite guidance and control issues within the framework of space projects for Earth Observation applications. Then, he specialized in the design and implementation of mission planning and programming functions for various space missions in the fields of Earth Observation, Science and Astronomy.

Julien Queyrel received his spacecraft mechanics engineering diploma from ISAE-SUPAÉRO and his MSc in astrophysics from the University of Toulouse in 2007. He completed his studies in 2010 when he obtained his PhD in astrophysics. Afterwards, he worked as a subcontractor for the French space agency (CNES) on electronic intelligence and mission programming. Since 2018, he has been working at ONERA as a research engineer on the use of numerical weather forecast models to predict microwave propagation through the troposphere.

F. Boniol, A. Chan-Hon-Tong,
A. Eudes, S. Herbin,
G. Le Besnerais, C. Pagetti,
M. Sanfourche
(ONERA)

E-mail: frederic.boniol@onera.fr

DOI: 10.12762/2020.AL15-06

Challenges in the Certification of Computer Vision-Based Systems for Civil Aeronautics

Computer vision techniques have made considerable progress in recent years. This advance now makes possible the practical use of computer vision in civil drones or aircraft, replacing human pilots. The question that naturally arises is then to provide a way to certify those types of systems at a given level of safety. The aim of the article is, firstly, to understand the gap between today's computer vision systems and the current certification standards; secondly, to identify the key activities that must be fulfilled in order to make computer-vision systems certifiable and, thirdly, to explore some recent works related to these key activities.

Introduction

Computer vision techniques have made considerable progress in recent years. One of the most recent successes in computer vision has been achieved through the development of Deep Learning methods. It seems unlikely that this trend will backtrack radically on short notice: most signal and data analysis approaches will now include somewhere in their processing pipeline one or several components that have been designed using machine learning techniques. Before the advent of Deep Learning, more conventional methods based on geometric vision already offered very interesting performances for autonomous localization.

The performance gain obtained by these techniques now makes possible the practical use of computer vision in complex systems, and particularly in surface vehicles and in civil drones or aircraft, replacing human pilots.

Driving Automation Systems for On-Road Motor Vehicles have been widely studied. A dedicated standard [80] has been published by SAE to propose a set of recommended practices and a taxonomy describing the full range of levels of driving automation in on-road motor vehicles. The concerns and risks associated with on-road autonomous vehicles are also discussed in [94]. To address these risks, the authors explore various strategies that can be adopted and emerging responses by governments. They show that, thus far, authorities have generally avoided binding measures and have focused on creating councils and work groups, in order to not slow down the development of autonomous vehicles.

In this article, we focus on the civil aeronautics domain. In this domain, the strategy is quite different. An aircraft (autonomous or not) cannot enter service without being certified from a safety point of view. One of the main rules to ensure safety is "see-and-avoid": it is the responsibility of the (human or artificial) pilot to detect any abnormal situation or any risk of collision and to ultimately take control of the vehicle.

The issue that naturally arises for allowing the use of computer vision in civil aeronautical vehicles is to provide a way to certify a given level of safety. This is a difficult issue for such processes, which are effective in their empirical domain of expertise, but it is often not possible to state why they are so.

What is a computer vision based system?

Let us begin by illustrating what a computer vision-based system is. As an example, let us consider a vision-based navigation system representative of embedded systems in robotics, drones, autonomous cars, or automated taxiway driving for an aircraft. A simplified generic architecture is sketched in Fig. 1. Visual information stems from two cameras (denoted as left and right camera) mounted together on a stereoscopic rig.

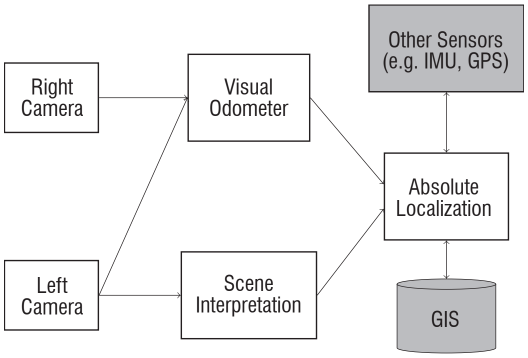


Figure 1 – A stereovision-based system

The Visual Odometer (VO) component exploits sequences of stereoscopic images to estimate the trajectory (position and orientation) with respect to some *relative* reference coordinate system.

The Scene Interpretation (SI) component uses frames from the left camera to build a description of the scene pertaining to the navigation task at hand. For instance, SI has been designed to provide bounding boxes (BB) around objects of interest, which are essentially of two types:

- Landmarks: objects referenced in a local GIS (Geographic information system) of the area;
- Obstacles.

SI should also be able to more precisely characterize detected objects of each type; for instance, providing an identification of landmarks (e.g., a traffic sign and its meaning, a ground sign and its class, etc.); and providing a category for each obstacle (e.g., moving/static object, person/car/truck, etc.).

The Absolute Localization box combines information from all other components to estimate the position/orientation of the mobile in an absolute world coordinate frame, such as WGS 84 (World Geodetic System 1984). The basic operation here is to match objects extracted by SI to landmarks in the GIS, so as to change and/or refine the estimated trajectory. Matching landmarks is aided by characteristics provided by SI and by the approximate 3D localization deduced from VO information. We will not enter into further details regarding the Absolute Localization component.

In this paper, we will focus on the two functions *VO* and *SI*. For both of these we will identify the gap between the current standards and their specifics. Indeed, if such a vision-based system is to be embedded into some operational system, these two components would have to be compliant with some certification framework, depending on the application field. The choice of these two functions is motivated by the fact that they represent two major trends in current vision resources. *VO* belongs to geometric vision, which is aimed at extracting geometric information from images, a field that has been theorized for instance in [41]. Although very different in their implementation details, many such geometric codes are used nowadays in robotic systems. *SI* is representative of the numerous recent codes for image-based scene understanding driven by machine learning techniques. Moreover, as said before, their combination opens the way to realistic vision-based systems.

Problem

The article focuses on the civil aeronautics domain. In this domain, an aircraft is allowed to enter in operation if the manufacturer has obtained a type certificate from the certification authorities. For that, the aircraft manufacturer must demonstrate the compliance of its product with the regulatory requirements [25]. An accepted means of compliance with the requirements is to rely on mature standards, such as the ARP 4754A [79] for the system's development process, or such as the DO 178C [75] for the software development process. When using these means to prove that a product is trustworthy, the certification activities consist in providing a detailed documentation, and justifications, that argue how the development process is indeed compliant with the standard.

The certification activities must cover all of the levels of the development process. In the embedded field, the development process is usually divided into four levels:

- *function*: specification of the expected behavior, the usage domain, and the constraints of an avionic function;
- *algorithm*: *i.e.*, the methods, the structure, the algorithmic principles, etc., used to fulfil the avionic function;
- *source code*: *i.e.*, the software modules, which are compiled and transformed into executable object code;
- *item*: *i.e.*, all of the low level components, whether they are hardware (e.g., processors, cameras, etc.) or software (e.g., middleware, kernel devices, etc.).

For instance, Table 1 illustrates these four levels for the VO and SI functions.

Level	VO	SI
Function	Estimates the relative 3D-position/orientation and provides error covariance	Predicts position in image coordinates and the category of objects, and associates them with a confidence score
Algorithm	Feature tracking in a frame flow + statistical estimation	Machine learning based predictor – Neural networks
Source code	C/C++ development	Development frameworks in Python, C/C++
Item	Executable object code + middleware + processor + camera, etc.	Executable object code + processor + GPU + libraries for neural networks + camera, etc.

Table 1 – VO and SI implementation

Existing certification approaches for avionic functions, as shown in Section "Certification practice for civil avionic systems", require strong relationships between levels (such as conformity and traceability) and strong properties (such as determinism). However, as discussed in Section "Certification practice for civil avionic systems", some of these properties are often not shared by vision systems. For instance, *VO* uses optimizer algorithms to compute the best position, and optimization partly relies on the random operation of outlier rejection (RANSAC, Random Sample Consensus, [29]). Similarly, *SI* is typically based on machine learning techniques applied to deep neural networks (DNN) [33]. In this case, it becomes difficult to ensure the traceability between each line of source code and the functional level.

Another difficulty arises from the notion of failure. In safety terminology, *random failure* refers to item failures only, and *systematic failure* refers to software bugs. However, even in the absence of item and software failure, the perception functions may behave abnormally due to "bad" external conditions (e.g., bad weather conditions) or "bad" internal choices (e.g., bad random operation). The difficulty arises from the fact that these "bad" conditions partly depend on the internal algorithms, making the safety analysis more difficult.

As a result, most of today's computer vision systems do not meet the current certification standards for civil aeronautical vehicles, although the evolution of technology makes it possible to integrate such perception systems into drones or aircraft.

Objectives and organization of the article

Following this observation, the objectives of the article are:

- First, to understand the gap between today's computer-vision systems and the current certification standards;
- Second, to identify the key activities to be fulfilled to make computer-vision systems more certifiable;
- And last, to explore some recent works related to these key activities.

The paper is organized as follows:

- The current certification practices in the civil aeronautical field are presented in Section "Certification practice for civil avionic systems". More particularly, we discuss two certification standards: the ARP-4754A [79] dedicated to safety issues (see Section "Safety design process"), and the DO-178BC [75] dedicated to software issues (see Section "Algorithm and software development process"). They are discussed with respect to computer-vision systems and we show that they are rather inappropriate for this type of systems. We also discuss in Section "Computer vision based system development process: a data driven design logic" one of the novelties of computer vision ("novelties" with respect to conventional certified systems); that is, its data driven nature: the behavior of the systems is mainly defined or tested via a great number of data (called dataset).
- Section "Developing specific certification objectives for computer-vision algorithms" then discusses a new certification approach (proposed by the *Overarching Properties* working group, and proposes 5 certification objectives dedicated to computer vision.
- Section "Visual odometry" (resp. 5) discusses the certification issues in the specific case of the *VO* function (resp. *SI*).
- Finally, Section "Conclusion and challenges" proposes a list of key scientific challenges to be explored to make the vision-based perception systems certifiable.

Certification practice for civil avionic systems

The two main certification standards that are concerned with civil avionic systems are: first, the ARP-4754A [79], which is a guideline for development processes under certification, with an emphasis on safety issues; and second, the DO-178BC [75], which provides guidance for developing software under certification.

Given that they are central in civil aeronautics, we briefly present these two standards in the two following subsections (ARP-4754A in Subsection "Safety design process", and DO-178C in Subsection "Algorithm and software development process") and we discuss their limitations with respect to computer-vision systems.

Safety design process

ARP-4754A design process

A safety critical development process is the imbrication of a *usual* development process (that ensures the functional correctness) with a safety assessment process (that ensures the safety requirements compliance). Figure 2 provides a schematic overview of the development process for a safety critical system compliant with ARP 4754A.

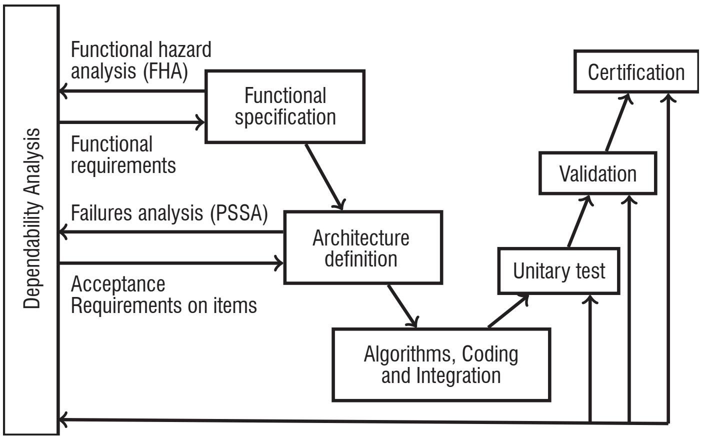


Figure 2 – A compliant ARP-4754A safety design process

The high-level functions define the main functionality expected from the system and are analyzed with regard to the risks that they may encounter through the FHA (Functional Hazard Analysis). For each risk, the experts must identify its causes and evaluate the severity of the consequences in dangerous situations. For instance, if a failure of a function could lead to a crash, it is classified as "catastrophic"; the function will not be lost with a probability lower than 10^{-9}/FH, and nothing less than a triple failure will lead to the loss of the function. If the failure of the function "only" causes serious or fatal injuries among the passengers, or could lead to physical distress of the crew, then the function is classified as "hazardous"; the probability of losing it will be lower than 10^{-7}/FH and only a double failure will lead to such a loss.

After such a risk analysis, the high-level functions are then refined as a *preliminary functional architecture* (second step of the development cycle in Figure 2). Each high-level function is *implemented* as a set of sub-functions providing the expected functionality. This architecture is analyzed through the PSSA (Preliminary System Safety Assessment) to check whether the requirements from the FHA can be fulfilled assuming some properties (such as independence, failure modes and propagation rules). This step is an iterative activity: if the functional architecture does not fit the requirements, the designers must propose a new architecture with additional redundancies.

Once a consolidated architecture has been found (at the end of the second step in Figure 2), the next phase is the selection of the hardware item, the allocation of levels of software criticality (called DAL for Dependability Assurance Levels) to each software function, and the coding of the functions and the platform services (third step). Five Dependability Assurance Levels are defined by the certification standards (by the DO178B), from DAL A (the highest criticality) to DAL E (the lowest criticality), with specific objectives and activities required for each level.

Then, during the ascent of the development cycle, several tests are applied and the SSA (System Safety Assessment) verifies that the hypotheses made in the previous steps are satisfied.

Applying this safety design process generally leads to a high level of safety for conventional avionic systems.

Application and limitations of the standard for perception systems

Now let us consider the *Absolute Localization* system. This system is a less conventional one in the sense that it involves computer vision. If we try to apply the current practices to this system, we obtain the following schematic reasoning:

- *Absolute Localization* is used for autonomous taxi driving. The FHA analysis (step one in Figure 2) concludes that the most risky situation is the failure condition FC = "*the function provides a wrong position without the error being detected*". If a FC occurs, it could lead to collision with other vehicles or with people on the taxiway. The severity of such a situation is classified as *hazardous* because it can cause serious injuries. Thus, the associated safety objectives are:
  - *no double failure should lead to the occurrence of a FC, and*
  - *the probability of occurrence of a FC should be less than* 10^{-7}*/FH.*
- Let us suppose that the chosen architecture (designed during the second step in Figure 2) of the perception system is that shown in Figure 1. *Absolute Localization* relies on *VO*, *SI* and on the *Other Sensors*. According to the contribution of each component to the whole function, the PSSA leads to new refined safety objectives for each component. In case of *VO* and *SI*, let us suppose that these refined safety objectives are:
  - *the probability of occurrence of an undetected erroneous output of VO (resp. SI) must be less than* 10^{-4}*/FH,*
  - *no common failure can lead to an undetected erroneous behavior of VO and SI, and*
  - *the software functions VO and SI must be developed in accordance with the DAL B objectives.*

A cause of an undetected erroneous *VO* behavior could be internal or external "failures" leading to an erroneous estimated covariance of the provided position. Likewise, a cause of an undetected erroneous *SI* output could be internal or external "failures" leading to a high score on false hypotheses.

The issue is then: what are the "failures" that can lead to *VO* or *SI* undetected erroneous outputs.

Algorithm associated hazards

As mentioned in the introduction, *random failures* refer to *hardware failures* and *systematic failures* refer to software bugs. In the domain of computer-vision, it is well admitted that vision algorithms may enter in failure modes even in the absence of those types of failure. For instance, external objects moving together in the same direction can fool the *VO* function. Similarly, an overexposed image can negate the *SI* function. Other internal causes, such as non-deterministic divergence of internal random solvers (usually used to speed up the convergence of the algorithms), could also lead to undetected erroneous outputs.

As a consequence, to apply the aeronautical safety design process (Figure 2) to computer-vision it is necessary to revisit the notion of "failure". Failures must be extended to *algorithm associated hazards*, that is to say, to any *internal or external ambiguous situations* where the algorithm is not able to behave correctly, even if there is no hardware or software failure.

The first difficulty is then to be able to identify, for a given vision-based perception system, all of the possible *algorithm associated hazards*.

Effect and failure modes of algorithm associated hazards

The second difficulty lies in the need to extend the safety analyses to take into account the *effects* of the *algorithm associated hazards*; that is to say, to determine what kind of hazard each algorithm is sensitive to, and what the associated failure modes are. These issues are new to the conventional safety aeronautical process.

Algorithm and software development process

With regard to the software level, the aim of the software assurance process is to provide evidence that the software components behave as expected by their requirements and do nothing else. In the commercial aircraft domain, the software assurance process is based on the certification standard titled "Software Considerations in Airborne Systems and Equipment Certification", known as DO-178C [76].

DO-178C

The DO-178C standard does not prescribe a specific development process, but identifies four mandatory steps:

- Development of High Level Requirements (HLR) from system requirements;
- Development of Low Level Requirements (LLR) and Software Architecture from the HLR requirements;
- Development of the source code;
- Production of an object code executable.

Certification objectives are then associated with each step. The schema depicted in Figure 3 summarizes all of these objectives for Dependability Assurance Levels A and B (the two highest ones). The five main points addressed by the software aeronautical certification standard are the following.

Requirements

The expected behavior of the software must be explicitly and completely defined by high level software requirements (the HLRs). For instance, in the case of a *VO* function, the HLRs are the functional requirement depicted in Table 1 (e.g., "the *VO* estimates the relative position and provides error covariance" and "any erroneous output is detected by a high covariance"). HLRs must then be refined into a software architecture (*i.e.*, the internal architecture of the *VO* function, shown in Figure 4) and low-level requirements (LLRs). In the case of *VO*, the LLRs describe the pseudo-code of each module of the function and the underlying methods (such as RANSAC). Like HLRs, LLRs must be complete and explicit. They must also be verifiable by an identified means.

Traceability and compliance

A second important certification objective is *downward and ascending traceability*. Downward traceability signifies the demonstration that a requirement of a given level is broken down into one or more requirements or software elements of the next level. Conversely, ascending traceability means the demonstration that a low level element corresponds to a requirement of the previous level. Together

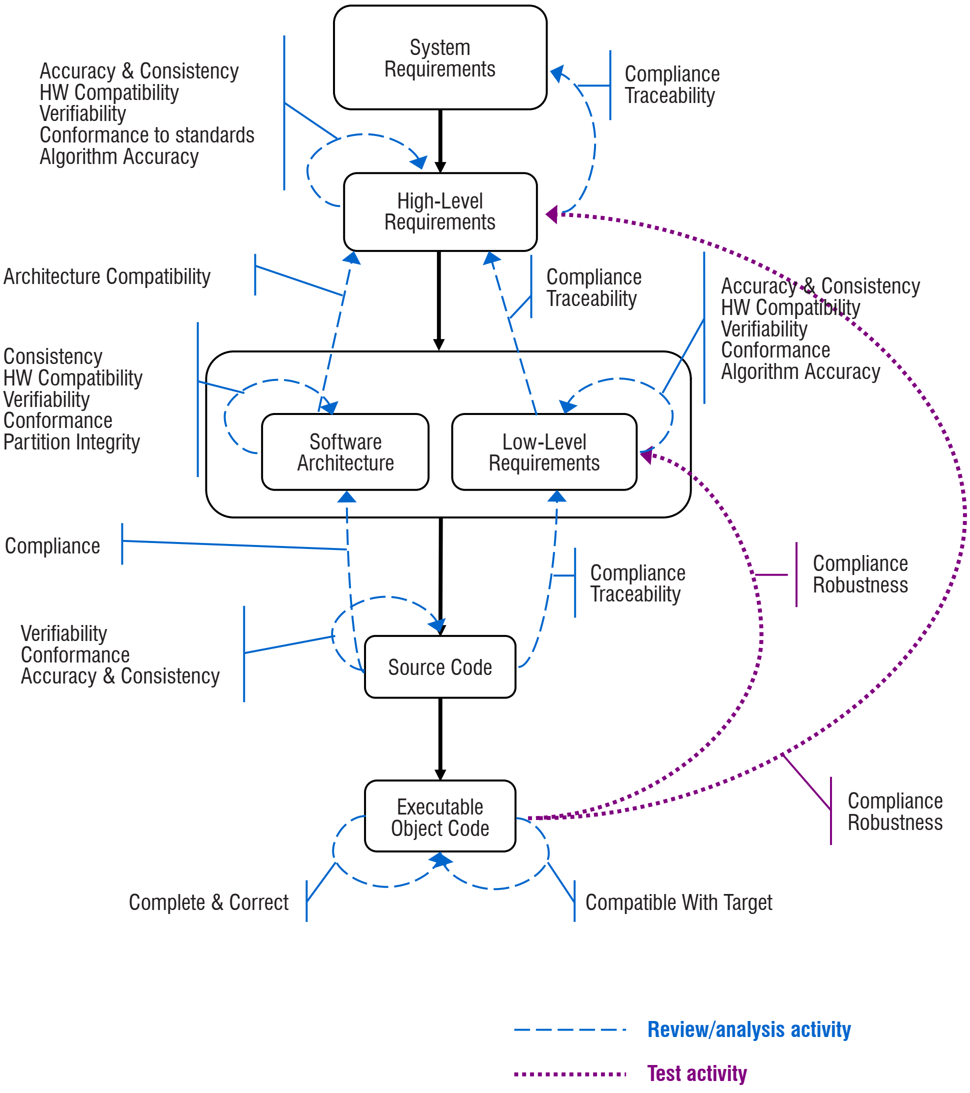


Figure 3 – Certification objectives required by DO-178C [76]

with traceability comes the compliance objective, which is the demonstration that the requirements or solution elements of a given level are correct with respect to the requirements of the previous level. In the aeronautical software certification scheme depicted in Figure 3, traceability and compliance concern more precisely:

- *traceability* between HLRs and system requirements and *compliance* of HLRs with system requirements;
- *traceability* between LLRs and HLRs, and *compliance* of LLRs with HLRs;
- *traceability* between source code and LLRs, and source code *compliance* with LLRs.

In other words, aeronautical certification requires evidence that all requirements are properly addressed, and that the source code does not contain unnecessary lines (*i.e.*, not justified by the requirements).

Coverage

A third strong certification objective is the coverage of all requirements and all of the source code during verification activities: each expected behavior related to a given requirement must be verified and, conversely, each part of the source code must be covered by a verification activity.

Determinism

Determinism of the software is a fourth key point. The expected behaviors must be deterministic in the sense that to each input stimulation must correspond a single response. From a mathematical point of view, the software must implement a *total function*.

Bounded execution time

Finally, an embedded system is by definition immersed in a real environment. It must respond in real time: faced with an external situation, the system must be able to find a suitable answer within a limited time; this time bound must be compatible with the time constraints of the operating conditions under which the system is used.

Application of all of these certification objectives to conventional civil avionic software, such as a flight control software, provides the high level confidence required for the most critical systems (classified as catastrophic or hazardous).

Application and limitations of the standard for perception systems

Unfortunately, as shown in [7], the certification objectives for civil avionic software outlined above pose several difficulties when trying to apply it to computer-vision algorithms. In particular, only *requirements* and *bounded execution time* objectives can be achieved. The three others must be dealt with in a different manner.

Ambiguity (opposite to compliance)

First, ambiguity is inherent to the real world that the system has to perceive. For instance, even for human actors it is difficult to interpret without ambiguity a real scene in an airport under difficult weather conditions. In other terms, it could happen that the questions "What do I see?", "Do I see any the object on the landing track?", "Am I matching the right features from two consecutive input images?" do not have unambiguous answers. Thus, the possibility of ambiguous situations make the algorithm difficult to test and validate. Therefore, it is difficult (if not impossible) to prove compliance between each level of the development process (*i.e.*, between LLR and HLR, and between the source code and LLR).

Algorithm associated hazards (opposite to compliance)

Second, as mentioned in Section "Safety design process", vision algorithms may return erroneous outputs even in the absence of hardware failures or software bugs. This is what we called "algorithm associated hazards". When such a hazard occurs, the system (*i.e.*, the source code) does not fulfill the intended behavior, leading to a loss of compliance between the source code and the requirements (the LLR and the HLR).

Indeterminism

Third, several vision algorithms explicitly use random methods. This is the case, for instance, for algorithms that use optimization techniques to extract features from a frame. The advantage of using randomness is to improve the convergence of the algorithms (and thereby reduce their computation time). However, it may lead to unrepeatable executions. Such unpredictability is a strong limitation for current certification objectives.

Coverage and traceability

Finally, as mentioned above, certification requires that the software implementation be completely covered in order to guarantee that each part of the code corresponds to an identified requirement. However, for perception systems implemented by neural networks (for instance aimed at detecting the landing track and detecting other aircraft in the airport) it is difficult to explicitly determine which part of the neural network is responsible for the track detection and which part is responsible for aircraft detection. This leads to a lack of traceability between requirements and source code.

To summarize, compliance, determinism and coverage requirements are difficult (if not impossible) to meet for computer-vision software. Therefore, we believe that the current certification standard for civil avionic software cannot apply to vision-based perception systems.

Computer vision based system development process: a data driven design logic

Another novelty, and issue, when developing computer vision based systems is the way in which they are calibrated and validated. For both VO and SI, the algorithms are designed and configured using large data sets. Identifying difficult cases is a key ingredient for building data-sets that can evaluate safety issues. Most of the available benchmarks, however, do not address the explicit definition of hazards, but rather favor the diversity of sources. There are at least two reasons for this situation. First, data acquisition or collecting is usually opportunistic, and is not able to fully control their content. Second, most such data-sets are aimed at ranking new algorithms in terms of some easy-to-compute discriminant performance index. The variety of data sources seeks to challenge the algorithms rather than to explore some predefined operational domain.

This issue has been investigated in [102] by means of the HAZOP method originating from the chemical process control industry, and codified since then in the IEC 61882 standard (IEC 61882:2001). As FHA, HAZOP applies to some systems operating within its environment, but is interface-oriented, given that hazards are formulated in terms of deviations of the input/output of the system with respect to their nominal values.

[102] applies the HAZOP method to computer vision (CV) expert knowledge and design CV-HAZOP, a checklist of more than 900 hazards that could affect generic computer vision functions. They provide a guideline for evaluating existing data-sets or design new ones with respect to their coverage of hazards, and apply it to in-depth estimation by stereovision. The authors have made the CV-HAZOP checklist freely available and intend to integrate contributions from the community to extend it collaboratively in the future. While such an analysis of CV certainly pertains to vision systems, it does not fully describe the dynamical and environmental aspects of a vision process that could be embedded within an autonomous system. It would be interesting to update the CV-HAZOP checklist in this direction. [23] considers an autonomous system operating in open unconstrained environments in which interactions may occur outside the intended mission scenarios. The authors propose Environmental Survey Hazard Analysis (ESHA) as a way to exhaustively account for such non-mission interactions.

All conclusions of the previous subsections meet a more general observation made by authors of [17], who claim that "the current standards may be inappropriate for very complex systems developed now and in the future".

Developing specific certification objectives for computer-vision algorithms

Given that current standards do not offer a proper way to deal with artificial intelligence, new approaches and methodologies have to be developed. To face these difficulties and to anticipate the development of embedded vision-based systems, recent works regarding certification have been proposed. The reader can refer to [7] for a detailed study on certification challenges for adaptive systems. The authors explore new solutions to improve trust in the behavior of such systems and to facilitate certification. Among these solutions, they recommend that new certification processes be studied and, in particular, the OPs (Overarching Properties), which are a very promising methodology from which we derive five high level objectives.

Development of new certification approaches

The starting observation is that new techniques and technologies are rapidly developed and are "vital for the modernization of avionic systems, so finding an [certification] approach that is more amenable to new technology trends and capabilities is crucial" [17]. Alternative approaches have been explored in [17, 45] by a consortium composed of the European (EASA) and American (FAA) certification authorities, an industrial panel, and two aerospace research institutes (NASA and ONERA). The new certification framework that they propose is based on three "*Overarching Properties*" (OPs for short) that are fundamental characteristics of the system being certified and of any sub-element of it.

The three *Overarching Properties* are:

- **Intent**: the intended behavior (*i.e.*, the requirements of the system) must be explicitly defined, and it must be correct and complete with respect to the desired behavior (*i.e.*, what the system is supposed to do from an external point of view). The first *Overarching Property* also requires that the usage domain of the system (called "foreseeable operating conditions" in [45]) be clearly defined.
- **Correctness**: the implementation of the system, that is, its architecture (composed of a hardware item, algorithms, source code, etc.) is shown to be correct with respect to the defined intended behavior in the defined usage domain.
- and, finally, **Innocuity**: the system may contain some parts that are not required by the intended behavior (for instance, because the implementation uses a previously developed item that offers more services than required for the specific usage of the system). In that case, the traceability requirement between the intended behavior and each part of the implementation is broken. This is not a problem anymore. However, it must be shown that these extra parts (e.g., extra lines of code, extra services, etc.) have no unacceptable safety impact on the system.

The framework defined by these three high-level *Overarching Properties* no longer requires *determinism*, *traceability* and *coverage*, three of the four major difficulties related to the DO-178C standard identified in the previous subsection. It focuses on more fundamental objectives. Therefore, we believe that the *Overarching Properties* are the appropriate certification framework for computer-vision.

Five high-level objectives

From now on, we consider the framework defined by the *Overarching Properties*. Therefore, new questions arise: Is it possible to refine the three *Overarching Properties* into certification objectives that are specialized for computer-vision? And, if so, what are these specialized objectives? Let us enumerate the remaining high-level tough certification objectives (remember that *determinism*, *traceability* and *coverage* are no longer explicitly required):

- First, an applicant wishing to certify a computer-vision device shall define the usage domain and the intended behavior under this usage domain (first *Overarching Property*).
- Second, the applicant shall identify all algorithm associated hazards and their effect on the system (see discussion at the end of Subsection "Application and limitations of the standard for perception systems").
- Third, the applicant shall show the correctness of the implementation (second *Overarching Property*).
- Last, the applicant shall show that no unnecessary part has an unacceptable safety effect on the system (third *Overarching Property*).

The two first points are related to a modeling problem: What kind of models are required and how can we be sure that the models are complete? The two last points require argumentation: How can correctness and innocuity be shown, and what kind of evidence is required for that purpose?

We propose to segment the potential activities that could contribute to the certification of vision-based perception systems into five families:

1. **Complete description of the intended behavior and of the usage domain.** To explicitly enumerate all of the possible situations and to define what the system should "see" is a challenging task for most perception systems, even in the restricted area of an airport. For systems based on learning techniques, both the intended behavior and the usage domain are defined by data-sets (the test base). The challenge in this case is to show that this data-set correctly samples the real world and that the sampling is tight enough not to miss significant situations.
2. **Safety hazard identification.** The second activity to be carried out is twofold: to list the possible hazards, the difficulty here arises from the fact that some of the hazards are related to the internal weaknesses of the algorithms; and to define *good* benchmarks, that is, benchmarks that contain all of the identified hazards.
3. **Run-time safety.** As stated at the end of Subsection "Application and limitations of the standard for perception systems", computer-vision algorithms can be fooled in some situations (the situations that we called *hazards*). The question in this case is "How can the algorithm be prevented from generating hazardous or unexpected behaviors?", which can be answered by developing specific functions used to detect bad operation and mitigation means. The third activity to be carried out is then to be able to define and develop appropriate detection functions and mitigation means that address all of the possible hazards.
4. **Requirement satisfaction assessment.** The goal of the fourth activity is to answer the question "How can we ensure that the instantiated algorithm actually implements the target function and does nothing else unacceptable from a safety point of view?", and develop means of validating & verifying that requirements are satisfied. Some sub-requirements can also be considered, such as: *stability* (*i.e.*, Is the algorithm stable to small changes in the environment, for instance, is there any adversarial image that the system is sensitive to?); *convergence* (*i.e.*, If the algorithm contains an internal loop, how can we ensure that this loop converges in bounded time?).
5. **Certification assessment methods.** The question to be answered is "How can we demonstrate to users and authorities that the algorithm is doing the right thing?" and propose methods/tools able to either show that the algorithm actually performs well on the current data, or that the process has been correctly designed. In other terms, the keyword here is "explainability": how to make the computer-vision algorithm explainable in order to convince both the user and the certification activity.

Note that this fifth activity does not stem from the current certification standards nor from the *Overarching Properties*. Explainability is not a usual objective in certification. However, we believe that when faced with complex systems in complex situations, it could be safer to reassure the user by giving him, if required, some explanation about the behavior of the system. Misinterpretation can cause inappropriate actions by the user. Therefore, making the perception more explainable makes it safer.

In the following, we discuss these five activities and the related state-of-the-art in the case of *VO* (Section "Visual odometry") and *SI* (Section "Vision-based scene interpretation").

Visual odometry

Overview

Visual odometry denotes the estimation of the ego-motion of a vision system from the sequence of images that it provides. VO belongs to the field of artificial vision because it is essentially the implementation on an on-board computer of a sense common to many animals. It has been the subject of numerous research studies since the 80s. Nowadays, the subject is considered as mature, since complete formalizations were proposed in the mid-2000s and a number of hardware/software realizations have been released since then. Among these realizations, we focus on eVO for "efficient Visual Odometer" [81], a stereovision-based odometer proposed at ONERA in 2013, [81]. eVO is indeed paradigmatic of several works on visual odometry and, moreover, it has been used in many robotic experiments conducted at ONERA, demonstrating its practical interest for autonomous systems.

Note that monocular odometers could have been considered (which make use of one camera only); however, stereovision systems offer a conceptually simple way to get 3D information from the world, which greatly facilitates the navigation task and also the qualification of the result. In other words, stereovision leads to a simpler topic for the present study about safety and certification.

Architecture

The general principle of VO is to locate a camera with respect to a known 3D map of the environment. eVO uses a stereorig: a set of two cameras rigidly assembled and separated by a known distance called the baseline. Stereorigs can be mounted on small autonomous platforms, for instance UAV, as shown in Fig. 5. They allow the system to construct at each instant a map of the visible environment.

A simplified version of eVO's architecture is presented in Fig. 4. In the initial step, a map is constructed by stereovision: some image features are extracted in the left frame, matched in the right one and associated with a 3D position by triangulation (green boxes in Fig. 4). The association of a 3D position and an image feature is called a landmark. The map is a cloud of *landmarks*.

When the system moves on, features are tracked in the left frame. Their apparent motions in the image are solely due to the ego-motion of the system, since they are supposed to be associated to fixed landmarks. In the process, some of the landmarks may leave the camera field of view. However, if a sufficient number of the landmarks are still visible, the pose (position and orientation) of the current left camera can be computed by comparing the 3D positions of landmarks and their current localization in the image plane. VO must also provide an estimation of the covariance of the error on the outputted pose. Such a characterization of the estimation is required to update the state of the system and fuse visual information with that from other navigation sensors (GPS, IMU, wheel odometers). All of these operations, which are represented by blue boxes in Fig. 4, run nowadays at a framerate (*i.e.*, 20Hz) even on small PCs embedded on UAVs such as the one shown in Fig. 5.

To summarize, eVO is the combination of two processes: pose computation running at 20 Hz and map building invoked at each keyframe, typically every 1 second. The following section details the operations involved from the perspective of certification.

Hazards associated with eVO

As mentioned in Section "Safety design process", hazard identification is a strong issue for safety analysis and then for certification. Hazards can come from algorithm weaknesses. We called such hazards "algorithm associated hazards". In the case of VO (Figure 4), three groups of modules are the source of such hazards: (1) track features, extract features, and stereo matching; (2) triangulation; and (3) compute pose and covariance.

Hazards associated with feature extraction, tracking and stereo-matching

These operations act on image pixels and, as such, they are both costly and critically dependent on image quality. "Good features" are a group of pixels that can be extracted unambiguously and tracked or matched with high accuracy [86]; for instance, corners appearing in a man-made environment [40]. While several recent proposals have been made to improve this step by using more robust features [6, 74, 77], or by using strategies to improve their dispersion within the image field of view, failure cases are still encountered, with several causes:

- Scene. Feature processing requires that the image contain localized and highly contrasted unambiguous details. Homogeneous or pseudo-periodic scenes can be found, for example,

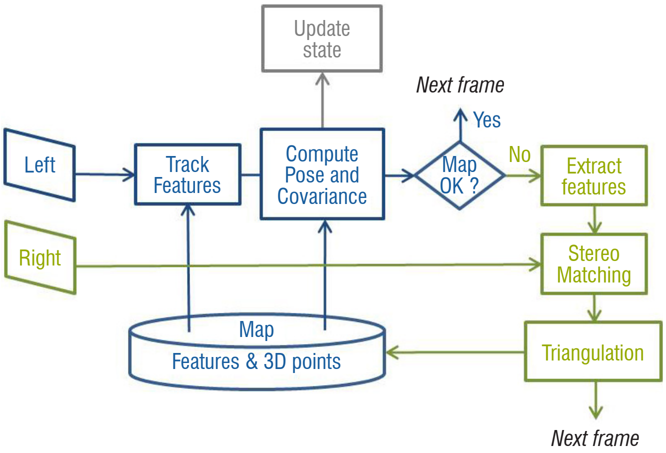


Figure 4 – Architecture of eVO. Blue boxes pertain to the estimation of the pose at each recorded left frame; green boxes concern stereo-reconstruction, and are activated when the map is to be updated (keyframe). The final output of eVO is an update of the state of the system including an increment of the trajectory and a new posterior covariance

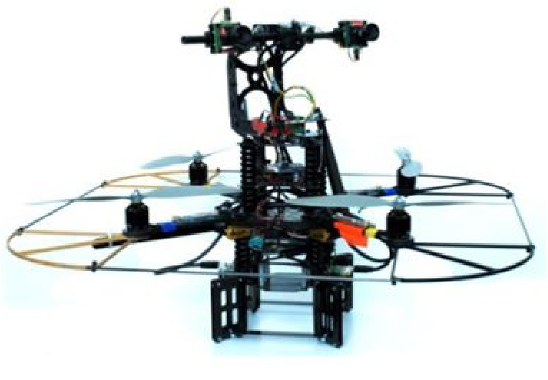


Figure 5 – An UAV equipped with a stereorig

in indoor environments. In addition, contrasts should be stable when the observer's point of view changes, which is not the case for reflective or semi-transparent scenes such as mirrors, glass or water surfaces.

- Illumination. Low illumination decreases contrast, focused light sources lead to unstable contrasts between illuminated areas and shadows, etc.
- Propagation conditions. Smoke, haze, rain or snow degrade useful and stable contrasts in the recorded images.
- Camera settings. Aperture, shutter, and gain are camera parameters that tune the range, contrast, noise and defocus, and motion blur affecting the image. When the system moves, going for instance from a bright to a dark area, they should be adapted in real time.
- Observer dynamics. When camera movements are too fast, the image quality becomes degraded due to motion blur. Reducing the shutter time is an option, but it also leads to an increase in the image noise. Moreover, fast and large rotations drastically modify the field of view and a large number of landmarks can be lost.

Some of these conditions can be detected by testing the input image (low contrast, high noise, etc.), but usually it is done on the output. Indeed, computing a quality score usually involves some costly filtering of the whole image field and most often a degraded quality will lead to an abnormally low number of extracted or tracked/matched features. Most efficient VO codes monitor at all times the number, and sometimes the spatial distribution, of extracted or tracked features. Observer dynamics can also be predicted thanks to accelerometers and gyroscopes included in modern inertial measurement units (IMU). We will return later to fusing image information with IMU or "visio-inertial" navigation, which has undergone a major evolution recently.

Hazards associated with triangulation

Triangulation theoretically amounts to locating the intersection of two 3D rays in 3D space. In practice, matching and calibration inaccuracies imply that the two rays do not cross. Only an approximate point can be found by means of a non-linear least-squares fit. Such inaccuracies can be considered as a source of "hazards" for navigation functions, for instance in the case of high-speed vehicles moving in scenes with highly varying depths with respect to the observer. To face this problem, solutions are explored in [54].

Hazards associated with pose and covariance computation

Pose computation is also a non-linear least-squares process calling for an iterative optimization. Initialization usually stems from an approximate linearized system, and it is important to ensure that it is not too far from the true pose [41]. The estimated pose must be accompanied by an estimation of the error, generally in the form of a covariance matrix. Two situations may lead to a detectable failure of the process.

First, the 3D map can be in a particular configuration leading to a degeneracy of pose computation, *i.e.*, the uniqueness and stability of the solution is no longer guaranteed. An example is the case of a planar surface. However, in many cases, there are tests to select the right solution, or stable solutions can be obtained from alternative estimation strategies, especially in the planar case.

Second, a least-squares estimation is highly sensitive to outlier data, which are unavoidable in video processing. A first answer is to use robust loss functions to mitigate the influence of outliers [104]. Robust estimators (also called M-estimators) lead to iterative optimization, but usually the extra computational load is limited because most vision problems are non-linear and already require an iterative linearization process. However, M estimation cannot cope with a high proportion of outliers, which is a situation that is common in practice. In such a case, a popular approach is RANSAC (Random Sample Consensus) [29]. RANSAC iterates on pseudo random samples of data points to form putative estimates, which are then tested on the whole dataset. Given that it is very efficient for outlier removal, RANSAC is widely used in embedded vision despite its non-deterministic nature.

However, pose estimation can also lead to inconsistency, a situation where the effective error is higher than the predicted error according to the estimated covariance. There are mainly two sources of inconsistency. The simple one is the case of a confusing scene leading to a high confidence in a wrongly estimated motion, like when someone sees the train on the track next to his own start and feels like he is moving in the opposite direction. Such situations are mostly momentary, but can destabilize the system. More tricky are the structural inconsistencies related to the non-linearity of pose or motion estimation from images. In such situations, the error can increase continuously while the estimated covariance remains low.

Difficulties regarding certification

Requirement satisfaction

If VO is part of a safety-critical function, demonstrating that the requirements associated with VO are met is a central issue for certification.

Formal approaches

VO is a particular instance of statistical estimation, where a quantity of interest, the state of the system, is involved in a criterion depending on some data (e.g., image features) and whose functional form derives from a statistical modeling of the various components (sensor noise, prior distribution on variables) and their relationships. Optimization of this criterion leads to the optimal estimate of the state given the data, with the (implicit) relationship between data and estimated state being referred to as the estimator. Modeling efforts allow the properties of the estimator to be theoretically characterized. Some properties concern the discrepancy between the estimated state and the true one, such as bias (e.g., systematic error) and variance (statistical dispersion). Bias and variance are usually associated with the performance of the estimation. They are, themselves, characterized by another level of properties, called structural properties. Efficiency refers to the optimality of bias and variance for the problem at hand; *i.e.*, that no other estimator can achieve lower values. Consistency expresses the fact that they correctly characterize the performance; that is to say, that the true state indeed lies within the interval of values defined by bias and variance. It clearly pertains to the safety of vision-based navigation: with a consistent estimator it is, for instance, possible to guarantee that the plane remains within some known bounds around the requested trajectory. Unfortunately, consistency is very difficult to assess for vision-based odometry or SLAM estimators. This is due to the non-linearity of the relationship between image data and state parameters. Also, as already mentioned, vision is prone to outliers, which are not accounted for in the problem modeling and lead to inconsistency. Hence, consistency is not a definitive answer to VO/SLAM safety issues, yet the vast literature on the subject includes relevant works; for instance, regarding consistency check techniques, which can be used as a run-time safety process [36].

Benchmarks

The subject of datasets and benchmarks has already been discussed in Sec. "Computer vision based system development process: a data driven design logic". It may be interesting though, to emphasize that benchmarking a VO algorithm is a difficult task. Acquiring real data on several trajectories with ground truth is a heavy and complicated burden, which, in practice, cannot be done for all operating conditions. It is even more difficult when accounting for the fact that this process is supposed to be embedded in a robotic platform, with available IMU and low computing power. However, the design of benchmarks is still considered as useful, at least for assessing algorithm performances: as an example, [21] recently proposed a benchmark relating to visual-inertial navigation for UAV, made publicy available as a "Euroc" dataset [11].

Like several other vision tasks, VO is the subject of open access benchmarks, the most popular being the Kitti dataset oriented towards autonomous driving[1]. The release of Kitti was originally motivated by extending the operational field of CV methods to real-life sequences of autonomous driving [30]. However, although this dataset has certainly contributed to improving the performance of recent CV algorithms, in particular in the field of urban visual navigation (and more precisely to navigation within a Western midsize town), it cannot be considered as a way to assess that an algorithm will behave correctly in other scenarios and environments, or even within the environment of the recordings. For instance, this dataset does not proceed from a systematic exploration of hazards.

Run-time safety

Detecting at run-time hazards and errors that can have a safety impact on the behavior of the system is required for certification.

Run-time safety check tests can be done at three levels: input (e.g., checking image quality), internal variables (e.g., number of tracked point features [81], [73]), and output (cross-validation with another sensor such as IMU, magnetometers array [14], etc.). It also includes tests about the status of internal operations, such as the monitoring of optimization processes and consistency checking [36]. Like for the problem mentioned in Sec. "Computer vision based system development process: a data driven design logic", of ensuring a complete coverage of hazards by a given database, an issue here is to guarantee a complete coverage of failures encountered at run-time. In this same way, [64, 47] formalize safety tests for hazard detection related to vision in a domain-specific language ViSaL (Vision Safety Language). ViSal allows the automatic generation of efficient code and opens the way to guaranteed safety check tests.

Vision-based scene interpretation

Overview

Scene interpretation is an expression that stands for a collection of functions, such as object detection and classification, semantic segmentation or object tracking, that take an image frame, or a video, as input data and produce a symbolic representation of its visual content, usually associating geometry (the *where* part) and semantics (the *what* part), and often qualified by a score (see Fig. 6). These functions have been addressed since the beginning of artificial intelligence and computer vision, with various paradigms. The modern approach, which is likely to last since it has demonstrated its capacity to equal or even surpass human performance in some contexts, involves a *machine learning* step able to specialize a complex parametric function to a *dataset* expected to be representative of the operating domain.

Description (white box)

Most of the current SI functions make use of *Convolutional Deep Networks*; *i.e.*, neural networks chaining a rather large number of layers with local two-dimensional filters. A typical example of such networks is depicted in Fig. 7.

The current trend of algorithm design is to integrate all of the necessary computations to complete the function in a common unifying deep network framework, making the learning step globally influence the whole chain, in a so-called *end-to-end* fashion. The resulting global function is therefore heavily dependent on the learning dataset that empirically specifies the function.

Difficulties regarding certification

The certification of software implementing a deep network should not modify current practices. Their architectures are homogeneous, exploit a small functional vocabulary (convolutional or fully connected layers, non-linear activation functions, pooling), make use of software development frameworks (Tensorflow, PyTorch, etc.) and specific libraries able to implement the network on a Graphical Processing Unit (GPU).

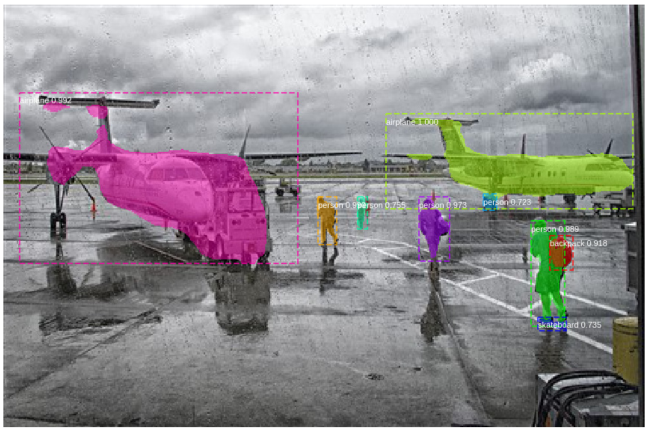


Figure 6 – Typical output of an SI algorithm, detecting objects of interest and their outlines. Obtained using Mask-RCNN [42]

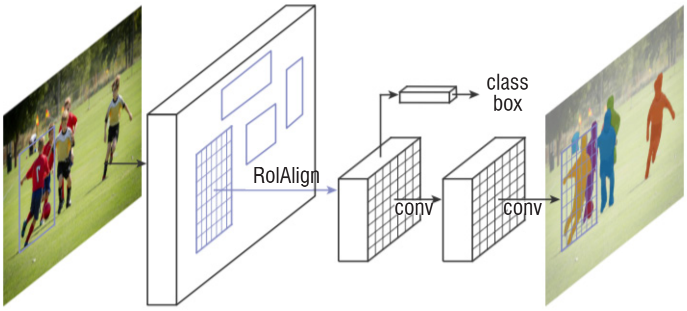


Figure 7 – A modern deep netw ork architecture for object detection [42]

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

However, the question of certifying data-driven AI algorithms – not software – is another matter. It appeared rather recently to be a very hot topic, since this kind of technology is expected to invade our life at rather short notice[2].

One can summarize the problems brought by data-driven algorithm design in the following way:

- Specification by examples: the high dimensions of input and output spaces involved in perception functions make their formal and complete specification almost impossible. The usual way to describe the target function is to provide a distribution of good operating samples; *i.e.*, a dataset of N input/output samples $\{X_i, Y_i\}_{i=1}^{N}$. This distribution is expected to describe both the input operating domain (what the possible input X_i are) and the desired output predictions (the system should produce Y_i when fed with X_i). This way of specifying the perception function thus assigns a central role to the quality and representativeness of the dataset.
- Probabilistic requirements: the approach of specification by examples has a direct consequence on the way functional requirements are described: they depend on some uncertainty representation that expresses the lack of knowledge about the exact operating domain at the time of the algorithm design. Classical ways to define requirements are performance metric objectives, such as precision/recall for detection or retrieval, classification accuracy for recognition, mean square error for localization, etc. Many usual metrics are presented as trade-offs between several measures. This requirement description approach leads to several issues: 1) How do they relate to the full system hazard analysis?, 2) How are the acceptable trade-offs defined?
- Validation by testing: one consequence of specification by examples is that validation also becomes data-driven. The question is to design fair evaluation protocols and metrics able to predict and estimate whether the requirements are satisfied. When dealing with machine learning, a key aspect is to find ways to compute unbiased estimates of the metrics and their variance, for instance by using cross validation.
- Robustness: deep networks are complex functional structures that are prone to instability or hyper sensibility that can be discovered, for instance, by adversarial optimization techniques. The question related to certification is to either assess a sufficient level of robustness, *i.e.*, invariance to perturbation, or to detect potential instabilities or "attacks".
- Operational domain assessment: Defining an operational domain through a dataset is inherently problematic and can be broken down into two issues: 1) How do we state whether a given input data will be correctly processed? 2) How do we describe the set of correctly processed input data – the operational domain itself?
- Usability of formal verification: This has been a central approach in the aerospace domain, and is suitable when the problem can be expressed as a series of formal properties that have to be jointly satisfied, making possible the application of generic solvers for verification. Several studies have proposed the adaptation of this paradigm to neural network architectures, usually for low-dimensional problems, but the question of their generic usability for perception in very large dimensional spaces remains.
- Intelligibility of the predictive process: DNNs follow a series of complex nonlinear transformations of an input space, with a role for each step that is hard to assign clearly. The overall process is considered opaque. This makes the justification of both bad and good behaviors difficult, which is an obstacle to convincing of hazard-free functioning.
- Evolutivity and operational domain extension: the use of machine learning techniques implies that the algorithm operational domain is restricted to what the dataset samples. Making the system evolve to a different context with new requirements will require a new learning phase, often with no non-regression guarantee.

State of the art

Several solutions to the above problems have been proposed, but mostly remain in the academic domain. We will follow the categorization described in "Developing specific certification objectives for computer-vision algorithms", with an emphasis on the last three to briefly give an idea of the current state of the art.

Run-time safety

Anomaly or novelty detection

A *safe* system should be able to warn its user when there is a risk of catastrophic consequences when exploiting its prediction; *i.e.*, when it may be false, suggesting that it be *rejected*. In a prediction system, there are mainly two causes of rejection: uncertainty – the input data can be associated to more than one prediction – or novelty – the input data has not been considered during the design phase or is abnormal with respect to the underlying models exploited for the prediction.

An uncertainty measure is a way to score prediction quality, and can be used either in combination with other scored predictions in a fusion step to improve the overall result, dynamically when using sequential filters for instance, or statically when exploiting redundancy. Another common usage of an uncertainty score is to consider it as a rejection indicator of abnormal behavior. We focus in this section on this last case.

Novelty, anomaly or outlier detection are synonyms of the same formal problem: deciding whether a given item of data belongs to an underlying distribution, usually described as samples, or as a set of characteristic prototypes. It does not address the question of designing a system that is robust to anomaly or outliers, but is aimed at equipping a predictor with an explicit rejection capacity or out-of-distribution detector. In machine learning, this problem is also known as the "one-class classification". The expression "anomaly detection" sometimes refers to a way of building "saliency" detectors [10] – an anomaly being a pattern considered different from most of the others – but is not used for rejection purposes.

Novelty detection is not a new problem, and is used in many applications, for instance in data stream analysis to detect intrusions (see [16, 61, 105, 72, 3] for various surveys). However, when data is highly dimensional, like images, applying generic methods is not powerful enough and depends on a projection over a much lower dimension feature space; e.g., Principal Component Analysis (PCA), auto-encoders or non-linear kernels, to make statistically relevant

[2] "A series of strategic themes [...] has to do with ethics, and the validation and certification of AI technologies, the aim being confidence by all stakeholders in their results: from validation in terms of theoretical proof to explicability, transparency, causality and fairness." p. 65 of Villani’s Report (https://www.aiforhumanity.fr/pdfs/MissionVillani_Report_ENG-VF.pdf)

inferences. [105] discusses the issue of high dimension and its relation to the dimensionality curse phenomenon.

Deep learning has been introduced in classical frameworks to better encode the data manifold, either for generic tasks (classification) or to specifically improve anomaly detection.

[15] describes a robust auto encoder that learns a nonlinear subspace that captures the majority of data points, while allowing some data to have arbitrary corruption, and evaluates their approach on three image datasets. [103] investigate two decision criteria (energy score and reconstruction error) for performing anomaly detection from an energy-based distribution representation computed on a deep network architecture. [26] presents a hybrid model where an unsupervised deep belief network (DBN) is trained to extract generic underlying features, and a one-class SVM is trained from the features learned by the DBN. [78] extends a one-class support vector approach to deep networks, using the same concept of a minimum volume hypersphere boundary.

Another series of works exploits or modifies the output scores before decision, and uses them to detect out-of-distribution data coming from datasets that contain classes different from those found in the in-distribution.

[43] shows the performance of a baseline approach on several datasets, relying on the idea that correctly classified examples tend to have greater maximum softmax probabilities than erroneously classified and out-of-distribution examples, allowing their detection. [53] describes a method improving the detectability of out-of-distribution from the output scores by adding a small perturbation to the input and output temperature scaling. [22] proposes a method that learns a confidence score jointly with the actual prediction by retraining the last layer of a classification network, and uses it on the task of out-of-distribution detection. [60] also learns a confidence coefficient from the inner layers of a classification network and prediction, but with another loss measuring pairwise distance between different classes. [50] exploits a hierarchical class structure to detect data coming from new classes using confidence-calibrated classifiers, data relabeling, and a leave-one-out strategy for modeling novel classes under the hierarchical taxonomy.

As a binary decision problem, the evaluation of novelty detection algorithms depends on measures of false positive/false negative tradeoffs (AUC under ROC curve, Precision at given Recall). Most evaluation frameworks exploit data acquired from "real" situations, e.g., by labelling several classes as outliers, or importing other datasets of similar origin and labelling them as novel (Cifar-10 vs. Imagenet). Algorithms are believed to be fairly compared under such settings. [12, 92] discusses the suitability of available benchmarks (datasets and metrics) and compares several algorithms using such metrics. Their evaluation, however, is limited to low-dimensional data, and whether their conclusion scales to higher-dimensional perceptual data is open.

However, using such evaluation approaches it is difficult to tell whether the state of the art of novelty detection algorithms is usable to assess on-line safety of data-driven perceptual algorithms.

Detecting adversarial examples

The discovery of adversarial examples has motivated the development of defense techniques able to counter or at least detect possible attacks. However, "few strong countermeasures exist for the many attacks that have been demonstrated" [34]. This can be a clear issue for APES safety, the fear being that attackers may purposely design malicious examples to fool the system.

There are have been mainly three different ways to address defense against adversarial attacks:

- *Modified training or input data:* Changes in training data for learning or inputs during testing. [20] detects adversarial examples by testing the validity of Neural Fingerprints, a set of fixed perturbations that are expected to have a controlled behavior when added to real data and not when added to an adversarial example. [96] studies a technique that augments training data with perturbations transferred from other models.
- *Modified networks or learning:* Modifying networks, e.g., by adding more layers/sub-networks, changing loss/activation functions, etc. For instance, [18] control the Lipschitz constant of each layer through regularization. [70, 69] exploit the notion of *distillation*, *i.e.*, the extraction of class probability vectors produced by a first model to train a second one of reduced dimensionality without loss of accuracy, to generate more regularized deep networks. [88] augments model parameter updates with worst-case perturbations of training data in a Wasserstein ball. [59] studies the adversarial robustness of neural networks through a robust optimization perspective.
- *Augmented networks:* Using external models as network add-ons when classifying unseen examples. [101] uses feature squeezed (pixel encoding depth reduction and spatial smoothing) data to compare predictions from the original and the squeezed images. If a large difference is found, the image is considered to be an adversarial example.

The objective of these is to enable the system to be robust to adversarial attacks, or simply raise an alert to initiate further mitigation means.

A rather large number of recent studies on adversarial example detectors exploit the same intuition that they are far from being a manifold of clean data and can be identified by an out-of-distribution method in a given subspace spanned by inner activation layers of a deep neural network. [49] exploits a convex outer approximation of the set of activations reachable through a norm-bounded perturbation for learning and testing. [62] detects adversarial examples by projecting the data to the learned manifold of clean images. [28] uses kernel density estimates and Bayesian uncertainty through a drop-out to detect out-of-distribution adversarial data. [55] learns a Radial Basis Function SVM to detect out-of-distribution data from the last stages of a deep network, where adversarial examples are expected to have the most different behavior. [52] defines a cascade classifier from convolutional filter outputs of various layers in a deep network to detect adversarial data. [58] uses local intrinsic dimension estimation of adversarial regions and applies it to the detection of adversarial examples. [51] proposes a method for detecting any abnormal samples based on computing the Mahalanobis distance between class conditional Gaussian distributions with respect to (low- and upper-level) features of the deep models obtained through Gaussian discriminant analysis.

The high interest of the research community has fostered several challenges in designing defense methods against adversarial attacks: for instance, NIPS 2017: Defense Against Adversarial Challenge Attack[3]

[3] https://www.kaggle.com/c/nips-2017-defense-against-adversarial-attack

and NIPS 2018 Adversarial Vision Challenge[4]. Benchmarks in these competitions are usually of medium size (number of samples and data dimension): cifar-10, MNIST, Tiny ImageNet, Traffic sign[5]. Those challenges often come with adversarial example generation toolboxes such as the adversarial robustness toolbox[6] [68] or CleverHans library[7] as baselines. A thorough benchmarking action is proposed in [91] with the objective of examining the existence of empirical trade-offs between robustness and accuracy using multiple robustness metrics, including distortion, success rate and transferability of adversarial examples[8]. Their conclusion is that low error networks are highly vulnerable to adversarial attacks and that network architecture has a larger impact on robustness than model size.

As regularly mentioned in papers, several attacks fool most of the currently proposed defenses, but are also increasingly detected. As is asked in [34], "Can we expect an arms race with attackers and defenders repeatedly seizing the upper hand in turn?", as is for instance instantiated in the NIPS 2018 Adversarial Vision Challenge.

Instead of playing such an endless game, a critical question regarding safety of APES would be to know whether defenses can be universal, and in what sense. If universality is not attainable, a secondary problem would be to state what kind of attack, *i.e.*, hazard, can be reliably defeated. Adversarial machine learning is in its infancy – many phenomena encountered in deep learning are not well understood – and is still not able to clearly define its operating domain.

Requirement satisfaction, coverage and robustness

The studies can be divided into three groups: evaluation benchmark design, adversarial example attack design, and formal verification.

Evaluation benchmark design

Many datasets are now available thanks to the availability of modern sensors and storing capacities. The CVonline site[9] maintains a rather up-to-date list of current sets, showing the variety of data and annotations that have been gathered.

Several specific domains have gathered a large amount of data, especially to be used as learning databases. This is the case, for instance, for data targeting autonomous vehicles (Berkeley Deep Drive[10], Cityscape[11], Kitti[12], etc.), or remote sensing[13].

One possibility to overcome the lack of data instantiating hazards is to simulate data[14]. Computer graphics simulation has been used for a long time in robotics, for instance, using modern game engines [85, 67]: data realism is achievable with such generators (see for instance [8] for scene synthesis for research on ADAS), but essentially depends on the models fed to the simulator. They are in practice very costly to create, and what is often exploited by those simulations is more the controlled diversity of situations than the realism of sense data.

More recently, style transfer techniques have been applied to enhance data quality from low resolution models, and have been shown to improve performance [87, 99, 5]. The goal of these approaches, however, is more to increase the number of learning samples with easily obtained annotations than to design a good test set.

All benchmarks come with associated evaluation metrics aimed at measuring the discrepancy between the algorithm output and the required ground truth. The current trend is to compute a series of measures, possibly correlated, each one being used to address either a certain type of phenomenon or specific input data, and select a master one for ranking. The proposed metrics are multidimensional: algorithms may fail in various ways and for various types of input data, which motivates the proposition of several corresponding measures.

Adversarial attacks

A complementary approach is to start from a given instantiated function and discover its possible failure cases through specific stress tests or *attacks*.

A particular and notorious approach to build hard examples for deep networks, the current state of the art approach for perceptual functions, is the creation of adversarial examples: they reveal the fact that specifically designed small perturbations may have a dramatic impact on algorithm behavior; *i.e.*, that current deep networks are unstable in several input dimensions.

Since the seminal articles of Szegedy *et al.* [93] and Goodfellow *et al.* [35] that have identified the phenomenon, adversarial examples, both from the attacking and defending sides, have generated a rather huge literature in a very short time. [2] is a recent survey in the computer vision domain, and contains more than 180 references. It distinguishes between white box [13] and black box [71] strategies, between universal [66] and image specific [65] attacks, and whether the fooled output is controlled; *i.e.*, whether its output predicted class is a parameter or not.

Whether adversarial examples are a real threat for real-world or embedded applications is still a debated question. [27] describes real world attacks and shows that simple stickers on road signs may fool the classifier for various viewing conditions.

However, some advocate that the theoretical existence of such a phenomenon is not critical for embedded applications like autonomous driving [56], especially for object detection [57] where the technique proposed in [27] is hard to reproduce.

Given the maturity of this research domain, it is hard to say whether adversarial examples are a real concern for safety issues, or whether their occurrence in real situations is negligible compared to other hazards [31]. However, the already large body of techniques developed can be used to tailor benchmarks of various difficulty levels or simply to improve the robustness of algorithms.

Formal verification

Deep networks are rather complex objects: their behavior is not fully understood, and there are no definite results stating the impact of

[4] https://www.crowdai.org/challenges/adversarial-vision-challenge
[5] http://benchmark.ini.rub.de/index.php?section=gtsrb&subsection=dataset
[6] https://github.com/IBM/adversarial-robustness-toolbox
[7] https://github.com/tensorflow/cleverhans
[8] https://github.com/huanzhang12/Adversarial_Survey
[9] http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm
[10] http://bdd-data.berkeley.edu/
[11] https://www.cityscapes-dataset.com/
[12] http://www.cvlibs.net/datasets/kitti/
[13] https://github.com/chrieke/awesome-satellite-imagery-competitions
[14] A list of resources exploiting simulated data for computer vision is given at https://github.com/unrealcv/synthetic-computer-vision.

optimization, architecture, and data sets on performance stability and accuracy. However, several approaches have attempted to adapt several formal results, or practice validation & verification techniques.

A first series of methods makes use of verification algorithms to evaluate the stability of a network; *i.e.*, their output invariance to perturbations at a given operating point. [46] presents work on verifying the absence of adversarial inputs in generic feed-forward multi-layer neural networks using Satisfiability Modulo Theory (SMT), while [48] develops Reluplex, a simplex formulation of local invariance for networks combining linear and ReLU type non-linearities. [95] formulates verification of piecewise-linear neural networks as a mixed integer program. Those verification processes are exponential in the number of features, and their scaling for large images is an issue. [39] presents a general recent account of a formal method developed to assess safety of deep networks.

A second series of studies examines a global network from a functional point of view, and measures stability through an evaluation of their Lipschitz constant [82, 100].

Finally, [19] takes a statistical learning perspective and extends the Probably Approximately Correct (PAC)-learning framework to account for the presence of adversaries.[97, 98] formally define machine learning safety in terms of risk, epistemic uncertainty, and the harm incurred by unwanted outcomes.

Those methods are related to the emerging topic of *Verified AI,* which proposes to extend the current validation & verification practices to AI [63]. Seshia *et al.* [84] identified five main challenges from a formal method perspective (environment modeling, formal specification, system modeling, computational engines, and correct-by-construction design), and defined several corresponding design principles.

Those principles target generic AI systems and are general, with a twist towards model-based approaches as a prerequisite of many formal methods. The question whether they are relevant to modern perceptual data-driven algorithms is open.

Certification assessment tools

Certification must be understood as a global process that may involve activities not necessarily directed towards operation design or control, but that may be used to assess the safety of the resulting system. When dealing with SI, which relies on machine learning techniques, two activities may help to improve certification.

Explainability

Explainability is the ability of a system to justify the cause or origin of its prediction by providing a dedicated representation: a text or a visual sign.

The idea of providing prediction processes with better intelligibility is not new, and is central to the symbolic approach of AI, sometimes referred as GOFAI (Good Old Fashioned AI) [9], which promotes explicit, *i.e.*, step-by-step understanding and reasoning in its models. The involvement of machine learning techniques in modern methods and the opacity of the resulting prediction processes has encouraged the development of mixed approaches that could benefit jointly from both worlds.

A prominent initiative is the XAI program from the DARPA [38], initiated in 2016, with the final objective of bringing to the user a series of elements that would make him trust and efficiently exploit the predictions made by the automated system. The declared objective of this project is to move the trade-off between process interpretability and performance.

Explainability of artificial intelligence is becoming a research domain in itself, led by various dedicated workshops. Several recent surveys give an idea of the state of the art in this matter: [83] addresses deep network visualization, [44, 1] present a recent literature analysis on deep network visual explanations through a user oriented perspective. [32, 37, 4] are other recent papers that give a broad view of the field.

The fact that a system is able to deliver reliable explanations or proof of good operation is an element that may be used to improve its trustworthiness. The values of explanations can be checked to verify in specific cases that everything is right.

Another use of explanations for authorities is to log them in recording devices for further analysis in case of failure. Explanations usually have a smaller size than the system inner states, and may encode informative features.

The black boxes produced by modern deep learning techniques are not meant to be intelligible – after all, their computing principle is to distribute *subsymbolic* information [90] among large sets of simple components – but they may be completed by side representations that refer faithfully to understandable behaviors. A residual and unsolved problem of explainability is its evaluation: How faithful can those representations be? And who is expected to understand them?

A last issue is related to the deployment of machine learning enabled components: Do they really need explainability? If it is accepted that interpretability can be increased at the expense of accuracy, given that such a trade-off is achievable and that interpretability is measurable, the trustworthiness gain may be worth it. It seems too soon, however, to state that explainability is really achievable and in what sense.

Good practices

The application of machine learning techniques to real SI situations has several pitfalls due to its complexity and to the large number of parameters that require selection, optimization and tuning. Therefore, one way to ensure that a given system is likely to be certified is to demonstrate that it has followed good design principles. There is no success-guaranteed methodology for the machine learning practice. Textbooks provide general principles, theorems and procedures that could help to reach low generalization error, for instance, but with no guarantee. General guidelines have been proposed by several authors [24, 89], but addressing more heuristic objectives rather than performance assessment.

Many modern algorithm proposals are evaluated using standard academic benchmarks that have their own biases and peculiarities: it is often very hard to state whether a given algorithm is really good, or simply better than another when performance figures are only a few percent higher with regard to a specific benchmark. It also appears rather difficult to reproduce the same results as described in a paper due to the experimental dimension of machine learning techniques, although the current trend in computer vision research, for instance, is to make the code available to the community for fair and transparent comparison: code publication should be considered as a mandatory good practice for certification.

Conclusion and challenges

The purpose of this article is to understand the challenges posed by the certification of computer vision-based systems for civil aeronautics. The first difficulty arises from the notion of failure. In standard avionic systems, failures usually refer to *hardware failures* and *systematic failures* usually refer to software bugs. However, vision algorithm may fail even in the absence of a hardware failure or systematic failures, for instance in the case of *adversarial* images or unexpected external conditions (e.g., overexposed images). The first difficulty encountered by certification is to identify all of the possible *algorithm associated hazards* and to show that they are covered by appropriate mitigation means. Faced with this difficulty, we have shown that the current certification standard for civil avionic software cannot apply to vision-based systems. We believe a solution could come from the framework of the *Overarching properties*. We have proposed in Section "Developing specific certification objectives for computer-vision algorithms" a first attempt to refine the three *overarching properties* to specific certification objectives for vision-based systems.

To continue in that direction, we identify five major challenges:

- *Hazard definition.* As stated above, hazard definition is the first main challenge: What is an *algorithm associated hazard*? Is there any typology of such hazards and is it possible to formally characterize them? Then, for a given vision algorithm, the next issue is how to identify the internal *weakness* of the algorithm; *i.e.*, the hazards that the algorithm is sensitive to.
- *Data driven Defined Intended Behavior.* As discussed in Section "Computer vision based system development process: a data driven design logic", building appropriate datasets is a key issue for certification: For a given vision algorithm, how can we build a dataset that covers the usage domain of the algorithm, and more specifically that covers the hazards that the algorithm is sensitive to and that can occur in the usage domain?
- *Hazard detection and mitigation.* The next question is how to detect and to mitigate, algorithm failures at run-time. For instance, in the case of adversarial images, how to detect that the algorithm misinterprets the situation.
- *Explainability.* As stated in Section "Vision-based scene interpretation", explainability could be a promising way to improve the trustworthiness of vision algorithms. The fact that a system is able to produce explanations that are understandable for human users and certification authorities could help to interact with the system and in some cases detect inappropriate behaviors.
- *Consistency.* Finally, failures can also be caused by internal inconsistency; that is, a situation where the effective error is higher than the predicted error computed by the algorithm according to the estimated covariance. However, consistency is very difficult to assess for vision algorithms, due to the non-linearity of the relationship between image data and state parameters. Correctly estimating the quality of the output of an algorithm is a key issue in the field of safety critical systems.

We believe that work on these five challenges could contribute to making vision algorithms usable in safety critical avionic systems. This objective will only be fulfilled if the safety and computer vision communities are able to build a shared research program. We wrote this document with the ambition of taking a step in that direction ■

Acknowledgements

This work has been funded by the PHYDIAS and SUPER projects and the ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANITI).

References

[1] A. Adadi, M. Berrada - *Peeking Inside the Black-Box: a Survey on Explainable Artificial Intelligence (XAI).* IEEE Access, 6:52138-52160, 2018.

[2] N. AKHTAR, A. MIAN - *Threat of Adversarial Attacks on Deep Learning in Computer Vision: a Survey*. IEEE Access, 6:14410-14430, 2018.

[3] L. AKOGLU, H. TONG, D. KOUTRA - *Graph Based Anomaly Detection and Description: a Survey.* Data mining and knowledge discovery, 29(3):626-688, 2015.

[4] A. B. ARRIETA, N. DÍAZ-RODRÍGUEZ, J. DEL SER, A. BENNETOT, S. TABIK, A. BARBADO, S. GARCÍA, S. GIL-LÓPEZ, D. MOLINA, R. BENJAMINS, R. CHATILA, F. HERRERA - *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI.* arXiv preprint arXiv:1910.1004, 2019.

[5] A. ATAPOUR-ABARGHOUEI, T. P. BRECKON - *Real-Time Monocular Depth Estimation using Synthetic Data with Domain Adaptation via Image Style Transfer.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1, 2018.

[6] H. BAY, T. TUYTELAARS, L. VAN GOOL - *Surf: Speeded up Robust Features*. European conference on computer vision, pages 404-417, 2006. Springer.

[7] S. BHATTACHARYYA, D. COFER, D. J. MUSLINER, J. MUELLER, E. ENGSTROM - *Certification Considerations for Adaptive Systems.* Technical report, NASA/CR-2015-218702, Rockwell Collins Ins. et NASA, 2015.

[8] D. BIEDERMANN, M. OCHS, R. MESTER - *Evaluating Visual ADAS Components on the COnGRATS Dataset.* Intelligent Vehicles Symposium (IV), 2016 IEEE, pages 986-991, 2016. IEEE.

[9] M. A. BODEN - *GOFAI*. Frankish, Keith and Ramsey, William M. Editors, editors, The Cambridge Handbook of Artificial Intelligence, pages 89-107. Cambridge University Press, 2014.

[10] A. BORJI, M.-M. CHENG, H. JIANG, J. LI - *Salient Object Detection: a Benchmark*. IEEE transactions on image processing, 24(12):5706-5722, 2015.

[11] M. BURRI, J. NIKOLIC, P. GOHL, T. SCHNEIDER, J. REHDER, S. OMARI, M. W. ACHTELIK, R. SIEGWART - *The EuRoC Micro Aerial Vehicle Datasets*. The International Journal of Robotics Research, 35(10):1157-1163, 2016.

[12] G. O. CAMPOS, A. ZIMEK, J. SANDER, R. J. G. B. CAMPELLO, B. MICENKOVÁ, E. SCHUBERT, I. ASSENT, M. E. HOULE - *On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study.* Data Mining and Knowledge Discovery, 30(4):891-927, 2016.

[13] N. CARLINI, D. WAGNER - *Adversarial Examples are not Easily Detected: Bypassing Ten Detection Methods.* Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pages 3-14, 2017. ACM.

[14] D. CARUSO, M. SANFOURCHE, G. LE BESNERAIS, D. VISSIÈRE - *Infrastructureless Indoor Navigation with an Hybrid Magneto-Inertial and Depth Sensor System.* Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on, pages 1-8, 2016. IEEE.

[15] R. CHALAPATHY, A. K. MENON, S. CHAWLA - *Robust, Deep and Inductive Anomaly Detection.* Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 36-51, 2017. Springer.

[16] V. CHANDOLA, A. BANERJEE, V. KUMAR - *Anomaly Detection: a Survey*. ACM Computing Surveys (CSUR), 41(3):15, 2009.

[17] J. CHELINI, J.-L. CAMUS, C. COMAR, D. BROWN, A.-P. PORTE, M. DE ALMEIDA, H. DELSENY - *Avionics Certification: Back to Fundamentals with Overarching Properties.* 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), 2018.

[18] M. CISSE, P. BOJANOWSKI, E. GRAVE, Y. DAUPHIN, N. USUNIER - *Parseval Networks: Improving Robustness to Adversarial Examples.* International Conference on Machine Learning, pages 854-863, 2017.

[19] D. CULLINA, A. N. BHAGOJI, P. MITTAL - *PAC-Learning in the Presence of Evasion Adversaries.* arXiv preprint arXiv:1806.01471, 2018.

[20] S. DATHATHRI, S. ZHENG, R. M. MURRAY, Y. YUE - *Detecting Adversarial Examples via Neural Fingerprinting.* arXiv preprint arXiv:1803.03870, 2018.

[21] J. DELMERICO, D. SCARAMUZZA - *A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots.* Memory, 10:20, 2018.

[22] T. DEVRIES, G. W. TAYLOR - *Learning Confidence for Out-of-Distribution Detection in Neural Networks.* arXiv preprint arXiv:1802.04865, 2018.

[23] S. DOGRAMADZI, M. E. GIANNACCINI, C. HARPER, M. SOBHANI, R. WOODMAN, J. CHOUNG - *Environmental Hazard Analysis – a Variant of Preliminary Hazard Analysis for Autonomous Mobile Robots.* Journal of Intelligent & Robotic Systems, 76(1):73-117, 2014.

[24] P. M. DOMINGOS - *A Few Useful Things to Know about Machine Learning.* Commun. acm, 55(10):78-87, 2012.

[25] EASA - *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25 - AMC 1309.* Technical Report, EASA, 2017.

[26] S. M. ERFANI, S. RAJASEGARAR, S. KARUNASEKERA, C. LECKIE - *High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning.* Pattern Recognition, 58:121-134, 2016.

[27] I. EVTIMOV, K. EYKHOLT, E. FERNANDES, T. KOHNO, B. LI, A.PRAKASH, A. RAHMATI, D. SONG - *Robust Physical-World Attacks on Machine Learning Models.* arXiv preprint arXiv:1707.08945, 2017.

[28] R. FEINMAN, R. R. CURTIN, S. SHINTRE, A. B. GARDNER - *Detecting Adversarial Samples from Artifacts.* arXiv preprint arXiv:1703.00410, 2017.

[29] M. A. FISCHLER, R. C. BOLLES - *Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.* Communications of the ACM, 24(6):381-395, 1981.

[30] A. GEIGER, P. LENZ, R. URTASUN - *Are we Ready for Autonomous Driving?* The KITTI Vision Benchmark Suite. Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[31] J. GILMER, R. P. ADAMS, I. GOODFELLOW, D. ANDERSEN, G. E. DAHL - *Motivating the Rules of the Game for Adversarial Example Research.* arXiv preprint arXiv:1807.06732, 2018.

[32] L. H. GILPIN, D. BAU, B. Z. YUAN, A. BAJWA, M. SPECTER, L. KAGAL - *Explaining Explanations: an Approach to Evaluating Interpretability of Machine Learning.* arXiv preprint arXiv:1806.00069, 2018.

[33] I. GOODFELLOW, Y. BENGIO, A. COURVILLE, Y. BENGIO - *Deep Learning, Volume 1*. MIT Press Cambridge, 2016.

[34] I. GOODFELLOW, P. MCDANIEL, N.S PAPERNOT - *Explaining and Harnessing Adversarial Examples.* CoRR, abs/1412.6572, 2014.

[35] I. J. GOODFELLOW, J. SHLENS, C. SZEGEDY - *Making Machine Learning Robust against Adversarial Inputs.* Commun. ACM, 61(7):56-66, 2018.

[36] M. C. GRAHAM, J. P. HOW, D. E. GUSTAFSON - *Robust Incremental Slam with Consistency-Checking*. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 117-124, 2015. IEEE.

[37] R. GUIDOTTI, A. MONREALE, F. TURINI, D. PEDRESCHI, F. GIANNOTTI - *A Survey of Methods for Explaining Black Box Models.* arXiv preprint arXiv:1802.01933, 2018.

[38] D. GUNNING - *Explainable Artificial Intelligence (xai).* Defense Advanced Research Projects Agency (DARPA), 2017.

[39] G. HAINS, A. JAKOBSSON, Y. KHMELEVSKY - *Towards Formal Methods and Software Engineering for Deep Learning: Security, Safety and Productivity for dl Systems Development.* 2018 Annual IEEE International Systems Conference (SysCon), pages 1-5, 2018.

[40] C. HARRIS, M. STEPHENS - *A Combined Corner and Edge Detector.* Alvey Vision Conference, pages 10-5244, 1988. Citeseer.

[41] C. HARRIS, M. STEPHENS - *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2003.

[42] K. HE, G. GKIOXARI, P. DOLLÁR, R. GIRSHICK - *Mask r-cnn.* Computer Vision (ICCV), 2017 IEEE International Conference on, pages 2980-2988, 2017. IEEE.

[43] D. HENDRYCKS, K. GIMPEL - *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks.* International Conference on Learning Representations, 2017.

[44] F. M. HOHMAN, M. KAHNG, R. PIENTA, D. H. CHAU - *Visual Analytics in Deep Learning: an Interrogative Survey for the Next Frontiers.* IEEE Transactions on Visualization and Computer Graphics, 2018.

[45] C. M. HOLLOWAY - *Understanding the Overarching Properties: First Steps.* Technical report, NASA Langley Research Center, 2018.

[46] X. HUANG, M. KWIATKOWSKA, S. WANG, M. WU - *Safety Verification of Deep Neural Networks.* International Conference on Computer Aided Verification, pages 3-29, 2017. Springer.

[47] J. T. M. INGIBERGSSON, D. KRAFT, U. P. SCHULTZ - *Explicit Image Quality Detection Rules for Functional Safety in Computer Vision.* VISIGRAPP (6: VISAPP), pages 433-444, 2017.

[48] G. KATZ, C. BARRETT, D. L. DILL, K. JULIAN, M. J. KOCHENDERFER - *Reluplex: an Efficient SMT Solver for Verifying Deep Neural Networks.* International Conference on Computer Aided Verification, pages 97-117, 2017. Springer.

[49] J. Z. KOLTER, E. WONG - *Provable Defenses Against Adversarial Examples via the Convex Outer Adversarial Polytope.* arXiv preprint arXiv:1711.00851, 2017.

[50] K. Lee, K. Lee, K. Min, Y. Zhang, J. Shin, H. Lee - *Hierarchical Novelty Detection for Visual Object Recognition.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1034-1042, 2018.

[51] K. LEE, K. LEE, H. LEE, J. SHIN - *A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks.* NIPS, 2018.

[52] X. LI, F. LI - *Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics.* ICCV, pages 5775-5783, 2017.

[53] S. LIANG, Y. LI, R. SRIKANT - *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks.* International Conference on Learning Representations, 2018.

[54] P. LINDSTROM - *Triangulation made Easy.* Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1554-1561, 2010. IEEE.

[55] J. LU, T. ISSARANON, D. A. FORSYTH - *SafetyNet: Detecting and Rejecting Adversarial Examples Robustly.* ICCV, pages 446-454, 2017.

[56] J. LU, H. SIBAI, E. FABRY, D. FORSYTH - *No Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles.* CVPRW, 2017.

[57] J. LU, H. SIBAI, E. FABRY, D. FORSYTH - *Standard Detectors aren't (Currently) Fooled by Physical Adversarial Stop Signs.* arXiv preprint arXiv:1710.03337, 2017.

[58] X. MA, B. LI, Y. WANG, S. M. ERFANI, S. WIJEWICKREMA, G. SCHOENEBECK, M. E. HOULE, D. SONG, J. BAILEY - *Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality.* International Conference on Learning Representations, 2018.

[59] A. MADRY, A. MAKELOV, L. SCHMIDT, D. TSIPRAS, A. VLADU - *Towards Deep Learning Models Resistant to Adversarial Attacks.* International Conference on Learning Representations, 2018.

[60] A. MANDELBAUM, D. WEINSHALL - *Distance-Based Confidence Score for Neural Network Classifiers.* arXiv preprint arXiv:1709.09844, 2017.

[61] M. MARKOU, S. SINGH - *Novelty Detection: a Review – Part 1: Statistical Approaches.* Signal processing, 83(12):2481-2497, 2003.

[62] D. MENG, H. CHEN - *Magnet: a Two-Pronged Defense against Adversarial Examples*. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 135-147, 2017. ACM.

[63] T. MENZIES, C. PECHEUR - *Verification and Validation and Artificial Intelligence.* Advances in computers, 65:153-201, 2005.

[64] J. T. I. MOGENSEN, D. KRAFT, U. P. SCHULTZ - *Declarative Rule-Based Safety for Robotic Perception Systems.* Journal of Software Engineering for Robotics, 8(1):17-31, 2017.

[65] S.-M. MOOSAVI-DEZFOOLI, A. FAWZI, P. FROSSARD - *Deepfool: a Simple and Accurate Method to Fool Deep Neural Networks.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2574-2582, 2016.

[66] S.-M. MOOSAVI-DEZFOOLI, A. FAWZI, O. FAWZI, P. FROSSARD - *Universal Adversarial Perturbations.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1765-1773, 2017.

[67] M. MUELLER, V. CASSER, J. LAHOUD, N. SMITH, B. GHANEM - *UE4Sim: a Photo-Realistic Simulator for Computer Vision Applications.* arXiv preprint arXiv:1708.05869, 2017.

[68] M.-I. NICOLAE, M. SINN, M. N. TRAN, A. RAWAT, M. WISTUBA, V. ZANTEDESCHI, I. M. MOLLOY, B. EDWARDS - *Adversarial Robustness Toolbox v0.3.0.* arXiv preprint arXiv:1807.01069, 2018.

[69] N. PAPERNOT, P. MCDANIEL - *Extending Defensive Distillation.* arXiv preprint arXiv:1705.05264, 2017.

[70] N. PAPERNOT, P. MCDANIEL, X. WU, S. JHA, A. SWAMI - *Practical Black-Box Attacks Against Machine Learning.* Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pages 506-519, 2017. ACM.

[71] N. PAPERNOT, P. MCDANIEL, I. GOODFELLOW, S. JHA, Z. B. CELIK, A. SWAMI - *Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks.* Security and Privacy (SP), 2016 IEEE Symposium on, pages 582-597, 2016. IEEE.

[72] M. A. F. PIMENTEL, D. A. CLIFTON, L. CLIFTON, L. TARASSENKO - *A Review of Novelty Detection*. Signal Processing, 99:215-249, 2014.

[73] H. ROGGEMAN, J. MARZAT, A. BERNARD-BRUNEL, G. LE BESNERAIS - *Autonomous Exploration with Prediction of the Quality of Vision-Based Localization.* IFAC-PapersOnLine, 50(1):10274-10279, 2017.

[74] E. ROSTEN, T. DRUMMOND - *Machine Learning for High-Speed Corner Detection.* European conference on computer vision, pages 430-443, 2006. Springer.

[75] RTCA, Inc. DO-178 ED-12B - Software Considerations in Airborne Systems and Equipment Certification. 2008.

[76] RTCA, Inc. DO-178 ED-12C - Software Considerations in Airborne Systems and Equipment Certification. 2011.

[77] E. RUBLEE, V. RABAUD, K. KONOLIGE, G. BRADSKI - *ORB: An Efficient Alternative to SIFT or SURF.* Computer Vision (ICCV), 2011 IEEE international conference on, pages 2564-2571, 2011. IEEE.

[78] L. RUFF, N. GOERNITZ, L. DEECKE, S. A. SIDDIQUI, R. VANDERMEULEN, A. BINDER, E. MÜLLER, M. KLOFT - Deep One-Class Classification. International Conference on Machine Learning, pages 4390-4399, 2018.

[79] SAE. Aerospace Recommended Practices ARP4754a - Development of Civil Aircraft and Systems. 2010. SAE.

[80] SAE. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. 2018. SAE.

[81] M. SANFOURCHE, V. VITTORI, G. LE BESNERAIS - *Evo: A Realtime Embedded Stereo Odometry for MAV Applications.* Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 2107-2114, 2013. IEEE.

[82] K. SCAMAN, A. VIRMAUX - *Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation.* arXiv preprint arXiv:1805.10965, 2018.

[83] C. SEIFERT, A. AAMIR, A. BALAGOPALAN, D. JAIN, A. SHARMA, S. GROTTEL, S. GUMHOLD - *Visualizations of Deep Neural Networks in Computer Vision: a Survey.* Transparent Data Mining for Big and Small Data, pages 123-144. Springer, 2017.

[84] S. A. SESHIA, D. SADIGH, S. S. SASTRY - *Towards Verified Artificial Intelligence*. arXiv preprint arXiv:1606.08514, 2016.

[85] S. SHAH, D. DEY, C. LOVETT, A. KAPOOR - *Airsim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles.* Field and service robotics, pages 621-635, 2018. Springer.

[86] J. SHI, C. TOMASI - *Good Features to Track.* Technical Report, Cornell University, 1993.

[87] A. SHRIVASTAVA, T. PFISTER, O. TUZEL, J. SUSSKIND, W. WANG, R. WEBB - *Learning from Simulated and Unsupervised Images through Adversarial Training.* 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2242-2251, 2017.

[88] A. SINHA, H. NAMKOONG, J. DUCHI - *Certifiable Distributional Robustness with Principled Adversarial Training*. International Conference on Learning Representations, 2018.

[89] L. N. SMITH - *Best Practices for Applying Deep Learning to Novel Applications.* arXiv preprint arXiv:1704.01568, 2017.

[90] P. SMOLENSKY - *Connectionist AI, Symbolic AI, and the Brain.* Artificial Intelligence Review, 1(2):95-109, 1987.

[91] D. SU, H. ZHANG, H. CHEN, J. YI, P.-Y. CHEN, Y. GAO - *Is Robustness the Cost of Accuracy? A Comprehensive Study on the Robustness of 18 Deep Image Classification Models.* arXiv preprint arXiv:1808.01688, 2018.

[92] L. SWERSKY, H. O. MARQUES, J. SANDER, R. J. G. B. CAMPELLO, A. ZIMEK - *On the Evaluation of Outlier Detection and One-Class Classification Methods.* Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on, pages 1-10, 2016. IEEE.

[93] C. SZEGEDY, W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. GOODFELLOW, R. FERGUS - *Intriguing Properties of Neural Networks.* ICLR, 2014.

[94] A. TAEIHAGH, H. LIM - *Governing Autonomous Vehicles: Emerging Responses for Safety, Liability, Privacy, Cybersecurity, and Industry Risks.* Transport Reviews, 1-26, 2018.

[95] V. TJENG, R. TEDRAKE - *Verifying Neural Networks with Mixed Integer Programming.* arXiv preprint arXiv:1711.07356, 2017.

[96] F. TRAMÈR, A. KURAKIN, N. PAPERNOT, I. GOODFELLOW, D. BONEH, P. MCDANIEL - *Ensemble Adversarial Training: Attacks and Defenses.* International Conference on Learning Representations, 2018.

[97] K. R. VARSHNEY - *Engineering Safety in Machine Learning.* Information Theory and Applications Workshop (ITA), 2016, pages 1-5, 2016. IEEE.

[98] K. R. VARSHNEY, H. ALEMZADEH - *On The Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products.* Big data, 5(3):246-255, 2017.

[99] T.-C. WANG, M.-Y. LIU, J.-Y. ZHU, A. TAO, J. KAUTZ, B. CATANZARO - *High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs.* 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 8798-8807, 2018.

[100] T.-W. WENG, H. ZHANG, P.-Y. CHEN, J. YI, D. SU, Y. GAO, C.-J. HSIEH, L. DANIEL - *Evaluating the Robustness of Neural Networks: an Extreme Value Theory Approach.* ICLR, 2018.

[101] W. XU, D. EVANS, Y. QI - *Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks.* arXiv preprint arXiv:1704.01155, 2017.

[102] O. ZENDEL, M. MURSCHITZ, M. HUMENBERGER, W. HERZNER - *How Good is my Test Data? Introducing Safety Analysis for Computer Vision.* International Journal of Computer Vision, 125(1-3):95-109, 2017.

[103] S. ZHAI, Y. CHENG, W. LU, Z. ZHANG - *Deep Structured Energy Based Models for Anomaly Detection.* International Conference on Machine Learning, pages 1100-1109, 2016.

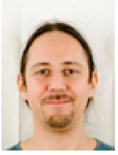
[104] Z. Zhang - *Parameter Estimation Techniques: a Tutorial with Application to Conic Fitting.* Image and vision Computing, 15(1):59-76, 1997.

[105] A. ZIMEK, E. SCHUBERT, H.-P. KRIEGEL -*A Survey on Unsupervised Outlier Detection in High-Dimensional Numerical Data.* Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5):363-387, 2012.

AUTHORS

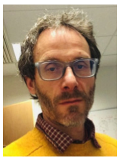
Frédéric Boniol graduated from a French High School for Engineers in Aerospace Systems (Suapero) in 1987. He holds a PhD in computer science from University of Toulouse (1997). Since 1989, he works on the modeling and verification of embedded and real-time systems. Up to 2008, he was professor at ENSEEIHT. He has now a research position at ONERA. His research interests include modeling languages for real-time systems, formal methods, and computer-aided verification applied to avionics systems.

Adrien Chan-Hon-Tong received a phd from université Paris Sorbonne in 2014, and enters at ONERA. He has worked from 2014 to 2018 on the development of object detection algorithm working on satellite images and/or UAV videos (mainly for military purpose). Since 2018, he focuses on safety issues raised by machine learning on critical plateform (autonomous driving, lethal autonomous weapon systems): performance evaluation on operationnal condition, robustness to adversarial attack, data poisoning...



Alexandre Eudes received his Phd degree in robotics and computer vision from university Blaise Pascal in 2010, on visual SLAM for real-time car localization applications. He is currently a research engineer at ONERA in the Image Vision leArning(IVA) unit of the Information processing and systems department (DTIS). His research problematic are focus on state estimation (Visual SLAM, multi-sensor fusion, decentralized estimation) for experimental application on autonomous navigation of a single or team of terrestrial and aerial robots.

Stéphane Herbin received an engineering degree from the Ecole Supérieure d'Electricité (Supélec), the M. Sc. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign, and the PhD degree in applied mathematics from the Ecole Normale Supérieure de Cachan. He was employed by Aérospatiale Matra Missiles (now MBDA) from 1998 to 2000. He joined ONERA in 2000, and has been working since then in the Information Processing and Modelling Department. His research adresses mainly the design of models and algorithms for data interpretation with a focus on images and videos.



Guy Le Besnerais graduated from ENSTA in 1989 and obtained the PhD degree from Université Paris Sud in 1993. Since 1994 works at ONERA, The French Aerospace Lab, now with the Information Processing and System Department (DTIS) in Palaiseau (91) with the grade of Research Director. He has obtained the "*Habilitation à diriger les recherches*" (HDR) in 2008 and is affiliated to University Paris-Saclay. His research activities include methodology for solving inverse problems, performance modeling for imaging measurement systems, embedded vision for robotics applications.

Claire Pagetti. Since 2005, Claire Pagetti is a researcher at ONERA and since 2007 an associate professor at ENSEEIHT. She holds a research chair in the ANITI project on "New certification approaches of AI based systems for civil aeronautics". She defended her habilitation (HDR) in 2015. Her fields of interest concern the safe implementation of control command avionic applications on avionic platforms. She has contributed to several industrial, European and French projects that lead to several publications, industrial development and a patent. She was responsible of the Torrents cluster and participated to French GDR groups. From September 2016 to August 2017, she was on a sabbatical at TUHH where she worked on WCET-aware compilation of synchronous programs with the WCC compiler

Martial Sanfourche received is Master of Science in Computer Science from Université de Cergy-Pontoise in 2001 then a Ph-D degree in image and signal processing from the Université de Cergy-Pontoise in 2005. After a postdoctoral position at CNRS-LAAS, he joined ONERA in 2007 where is now a research engineer in computer vision. His current research interest include online and offline visual localization and mapping for robotic systems.

M. Nugue, J.-M. Roche,
G. Le Besnerais, C. Trottier,
R. W. Devillers
(ONERA)

J. Pichillou
(CNES)

A. Chan-Hon-Tong, A. Boulch,
A. Hurmane
(ONERA)

E-mail: adrien.chan_hon_tong@onera.fr

DOI: 10.12762/2020.AL15-07

Scaling Up Information Extraction from Scientific Data with Deep Learning

This paper presents two use cases where deep learning is able to help scientists by removing the burden of manual review of large volumes of physical data. Such examples highlight why deep learning could become a transverse tool across many scientific fields.

Introduction

Designing an experimental protocol to validate a scientific hypothesis, and analyzing the resulting data is an intellectual process that cannot be solved by current artificial intelligence. However, in this process at the edge of the human mind, there are often sub-tasks that are repetitive, time-consuming and that do not involve expert skill or knowledge. This paper is aimed at showing that existing artificial intelligence, like deep learning, can solve such tasks.

For example, there are common scientific issues where understanding a physical phenomenon is slowed down by the need to extract information from large volumes of physical data, and in this context the application of deep learning can be very relevant. This article presents such scientific use cases, and the deep learning algorithms that have been able to scale information extraction where humans alone have not.

Specifically, this article focuses on two successful ONERA scenarios linked to the DELTA project[1]: solid-propellant combustion and material resistance.

Solid-propellant combustion analysis by shadowgraphy

Solid propellants are widely used for spatial and military applications. For example, they are used for the first stage of the Ariane V launcher. Classically, aluminum particles are included in the composite to increase the efficiency of the solid-propellant combustion, improving the thrust by 10% [4]. However, aluminum particles can trigger various negative effects, such as diphasic losses, film formation on the nozzle or pressure instability (e.g., thermoacoustic instabilities [6]).

Understanding the physical phenomena associated with burning aluminum droplets is a critical issue to design new generations of engines and/or new solid propellant compositions. Typically, numerical simulation results are highly dependent on two critical input parameters: the initial size distribution of the droplets, and their initial velocity distribution when they leave the burning surface [7]. Defining precise representative droplet size distributions is not as easy as it may seem since agglomeration phenomena [5] strongly modify the droplet diameters due to the aluminum powders introduced in the composite material. Similarly, accurate velocity data are not available today under representative burning propellant conditions (for instance, recent data were published [3] but only for 0.1MPa burning conditions). Hence, obtaining access to these two data would be a breakthrough to update physical models and the numerical simulations for solid-propellant combustion.

For the past couple of years, the ONERA solid propellant research team has been using an experimental setup to characterize aluminum droplet combustion under relevant solid propellant conditions [15]. High-speed visualization of droplets is achieved via a focused shadowgraphy diagnostic (see Figure 1). The diagnostic

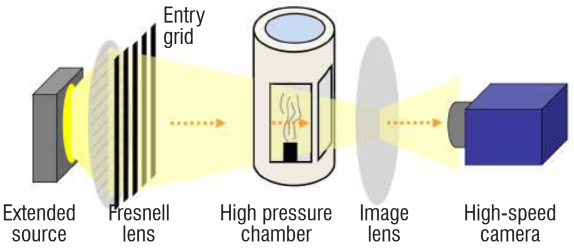


Figure 1 – Shadowgraphy setup enabling aluminum droplets to be visualized during solid-propellant combustion

[1] https://delta-onera.github.io/

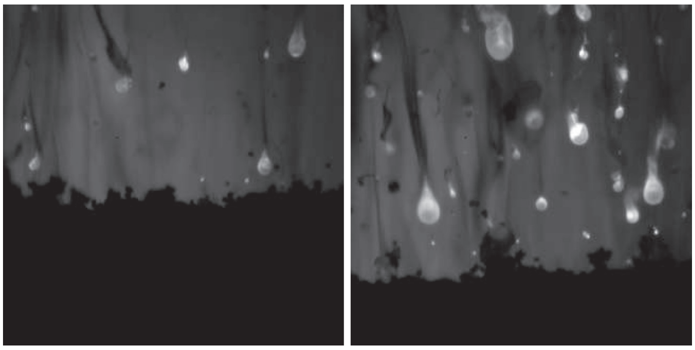


Figure 2 – Images obtained by shadowgraphy

is sensitive to refractive index gradients, leading to a good contrast between the liquid aluminum droplets and the surrounding hot gases (see Figure 2).

The experimental images in Figure 2 show that ejected burning aluminum droplets are clearly visible to human eyes as bright grey balloons over the dark background. Thus, from a theoretical point of view, there is no obstacle for obtaining the size and speed of particles: *it only requires somebody to scan the existing* experimental images and to extract the data.

However, each combustion test generates between 1,000 and 10,000 images, each showing several dozen aluminum droplets. Considering the volume of data, extracting the desired parameters is impossible using human hands/eyes. In other words, today, combustion scientists know how to generate data containing the information but not how to extract the information from this large volume of data. Making progress on an algorithm enabling this information extraction could allow this scientific obstacle to be overcome.

Modeling some material properties with microscopy

The ability to predict the fracture behavior of a material under strain is essential to design a car, building or plane. However, in aeronautics, the use of a large variety of materials (especially composite materials) makes it difficult to derive the fracture behavior solely from solid mechanic physical equations. Thus, fracture mechanics sometimes relies on experimental data to characterize a special kind of material [9].

Such an experiment consists in measuring a material state at different levels of strain, and producing a stress versus strain curve. Measurements can include acoustic measurements, image correlation (when it is possible to obtain an image before deformation), thermal imaging, or microscopy imaging.

A typical example of such a microscopy experiment is provided in Figure 3. Carbon epoxy laminates enhance damages within their plies. These damages are observed at all 3 scales of the material, i.e., the microscale (fiber and matrix), the mesoscale (ply), and the macroscale (laminate). The main damages that can appear during a tension test of a [0/90]*s* laminate are illustrated in Figure 3. First of all, when damage at the microscale appears, it is in the form of fiber/matrix debonding or micro cracks within the resin (3.1 in the figure). Loading the sample causes the damages at the fiber scale to grow upward and leads to a transverse matrix crack at the ply scale (3.2) parallel to the fiber direction. These cracks cross the entire thickness of the ply. In some composites, micro delamination can be observed at the crack tips. Finally, when the strength limit of the sample is reached, the crack density increases (3.3) and leads to fiber failure due to redistribution of the load (3.4).

Here again, scientists have designed an experimental protocol allowing the behavior of these materials to be modelled under stress. However, this involves the analysis of all resulting data to count and/or characterize all fractures. Unfortunately, the volume of resulting data is also too large to be processed by human hands/eyes in these experiments. Again, making progress on the development of an algorithm enabling this information to be extracted could allow this scientific obstacle to be overcome.

A few words about deep learning

Deep learning (which is becoming the state-of-the-art in image classification with [10]) consists simply of neural networks (which existed at least since [12]) with a large number of layers. This simple enlargement of neural networks, allowed by the increase in available GPU, eventually leads to very impressive performance gap with previous neural networks attempts. However, deep learning has been successfully applied to many tasks: image classification [10], object detection in images [8], natural image segmentation [1], semantic segmentation in remote sensing images [11], natural language processing like automatic translation [22], sound classification [17], cyber intrusion detection [21], malware classification [25], games (with a reinforcement learning framework) [19], medical diagnosis [2], etc.

Supervised classification: More generally, deep learning is a sub-field of supervised machine learning, which is a sub-field of machine learning, which in turn is a sub-field of artificial intelligence. The simplest form of supervised machine learning is binary supervised classification: it consists in estimating an unknown function f of $\mathbf{R}^D$ to $\{-1,1\}$ from training data; i.e., a set of samples $(x_1, y_1 = f(x_1)),\ldots,(x_N, y_N = f(x_N))$. A learning step in this training data leads to a function $\hat{f}$ (classically $\hat{f}(x) = g(x,\hat{w})$ and learning consists in selecting weights w to set up a function family $g(x,w)$). This function $\hat{f}$ is an approximation

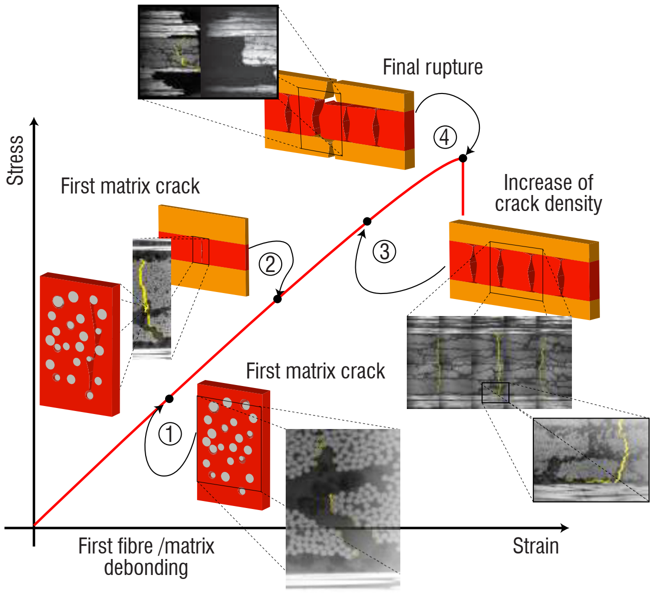


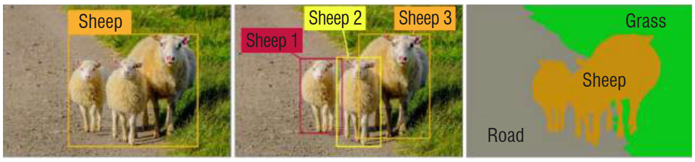
Figure 3 – Typical experiment to characterize a material

f accepting that $\hat{f}$ may be different from f. Usually, the quality of the approximation is evaluated by comparing $\hat{f}$ and f for a disjoint set of samples known as the test data. Of course, given that multiple functions f will lead to the same training data, there is no mathematical possibility of finding the correct function in the discrete case (this statement is confirmed in [23]). However, in practice, given that the function f to be approximated generally has a good regularity property, in practice very interesting results can be achieved.

For example, the function f could use an image as input and provide *cat* or *dog* as the output depending on whether the image is *an image of a cat* or *an image of a dog*. Using a human being, f can be trivially evaluated on a particular image just by asking the human whether this is *an image of a cat* or *an image of a dog*. However, nobody is able to compute this function exactly, even with an arbitrarily powerful computer, since the steps leading to the decision are unknown. However, it is possible to collect training data and to approximate this function quite efficiently, especially with deep learning.

Supervised segmentation: A very natural extension to supervised classification on images is supervised segmentation, where the goal is to classify each pixel of an input image (see Figure 4).

Contrarily to most other classifiers, deep learning based classifiers are much more flexible. Simply changing a few layers (or even just changing the layer structure) allows a classifier to be converted into a segmentation algorithm. In addition, specific networks for segmentation have been designed, like UNET [18], whose architecture is presented in Figure 5. The first half of the network is considered to be an encoder, i.e., it extracts more and more relevant information from the image, but loses more and more precise spatial information. Then, the second half would decode the decoder; i.e., it restores precise spatial location by combining different levels of encoded information. At the end of the process, the network produces a map with the same spatial size as the input image, in which each pixel is associated with a likelihood per class.



Classification + Localization   Object Detection   Semantic Segmentation

Figure 4 – Illustration of different natural extensions of classification

This framework is especially relevant for the two scientific use cases presented in this paper. Indeed, in these two use cases, relevant physical information can be obtained from a segmentation map.

Furthermore, when processing scientific data, deep learning can safely assume that hacked data will not be encountered (indeed, sensibility to adversarial attacks [13, 24, 16, 20, 14] is an important issue for deep learning).

Particle segmentation by deep learning

As presented in Figures 1 and 2, the shadowgraphy diagnostic provides experimental images of the solid-propellant combustion showing aluminum droplets. Thus, a straightforward idea is to apply UNET (see Figure 4) to shadowgraphy images (Figure 2).

First, given that we are considering supervised segmentation, it is unfortunately unavoidable to consider the issue of designing a semantic segmentation dataset. Thus, this step consists in manually annotating some images for semantic segmentation; i.e., providing a class for each pixel.

This is a very repetitive and time-consuming task. The good news is that, contrarily to common thinking, UNET does not require too many images to reach a relevant state depending on the type of data.

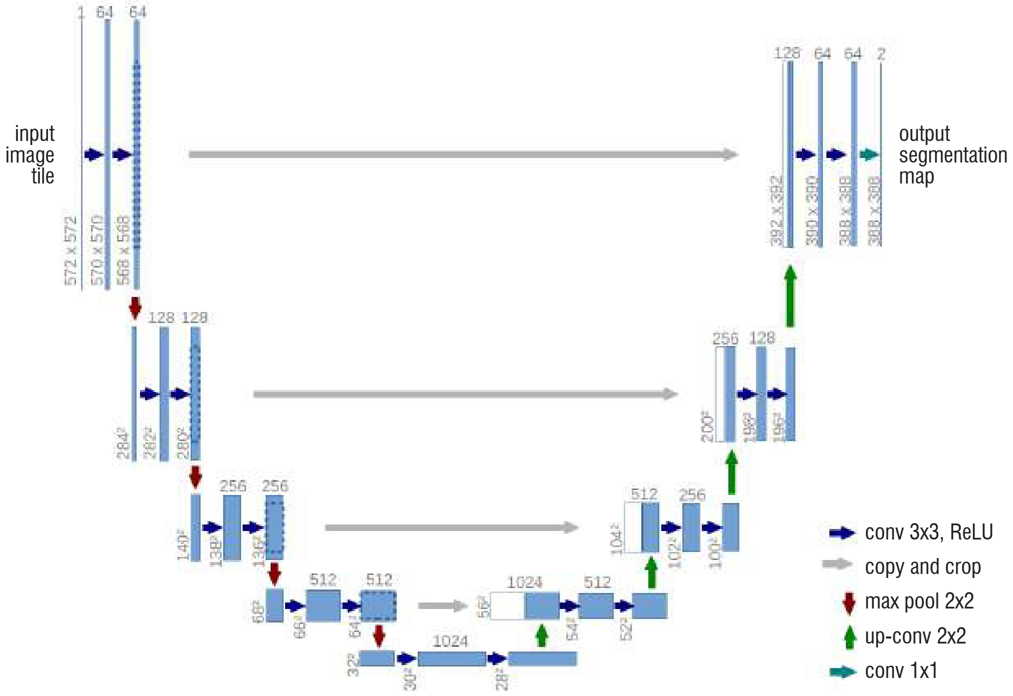


Figure 5 – UNET is a representative example of a segmentation network

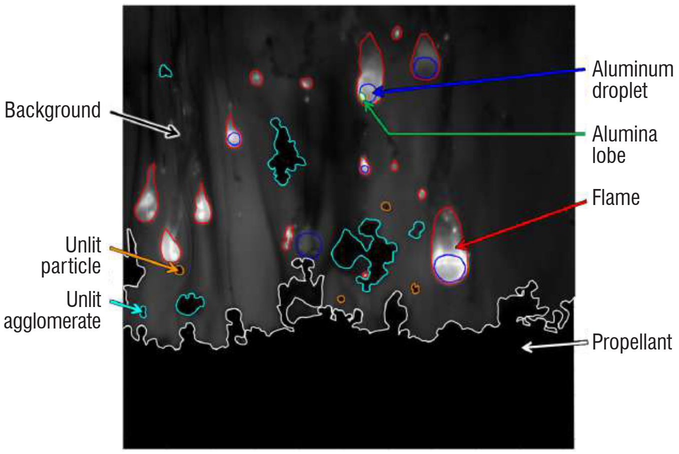
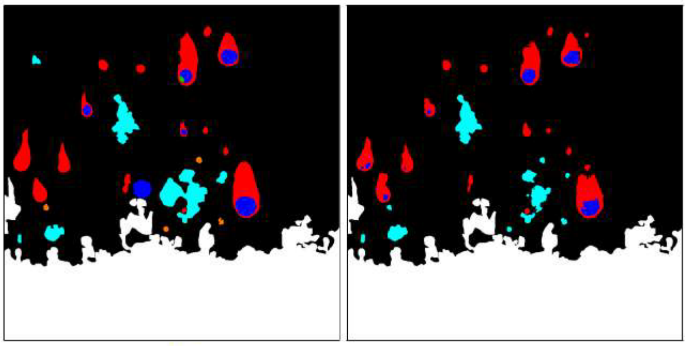


Figure 6 – Illustration of the manual annotation required to apply UNET



(a) VT

(b) U-Net

Figure 7 – Illustration of a UNET prediction on a testing image

Given that each pixel leads to a decision in semantic segmentation, an image corresponds to 1 million data points. This is a number quite close to the size of the well-known IMAGENET dataset. Off course, pixels are much more correlated than the IMAGENET examples, but, on the other hand, the variability of shadowgraphy images may be lower than the variability of a natural image. Here, the difficulty is mainly due to the low contrast, irregular shape and very complex features of shadowgraphy images. With these ideas in mind, we only annotate 30 images from 3 experimental videos (10 images per video).

Then, we perform a cross-validation: we learn from 2 videos (20 images) and test the model on the last video (10 images). This way, we can evaluate the generalization to new videos. Eventually, the algorithm is able to generalize (quite fairly) to new videos: see Figure 7.

Now, our goal is to estimate the droplet size distribution during combustion. A perfect segmentation of the images enables the size distribution to be evaluated accurately. However, predicted segmentation will always be more or less noisy, and it is not trivial to derive the resulting noise in the droplet size distribution. Also, estimating the distribution on a small subset of the images (e.g., the annotated data) may not be sufficiently accurate. Thus, we cannot use as a metric the distance between the distribution estimated by annotation and that obtained by the algorithm. Hence, we propose here to only quantify the segmentation (and not the resulting size distribution). In our cross-validation evaluation, the multi-class accuracy of UNET (i.e., the number of pixels correctly predicted divided by the number of pixels) is above 90% (training is not convex, so multiple runs do not lead exactly to the same accuracy even if variance is not an issue).

After post-processing, the resulting size distributions with different post processing (estimated on all frames of the video) are presented in Figure 8 alongside the size before burst (which is known to change during burst). Even though none of these curves can be considered as reference distributions, we can see that they seem consistent. Estimated distributions have a greater number of large particles, which is expected due to the aggregation phenomenon during burst.

As a partial conclusion, although the developed process is not complete (particularly with regard to guaranteeing the estimated distribution), it leads to very promising results and already allows input data to be refined for numerical simulations of solid-propellant combustion.

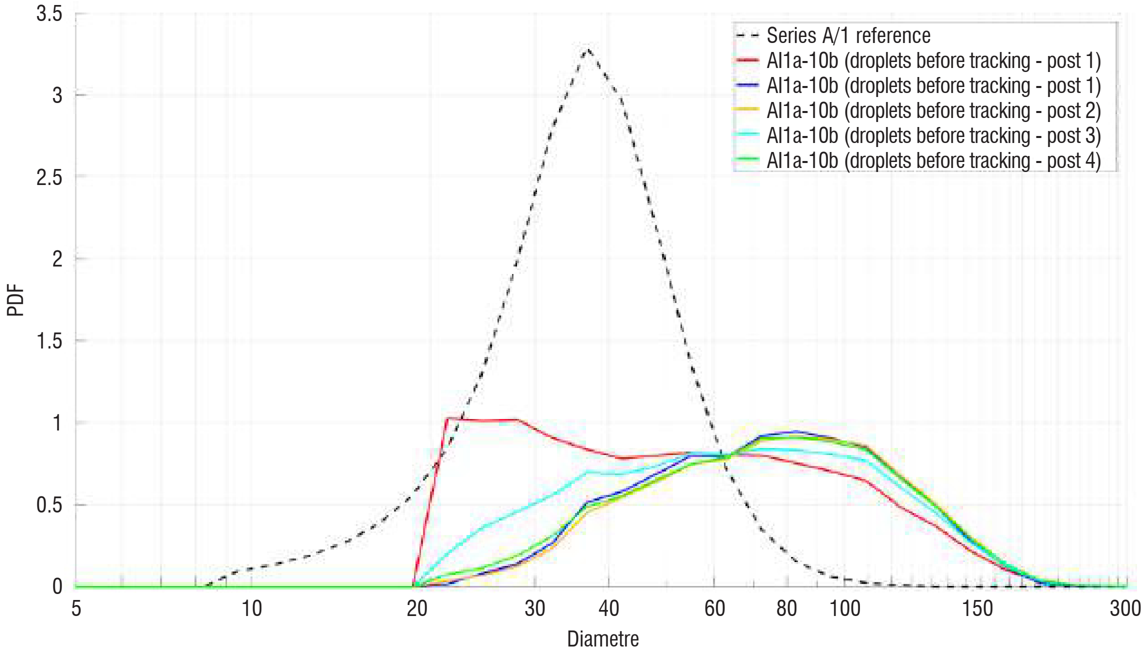


Figure 8 – Illustration of the extracted size distribution

Crack segmentation by deep learning

Like in the previous use case, UNET is a straightforward idea for crack segmentation. Here, other algorithms from the computer vision community could be relevant. Indeed, the semantic object of interest is not defined by itself but by opposition to a normal situation. Thus, abnormality detection, or even, edge detection could have been investigated at first glance. However, abnormality detection is currently less mature than supervised detection, and edge detection may be deceiving considering that a non-uniform material leads to strong contrast. Thus, using supervised deep learning to capture the structure of cracks, which are the only edges of interest, is a good shot.

The processing of these images is considered as a binary semantic segmentation problem where the goal is to give to each pixel either a crack tag or a background tag.

These data are in a way harder to obtain than shadowscopy ones (from a deep-learning point of view) because of the unbalance of crack pixels vs. background pixels, and the required spatial accuracy (as cracks are linear area not delimited surfacique object).

Indeed, the number of background pixels is much greater than the number of crack pixels. Thus predicting all pixel as background leads to a very small error i.e. a good local minimum.

However, applying UNET with an *ad hoc* weighting of error proved to be sufficient to learn a relevant model for crack segmentation. The weighting consists in penalizing an error on a crack pixel much more than an error on a background pixel. This way, the network is forced to go outside the local minimum where all pixels are background ones.

The resulting performances of the network are quite interesting as shown by Figures 9 and 10.

Conclusion

In this paper, we focus on two real-life use cases where the size of the experimental data is an obstacle in the understanding of a physical phenomenon. We show how, in these two contexts, deep learning was successfully applied to help scientists to achieve the processing of the experimental data.

However, this success is not that surprising, since the selected experimental data are images, which is one of the primary areas of deep learning development. However, this article still shows that state-of-the-art deep learning is becoming more and more mature for use in processing scientific experimental data ■

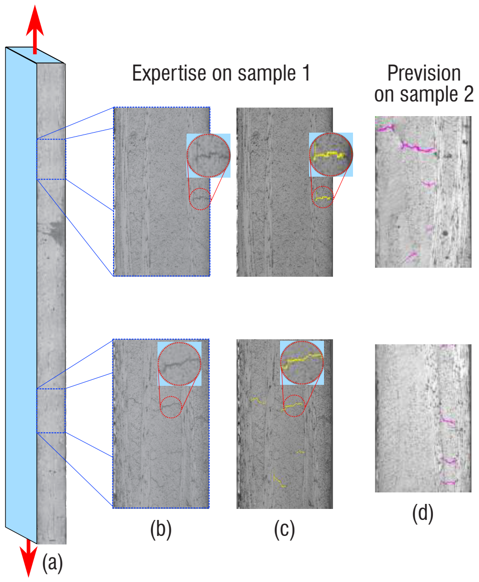


Figure 9 – Application of UNET to detect cracks on a glass/epoxy 2D woven composite. (a) Microscopic observation of the entire sample under tension, (b) Zoom on cracks, (c) Manual assessment of the crack (in yellow), (d) Cracks predicted by our UNET model (in pink).

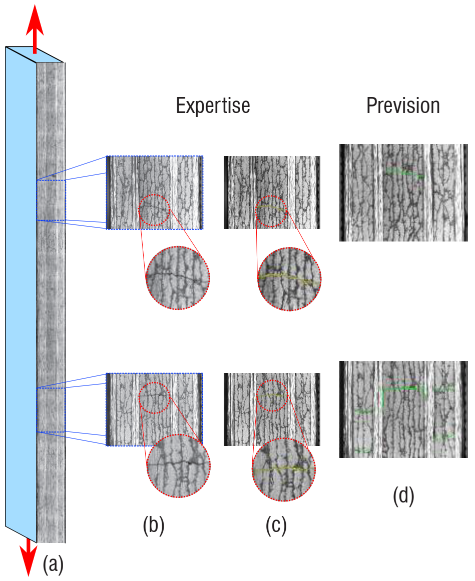


Figure 10 – Application of UNET to detect cracks on a carbon/epoxy [0/90] laminated composite. (a) Microscopic observation of the entire sample under tension, (b) Zoom on cracks, (c) Manual assessment of a crack (in yellow), (d) Cracks predicted by our UNET model (in green).

Acknowledgement

We wish to thank Antoine HURMANE and Francois Henri LEROY, for their help with this paper. Alexandre BOULCH is now a researcher at Valeo, but this work was done while he was working at ONERA.

References

[1] V. BADRINARAYANAN, A. HANDA, R. CIPOLLA - *Segnet: A deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling.* arXiv preprint arXiv:1505.07293, 2015.

[2] O. BERNARD, A. LALANDE, C. ZOTTI, F. CERVENANSKY, X. YANG, P.-A. HENG, I. CETIN, K. LEKADIR, O. CAMARA, M. A. G. BALLESTER, *et al.* - *Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: is the Problem Solved?* IEEE transactions on medical imaging, 37(11):2514-2525, 2018.

[3] Y. CHEN, D. GUILDENBECHER, K. HOFFMEISTER, M. COOPER, H. STAUFFACHER, M. OLIVER, E. WASHBURN - *Study of Aluminum Particle Combustion in Solid Propellant Plumes using Digital In-Line Holography and Imaging Pyrometry*. Combustion and Flames, 182:225-237, 2017.

[4] Y. FABIGNON, J. ANTHOINE, D. DAVIDENKO, R. DEVILLERS, J. DUPAYS, D. GUEYFFIER, J. HIJLKEM, N. LUPOGLAZOFF, J.-M. LAMET, L. TESS, A. GUY, C. ERADS - *Recent Advances in Research on Solid Rocket Propulsion.* AerospaceLab Journal, 11:AL11-13, 2016.

[5] S. GALLIER - *A stochastic Pocket Model for Aluminium Agglomeration in Solid Propellants.* Propellants Explos. Pyrotech, 34:97-105, 2009.

[6] S. GALLIER, F. GODFROY - *Aluminum Combustion Driven Instabilities in Solid Rocket Motors.* Journal of propulsion and power, 25(2):509-521, 2009.

[7] A. GENOT, S. GALLIER, T. SCHULLER - *Thermo-Acoustic Instabilities Driven by Fuel Droplet Lifetime Oscillations.* Proceedings of the Combustion Institute, 37(4):5359-5366, 2019.

[8] R. GIRSHICK - *Fast R-CNN*. Proceedings of the IEEE International Conference on Computer Vision, pages 1440-1448, 2015.

[9] C. HUCHETTE - *Sur la complémentarité des approches expérimentales et numériques pour la modélisation des mécanismes d'endommagement des composites stratifiés.* Thesis, 2005.

[10] A. KRIZHEVSKY, I. SUTSKEVER, G. E. HINTON - *Imagenet Classification with Deep Convolutional Neural Networks.* Advances in neural information processing systems, pages 1097-1105, 2012.

[11] A. LAGRANGE, B. LE SAUX, A. BEAUPERE, A. BOULCH, A. CHAN-HON-TONG, S. HERBIN, H. RANDRIANARIVO, M. FERECATU - *Benchmarking Classification of Earth-Observation Data: from Learning Explicit Features to Convolutional Networks.* 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pages 4173-4176. IEEE, 2015.

[12] Y. LECUN, B. E. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. E. HUBBARD, L. D. JACKE - *Hand-Written Digit Recognition with a Back-Propagation Network.* Advances in Neural Information Processing Systems, pages 396-404, 1990.

[13] S. M. M. DEZFOOLI, A. FAWZI, O. FAWZI, P. FROSSARD - *Universal Adversarial Perturbations.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[14] A. NGUYEN, J. YOSINSKI, J. CLUNE - *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 427-436, 2015.

[15] M. NUGUE, R. W. DEVILLERS, G. LE BESNERAIS, N. CESCO - *Particle Detection and Size Evaluation in Solid Propellant Flames via Experimental Analysis to Improve Two-Phase Flow Simulation in Rocket Motors.* Space Propulsion 2016 Conference, Rome, page 3124722, 2016.

[16] N. PAPERNOT, P. MCDANIEL, S. JHA, M. FREDRIKSON, Z. B. CELIK, A. SWAMI - *The Limitations of Deep Learning in Adversarial Settings.* Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, pages 372-387. IEEE, 2016.

[17] K. J. PICZAK - *Environmental Sound Classification with Convolutional Neural Networks.* 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pages 1-6. IEEE, 2015.

[18] O. RONNEBERGER, P. FISCHER, T. BROX - *U-Net: Convolutional Networks for Biomedical Image Segmentation.* International Conference on Medical image computing and computer-assisted intervention, pages 234-241. Springer, 2015.

[19] D. SILVER, J. SCHRITTWIESER, K. SIMONYAN, I. ANTONOGLOU, A. HUANG, A. GUEZ, T. HUBERT, L. BAKER, M. LAI, A.BOLTON, *et al.* - *Mastering the Game of Go Without Human Knowledge.* Nature, 550(7676):354, 2017.

[20] C. SZEGEDY, W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. J. GOODFELLOW, R. FERGUS - *Intriguing Properties of Neural Networks.* technical report arxiv:1312.6199, 2013.

[21] A. TUOR, S. KAPLAN, B. HUTCHINSON, N. NICHOLS, S. ROBINSON - *Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams.* Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[22] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER, I. POLOSUKHIN - *Attention is All you Need.* Advances in neural information processing systems, pages 5998-6008, 2017.

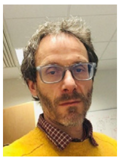
[23] D. H. WOLPERT - *The Lack of a Priori Distinctions Between Learning Algorithms.* Neural computation, 8(7):1341-1390, 1996.

[24] C. XIE, J. WANG, Z. ZHANG, Y. ZHOU, L. XIE, A. YUILLE - *Adversarial Examples for Semantic Segmentation and Object Detection.* The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[25] Z. YUAN, Y. LU, Z. WANG, Y. XUE - *Droid-Sec: Deep Learning in Android Malware Detection*. ACM SIGCOMM Computer Communication Review, volume 44, pages 371-372. ACM, 2014.

Matthieu Nugue obtained his Engineering Degree in Fluid Mechanics and Energy from *Polytech Marseille*, and his Master's degree in two-phase flow from Aix-Marseille University in 2015. From 2015 to 2018 he accomplished his PhD, co-funded by the CNES and ONERA, in the field of image processing in the Rocket Propulsion Modeling Team (MPF) at ONERA. From 2018 to 2019 he worked on computer vision for robotics applications as a member of the innovation team at Audensiel. Since 2020, he has been working with the research and development team at Reminiz on deep learning for video analysis.

Jean-Michel Roche
Research engineer
Head of the "*Manufacturing & non-destructive testing of composite materials*" research team.
Graduated in Aerospace Engineering (*École Centrale de Lyon*)
PhD thesis in Acoustics (ONERA).
He has been working on non-destructive inspection and structural health monitoring of materials and structures for over 10 years.



Guy Le Besnerais graduated from ENSTA in 1989 and obtained his PhD degree from *Université Paris Sud* in 1993. Since 1994 he works at ONERA – The French Aerospace Lab, currently with the Information Processing and System Department (DTIS) in Palaiseau (91) as Research Director. He obtained the "*Habilitation à diriger les recherches*" (HDR) in 2008 and is affiliated to University of Paris-Saclay. His research activities include methodology for solving inverse problems, performance modeling for imaging measurement systems, and embedded vision for robotics applications.

Camille Trottier
Research engineer in the "Manufacturing & non-destructive testing of composite materials" research team.
(2013 - Université du Mans) Graduated in Optics and Non-destructive evaluations.
(2013 - 2016 / EDF) PhD thesis in ultrasound for non-destructive evaluations.
(2016 - 2018 / THALES) Engineer specializing in signal processing and radar algorithms.
(2018 - to date/ ONERA) He has conducted research on the inspection, characterization and monitoring of damage in composite and metallic materials, using non-destructive techniques (thermal & ultrasonic).

Robin Devillers has been a research engineer at ONERA since 2013. He is in charge of experimental measurements associated with solid propellant studies, ranging from material characterization to combustion imaging. He worked previously for 4 years at the National Research Center of Canada in Ottawa, on optical diagnostics for soot emission characterization in gas flares, with measurement campaigns performed in oil refineries in Mexico. He obtained his PhD from *École Centrale Paris* for his work on laser techniques for temperature measurements in engines at IFP. He has a MSc and an engineering degree from *École Centrale Paris*.

Julien Pichillou obtained his Energetics and Propulsion engineering diploma from INSA de Rouen in 2000, and has been a thermal engineer specializing in space launcher applications (thermal insulation development, thermal environment definitions) for 14 years. Since 2015 he has been employed as a solid propulsion engineer in the CNES launcher directorate, working on P120C solid rocket motor development and on thrust oscillations in SRM.

Adrien Chan-Hon-Tong has been a researcher in machine learning at ONERA since 2014 in the IVA team. He has worked from 2014 to 2018 on the development of an object detection algorithm, working on satellite images and/or UAV videos (for military purposes). Since 2018, he has been focusing on safety issues raised by machine learning on critical platforms (autonomous driving, lethal autonomous weapon systems): performance evaluation of operational conditions, robustness to adversarial attack and data poisoning.

Alexandre Boulch received his engineering degree from the *École Polytechnique*, his M.Sc in Computer Vision and Machine Learning from ENS Cachan and his PhD from the Eastern Paris University (UPE) with a thesis focusing on geometric and semantic reconstruction for buildings. His research fields are computational geometry and machine learning for remote sensing data, including Lidar and satellite imagery.

Antoine Hurmane graduated (Engineering Diploma) from the French Institute for Advanced Mechanics (IFMA), and received a Master Degree from Blaise Pascal University (Clermont- Ferrand France) in 2011. He received his PhD in mechanics from the University of Technology of Compiègne in 2015. Since then he has been working at ONERA – The French Aerospace Lab as a research scientist in the Composite Materials and Structures Department. His research includes the development of damage and failure approaches for composite materials, and experimental characterization of damage and failure of composite materials.

E. Colin Koeniguer, G. Le Besnerais, A. Chan Hon Tong, B. Le Saux, A. Boulch, P. Trouvé, R. Caye Daudt, N. Audebert, G. Brigot, P. Godet, B. Le Teurnier, M. Carvalho, J. Castillo-Navarro
(ONERA)

E-mail: elise.koeniguer@onera.fr

DOI: 10.12762/2020.AL15-08

Recent Examples of Deep Learning Contributions for Earth Observation Issues

The purpose of this article is to take stock of the progress made in remote sensing thanks to the recent development of deep learning techniques. This assessment is made by means of a systematic presentation of the various activities carried out at ONERA in remote sensing imagery using deep learning methods. It covers a large part of the observation problems: filtering, object detection, land-use classification, change detection, and biomass estimation. In light of these activities, we highlight the practical challenges of deep learning, mainly physical feature definition and training database construction. Some directions for future research are also given, such as the development and use of dedicated remote sensing platforms, hybrid supervised/unsupervised strategies, and the further exploitation of multimodal/multitemporal data.

Introduction

The rise of open-data makes it possible to promote the use of deep-learning in many areas. Also, many experts around the world are wondering about the progress that deep-learning methods will allow in the coming decades, and seek to predict when the performances of automatic algorithms will exceed those of humans, in several tasks: for translating; writing text, driving vehicles, performing surgical operations, etc. It would therefore seem that the more the field corresponds to a broad need, the more deep-learning allows rapid improvements.

In this context, the field of remote sensing is a much smaller field today in terms of the number of users and needs, than more conventional domains such as language, classical images, or text. However, even though the concerned audience is less extended, remote sensing data processing is undergoing the same revolution as other sectors of big data. The number and the diversity of sensors increase very quickly, at two levels:

- The rise of open data, in particular through data from the European Copernicus observation program, which delivers free images acquired since the end of 2014, and produces a petabyte of data every six months [39].
- The development of commercial-type data and the democratization of satellite systems, with more and more launching constellations of micro or even nano-satellites [15, 11].

This entrance into the big data movement promises exciting developments of deep learning methods for remote sensing images. All the more so since remote sensing data are mostly images, and since the most significant improvements in deep learning were made recently on images in computer vision through the ImageNet dataset [29].

Implementing deep learning methods requires a certain number of choices to be made: which databases and how to access them, which servers or computing power, and which learning architectures for a given function.

First, to successfully implement a deep learning application, it is necessary to have access to massive datasets. These are essential to the performance of artificial intelligence systems and, therefore, highly strategic. The quantity and nature of these data will guide the technological choices. Similarly, if the data are subject to protection constraints, such as military data, they will have to be processed on dedicated servers, which then directs the technological choices.

In any case, the data-processing tool remains the core of a learning method. Many companies compete in this market: big names like Amazon; Google with TensorFlow, Caffe (a project initiated at the University of California at Berkeley), and Torch (widely used and improved by Facebook engineers), can be used for remote sensing images.

However, the functions deduced from the remote sensing images that we are trying to learn are of a very different nature from that of those for which these large classical networks have been developed. For example, we can mention some specificities of remote sensing images compared to the standard images of the web:

- A remote sensing image contains multiple objects, spatially distributed and organized on several pixels. When we deal with remote sensing classification, it is not a label of a single image that we are trying to find, but rather different labels for all of the pixels or group of pixels in the image.
- Remote sensing images do not look like conventional images. Full-wave LIDAR images, for example, correspond to a complete profile describing the vertical structure of the object for each pixel. Radar images or SAR images are subject to particular statistics due to speckle, and can highlight particular physical effects that are invisible in optics (for example, humidity effects). Hyperspectral images contain spectral information and are therefore datacube images. Remote sensing time series are more and more easily accessible, when several acquisitions of the same scene are repeated over time.
- Finally, some functions are specific to this field. This, for example, is the case of the detection of changes between two images, to determine the differences between two acquisitions of the same scene, or the transfer of modalities: we try to move from one type of images to another when the information that they contain is related to the same underlying physical phenomena. Note that these functions are also important in other domains, such as the medical domain, where deep learning methods have been well developed.

ONERA conducts a large number of works in remote sensing imagery, on the one hand, and in artificial intelligence on the other. This dual competence makes it possible to ensure expertise in the use of deep learning for remote sensing applications by identifying the particular difficulties and also the opportunities.

The purpose of this article is twofold:

- First of all, to show, for a certain number of different cases studied at ONERA, how machine learning allows significant improvements in performance on functions based on the use of remote sensing images.
- Then to analyze, for each case, the major difficulties encountered in the implementation of these AI methods.

Compared with other articles summarizing the contributions of deep learning to remote sensing, this article is not intended to be exhaustive on the existing methods in the field, but rather to reveal recent and original results obtained specifically at ONERA, either in terms of methods, or in terms of the application scenario. Most of these methods have to face the challenges of the field of deep learning, such as the unsupervised or weakly-supervised paradigm, in order to prevent the need and the cost of annotation, and the issues arising from the lack of interpretability of such approaches and their perspectives with regard to the Earth observation domain.

The choice and implementation of network architectures depend on both the types of input data and the function to be implemented. Thus, in the second section, we first analyze what the state of the art is regarding the use of deep learning for the main functionalities of remote sensing. Then, the following sections are therefore organized around several fundamental studies, first by the type of function envisaged, concerning a type of specific data.

The applications are organized by hierarchical levels, from the lowest level to the highest level. Five types of functions are envisaged, for which the difficulty of the task and/or the abundance of images has been considered as a significant argument for the use of a learning method.

The successive sections of this article deal with co-registration of heterogeneous images, image quality enhancement (particularly SAR image filtering for radar images subject to speckle noise), land cover classification, vehicule detection, change detection, 3D sensing and estimation. Then concluding remarks are presented.

Related works

The creation of large-scale image databases, such as the pioneering ImageNet [29, 46] published by Stanford University, enabled an impressive shift in the way that image processing is considered. Neural network algorithms could then be applied to images. Indeed, they had made considerable progress in other fields where abundant training data was available. However, their implementation for images also relied on a recent technical advance: using Graphical Processing Units (GPUs) for general programming. Soon, deep learning [33, 38] brought a significant performance gap. Deep initially referred to the depth of the neural networks, which comprised many hidden layers. However, deep also means that the processing function is trained end-to-end, from data to expected result. From a learning point of view, this was considered much more satisfactory than previous processing pipelines designed by experts. In particular, the feature extraction was then trained, and yielded much better features than previous hand-crafted ones. In the following years, image processing underwent tremendous changes. Not only were tasks for which machine learning was already often considered then successfully addressed by deep networks, but traditional analytically-solved tasks became trainable.

Also, can we consider that remote sensing images are always like those of human vision? For traditional optical images, it is legitimate to think so. For other types of sensors, it is less obvious, because:

- Data are sometimes multi-modal,
- Data are geolocalized; they contain geographical maps rather than an object map,
- The time variable is becoming critical,
- In many cases, remote sensing is aimed at estimating geophysical parameters rather than detecting or classifying objects,
- Some images contain physical information that is different from visible information, such as SAR images or full-wave LIDARs.

Using deep learning for remote sensing came a little later than it did for computer vision; nevertheless, today, deep learning is widespread in the field and often establishes a new state of the art. Also, for each type of functionality, it is necessary to analyze what progress has been made in this area.

Up to now, the main remote sensing functions can be categorized as image processing, pixel-based classification or segmentation, target recognition, and scene understanding. A helpful review of state-of-the-art results of deep learning in remote sensing for several applications is given in [54], especially for hyperspectral image analysis for land cover/use classification and anomaly detection, SAR interpretation, high-resolution optical image interpretation, and fusion. In [53], the review focuses more specifically on classification techniques. To date, the auto-encoder (AE), the CNN, the Deep Belief Network (DBN), and Recurrent NN (RNNs) have been the four mainstream DL architectures used in the field of remote sensing. RNN is primarily used for analyzing non-stationary processes, CNN for classification tasks; and DBN or generative AEs for all other tasks, in particular unsupervised ones.

Both states of the art in [54] and [53] confirm that, as in other fields, deep learning is making remote sensing advance, even though progress is recent and further improvements can be expected. The general feeling is that, in upcoming years, we can still expect great advances in remote sensing thanks to deep learning. It also has limitations and raises new challenges: the lack of annotated data, the difficulty in deploying and transferring models under various conditions at global scales, and also the taking into account of sensor physics and purely algorithmic tasks.

Since our first deep learning works [34, 2], we have made progress on several tasks at ONERA, with research on standard tasks, such as classification or object detection, as well as on topics that have been addressed very rarely, such as co-registration, or 3D estimation. We detail below our recent advances in this area over the last 5 years.

Co-registration

Co-registration of heterogeneous images is useful in various remote sensing image fusion applications, since a gain is expected from the synergy of sensors. Relevant applications are numerous, whether for land classification, for agriculture, or forestry applications. Some applications require a pixel precision and, generally, the terrain correction applied for georeferencing is not good enough. The residual bias arises from the impact of the imprecision on orbital or DTM (Digital Terrain Model) parameters during the mapping. The influence of the relief on such registration is non-rigid and, therefore, requires the estimation of a dense motion field.

In remote sensing, deep learning methods for co-registration are not numerous. [41] is a feature-based approach that proposes an architecture derived from a Siamese Neural Network (SNN) trained to select precise and reliable points of correspondence between the two images.

Recently at ONERA, we proposed the investigation of image-based deep learning approaches taking into account all of the pixels of an image. We then proposed the adaptation of PWC-Net, a CNN already developed in Computer Vision, in order to make it efficient for heterogeneous images, such as a couple of SAR/optical images. All of the performances of our tests were compared with a reference algorithm for optical flow developed at ONERA, GeFolki.

A first significant challenge was to constitute the training base. For this purpose, we used the Google Earth Engine (GEE) data platform, able to handle both optical Sentinel-2 (S2) and radar Sentinel-1 (S1) images. We selected georeferenced images assumed to be well co-registered together. Using the platform, we were able to define large footprints common to S1 and S2. For S2 images, we choose dates with the weakest cloud cover. Around the corresponding acquisition dates, the S1 radar images have been filtered temporally to reduce the effects of speckle and increase their signal-to-noise ratio. A systematic coverage of the entire French territory has been established; this is to ensure a representative diversity of all of the landscapes encountered, such as agricultural, city, forest, and mountain areas. Then, we also artificially generated, for each pair, realistic deformations whose amplitude is modulated spatially by the relief, given by the SRTM product downloaded on the same footprint.

We have considered FlowNetS [31] and PWC-Net [48], two Convolutional Neural Networks (CNN) commonly used for optical flow estimation in computer vision, PWC-Net being state of the art.

FlowNetS can take into account any type of input image. Without modifying its architecture, it could be applied successfully to our images. It proved robust for estimating flows. On the contrary, PWC-Net is conceptually defined for two entries of the same type, in particular with a shared encoding between the entries through the application of the same encoder. Such encoding makes no sense for different images. For this reason, we propose a modification of the architecture with two independent encoders. We separate the Siamese contracting parts of PWC-Net into two different contracting parts performing two different operations. Those two contracting parts share the same architecture but can have different weights. Thus, there is a contracting part specialized for SAR images and a second one specialized for optical images. Moreover, we decided to remove the lowest resolution stage of the architecture for both the contracting and the expending parts, since it can estimate mainly large deformations. Finally, we propose to assist the training by simultaneously using three different loss functions exploiting the different combinations of contracting parts that we can use: optics/optics, radar/optics, and radar/radar. We call our new architecture PWC-Net-multimodal.

Results have been tested on a new database, and compared with that of the GeFolki algorithm [13], which is an optical flow method without machine learning. Deep learning architecture performed better, not only on data close to the training data set, but also on data acquired with different sensors, with significantly higher resolution. The expected prediction error lies between 0.7 and 1.1 pixels for the different deep learning architectures and for different scenarios, whereas it lies between 2.3 and 3.4 for GeFolki.

Concerning FlowNet, PWCNet, and PWC-Net-multimodal, we have shown that the results obtained using the three methods are close, with a better result having been achieved with our PWC-Net-multimodal method. Furthermore, the PWC-Net-multimodal method is more robust to train with excellent repeatability, while the original PWC-Net does not converge every time we try to train it with heterogeneous data.

One example of the result obtained by PWC-Net-multimodal is given in Figure 1. The first column in Figure 1 shows a mosaic of the optical and radar images before and after registration. The optical-radar junctions of the mosaic highlight some structures that are shifted before registration, and that our algorithm manages to align well. The middle column represents the norm of the flow and its direction in color. The top image gives the ground truth, and the bottom image gives estimated results. We see that the estimated flow has the right direction but still lacks spatial details. However, the absolute flow errors remain less than 2 pixels, and the relative error remains below 20%.

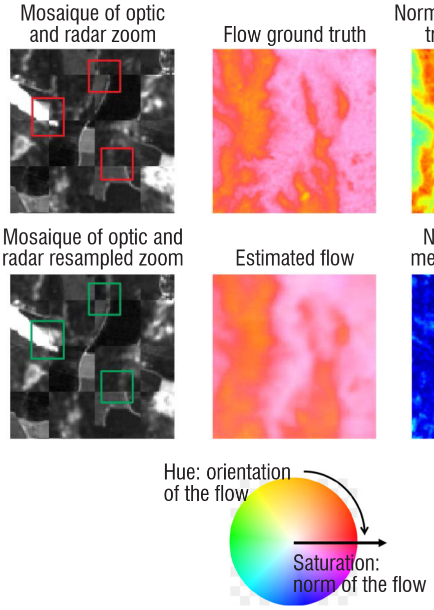


Figure 1 – Mosaic of Sentinel-1 and Sentinel-2 images of the same zone (First column). The ground truth flow and the flow estimated with PWC-Net trained with our dataset (middle column). The quadratic norm of the ground truth flow and the norm of the error of the estimation (last column).

Perspectives

Given that databases with different modalities such as SAR/optical images are still being widely developed, progress is likely to continue thanks to deep learning. Methods should take advantage of the specificity of the different modalities. For example, the different polarizations for radar images can be used together to improve efficiency. We also plan to test and improve robustness to the possible presence of clouds or changes. Finally, we have to tackle the co-registration problems for high-resolution images by considering a new formulation taking into account the 3D aspects, since the resolved 3D elements such as buildings can have different projections, without any bijective relationship between them.

Image quality enhancement

The notion of noise filtering is particularly crucial for radar images because these images have an inherent speckle noise. Many algorithms strive to remedy this noise through the speckle filtering operation. Up until now, all of the algorithms exploited spatial information. Now, as time-series become available, temporal information can also be used.

Speckle filtering methods usually fall into two categories: noise modeling and data-based approaches. The later includes machine learning methods. The amount of available data and the difficulty in modeling generic de-noising methods make the use of deep learning an already efficient solution [52, 50]. However, most proposed solutions rely on supervised methods, and thus on the description of ground truth, which is, in this case, the achieved goal at the filtering output. An essential difficulty is knowing how to describe what is meant by ground truth in this case. Obtaining training datasets by using simulation is one of the possible remedies, but transfer to real SAR data remains a significant challenge.

The originality of the work undertaken at ONERA in this area is the use of time series to avoid having to provide an objective [12]. We propose to use the redundancy of the data in the stack, and to formulate the problem as follows: given two realizations I_1 and I_2 of the same scene, let us learn to predict I_2 from I_1. The transfer function itself performs the filtering of the random part of the signal, keeping only the deterministic part.

We have tested several networks and several loss functions, and we have also compared our results with other spatiotemporal filtering methods, such as BM3D [26] and SAR-BM3D [44]. The best results are achieved using dilated convolutional networks and histogram loss, which is defined by a distance ℓ_2 on the histogram vector of a given pixel $x \in X$ in its neighborhood N_x. Then, the gradients for backpropagation are the differentiation of the previous distance. To scale up to large images, we do not feed the network with the whole image but rather with patches, with possible overlapping to prevent border effects.

The learning phase focused on an SLC Sentinel-1 image stack around Saclay, 20 km south of Paris. Figure 2 presents the results for Valencia, a scene that is not part of the auto-encoding set (red and blue channels for VV and green channel for VH). The function has removed most of the noise, e.g., around the harbor.

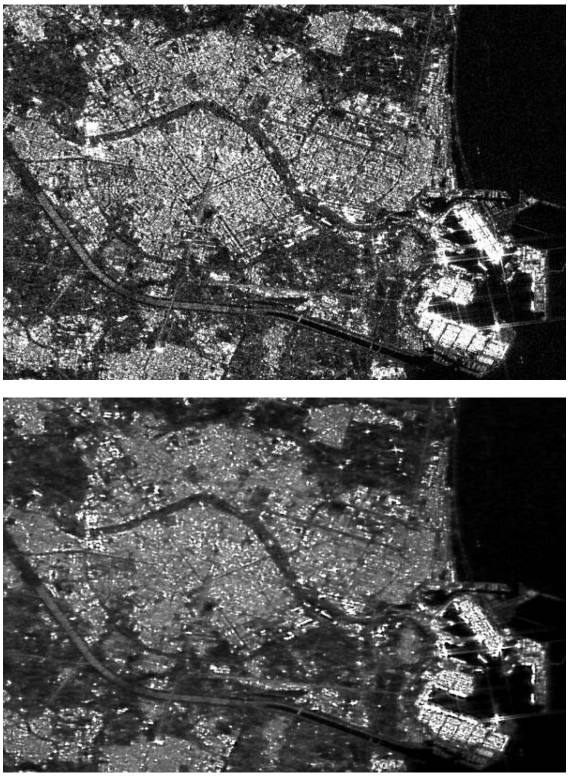


Figure 2 – Filtering results over images of Valencia with a network trained on Paris. Top: original image, bottom filtered image.

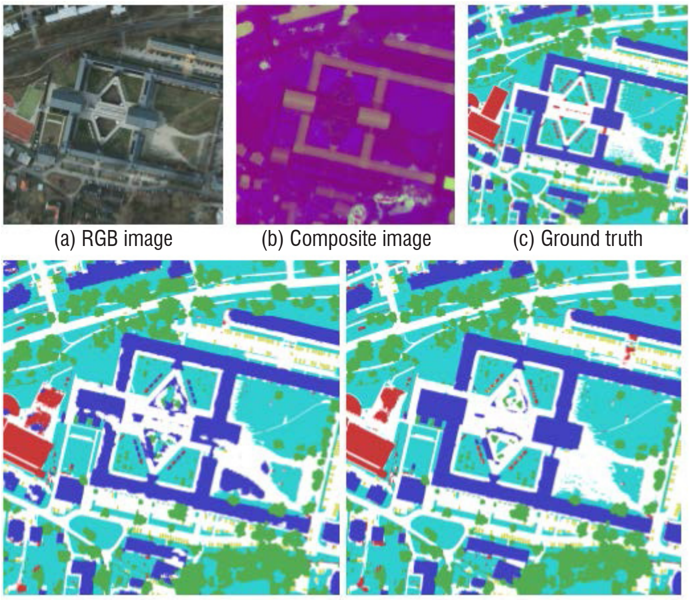


Figure 3 – Semantic segmentation

Although the results are not perfect, the ability to generalize has been demonstrated, since the network has been successfully applied, having been learned on a very different site and for different resolutions.

Perspectives

The next steps are to manage the way in which knowledge of some of the metadata is included, such as polarization, incidence angle, and also the possibility of mixing this unsupervised approach with supervised ones, through the use of image databases for which speckle noise is added artificially.

Land cover classification

Classification tasks in remote sensing benefited the most from the deep learning trend [53, 54, 10]. Although image classification triggered an interest in deep neural networks in computer vision, remote sensing tasks have their peculiarities. They include a focus on pixel classification or semantic segmentation (partly tied to the fact that remote sensing images are very large, while being only small portions of the Earth Surface), the variety of imaging techniques (RGB, multispectral, LIDAR, SAR, etc.), and the variety of potential ground-truths, which lies in the existing maps of all sorts.

The abundance of remote sensing data enables us to extract relevant information through deep learning, for the automatic **semantic mapping** of the Earth from multimodal, aerial and satellite data, in urban or rural environments [40, 16, 3]. Primarily, we aim to automatically map the land cover and land use of large-scale scenes using all available data. Therefore, we have proposed new neural networks[1] to deal with highly heterogeneous multimodal data, such as LIDAR and optical acquisitions [8]. Thanks to a double-flow architecture and to the introduction of a new neuronal block called residual correction [4, 5], our model has improved upon the state of the art achieved with the ISPRS Vaihingen and Potsdam datasets [45] for various classes, such as roads, buildings, vegetation and vehicles, and with another benchmark dataset for buildings all over the world [32]. An example of a segmentation result is shown in Fig. 3 on the Postdam dataset.

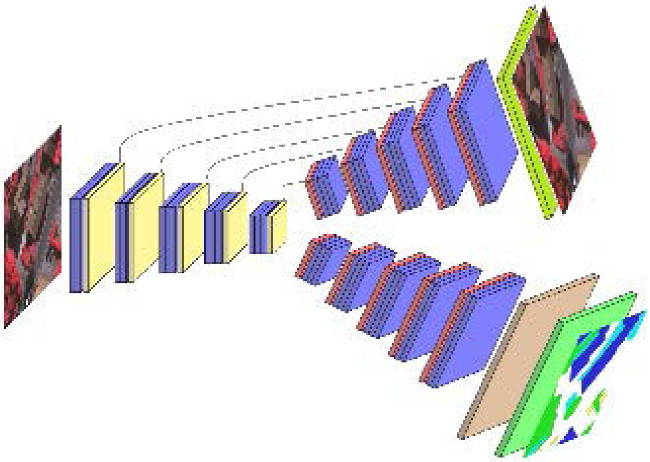


Figure 4 – BerundaNet, a multi-task neural network for semi-supervised semantic segmentation

In the field of **hyperspectral images** (HS), we explored models with various kinds of convolutions more suited to the specific data structure. Indeed, we consider HS images as data cubes rather than rasters of standard images. We have shown that 3D convolutions are more suited to HS data if enough da ta are available [10]. We then released DeepHyperX[2], an open-source toolbox that enables the scientific community to investigate how deep learning tools can participate in particular HS imaging classification problems. Finally, data scarcity is a common issue in vegetation or mineral studies. To enrich the databases used for training algorithms, we have proposed a method for the synthesis of realistic spectra based on generative adversarial models [9].

Geospatial data include huge volumes of ortho-rectified images and maps of different kinds (geographic but also political or theme-specific). It was necessary to find ways to leverage them for training. For instance, we proposed the inclusion of prior knowledge from OpenStreetMap in the learning process, thus showing the ability of neural networks to take advantage of heterogeneous sources of information [6]. We also proposed to encode the spatial shapes and relationships between classes through Distance Transform Regression [1]. The production of thematic maps often comes up against the high-level semantics of the expected classes. For example, in order to find solutions to characterize urban heat islands from the sky, we organized a benchmark for Local Climate Zone (LCZ, [47]) classification, spanning various cities around the world. It showed that although deep networks are efficient for quickly producing averagely-good maps, adding expert knowledge with more standard approaches like boosting or random forests were highly valuable to obtain quality maps [51].

Perspectives

We now seek to benefit from the unexploited, unlabeled data. To this end, we investigate semi-supervised architectures, such as that in Fig. 4, able to learn image characteristics from the unlabeled images available for every location to regularize land-use and land-cover classification (urban fabric, wetlands, forests, fields) [19, 20]. A related topic of interest is weakly-supervised learning, to learn with unreliable ground-truth or classes that are not visually homogeneous [23, 21].

[1] https://github.com/nshaud/DeepNetsForEO

[2] https://github.com/nshaud/DeepHyperX

Vehicle detection

Object detection in an image consists in locating all instances of the object of interest in an image. In this task, the input of the algorithm is an image, and the output is a set of locations. The quality of the algorithm output is measured by comparing the produced set of locations with the ground truth (the known set of locations of the objects). Although this problem seems very understandable, there are many ways to quantify the results, depending on the nature of the requested location. Classically, detection is aimed at placing bounding boxes on vehicles, predicting both location and scale, and ensuring one-to-one matching. This task allows, for example, vehicles to be counted.

In particular experimental settings, typically in a low-resolution context or for hard-to-understand sensors, ground truth can be obtained by monitoring vehicles on the ground, and operators and algorithms are complementary. However, most of the time, humans excel at locating vehicles in images when the resolution is higher than 20 cm per pixel. Indeed, ground truth is usually obtained just by manual inspection of the image. Also, because humans are outstanding at this task, there is no qualitative advantage of using algorithms instead of concentrated operators.

The advantage of a detection algorithm thus lies either in the ability to automate the detection of vehicles in large numbers, or to improve the performance of the detection in rare modalities among all available remote sensing data. This task could bring a real technological breakthrough, notably allowing a better town planning policy and, of course, providing valuable information for intelligence.

Deep learning is especially relevant for both accuracy and scalability. First, the performance achieved by deep learning in vehicle detection is at least as high as the performance achieved by other kinds of algorithms, such as those described in [36, 34]. Typically, given a minimal set of images, designing an *ad hoc* detector to detect vehicles for this specific context is often possible.

However, deep learning is generic and incremental: it is increasingly accurate when fed with more training data. Then, training is just linear in relation to the size of the training database, and both training and testing are very fast on hardware like GPU cards designed for deep learning. In addition, we can use a shared deep learning pipeline for multiple purposes: typically [7] offers a way to achieve detection as post-processing of land cover classification.

To reach its own opinion of the results obtained, ONERA implemented various different architectures on different datasets. In particular, ONERA has developed a manually annotated database of 20,000 vehicles on 20 cm resolution aerial images from the ORTHO HR ® produced by the IGN (National Institute of Geographic and Forest Information) in partnership with local authorities.

The first results of car detection were obtained using the approach described in [7], on these images. Figure 5 illustrates one example of detection results. We have conducted other works on Pléiades datasets at 50 cm resolution, or for aircraft type targets. This way, on highly-resolved images better than 10 cm per pixel and on large datasets, deep learning overrules the state of the art [7, 42] providing performances as high as 94% of F-score for [42] – 86% for [7]. However, today, deep learning is not sufficiently accurate to keep its promises with regard to classical remote sensing images (less than 20 cm per pixel) and, besides, suffers from a lack of large, structured, annotated and free datasets at this resolution.

Perspectives

More than on network architectures, it is on the development of better-constructed large databases that efforts are expected. The simulation of various scenes, including diverse targets under diverse lighting conditions, could also play a role in this context and help to reach that operational quality soon.

Change detection

Change detection is aimed at finding the changes between two co-registered images taken at different times [35]. It is often tackled at the pixel level by semantic segmentation approaches. It is an example of a dense classification problem, where we attempt to assign a label to each pair or sequence of corresponding pixels. Depending on the desired application, the assigned labels may be binary, change or no change, or they may contain semantic information about the changes that have happened, such as deforestation, urban expansion, or water loss.

At ONERA, we have recently achieved state-of-the-art results in change detection using state-of-the-art machine learning techniques. For this purpose, a dataset has been developed to train and benchmark various different change detection algorithms for change detection [22]: the ONERA Sentinel Change Detection dataset (OSCD). It contains several multispectral image pairs extracted from Sentinel-2 acquisitions and manually created binary change labels for all pixels in all image pairs. OSCD has also been released publicly[3] so that scientists all over the world can accurately compare their proposed algorithms quantitatively and together

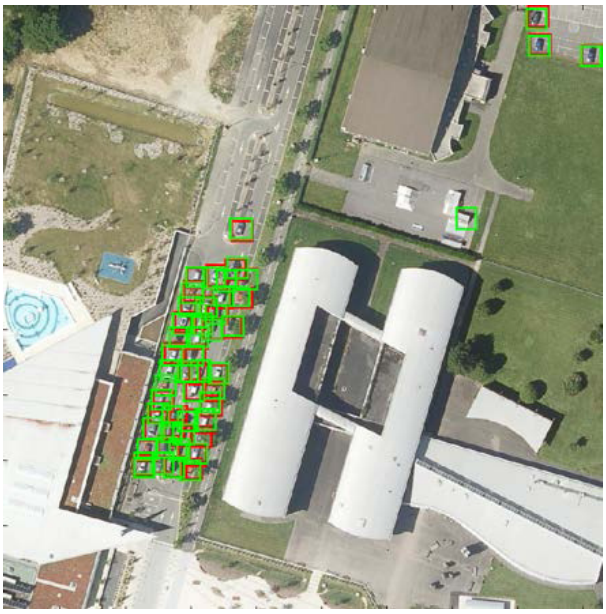
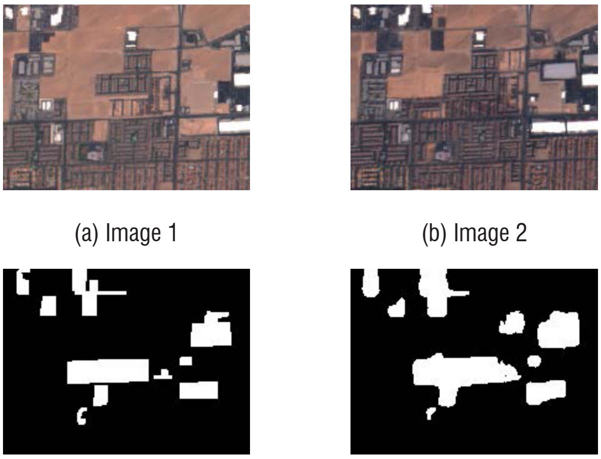


Figure 5 – Detection results obtained using the approach described in [7] implemented through a UNet, applied over aerial photography, from IGN ORTHO HR ®, at 20 cm resolution. Red boxes: detection, Green boxes: Ground Truth.

[3] http://dase.grss-ieee.org/



(a) Image 1

(b) Image 2

(c) Change map

(c) Estimated change map

Figure 6 – Example of a satellite image pair of Las Vegas, true change map and estimated change map using a convolutional neural network

develop ever-improving change detection methods. Example results are given in Figure 6.

The change detection methods proposed in [27] surpass related methods in both accuracy and speed. They are extensions of fully convolutional encoder-decoder networks using skip connections and, most notably, a *Siamese* extension of a traditional encoder-decoder architecture using heuristics specific to the problem of change detection achieved the best results.

More recent works have pushed the boundaries of state-of-the-art change detection methods even further. A new dataset, called High-Resolution Semantic Change Detection (HRSCD) dataset has been generated, and will also be made available to the scientific community. This dataset is more than 3000 times larger than any other change dataset openly available, and it contains high-resolution (50 cm per pixel) images. It contains not only change labels for all pixels, but also land cover information.

This dataset enables research in change detection to go much further. First, multitask learning was carried out using the HRSCD dataset, and we show that simultaneously learning to detect changes and to classify the terrain in the images led to better network performances [28]. Wea kly supervised learning techniques were also proposed to deal with the label noise inherent to automatically generated data [23]. Using a combination of iterative learning, classification filtering, and the newly proposed guided anisotropic diffusion post-processing method, an encoder-decoder fully convolutional neural network was trained to obtain excellent change detection performances.

Recently, given the availability of time-series, we have extended change detection to **activity detection** over a given period. The volume of available data for this specific problem is much smaller than for other remote sensing tasks. Indeed, in the case of optical images where clouds can be an issue, gathering a sequence of cloud-free images is much harder than finding a single image, especially in humid regions. The volume of labeled data for activity detection is also scarcer than those available for other problems, which limits the complexity of the machine learning models that we can use, since more complex models need large amounts of data to avoid overfitting. Since Sentinel-1 (S1) radar data do not suffer from cloud cover, they allow for more accessible collection and processing of stacks of images. We have developed the REACTIV algorithm on this basis: by exploiting the particular statistical properties of the radar images, the algorithm allows us to obtain unmatched change detection performance, superior to that obtained by the previous supervised deep learning approach applied to the same scenario, but on optical Sentinel-2 images [25]. We have evaluated these performances on a set of data from the Saclay plateau, chosen for a large number of construction sites present during the analysis period.

Perspectives

On the one hand, we have developed a very robust change detection algorithm with the exploitation of SAR time-series, but the interpretation of the change remains difficult using this data. On the other hand, thanks to the optical-imagery datasets that we released for the community, deep networks for change detection have emerged [27, 43]. However, a more massive dataset could improve results still further. Therefore, our next efforts will be devoted first to improving database creation by using robust automatic detection on radar images to select change key positions in optical images. Then, we will investigate more sophisticated and accurate deep learning methods for semantic change detection and high-resolution change detection based on time-series.

3D Sensing and Estimation

Obtaining 3D data through imaging is an area where the traditional deep learning used for image processing does not apply directly. However, we have invested in this area by using our expertise in the conception and processing of data from advanced sensors. Two areas of research concern machine learning for 3D: LIDAR and SAR data fusion for 3D forest structure studies; and 3D model estimation from the sky.

The goal of [14] is to predict the structural parameters of forests on a large scale using remote sensing images. LIDAR and polarimetric interferometric SAR sensors are both interesting for estimating forest biomass. However, if LIDARs offer excellent vertical accuracy, they suffer from their lack of spatial coverage. On the other hand, SAR systems have extensive coverage and ground spatial accuracy, but reduced vertical precision. Therefore, the approach is to extend the accuracy of LIDAR full waveforms to a larger area covered by polarimetric and interferometric (PolInSAR) synthetic aperture radar images using machine learning methods.

We proposed in [14] a set of PolInSAR parameters, computed for each pixel, which is likely to have strong correlations with the LIDAR density profiles on forest stands. These features were used as input data to learn a set of forest LIDAR features: the canopy height, the vertical profile type, and the canopy cover. The approach has investigated several methods of machine learning for this purpose:

- Random forest methods for class classification of vegetation profiles.
- Classical SVN methods, then perceptron methods for estimating canopy height, and canopy density.
- Finally, CNN methods were also tested for estimating canopy height.

In the latter case, the results were not better than those obtained with other traditional machine learning methods. The small number of data in the learning base can explain this. Also, the chosen input was not a simple vector, but was transformed into a two-dimensional space and interpreted artificially as an image, even though it does not correspond to real objects but rather to pure mathematical representations used to feed CNN networks.

Nevertheless, perceptrons give very encouraging results. Neural networks give the best performances in terms of RMSE on the estimated tree heights, as represented in Fig. 7, and they are the most faithful on the fidelity of the estimated statistical distributions, as shown in Fig. 8.

This work has also demonstrated the importance of the choice of input descriptors: the performances are better with descriptors judged to have the most physical meaning. This work makes us think that deep learning allows increased performance, but not necessarily based on images, even if we acquired the data in that way.

A second axis consists of scene understanding from standard optical imagery. Additionally to semantics (see Sections 5 and 6), providing the local height, for example, as with Digital Surface Models (DSMs), is useful for many applications, such as urban planning, telecommunications, aviation, and intelligent transport systems. Multi-view stereo [30] was the means of choice to obtain these products, but today deep learning approaches also offer competitive performances [37]. We address this problem by using a Multi-Task Learning (MTL) deep network that estimates both height and semantic maps simultaneously from a single aerial image [18]. Our approach is built on powerful models that we developed previously for depth prediction from a single image taken from the ground [17].

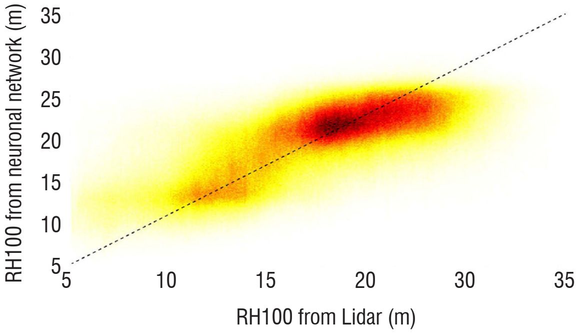


Figure 7 – Tree height estimation precision using neural network

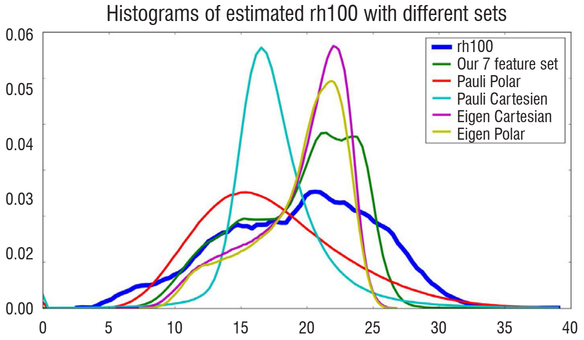


Figure 8 – Tree height estimation using different machine learning algorithms

Precisely, we adapted D3-Net [17] to a multi-task architecture by adding a semantic classification decoder to the original depth estimation one. As shown in Figure 9, the contractive new decoder layers are common to both semantics and height estimation. On the contrary, layers of the decoders are specific for each objective and generate, respectively, as many channels as classes for semantics and one channel for height. We have evaluated each output with a corresponding loss function: we have adopted the absolute error ($L1$) for height regression and the cross-entropy loss (LCE) for semantics evaluation. We have also implemented various mechanisms for multi-task optimization, such as Grad-Norm [24].

Figure 10 shows maps generated from the 2018 Data Fusion Contest (DFC2018 [49]) dataset. In general, the network produces nearly

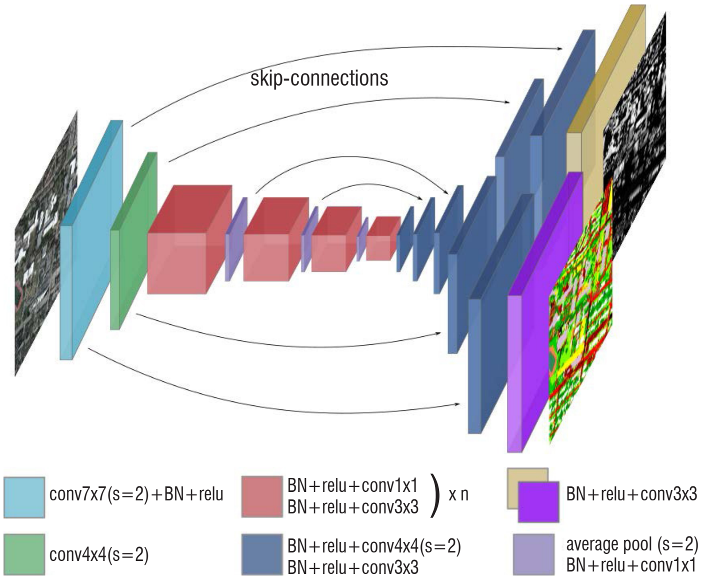


Figure 9 – Architecture of our MTL model for height regression and semantic classification. On the left most layers share parameters between all tasks and on the right most layers are task-specific.

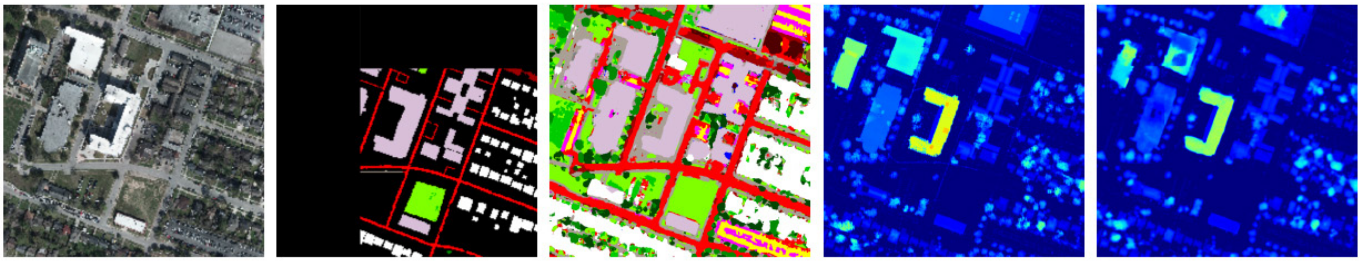


Figure 10 – From left to right, input RGB image, semantic ground-truth and prediction. Black represents no information. Height ground-truth and prediction evaluated for DFC2018 data over Houston.

accurate heights for ground, residential buildings, and vegetation, while some structures are more challenging, like high buildings or stadiums. Indeed, these classes in bird-view images have various shapes, colors, and heights, which make precise estimation difficult. We can note that semantics are detailed, with dense cartography even when the ground truth labeled only a few objects. Following Section 5, we have shown that multi-task learning allows performance to be improved for both tasks, or in other words, that simultaneous height estimation helps classification considerably.

Perspectives

Our works on the prediction of biomass will continue through the scaling-up of this kind of algorithm by using future satellite missions, such as BIOMASS, Tandem-L, or Ni-SAR. Height-estimation deep networks are only one example of a network able to translate one modality into another: predictors of LIDAR-like point-clouds or SAR-processing networks are also on our agenda. To this end, the understanding of the physics behind the sensor benefits from every bit of available information to build a better 3D estimate. More generally, the estimation of the 3D structure with precise levels of detail using a variety of sensors is a crucial step for creating models of the world that enable environmental sustainability or development of smart cities.

Conclusion

This paper provided an overview of the performance gains obtained today in remote sensing through the use of deep learning techniques. It has demonstrated significant gains in the areas closest to those of computer vision: classification and detection of vehicles in optical images. They are probably the most impressive results because more effort has been made, and also because the transfer of techniques from computer vision to remote sensing is more comfortable to do.

For other more exotic applications, such as multitemporal change detection, LIDAR to radar data transfer for biomass estimation, and 3D estimation, deep learning methods are being developed. The first results are encouraging and prove the feasibility of the methods. Thus, today, it seems that the contribution of deep learning no longer needs to be demonstrated. The expected gains in the future relate mainly both to the development of new architectures or new learning schemes and to the way of forming the learning base.

Hence, future expectations will focus on topics that are absent today from computer vision: for example, the taking into account of complex signals such as SAR images, multimodal data, sparse data such as hyperspectral images. Moreover, taking into account the physics knowledge or the part of the signal that is useful for the intended application seems to be essential and seems essential to the quality of the result.

The other issues concern the organization and constitution of databases. This point is a real difficulty. To obtain annotated databases, in the case of the web, we can count on the annotations of millions of users. In remote sensing, this is not the case. Not to mention the difficulties due to data that are confidential and cannot necessarily benefit from these public platforms. The number of data is not necessarily the most limiting factor: in fact, the images have large dimensions of several tens of thousands of pixels, which makes it possible to process them in a large number of smaller images. Also, a promising avenue is to use approaches that mix supervised learning with unsupervised learning.

However, in any case, it is be necessary to rely on the development of dedicated platforms, and their capacity to interface with the data processing cloud type deep learning, as well as to ingest data from various sources of geographic information, to benefit from all of the advances of this sector of machine learning ■

Acknowledgements

The authors thank the ISPRS for making the Vaihingen and Potsdam datasets available and for organizing the semantic labeling challenge. The authors would also like to thank the National Center for Airborne Laser Mapping and the Hyperspectral Image Analysis Laboratory at the University of Houston for acquiring and providing the DFC2018 data used in this study, and the IEEE GRSS Image Analysis and Data Fusion Technical Committee.

References

[1] N. AUDEBERT, A. BOULCH, B. LE SAUX, S. LEFÈVRE - *Distance Transform Regression for Spatially-Aware Deep Semantic Segmentation*. Computer Vision and Image Understanding, 189:102809, 2019.

[2] N. AUDEBERT, A. BOULCH, H. RANDRIANARIVO, B. LE SAUX, S. LEFÈVRE, R. MARLET - *Deep Learning for Urban Remote Sensing.* Joint Urban Remote Sensing Event (JURSE). IEEE, March 2017.

[3] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - *How Useful is Region-Based Classification of Remote Sensing Images in a Deep Learning Framework?* International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2016.

[4] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - *Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks.* Asian Conference on Computer Vision (ACCV). IEEE, November 2016.

[5] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - *Fusion of Heterogeneous Data in Convolutional Networks for Urban Semantic Labeling.* Joint Urban Remote Sensing Event (JURSE). IEEE, March 2017.

[6] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - *Joint Learning from Earth Observation and Openstreetmap Data to get Faster Better Semantic Maps.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops / EarthVision, July 2017.

[7] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - *Segment-Before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images.* Remote Sensing, 9(4):368, 2017.

[8] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - *Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks.* ISPRS Journal of Photogrammetry and Remote Sensing, 140:20 – 32, 2018.

[9] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - Generative Adversarial Networks for Realistic Synthesis of Hyper-Spectral Samples. International Geo- science and Remote Sensing Symposium (IGARSS). IEEE, 2018.

[10] N. AUDEBERT, B. LE SAUX, S. LEFÈVRE - Deep Learning for Classification of Hyperspectral Data: A Comparative Review. IEEE Geoscience Remote Sensing Magazine, 7(1), 2019.

[11] W. BLACKWELL, G. ALLEN, C. GALBRAITH, T. HANCOCK, R. LESLIE, I. OSARETIN, L. RETHERFORD, M. SCARITO, C. SEMISCH, M. SHIELDS, *et al* - *Nanosatellites for Earth Environmental Monitoring: the MicroMAS Project.* Microwave Radiometry and Remote Sens- ing of the Environment (MicroRad), pages 1-4. IEEE, 2012.

[12] A. BOULCH, P. TROUVÉ, E. COLIN-KOENIGUER, F. JANEZ, B. LE SAUX - *Learning Speckle Suppression in SAR Images without Ground Truth: Application to Sentinel-1 Time-Series.* International Geoscience and Remote Sensing Symposium (IGARSS), pages 2366-2369. IEEE, 2018.

[13] G. BRIGOT, E. COLIN-KOENIGUER, A. PLYER, F. JANEZ - *Adaptation and Evaluation of an Optical Flow Method Applied to Coregistration of Forest Remote Sensing Images.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9(7):2923-2939, 2016.

[14] G. BRIGOT, M. SIMARD, E. COLIN-KOENIGUER, A. BOULCH - *Retrieval of Forest Vertical Structure from Polinsar Data by Machine Learning Using Lidar-Derived Features.* Remote Sensing, 11(4):381, 2019.

[15] E. BUCHEN, D. DE-PASQUALE - *2014 Nano/Microsatellite Market Assessment.* SpaceWorks Enterprises, 12, 2014.

[16] M. CAMPOS-TABERNER, A. ROMERO-SORIANO, C. GATTA, G. CAMPS-VALLS, A. LAGRANGE, B. LE SAUX, A. BEAUPÈRE, A. BOULCH, A. CHAN-HON-TONG, S. HERBIN, H. RANDRIANARIVO, M. FERECATU, M. SHIMONI, G. MOSER, D. TUIA - *Processing of Extremely High-Resolution LIDAR and RGB Sata: Outcome of the 2015 IEEE GRSS Data Fusion Contest – Part a: 2-D Contest.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9(12):5547–5559, 2016.

[17] M. CARVALHO, B. LE SAUX, P. TROUVÉ-PELOUX, A. ALMANSA, F. CHAMPAGNAT - *On Regression Losses for Deep Depth Estimation*. International Conference on Image Process- ing (ICIP), pages 2915-2919. IEEE, 2018.

[18] M. CARVALHO, B. LE SAUX, P. TROUVÉ-PELOUX, A. ALMANSA, F. CHAMPAGNAT - *Multi-Task Learning of Height and Semantics from Aerial Images.* IEEE Geosci. and Remote Sensing Letters, 2019.

[19] J. CASTILLO, B. LE SAUX, A. BOULCH, S. LEFÈVRE - *Réseaux de neurones semi-supervisés pour la segmentation sémantique en télédétection.* Colloque GRETSI, 2019.

[20] J. CASTILLO-NAVARRO, N. AUDEBERT, A. BOULCH, B. LE SAUX, S. LEFÈVRE - *What Data are Needed for Deep Learning in Earth Observation?* Joint Urban Remote Sensing Event (JURSE). IEEE, May 2019.

[21] R. CAYE DAUDT, A. CHAN-HON-TONG, B. LE SAUX, A. BOULCH - *Learning to Understand Earth-Observation Images with Weak and Unreliable Groundtruth.* International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2019.

[22] R. CAYE DAUDT, B. LE SAUX, A. BOULCH, Y. GOUSSEAU - *Urban Change Detection for Multispectral Earth Observation using Convolutional Neural Networks.* International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2018.

[23] R. CAYE DAUDT, B. LE SAUX, A. BOULCH, Y. GOUSSEAU - *Guided Anisotropic Diffusion and Iterative Learning for Weakly Supervised Change Detection.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops / EarthVision, June 2019.

[24] Z. CHEN, V. BADRINARAYANAN, C.-Y. LEE, A. RABINOVICH - *Gradnorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.* International Conference on Machine Learning (ICML), 2018.

[25] E. COLIN KOENIGUER, J.-M. NICOLAS - *Change Detection in SAR Time-Series Based on the Coefficient of Variation*. arXiv preprint arXiv:1904.11335, 2019.

[26] K. DABOV, A. FOI, V. KATKOVNIK, K. EGIAZARIAN - *Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering*. IEEE Transactions on Image Processing, 16(8):2080-2095, 2007.

[27] R. CAYE DAUDT, B. LE SAUX, A. BOULCH - *Fully Convolutional Siamese Networks for Change Detection*. International Conference on Image Processing (ICIP). IEEE, 2018.

[28] R. CAYE DAUDT, B. LE SAUX, A. BOULCH, Y. GOUSSEAU - *Multitask Learning for Large-Scale Semantic Change Detection*. Computer Vision and Image Understanding, 187:102783, 2019.

[29] J. DENG, W. DONG, R. SOCHER, L.-J. LI, K. LI, L. FEI-FEI - *ImageNet: A Large-Scale Hierarchical Image Database.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248-255. IEEE, 2009.

[30] G. FACCIOLO, C. DE FRANCHIS, E. MEINHARDT-LLOPIS - *Automatic 3D Reconstruction from Multi-Date Satellite Images.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops / EarthVision, 2015.

[31] P. FISCHER, A. DOSOVITSKIY, E. ILG, P. HÄUSSER, C. HAZIRBAŞ, V. GOLKOV, P. VAN DER SMAGT, D. CREMERS, T. BROX - *FlowNet: Learning Optical Flow with Convolutional Networks*. International Conference on Computer Vision (ICCV). IEEE, December 2015.

[32] B. HUANG, K. LU, N. AUDEBERT, A. KHALEL, Y. TARABALKA, J. MALOF, A. BOULCH, B. LE SAUX, L. COLLINS, K. BRADBURY, *et al.* - *Large-Scale Semantic Classification: Outcome of the First Year of Inria Aerial Image Labeling Benchmark.* International Geoscience and Remote Sensing Symposium (IGARSS), pages 6947-6950. IEEE, 2018.

[33] A. KRIZHEVSKY, I. SUTSKEVER, G. E. HINTON - *Imagenet Classification with Deep Convolutional Neural Networks.* Advances in Neural Information Processing Systems, pages 1097-1105, December 2012.

[34] A. LAGRANGE, B. LE SAUX, A. BEAUPERE, A. BOULCH, A. CHAN-HON-TONG, S. HERBIN, H. RANDRIANARIVO, M. FERECATU - *Benchmarking Classification of Earth-Observation Data: from Learning Explicit Features to Convolutional Networks.* International Geoscience and Remote Sensing Symposium (IGARSS), pages 4173-4176. IEEE, July 2015.

[35] B. LE SAUX, H. RANDRIANARIVO - *Urban Change Detection in SAR Images by Interactive Learning.* International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, July 2013.

[36] B. LE SAUX, M. SANFOURCHE - *Robust Vehicle Categorization from Aerial Images by 3D-Template Matching and Multiple Classifier System.* IEEE Int. Symposium on Image and Signal Processing and Analysis (ISPA), September 2011.

[37] B. LE SAUX, N. YOKOYA, R. HANSCH, M. BROWN, G. HAGER - *2019 Data Fusion Contest [Technical Committees]*. IEEE Geoscience and Remote Sensing Magazine, 7(1):103-105, 2019.

[38] Y. LECUN, Y. BENGIO, G. HINTON - Deep Learning. Nature, 521(7553):436-444, May 2015.

[39] S. LI, S. DRAGICEVIC, F. A. CASTRO, M. SESTER, S. WINTER, A. COLTEKIN, C. PETTIT, B. JIANG, J. HAWORTH, A. STEIN, *et al* - *Geospatial Big Data Handling Theory and Methods: a Review and Research Challenges.* ISPRS Journal of Photogrammetry and Remote Sensing, 115:119-133, 2016.

[40] D. MARMANIS, J. D. WEGNER, S. GALLIANI, K. SCHINDLER, M. DATCU, U. STILLA - *Semantic Segmentation of Aerial Images with an Ensemble of CNNs.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 3:473, 2016.

[41] N. MERKLE, W. LUO, S. AUER, R. MÜLLER, R. URTASUN - *Exploiting Deep Matching and SAR Data for the Geo-Localization Accuracy Improvement of Optical Satellite Images.* Remote Sensing, 9(6):586, 2017.

[42] T. N. MUNDHENK, G. KONJEVOD, W. A. SAKLA, K. BOAKYE - *A3 Large Contextual Dataset for Classification, Detection and Counting of Cars With Deep Learning.* European Conference on Computer Vision (ECCV), pages 785-800. IEEE/CVF, 2016.

[43] M. PAPADOMANOLAKI, S. VERMA, M. VAKALOPOULOU, S. GUPTA, K. KARANTZALOS - *Detecting Urban Changes with Recurrent Neural Networks from Multitemporal Sentinel-2 Data.* International Geoscience and Remote Sensing Symposium (IGARSS), pages 214-217. IEEE, 2019.

[44] S. PARRILLI, M. PODERICO, C. V. ANGELINO, L. VERDOLIVA - *A Nonlocal SAR Image Denoising Algorithm Based on LLMMSE Wavelet Shrinkage.* IEEE Transactions on Geoscience and Remote Sensing, 50(2):606-616, 2011.

[45] F. ROTTENSTEINER, G. SOHN, J. JUNG, M. GERKE, C. BAILLARD, S. BENITEZ, U. BREITKOPF - *The ISPRS Benchmark on Urban Object Classification and 3D Building Reconstruction.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, I-3(1):293-298, 2012.

[46] O. RUSSAKOVSKY, J. DENG, H. SU, J. KRAUSE, S. SATHEESH, S. MA, Z. HUANG, A. KARPATHY, A. KHOSLA, M. BERNSTEIN, *et al.* - *ImageNet Large Scale Visual Recognition Challenge*. International journal of computer vision, 115(3):211-252, 2015.

[47] I. D. STEWART, T. R. OKE - *Local Climate Zones for Urban Temperature Studies.* Bulletin of the American Meteorological Society, 93(12):1879-1900, 2012.

[48] D. SUN, X. YANG, M.-Y. LIU, J. KAUTZ - *PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8934-8943, 2018.

[49] Y. XU, B. DU, L. ZHANG, D. CERRA, M. PATO, E. CARMONA, S. PRASAD, N. YOKOYA, R. HÄNSCH, B. LE SAUX - *Advanced Multi-Sensor Optical Remote Sensing for Urban Land Use and Land Cover Classification: Outcome of the 2018 IEEE GRSS Data Fusion Contest.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, pages 1-16, 2019.

[50] D. YANG, J. SUN - *BM3D-Net: a Convolutional Neural Network for Transform-Domain Collaborative Filtering.* IEEE Signal Processing Letters, 25(1):55-59, 2017.

[51] N. YOKOYA, P. GHAMISI, J. XIA, S. SUKHANOV, R. HEREMANS, C. DEBES, B. BECHTEL, B. LE SAUX, G. MOSER, D. TUIA - *Open Data for Global Multimodal Land Use Classification: Outcome of the 2017 IEEE GRSS Data Fusion Contest.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(5):1363-1377, May 2018.

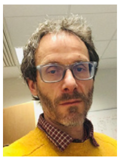
[52] K. ZHANG, W. ZUO, S. GU, L. ZHANG - *Learning Deep CNN Denoiser Prior for Image Restoration.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3929-3938, 2017.

[53] L. ZHANG, L. ZHANG, B. DU - *Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art.* IEEE Geoscience and Remote Sensing Magazine, 4(2):22-40, 2016.

[54] X. XIANG ZHU, D. TUIA, L. MOU, G.-S. XIA, L. ZHANG, F. XU, F. FRAUNDORFER - *Deep Learning in Remote Sensing: a Comprehensive Review and List of Resources*. IEEE Geoscience and Remote Sensing Magazine, 5(4):8-36, 2017.

AUTHOR

Elise Colin Koeniguer graduated from Supelec, and received her M.Sc. in nuclear physics and particle physics in 2002, and her Ph.D. in 2005 from the University of Paris 6, with a thesis focusing on polarimetric and interferometric radar images. She obtained the degree of Habilitation à Diriger les Recherches (HDR) in 2014 from Paris-Saclay University. She is a Research Engineer at Onera. Her research fields include image processing and computer vision for remote sensing images, with applications to forest and urban areas, as well as polarimetry in the optical and microwave domains.



Guy Le Besnerais graduated from ENSTA in 1989 and obtained his PhD degree from *Université Paris Sud* in 1993. Since 1994 he has been working at ONERA, The French Aerospace Lab, currently in the Information Processing and Systems Department (DTIS) in Palaiseau (91) as Research Director. He obtained the "*Habilitation à Diriger les Recherches*" (HDR) in 2008 and is affiliated to Paris-Saclay University. His research activities include methodology for solving inverse problems, performance modeling for imaging measurement systems, and embedded vision for robotics applications.

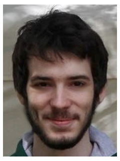
Adrien Chan-Hon-Tong received his M.Sc. from the École Polytechnique in 2011 and his Ph.D. from UPMC in 2014. His research field at the ONERA is computer vision applied to remote sensing, focusing on safety issues such as adversarial attacks, data poisoning, and performance in an operational context.

Bertrand Le Saux (bls@ieee.org) received his M. Eng. in 1999 and his M.Sc. in 1999, both from the INP Grenoble, and his Ph.D. from the University of Versailles/Inria Rocquencourt in 2003. He obtained his *Habilitation à Diriger les Recherches* in 2019 from Paris-Saclay University and is currently a research scientist in the Information Processing and Systems Department at ONERA. His research interests focus on developing machine learning and deep learning methods for remote sensing, (flying) robotics and 3D vision. He teaches machine learning, image processing, and computer vision at the *Institut d'Optique* Graduate School and ENSTA ParisTech. From 2015 to 2019, he was co-chair and later chair of the Image Analysis and Data Fusion Technical Committee of the IEEE GRSS. Since 2019, he has been an associate editor of the IEEE Geoscience and Remote Sensing Letters. He has also been a research fellow at the CNR Pisa (2004), the University of Bern (2005), and the ENS Cachan (2006-2007).

Alexandre Boulch received his M. Eng. from the École Polytechnique, his M.Sc. From the ENS Cachan, and his Ph.D. from *Université Paris Sud*). He received a Ph.D. in geometric and semantic reconstruction for buildings from the ENPC in 2014. For five years, he worked on remote sensing and computational geometry at ONERA. His research included machine learning for semantic segmentation using various data, such as Lidar and satellite imagery. He is currently a research scientist working with Valeo.ai, a research team dedicated to autonomous and assisted vehicles. He has mainly worked on 3D perception and scene analysis.

Pauline Trouvé graduated from the *Institut d'Optique* Graduate School in 2009 and obtained her Ph.D. degree in signal and image processing from the *École Centrale de Nantes* in 2012. Since 2012 she has been working in the Information Processing and Modeling Department at ONERA, the French Aerospace Lab. Her research field at ONERA concerns image processing (deblurring and denoising), optical design, and performance modeling for imaging systems. In particular, she has been working on an unconventional imaging system design using a joint digital/optical performance model. Her work is applied in computational photography and computer vision, especially in the field of 3D applications.

Rodrigo Caye Daudt received his degree in Electrical Engineering from the *Universidade de Brasília* in 2015, and his joint master's degree in Computer Vision and Robotics from the *Université de Bourgogne*, the *Universitat de Girona*, and Heriot-Watt University in 2017. Since 2017 he has been a Ph.D. student at Télécom Paris' Information Processing and Communication Laboratory (LTCI) and in ONERA's Information Processing and Systems Department (DTIS). His research studies the application of computer vision and machine learning methods to multitemporal remote sensing images.



Nicolas Audebert received his M. Eng. degree from Supélec (2015), his M.Sc. in Computer Science from *Université Paris-Sud* (2015), and his Ph.D. from *Université Bretange-Sud* (2018) in the field of deep learning for multimodal remote sensing. He is currently an assistant professor at the CNAM in the Vertigo team. His research interests focus on machine learning for image understanding, Earth observation, and data fusion.

Guillaume Brigot received his M. Eng. degree from Supélec (2014), his M.Sc. in Electromagnetism and Communication from *Université Paris-Sud* (2015), and his Ph.D. in Information Sciences from Paris-Saclay University in 2017, in the fields of machine learning and fusion of LIDAR and RADAR. He is currently an image processing engineer at MBDA.

Pierre Godet graduated from the *Institut d'Optique* Graduate School and received his Master's degree from *Université Paris-Saclay* in the field of signal and image processing. He is currently a Ph.D. student at ONERA in the Department of Information Processing and Systems, in the field of learning-based approaches for multi-frame optical flow estimation.

Benjamin Le Teurnier received his engineering degree from the *Institut d'Optique* Graduate School and his Master's degree from *Université Paris-Saclay* in the field of signal and image processing in 2019. He started a Ph.D. at the *Laboratoire Charles-Fabry* of *Université Paris-Saclay* the same year, in the field of polarimetric imaging.

Marcela Carvalho received her engineering diploma in Mechatronics in 2014 from the *Universidade de Brasilia*, followed by a Master diploma in Robotics in 2015 from the *Université de Montpellier*, and a Ph.D. degree in 2009 in the field of deep learning from Defocus. Her research activities concern robotics and computer vision. During her undergraduate studies, she participated in the creation of the Droid Team, a robotics team that participates in Brazilian national and International intelligent robotics competitions, and co-founded the Lua Startup in 2016.

Javiera Castillo Navarro holds an engineering degree from *École Centrale Paris* and from the *Universidad de Chile* (2017), and she received her M.Sc. in Applied Mathematics from the *Universidad de Chile* (2017) and her M.Sc. in Data Science from *Université Paris-Saclay* (2018). She is currently a Ph.D. student at ONERA, working on deep learning for Earth observation.

T. Mylvaganam
(Department of Aeronautics,
Imperial College London, UK)

H. Piet-Lahanier
(ONERA)

E-mail: t.mylvaganam@imperial.ac.uk

DOI: 10.12762/2020.AL15-09

Robust consensus-seeking via a multi-player nonzero-sum differential game

Considering a class of linear multi-agent systems, we study the problem of consensus-seeking in the presence of an exogenous signal, possibly representing a disturbance. We formulate the *robust consensus-seeking* problem as a nonzero-sum differential game. Exact solutions are characterized by a system of coupled differential Riccati equations (in the finite-horizon case) or algebraic Riccati equations (in the infinite-horizon case), whereas approximate solutions are characterized in terms of coupled matrix *inequalites*. Simulations for two examples are presented to illustrate the resulting performances: one example concerns consensus among agents described by single-integrator dynamics and the other concerns formation flight of unmanned aerial vehicles.

Introduction

Multi-agent systems have gained interest for a wide range of applications including, but not limited to, robotics (see, e.g. [11] and references therein), opinion dynamics (see, e.g. [6] and references therein) and power systems (see, for instance, [12], [22]). Notably, in part due to recent technological advances related to unmanned aerial vehicles (UAVs) and small satellites, multi-agent systems play a major role in a large variety of aerospace applications. For instance, the control of relative distances and orientations between multiple spacecraft to achieve a desired formation is considered in [17]. The spacecraft formation-following problem is also considered in [20], where a graph theoretic formulation of the leader-following approach (introduced in [39]) is provided and solved by means of linear matrix inequalities.

Consensus-seeking for multi-agent systems describes problems in which agents are required to "reach an agreement" on a certain value and is a particularly active research domain, see e.g. [32]. Consensus control aims at driving the states of all agents to reach a common value and plays a major role in various applications, such as formation flight, cooperation in networks, and fault detection and identification (see, for instance, [27], [32], and [34]). In aerospace applications, this topic is of particular interest given that several important problems, such as synchronization and formation control, can be formulated as (dynamic) consensus problems, as seen for instance in [26], [33], [41], [42], [43]. The performances of any consensus protocol are basically sensitive to the presence of persistent perturbations or potential information failures. Various works, such as [31] and [40], have been dedicated to increasing resilience of the controlled systems against various perturbation sources.

Many of the aforementioned consensus-based approaches address the issue of determining control actions for individual agents in a *distributed* manner (e.g., based on neighbor-to-neighbor communication). However, important aspects such as robustness and/or optimality are often recognized, but not addressed (see, e.g. [38]). In this paper we consider the problem of *robust consensus-seeking*, namely the problem of seeking consensus among members of a multi-agent system in the presence of disturbances. In particular, we consider a class of multi-agent systems, described by linear dynamics, and study the problem of consensus control in the presence of an exogenous input, representing a disturbance. The main contribution of this paper is the formulation of the robust consensus-seeking problem as a nonzero-sum differential game with *multiple* players – a formulation which, differently from most existing results concerning consensus problems, enables the consideration of scenarios in which agents are influenced by uncertain and unmodeled perturbations. As will be demonstrated in this paper, game theory provides a convenient framework to evaluate the discrepancy resulting from antagonist environments, and to define a reactive control that provides a suitable compromise between performance and robustness. The motivations of the game-theoretic formulation are twofold. Firstly, since the game theoretic framework essentially models strategic decision making, it allows for the elegant characterization of possibly conflicting goals (such as, for instance, robustness and optimality). Secondly, recent developments in the control engineering field indicate that game theory can serve as a useful tool to systematically design distributed controllers. Although promising preliminary results are available (see, e.g. [8], [9], [16], [21]), the issue of distributed control design is not addressed in this paper, since game theory-based approaches to distributed control design have yet to be fully developed. Once the results are more mature, the formulation of the consensus-seeking problem as a nonzero-sum differential game provided in this paper can be more readily integrated with game theory-based methods for distributed control design in the future. Moreover, by capturing the performance – in terms of optimality *and* robustness – of the closed-loop system in the absence of communication constraints, the results presented herein may constitute a benchmark for new distributed control methods.

Concerning robustness, it is well-known that H_∞ control can be considered as a two-player zero-sum differential game (see e.g. [4]). Concerning robustness *and* optimality on the other hand, mixed H_2/H_∞ control cannot be described using the same framework due to the inherent trade-off between the two objectives. To reflect the presence of this trade-off, in [18] the classical mixed H_2/H_∞ control problem has been formulated as a two-player *nonzero-sum* differential game. Whereas linear systems are considered in [18], the nonlinear counterparts of the differential game formulation of mixed H_2/H_∞ control have been explored in [19], [25].

Differently from the framework considered in [18], where the control problem involves a *single* optimization criterion and a single robustness criterion, herein we consider a setting with *multiple* optimization criteria. This formulation is adopted to reflect practical scenarios in which each agent has an *individual* objective (e.g., to reach a particular position relative to neighbors in an optimal manner), while it is desired for the system, as a whole, to satisfy a certain robustness property. More precisely, the overall multi-agent system is represented by a dynamical system with several inputs – one for each agent – and each agent is associated with an individual cost functional designed to encourage consensus with neighboring agents in the presence of a disturbance.

The remainder of the paper is organized as follows. The robust consensus-seeking problem is defined and formulated as a nonzero-sum differential game in § "Robust consensus-seeking". Exact solutions of the differential game are characterized, both in the finite-horizon and infinite-horizon cases, in terms of coupled differential Riccati equations and coupled algebraic Riccati equations (AREs), respectively, in § "Exact solutions ". Noting that solutions to the coupled AREs, which arise in the context of infinite-horizon differential games may, in general, be difficult to obtain, approximate solutions to the differential game are characterized by means of matrix inequalities (instead of the AREs) in § "Approximate solutions". Simulations corresponding to two examples are presented in § "Simulations" to illustrate the performances of the resulting controllers. One example, presented in § "Consensus at the origin", concerns consensus-seeking among agents described by single-integrator dynamics. The second example concerns a problem of formation flight of UAVs and is presented in § "Application to UAV formation flight". Finally, some concluding remarks are provided.

Notation

Standard notation is adopted throughout this paper. $\mathbb{R}$ denotes the set of real numbers, whereas $\mathbb{C}$ denotes the set of complex numbers and $\mathbb{C}^-$ denotes the open left-half complex plane. Given a square matrix $M \in \mathbb{R}^{n\times n}$, its spectrum is denoted by $\sigma(M)$. The identity matrix is denoted by I. Given a vector $v \in \mathbb{R}^n$, its Euclidean norm is denoted by $\| v \|$.

Robust consensus-seeking: A differential game formulation

Consider a set of $N > 1$ agents, where the dynamics of each agent i, $i = 1,\dots,N$, is described by

$$\dot{x}_i = A_i x_i + B_i u_i + B_{N+1}\omega \quad (1)$$

where $x_i(t) \in \mathbb{R}^n$ is the state vector of Agent i, $u_i(t) \in \mathbb{R}^m$ is its control input, $\omega \in \mathbb{R}^r$ is an exogenous input representing a disturbance or perturbation common to all agents, and $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times m}$, for $i = 1,\dots,N$, and $B_{i+1} \in \mathbb{R}^{n\times r}$ are constant matrices.

The individual states x_i, $i = 1,\dots,N$, can be combined (in a manner to be specified) to form a *global state* $X \in \mathbb{R}^{Nn}$ with the *global system* described by linear dynamics of the form

$$\dot{X} = A^g X + B_1^g u_1 + \ldots + B_N^g u_N + B_{N+1}^g \omega \quad (2)$$

where the matrices A^g and B_i^g, $i = 1,\dots,N$, are specified according to the definition of the global state. The global state could, for instance, be defined as the simple aggregate of all individual states (as considered in § "Consensus at the origin") or in terms of an error variable, for example, in terms of relative differences between neighboring agents.

In this paper we consider the case in which each agent seeks to reach a consensus with its neighboring agents subjected to the disturbance ω. The connectivity between agents is described by a directed graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V} = \{1,\dots,N\}$ is the set of vertices and $\mathcal{E}$ is the edge set. Each vertex corresponds to an agent and, if $(j,i) \in \mathcal{E}$, Agent j is said to be a neighbor of Agent i. Since we consider directed graphs, $(j,i) \in \mathcal{E}$ does not necessarily imply $(i,j) \in \mathcal{E}$. We assume that a connection between two agents, *i.e.,* $(j,i) \in \mathcal{E}$, implies that the i-th Agent has (through measurements or some form of communication) access to the state x_j of the j-th Agent. Let $\mathcal{N}_i$ be the set of neighbors of Agent i and let N_i denote the cardinality of the set, *i.e.,* $N_i = |\mathcal{N}_i|$. In the following we adopt the convention that Agent i is included in its own neighbor set only if explicitly stated, *i.e,* if $(i,i) \in \mathcal{E}$, for $i = 1,\dots,N$. Let $e_i(t)$ denote the consensus error of Agent i, namely e_i is given by

$$e_i = \left(x_i - \frac{1}{N_i} \sum_{j\in\mathcal{N}_i} x_j \right) \quad (3)$$

and let $\| Z(t) \|^2$ denote a penalty variable given by

$$\|Z\|^2 = \sum_{i=1}^{N} \|e_i\|^2 .$$

In the remainder of this paper, u_i denotes a feedback control strategy, namely $u_i = u_i\left(x(t)\right)$, for $i = 1,\ldots,N$. The robust consensus-seeking problem is defined as follows.

Problem 1: robust consensus-seeking
Consider a system described by the dynamics (2). Determine feedback control laws u_i^*, $i = 1,\ldots,N$, such that the following conditions hold.

(C1) When the worst-case disturbance $\omega^*\left(x(t)\right)$ and the control actions u_j^*, $j = 1,\ldots,N$, $j \neq i$, are applied, u_i^* is such that the state is regulated to minimize the cost functional

$$J_i\left(X(0), u_1^*,\ldots,u_{i-1}^*, u_i, u_{i+1}^*,\ldots,u_N^*,\omega^*\right) = \int_0^T \left(\|e_i\|^2 + \|u_i\|^2\right)\mathrm{d}t \tag{4}$$

where the first term $\|e_i\|^2$ represents a running cost and the second term represents a penalty on the control effort of the i-th agent, $i = 1,\ldots,N$;

(C1) The disturbance is attenuated by γ with respect to the mean-square error

$$\left(\|Z\|^2 + \sum_{i=1}^{N}\|u_i\|^2\right),$$

for $0 < \gamma < 1$. Namely,

$$\int_0^T\left(\|Z\|^2 + \sum_{i=1}^{N}\|u_i\|^2\right)\mathrm{d}t \leq \gamma^2\int_0^T\|\omega\|^2\,\mathrm{d}t$$

for any $\omega \in \mathcal{L}_2$, $\omega \neq 0$.

Condition (C1) represents an individual optimality criterion for each agent, whereas Condition (C2) represents a robustness criterion for the global system.

Problem 1 can be interpreted as a multi-player version of the mixed H_2/H_∞ control problem and, following the approach of [18], it can be recast as a nonzero-sum differential game with $(N+1)$ players. To this end, let

$$J_{N+1}\left(X(0), u_1,\ldots,u_N,\omega\right) = \int_0^T\left(\gamma^2\|\omega\|^2 - \|Z\|^2 - \sum_{i=1}^{N}\|u_i\|^2\right)\mathrm{d}t. \tag{5}$$

Problem 2: nonzero-sum differential game formulation
Consider System (2). Determine a set of feedback strategies

$$\mathcal{U}^* = \left(u_1^*,\ldots,u_N^*,w^*\right)$$

that renders the zero equilibrium of System (2) stable in closed-loop with $\mathcal{U}^*$ and that satisfies the Nash equilibrium inequalities

$$J_i\left(X(0),\mathcal{U}^*\right) \leq J_i\left(X(0),\mathcal{U}_{u_i}\right), \tag{6}$$

and

$$J_{N+1}\left(X(0),\mathcal{U}^*\right) \leq J_{N+1}\left(X(0),\mathcal{U}_\omega\right), \tag{7}$$

where $\mathcal{U}_{u_i} = \left(u_1^*,\ldots,u_{i=1}^*,u_i,u_{i+1}^*\ldots,u_N^*,\omega^*\right)$ with $u_i \neq u_i^*$, for $i = 1,\ldots,N$, and $\mathcal{U}_\omega = \left(u_1^*,\ldots,u_N^*,\omega\right)$, with $\omega \neq \omega^*$, are sets of stabilizing feedback strategies.

The control strategies $\mathcal{U}^*$, namely the control inputs u_i^*, $i = 1,\ldots,N$, and disturbance w^* satisfying (6), $i = 1,\ldots,N$ and (7), constitute the Nash equilibrium strategies of the differential game in Problem 2. Considering the Nash equilibrium inequalities, it is clear that (6), $i = 1,\ldots,N$, correspond to Condition (C1), $i = 1,\ldots,N$, of Problem 2. Moreover, if $J_{N+1}\left(X(0),\mathcal{U}^*\right) > 0$, it follows from (7) that $J_{N+1}\left(X(0),\mathcal{U}_\omega\right) > 0$, for all $\omega \in \mathcal{L}_2$, thus satisfying Condition (C2) of Problem 2.

Exact solutions to the nonzero-sum differential game

Problem 2 constitutes a nonzero-sum differential game for which solutions, found using the dynamic programming method, are characterized by coupled Riccati differential equations (in the finite-horizon case) or coupled algebraic Riccati equations (in the infinite-horizon case). For more details on linear quadratic differential games see, for instance, [5], [35]. The game theoretic formulation in Problem 2 is particularly appealing because it naturally captures the trade-off between optimality and robustness (see, for instance, [1]). A solution of Problem 2 (considering linear feedback strategies only[1]) is provided in the following.

Assumption 1
The global state is constructed in a manner such that the running costs and terminal costs can be written as

$$q_i(X) \triangleq \|e_i\|^2 = X^{\mathrm{T}}Q_iX,$$

where $Q_i = Q_i^{\mathrm{T}} \geq 0$, for $i = 1,\ldots,N$.

Clearly, a consequence of Assumption 1 is that $\|Z\|^2$ can be written in the form

$$q_{N+1}(X) \triangleq \|Z\|^2 = X^{\mathrm{T}}Q_{N+1}X$$

where $Q_{N+1} = \sum_{i=1}^{N} Q_i$.

Proposition 1
Consider the global system (2), and the cost functionals (4), $i = 1,\ldots,N$, and (5). Suppose that we can find $P_i : \mathbb{R} \to \mathbb{R}^{Nn\times Nn}$, such that $P_i(t) = P_i(t)^{\mathrm{T}} \geq 0$, $i = 1,\ldots,N$, and $P_{N+1}(t) = P_{N+1}(t)^{\mathrm{T}} \leq 0$ satisfying the coupled Riccati differential equations

$$\begin{aligned}
-\dot{P}_i(t) &= Q_i + P_i(t)A^g + A^{g\mathrm{T}}P_i(t) - P_i(t)B_i^gB_i^{g\mathrm{T}}P_i(t)\\
&-\sum_{j=1,j\neq i}^{N}\left(P_i(t)B_j^gB_j^{g\mathrm{T}}P_j(t) + P_j(t)Bj^gB_j^{g\mathrm{T}}P_i(t)\right)\\
&-\gamma^{-2}P_i(t)B_{N+1}^gB_{N+1}^{g\;\mathrm{T}}P_{N+1}(t)\\
&-\gamma^{-2}P_{N+1}(t)B_{N+1}^gB_{N+1}^{g\;\mathrm{T}}P_i(t)\\
P_i(T) &= 0
\end{aligned} \tag{8}$$

for $i = 1,\ldots,N$, and

[1] In general, linear quadratic nonzero-sum differential games can admit *nonlinear* solutions (see, for instance, [2]). However, as is commonly done (see, for instance, [13]), only linear feedback strategies are considered here.

$$
\begin{aligned}
&-\dot{P}_{N+1}(t) = -Q_{N+1} + P_{N+1}(t)A^{g} + A^{g\mathrm{T}}P_{N+1}(t)\\
&-\gamma^{-2}P_{N+1}(t)B^{g}_{N+1}B^{g}_{N+1}{}^{\mathrm{T}}P_{N+1}(t)\\
&-\sum_{j=1}^{N}\left(P_j(t)B_j^{g}B_j^{g\mathrm{T}}P_j(t) - P_{N+1}(t)B_j^{g}B_j^{g\mathrm{T}}P_j(t)\right)\\
&-\sum_{j=1}^{N}P_j(t)B_j^{g}B_j^{g\mathrm{T}}P_{N+1}(t)\\
&P_{N+1}(T)=0
\end{aligned}
\tag{9}
$$

Then, the following statements hold:

i. The Nash equilibrium strategies are given by

$$
\begin{aligned}
u_i^{*} &= -B_i^{g\mathrm{T}}P_i(t)X\\
\omega^{*} &= -\gamma^{-2}B^{g}_{N+1}{}^{\mathrm{T}}P_{N+1}(t)X
\end{aligned}
\tag{10}
$$

for $i=1,\ldots,N$;

ii. In the case that $u=u^{*}$ and $X(0)=0$, Condition (C2) of Problem 2 is satisfied for any continuous function $\omega\in\mathcal{L}_2$.

Proof

Proposition 1 is essentially a multi-player version of the result in ([18], Theorem 2.1) and, as such, the proof is similar to the proof of the sufficient conditions provided therein. The proof consists of two main steps in which we demonstrate that Claims (i) and (ii) hold true, respectively.

As in [18], the statement (i) can be demonstrated by completion of squares[2]. Let us consider first the cost functionals (4), $i=1,\ldots,N$.

From the boundary condition $P_i(T)=0$ it follows that

$$
\begin{aligned}
&J_i\left(X(0),u_1,\ldots,u_N,\omega\right) - X(0)P_i(0)X(0)\\
&=\int_0^T X^{\mathrm{T}}Q_iX + \left\|u_i\right\|^2 + \frac{\mathrm{d}}{\mathrm{d}t}\left(X^{\mathrm{T}}P_iX\right)\mathrm{d}t.
\end{aligned}
\tag{11}
$$

Substituting the system dynamics (2) and the time derivative of P_i given in (8), the above relation is transformed into

$$
\begin{aligned}
&J_i\left(X(0),u_1,\ldots,u_N,\omega\right) - X(0)P_i(0)X(0)\\
&=\int_0^T\left(\begin{array}{c}\left\|u_i-u_i^{*}\right\|^2 + 2\displaystyle\sum_{j=1,j\neq i}^{N}X^{\mathrm{T}}P_iB_j^{g}\left(u_j-u_j^{*}\right)\\ +2X^{\mathrm{T}}P_iB^{g}_{N+1}\left(\omega-\omega^{*}\right)\end{array}\right)\mathrm{d}t
\end{aligned}
\tag{12}
$$

To demonstrate that u_i^{*} is the Nash equilibrium strategy of the i-th agent, note that

$$
J_i\left(X(0),\mathcal{U}_{u_i}\right) - X(0)P_i(0)X(0) = \int_0^T\left\|u_i-u_i^{*}\right\|^2\mathrm{d}t,
$$

which is minimized when $u_i=u_i^{*}$. Namely, u_i^{*} is such that Inequality (6) is satisfied, for $i=1,\ldots,N$. Considering the cost functional (5), following the same steps, it can be shown that

$$
\begin{aligned}
&J_{N+1}\left(X(0),u_1,\ldots,u_N,\omega\right) - X(0)^{\mathrm{T}}P_{N+1}(0)X(0)\\
&=\int_0^T\gamma^2\left\|\omega-\omega^{*}\right\|^2 - \sum_{j=1}^{N}\left(\left\|u_j^{*}\right\|^2 - \left\|u_j\right\|^2\right)\\
&\quad+2\sum_{j=1}^{N}\left(X^{\mathrm{T}}P_{N+1}B_j\left(u_j-u_j^{*}\right)\right)\mathrm{d}t.
\end{aligned}
$$

Once again, it follows that

$$
\begin{aligned}
&J_{N+1}\left(X(0),\mathcal{U}_{\omega}\right) - X(0)^{\mathrm{T}}P_{N+1}(0)X(0)\\
&\qquad=\gamma^2\int_0^T\left\|\omega-\omega^{*}\right\|^2\mathrm{d}t,
\end{aligned}
$$

is minimized when $\omega=\omega^{*}$, *i.e.* ω^{*} satisfies Inequality (7), thus completing the proof of the statement (i).

The second part of the claim is demonstrated by noting that

$$
J_{N+1}\left(X(0),\mathcal{U}^{*}\right) = X(0)^{\mathrm{T}}P_{N+1}(0)X(0)
$$

Thus, for the initial condition $X(0)=0$, the cost associated with the worst-case disturbance is zero, *i.e.*, $J_{N+1}(X(0),\mathcal{U}^{*})=0$. Therefore, it follows from (7) that any disturbance $\omega\in\mathcal{L}_2$ is such that

$$
J_{N+1}\left(X(0),\mathcal{U}_{\omega}\right)\geq 0.
$$

Condition (C2) then follows from the definition of the cost functional (5), which concludes the proof.

Remark 1

The so-called value functions

$$
V_i\left(X(t)\right) = X(t)^{\mathrm{T}}P_i(t)X(t),
$$

for $i=1,\ldots,N+1$, are such that

$$
J_i\left(X(0),\mathcal{U}^{*}\right) = X(0)^{\mathrm{T}}P_i(0)X(0).
$$

Noting that

$$
J_i\left(X(0),\mathcal{U}^{*}\right)\geq 0,
$$

for $i=1,\ldots,N$, it is clear that $P_i(t)\geq 0$ for $t\geq 0$. Similarly, since

$$
\begin{aligned}
&J_{N+1}\left(X(0),\mathcal{U}^{*}\right) = \int_0^T\gamma^2\left\|\omega^{*}\right\|^2\mathrm{d}t - \sum_{i=1}^{N}J_i\left(X(0),\mathcal{U}^{*}\right)\\
&\leq J_{N+1}\left(X(0),u_1^{*},\ldots,u_N^{*},0\right)\\
&=-\sum_{i=1}^{N}J_i\left(X(0),u_1^{*},\ldots,u_N^{*},0\right)\leq 0,
\end{aligned}
$$

it is clear that $P_{N+1}(t)\leq 0$ for $t\geq 0$.

In the infinite-horizon case, *i.e.*, in the limit as $T\to\infty$, the Nash equilibrium solution (10) requires the solution of coupled *algebraic* Riccati equations (AREs) instead of the coupled Riccati differential equations (8), $i=1,\ldots,N$, and (9).

[2] Alternatively, the property in (i) can be demonstrated by applying the dynamic Programming principle (see, *e.g.* [7]).

Proposition 2

Consider the global system (2) and suppose that we can obtain a solution $P_i = P_i^{\mathrm{T}} \geq 0$, $P_{N_1} = P_{N_1}^{\mathrm{T}} \leq 0$ of the (static) coupled AREs

$$\begin{aligned} & Q_i + P_i A^g + A^{g\mathrm{T}} P_i - P_i B_i^g B_i^{g\mathrm{T}} P_i \\ & - \sum_{j=1, j\neq i}^{N} \left(P_i B_j^g B_j^{g\mathrm{T}} P_j + P_j Bj^g B_j^{g\mathrm{T}} P_i \right) \\ & -\gamma^{-2} P_i B_{N+1}^g B_{N+1}^{g}{}^{\mathrm{T}} P_{N+1} \\ & -\gamma^{-2} P_{N+1} B_{N+1}^g B_{N+1}^{g}{}^{\mathrm{T}} P_i = 0 , \end{aligned} \tag{13}$$

for $i = 1, \dots, N$, and

$$\begin{aligned} & -Q_{N+1} + P_{N+1} A^g + A^{g\mathrm{T}} P_{N+1} \\ & -\gamma^{-2} P_{N+1} B_{N+1}^g B_{N+1}^{g}{}^{\mathrm{T}} P_{N+1} - \sum_{j=1}^{N} P_j B_j^g B_j^{g\mathrm{T}} P_j \\ & - \sum_{j=1}^{N} \left(P_{N+1} B_j^g B_j^{g\mathrm{T}} P_j + P_j B_j^g B_j^{g\mathrm{T}} P_{N+1} \right) = 0. \end{aligned} \tag{14}$$

Then, the following statements hold:

i. If the communication graph $\mathcal{G}$ is such that

$$\sum_{i=1}^{N} Q_i > 0 . \tag{15}$$

Then, the origin of System (2) in closed-loop with the feedback strategies

$$\begin{aligned} u_i^* &= -B_i^{g\mathrm{T}} P_i X , \\ \omega^* &= -\gamma^{-2} B_{N+1}^{g}{}^{\mathrm{T}} P_{N+1} X , \end{aligned} \tag{16}$$

for $i = 1, \dots, N$, with P_i, $i = 1, \dots, N+1$, satisfying (13), $i = 1, \dots, N$, and (14), is stable;

ii. The Nash equilibrium strategies corresponding to Problem 2 in the infinite horizon case, *i.e.,* in the limit as $T \to \infty$, are given by (16), for $i = 1, \dots, N$.

Proof

The proof essentially consists of two steps. To demonstrate stability of the closed-loop system, *i.e.,* statement (i) of the proposition, note that the summation of the N first AREs (13), $i = 1, \dots, N$, yields the relation

$$\sum_{j=1}^{N} \left(P_j A_{cl}^g + A_{cl}^{g\mathrm{T}} P_j \right) + \sum_{j=1}^{N} \left(Q_j + P_j B_j^g B_j^{g\mathrm{T}} P_j \right) = 0 \tag{17}$$

where A_{cl}^g is the matrix describing the closed-loop system, namely $A_{cl}^g = A^g - \sum_{j=1}^{N} B_j^g B_j^{g\mathrm{T}} P_j - \gamma^{-2} B_{N+1}^g B_{N+1}^{g}{}^{\mathrm{T}} P_{N+1}$. Let $V_i(X) = X^{\mathrm{T}} P_i X$ denote the *value function* associated with the *i*-th agent (as in Remark 1), for $i - 1, \dots, N$, and let $W(X) = \sum_{j=1}^{N} V_i$. Note that

$$W(X(0)) = \sum_{j=1}^{N} J_i \left(X(0), \mathcal{U}^* \right) > 0 ,$$

by Assumption (15). Moreover, along the trajectories of the closed-loop system the time derivative of the function W is given by

$$\dot{W} = X^{\mathrm{T}} \left(A_{cl}^{g\mathrm{T}} \sum_{j=1}^{N} P_j + \sum_{j=1}^{N} P_j A_{cl}^g \right) X .$$

It follows from (17) that

$$\dot{W} = -\sum_{j=1}^{N} X^{\mathrm{T}} \left(Q_j + P_j B_j^g B_j^{g\mathrm{T}} P_j \right) X \leq -\sum_{j=1}^{N} X^{\mathrm{T}} Q_j X .$$

Thus, from Assumption (15), $\dot{W} < 0$ for all $X \neq 0$ and stability of the closed-loop system follows from standard Lyapunov arguments.

The second statement follows directly from the same arguments in the proof of Proposition 1, noting that $\lim_{t\to\infty} X(t) = 0$ by stability. Consequently, the relations (11), $i = 1, \dots, N$, and (12) hold with P_i, $i = 1, \dots, N+1$, the (static) solutions of (13), $i = 1 \dots, N$, and (14). Consider first the cost functionals (4), $i = 1, \dots, N$. It can be shown, using (13), that

$$\begin{aligned} & \frac{\mathrm{d}}{\mathrm{d}t} \left(X^{\mathrm{T}} P_i X \right) = -X^{\mathrm{T}} Q_i X - \left\| u_i \right\|^2 + \left\| u_i - u_i^* \right\|^2 \\ & + \sum_{j=1, j\neq i}^{N} \left(X^{\mathrm{T}} P_i B_j^g \left(u_j - u_j^* \right) + \left(u_j - u_j^* \right) B_j^{g\mathrm{T}} P_i X \right) \\ & + X^{\mathrm{T}} P_i B_{N+1}^g \left(\omega - \omega^* \right) + \left(\omega - \omega^* \right)^{\mathrm{T}} B_{N+1}^{g}{}^{\mathrm{T}} P_i X , \end{aligned}$$

for $i = 1 \dots, N$. Thus, it follows (as in the finite-horizon case) that

$$J_i \left(X(0), \mathcal{U}_{u_i} \right) - X(0)^{\mathrm{T}} P_i X(0) = \int_0^T \left\| u_i - u_i^* \right\|^2 \mathrm{d}t$$

for $i = 1, \dots, N$. Similar considerations for the cost functional (5), yield (as in the finite-horizon case)

$$\begin{aligned} & J_{N+1} \left(X(0), \mathcal{U}_{\omega} \right) - X(0)^{\mathrm{T}} P_{N+1} X(0) \\ & \qquad = \gamma^2 \int_0^T \left\| \omega - \omega^* \right\|^2 \mathrm{d}t , \end{aligned}$$

which demonstrates statement (ii) and thus completes the proof.

Remark 2

The condition (15) is standard in the context of infinite-horizon differential games (and infinite-horizon optimal control) as seen, for instance, in [5, 37]. The connection between this condition and the topology of the underlying graph $\mathcal{G}$ will be investigated in future work.

Approximate solutions to the nonzero-sum differential game

Solutions to coupled AREs (such as (13), $i = 1, \dots, N$, and (14)) which arise in the context of linear quadratic differential games (see, for example [35], [13]) are oftentimes difficult to obtain. Coupled AREs have, for instance, been considered in [18], [15], [3] and solutions for particular classes of problems have been provided in [29], [30], [13], [14], [23]. In [24] it has been demonstrated that solving algebraic Riccati *inequalities* instead of *equalities* yields an approximate solution – in terms of a so-called ε_α-Nash equilibrium solution – to a differential game. The space of solutions of the inequalities contain the solutions of the original equalities and are, as a consequence, sometimes more readily solved. The notion of α-admissible strategies (introduced in [24]) is recalled and approximate solutions, *similar* to α-admissible strategies, are provided for Problem 2 in what follows. Solving inequalities instead of (13), $i=1,\ldots,N$ and (14) can be interpreted as solving a differential game subject to "modified cost functionals". While in general the results may differ significantly from the original game, the notion of an ε_α-Nash equilibrium solution enables us to relate the resulting feedback strategies to the original game.

Definition 1 [24]
A set of linear[3] state feedback control inputs

$$\mathcal{U}=(u_1,\ldots,u_N,\omega)$$

is said to be α-admissible, with $\alpha>0$, if the origin of System (2) in closed-loop with $\mathcal{U}$ is such that

$$\sigma\left(\bar{A}_{cl}+\alpha I\right)\in\mathbb{C}^-,$$

where $\bar{A}_{cl}$ is the matrix describing the closed-loop system.

Proposition 3
Consider the global system (2) and suppose that we can obtain $P_i=P_i^{\mathrm{T}}\geq 0$, $i-1,\ldots,N$, and $P_{N+1}=P_{N+1}^{\mathrm{T}}\leq 0$ satisfying the inequalities

$$\begin{aligned}&Q_i+P_iA^g+A^{g\mathrm{T}}P_i-P_iB_i^gB_i^{g\mathrm{T}}P_i\\&-\sum_{j=1,j\neq i}^{N}\left(P_iB_j^gB_j^{g\mathrm{T}}P_j+P_jBj^gB_j^{g\mathrm{T}}P_i\right)\\&-\gamma^{-2}P_iB_{N+1}^gB_{N+1}^{g\ \mathrm{T}}P_{N+1}\\&-\gamma^{-2}P_{N+1}B_{N+1}^gB_{N+1}^{g\ \mathrm{T}}P_i\leq 0,\end{aligned}\tag{18}$$

for $i=1,\ldots,N$, and

$$\begin{aligned}&-Q_{N+1}+P_{N+1}A^g+A^{g\mathrm{T}}P_{N+1}\\&-\gamma^{-2}P_{N+1}B_{N+1}^gB_{N+1}^{g\ \mathrm{T}}P_{N+1}-\sum_{j=1}^{N}P_jB_j^gB_j^{g\mathrm{T}}P_j\\&-\sum_{j=1}^{N}\left(P_{N+1}B_j^gB_j^{g\mathrm{T}}P_j+P_jB_j^gB_j^{g\mathrm{T}}P_{N+1}\right)\geq 0.\end{aligned}\tag{19}$$

Moreover, suppose that the communication graph $\mathcal{G}$ is such that (15) holds. Then, the set of feedback strategies $\mathcal{U}^*=\left(u_1^*,\ldots,u_N^*,\omega^*\right)$, with u_i^* and ω^* given by (16), $i=1,\ldots,N$, with P_i and P_{N+1} satisfying (18), $i=1,\ldots,N$, and (19), respectively, are such that the following statements hold:

i. System (2) in closed-loop with $\mathcal{U}^*$ is stable.
ii. Considering the infinite-horizon case, *i.e.,* in the limit as $T\to\infty$, the inequalities

$$J_i\left(X(0),\mathcal{U}^*\right)\leq J_i\left(X(0),\mathcal{U}_{u_i}\right)+\varepsilon_\alpha\tag{20}$$

are satisfied with $\varepsilon_\alpha>0$, parameterized in $\alpha>0$ and $X(0)$, for any α-admissible set of strategies $\mathcal{U}_{u_i}=\left(u_1^*,\ldots,u_{i-1}^*,u_i,u_{i+1}^*,\ldots,u_N^*,\omega^*\right)$, for any $\alpha>0$ and for $i=1,\ldots,N$.

iii. In the case where $u_i=u_i^*$ and $X(0)=0$, Condition (C2) of Problem 2 is satisfied for any continuous function $\omega\in\mathcal{L}_2$.

Proof
Stability can be demonstrated following the same steps used in the first part of the proof of Proposition 2. The statement (ii) can be demonstrated following steps similar to those provided in [24], Proposition 2. Namely, consider the inequalities (18), $i=1,\ldots,N$. These inequalities imply that there exist matrices $\Upsilon_i=\Upsilon_i^{\mathrm{T}}\geq 0$, such that

$$\begin{aligned}&Q_i+\Upsilon_i+P_iA^g+A^{g\mathrm{T}}P_i-P_iB_i^gB_i^{g\mathrm{T}}P_i\\&-\sum_{j=1,j\neq i}^{N}\left(P_iB_j^gB_j^{g\mathrm{T}}P_j+P_jBj^gB_j^{g\mathrm{T}}P_i\right)\\&-\gamma^{-2}P_iB_{N+1}^gB_{N+1}^{g\ \mathrm{T}}P_{N+1}\\&-\gamma^{-2}P_{N+1}B_{N+1}^gB_{N+1}^{g\ \mathrm{T}}P_i=0,\end{aligned}$$

for $i=1,\ldots,N$. Similarly, Inequality (19) implies that there exists a matrix $\Upsilon_{N+1}=\Upsilon_{N+1}^{\mathrm{T}}\geq 0$ such that

$$\begin{aligned}&-Q_{N+1}-\Upsilon_{N+1}+P_{N+1}A^g+A^{g\mathrm{T}}P_{N+1}\\&-\gamma^{-2}P_{N+1}B_{N+1}^gB_{N+1}^{g\ \mathrm{T}}P_{N+1}-\sum_{j=1}^{N}P_jB_j^gB_j^{g\mathrm{T}}P_j\\&-\sum_{j=1}^{N}\left(P_{N+1}B_j^gB_j^{g\mathrm{T}}P_j+P_jB_j^gB_j^{g\mathrm{T}}P_{N+1}\right)=0.\end{aligned}$$

It follows that the feedback strategies $\mathcal{U}^*$ are the Nash equilibrium strategies of a nonzero-sum differential game with the *modified* cost functionals

$$\tilde{J}_i\left(X(0),\mathcal{U}\right)=J_i\left(X(0),\mathcal{U}\right)+\int_0^\infty X^{\mathrm{T}}\Upsilon_iX\mathrm{d}t,\tag{21}$$

for $i=1,\ldots,N$, and

$$\tilde{J}_{N+1}\left(X(0),\mathcal{U}\right)=J_{N+1}\left(X(0),\mathcal{U}\right)-\int_0^\infty X^{\mathrm{T}}\Upsilon_{N+1}X\mathrm{d}t,\tag{22}$$

wherein $\mathcal{U}=\left(u_1,\ldots,u_N,\omega\right)$. Let $\hat{X}(t)$ denote the trajectory of System (2) in closed-loop with the α-admissible set of strategies $\hat{\mathcal{U}}_{u_i}=\left(u_1^*,\ldots,u_{i-1}^*,\hat{u}_i,u_{i+1}^*,\ldots,u_N^*,\omega^*\right)$, where $\hat{u}_i=K_iX$ is such that the closed-loop system has the minimum possible decay rate (specified by $\alpha>0$). It is straightforward to see that

$$J_i\left(X(0),\mathcal{U}^*\right)\leq\tilde{J}_i\left(X(0),\mathcal{U}^*\right)\leq\tilde{J}_i\left(X(0),\hat{\mathcal{U}}_{u_i}\right),$$

since $\Upsilon_i\geq 0$, $i=1,\ldots,N$. Namely, the relation

$$J_i\left(X(0),\mathcal{U}^*\right)\leq J_i\left(X(0),\hat{\mathcal{U}}_{u_i}\right)+\int_0^\infty\hat{X}^{\mathrm{T}}\Upsilon_i\hat{X}\mathrm{d}t\tag{23}$$

holds and, exploiting α-admissibility of the set of strategies $\hat{\mathcal{U}}_{u_i}$ the second term on the right-hand side of (23), which accounts for an *additional running cost*, can be bounded from above. To this end, let

[3] While we limit our at tention to linear feedback strategies, the notion can be defined for general (possibly nonlinear) strategies as in [24].

$\hat{A}_{cl,\alpha}$ denote the matrix describing System (2) in closed-loop with $\hat{\mathcal{U}}_{u_i}$ and note that since $\hat{\mathcal{U}}_{u_i}$ is α-admissible, the Lyapunov equation

$$\hat{P}_{i,\alpha}\hat{A}_{cl,\alpha} + \hat{A}_{cl,\alpha}^{\mathrm{T}}\hat{P}_{i,\varepsilon} + \Upsilon_i = 0\,,$$

has a unique solution $\hat{P}_{i,\varepsilon} = \hat{P}_{i,\varepsilon}^{\mathrm{T}} \geq 0$. Moreover, the function $\hat{V}_i = \hat{X}^{\mathrm{T}}\hat{P}_{i,\varepsilon}\hat{X}$ is such that

$$\dot{\hat{V}}_i(\hat{X}) = \hat{P}_{i,\varepsilon}\hat{A}_{cl,\alpha} + \hat{A}_{cl,\alpha}^{\mathrm{T}}\hat{P}_{i,\varepsilon} = -\hat{X}\Upsilon_i\hat{X}\,.$$

Integrating both sides of this relation from zero to infinity (noting that $\lim_{t\to\infty} X(t) = 0$ since $\hat{\mathcal{U}}_{u_i}$ is α-admissible), yields

$$\hat{V}(\hat{X}(0)) = \hat{X}(0)^{\mathrm{T}}\hat{P}_{i,\varepsilon}\hat{X}(0) = \int_0^\infty \hat{X}\Upsilon_i\hat{X}\,\mathrm{d}t\,.$$

It follows from (23) that

$$J_i(X(0),\mathcal{U}^*) \leq J_i(X(0),\hat{\mathcal{U}}_{u_i}) + X(0)^{\mathrm{T}}\max_i\{P_{i,\varepsilon}\}X(0)\,,$$

$i = 1,\ldots,N$. The modified cost functional (22), on the other hand, is such that

$$\tilde{J}_{N+1}(X(0),\mathcal{U}^*) = X(0)^{\mathrm{T}}P_{N+1}X(0) \leq 0\,.$$

It follows that

$$\tilde{J}_{N+1}(0,\mathcal{U}_\omega) \geq \tilde{J}_{N+1}(0,\mathcal{U}^*) = 0\,,$$

which implies that the inequality

$$\begin{aligned}
&\gamma^2\int_0^\infty \|\omega\|^2\,\mathrm{d}t \\
&\geq \int_0^\infty\left(\|Z\|^2 + \sum_{i=1}^N\|u_i^*\|^2\right)\mathrm{d}t + \int_0^\infty X^{\mathrm{T}}\Upsilon_{N+1}X\mathrm{d}t \\
&\geq \int_0^\infty\left(\|Z\|^2 + \sum_{i=1}^N\|u_i\|^2\right),
\end{aligned}$$

is satisfied, for any $\omega \in \mathcal{L}_2$ when $X(0) = 0$, which concludes the proof.

Remark 3

Interestingly, considering Problem 2, the results in Proposition 3 entail that the "optimality criteria" in (C1) are solved approximately, whereas the "robustness" criterion in (C2) is solved exactly. This result is in line with the observation that there is, in general, a trade-off between optimality and robustness.

Remark 4

A set of matrices P_i, $i = 1,\ldots,N$, satisfying the coupled AREs (13), $i = 1,\ldots,N$, and (14) also satisfy the inequalities (18), $i = 1,\ldots,N$, (19). That is, the space of solutions of the inequalities is larger – and includes – the space of solutions of the coupled AREs.

Simulations

In this section, we present two numerical examples to illustrate the theoretical results presented in the previous sections. The first example concerns a simple, scalar consensus problem, whereas the second example concerns formation control for a fleet of autonomous vehicles. While the two examples are considered separately in the following, both involve the same number of agents and communication graph.

Consensus at the origin

Consider first the case in which we wish to steer a group of agents towards a common consensus value which is fixed *a priori*. Without loss of generality, we consider this common value to be $x_i = 0$, $i = 1,\ldots,N$. Towards this end we construct the global state simply as the collection of the individual states of each agent, namely

$$X = \left[x_1^{\mathrm{T}}, x_2^{\mathrm{T}},\ldots,x_N^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{24}$$

resulting in the global system described by (2) with

$$A^g = \mathrm{blockdiag}\{A_1,\ldots,A_N\}\,,$$

$$B_1^g = \left[B_1^{\mathrm{T}},0,\ldots,0\right]^{\mathrm{T}},\ldots,B_N^g = \left[0,\ldots,0,B_{N+1}^{\mathrm{T}}\right]^{\mathrm{T}},$$

and

$$B_{N+1}^g = \left[B_{N+1}^{\mathrm{T}}, B_{N+1}^{\mathrm{T}},\ldots,B_{N+1}^{\mathrm{T}}\right]^{\mathrm{T}}.$$

Consider the case, similar to the one presented in [10], of four agents. Each agent $i \in \{1,\ldots,4\}$ is described by a scalar state $x_i \in \mathbb{R}$ and satisfies the dynamics

$$\dot{x}_i = B_i u_i + B_5\omega\,,$$

i.e., $A_i = 0$, for $i = 1,\ldots,N$, and consider the case in which the matrices B_i are identical for all $i = 1,\ldots,4$. The communication graph is described by the edge set

$$\mathcal{E} = \{(1,1);(1,2);(1,3);(2,2),(2,3);(3,3);(3,1);(3,4)\}\,,$$

and the corresponding running costs for each agent i, $i = 1,\ldots,4$, are defined by the matrices

$$Q_1 = \begin{bmatrix} \frac{5}{4} & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\quad Q_2 = \begin{bmatrix} \frac{1}{4} & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{5}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Q_3 = \begin{bmatrix} \frac{1}{9} & 0 & -\frac{1}{3} & 0 \\ 0 & \frac{1}{9} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{22}{9} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\quad Q_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & \frac{5}{4} \end{bmatrix}.$$

Consider the case in which $\gamma = 0.5$, $B_i = 1$, for $i = 1,\ldots,N$, and $B_5 = 0.1$ and consider Problem 2 in the infinite-horizon case. The set of matrices

$$P_1 = \begin{bmatrix} 1.1287 & 0.0074 & -0.1640 & 0.0189 \\ 0.0074 & 0.0005 & 0.0020 & 0.0004 \\ -0.1640 & 0.0020 & 0.0672 & -0.0004 \\ 0.0189 & 0.0004 & -0.0004 & 0.0005 \end{bmatrix},$$

$$P_2 = \begin{bmatrix} 0.0888 & -0.2087 & 0.0038 & -0.0009 \\ -0.2087 & 1.1485 & 0.0047 & 0.0185 \\ 0.0038 & 0.0047 & 0.0007 & 0.0004 \\ -0.0009 & 0.0185 & 0.0004 & 0.0005 \end{bmatrix},$$

$$P_3 = \begin{bmatrix} 0.0407 & -0.0036 & -0.1142 & -0.0003 \\ -0.0036 & 0.0416 & -0.1054 & -0.0001 \\ -0.1142 & -0.1054 & 1.5944 & 0.0220 \\ -0.0003 & -0.0001 & 0.0220 & 0.0006 \end{bmatrix},$$

$$P_4 = \begin{bmatrix} 0.0006 & 0.0005 & 0.0025 & 0.0054 \\ 0.0005 & 0.0005 & 0.0020 & 0.0072 \\ 0.0025 & 0.0020 & 0.0672 & -0.1669 \\ 0.0054 & 0.0072 & -0.1669 & 1.1559 \end{bmatrix},$$

$$P_5 = \begin{bmatrix} -1.2426 & 0.2205 & 0.2906 & -0.0062 \\ 0.2205 & -1.1750 & 0.1155 & -0.0088 \\ 0.2906 & 0.1155 & -1.7071 & 0.1654 \\ -0.0062 & -0.0088 & 0.1654 & -1.1390 \end{bmatrix},$$

constitutes a solution[4] of the coupled AREs (13), $i = 1,\dots,N$, and (14). Note that the condition (15) is satisfied. The performance of the resulting feedback control laws u_i^* given in (16) is evaluated through a series of simulations. In all plots blue indicates Agent 1, green indicates Agent 2, purple indicates Agent 3 and cyan indicates Agent 4, whereas red indicates the quantities relating to the disturbance, *i.e., Player 5* in the differential game defined in Problem 2. Note that the running cost associated with Player 5 is given by $Q_5 = \sum_{i=1}^{N} Q_i$.

Consider first the case in which the state of the system is perturbed, such that the initial states are $x_1(0) = 1$, $x_2(0) = 2$, $x_3(0) = 0$ and $x_4(0) = -1$. Suppose that the system is influenced by a disturbance of the form $\omega = \omega_k = k\omega^*$, where $k \in \mathbb{R}$ is a constant parameter. Let $\mathcal{U}_{\omega_k}$ denote the set of strategies

$$\mathcal{U}_{\omega_k} = \left(u_1^*,\dots,u_N^*,\omega_k\right).$$

The cost $J_{N+1}\left(X(0),\mathcal{U}_{\omega_k}\right)$ obtained for various values of k, when all four agents apply the feedback control strategies u_i^* given in (16), $i = 1,\dots,4$, is presented in Figure 1, where the minimum cost is indicated by the black diamond marker and corresponds to the value $k = 1$. Similarly, consider the four cases in which the system is influenced by the worst-case disturbance ω^* and each agent i, for $i = 1,\dots,4$, adopts a control input of the form $u_i = u_{i,k} = ku_i^*$ (where $k \in \mathbb{R}$ is again a constant parameter) while all other agents j, $j = 1,\dots,4$, $j \neq i$ adhere to the Nash equilibrium control laws u_j^*. Let $\mathcal{U}_{u_{i,k}}$ denote the set of strategies

$$\mathcal{U}_{u_{i,k}} = (u_1^*,\dots,u_{i=1}^*,u_{i,k},u_{i+1}^*,\dots,u_N^*,\omega^*).$$

The variations of the resulting costs, namely $J_i(X(0),\mathcal{U}_{i,k})$, with the parameter k are shown in Figure 2, for Agent 1 (top, left), Agent 2 (top, right), Agent 3 (bottom, left) and Agent 4 (bottom, right). In each plot the minimum cost is indicated by the black diamond marker and corresponds to $k = 1$. The latter result serves as an illustration that Condition (C1) of Problem 2 is satisfied by the Nash equilibrium solution of the differential game defined in Problem 2.

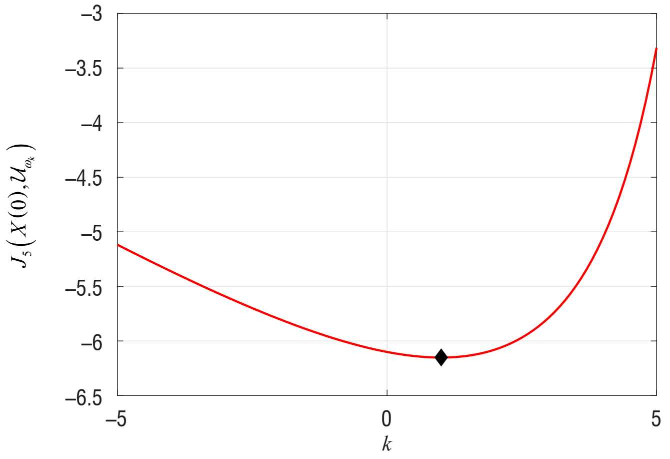


Figure 1 - Variation of $J_{N+1}\left(X(0),\mathcal{U}_{\omega_k}\right)$ as a function of k

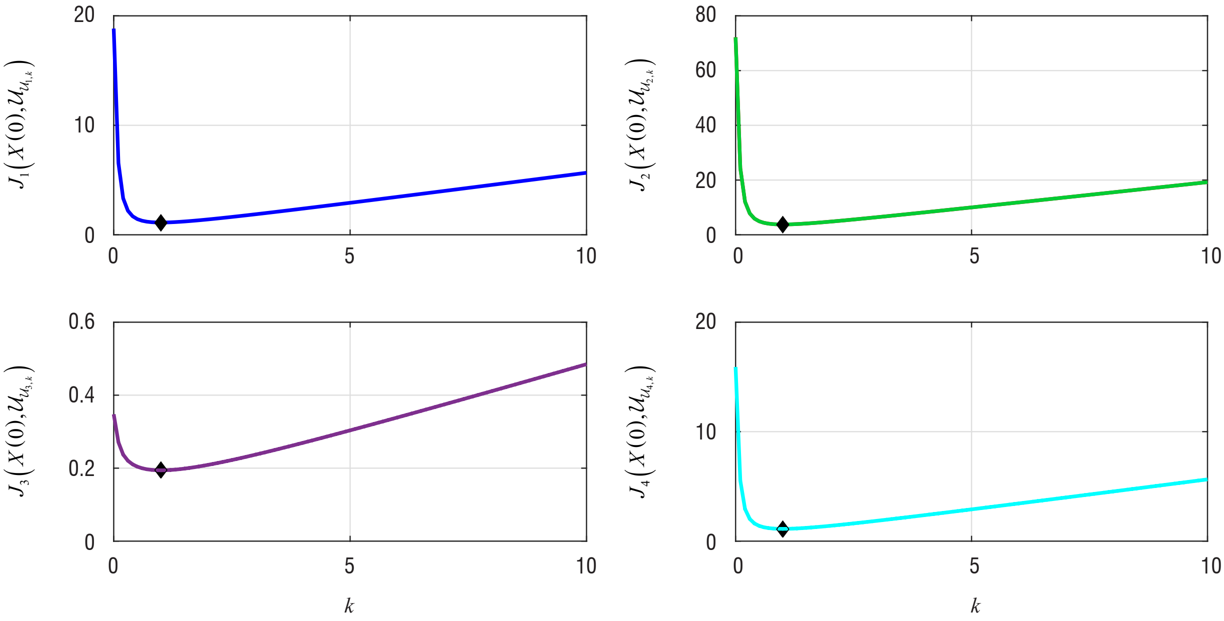


Figure 2 - The variations of $J_i\left(X(0),\mathcal{U}_{u_{i,k}}\right)$ as a function of k for Agent 1 (top, left), Agent 2 (top, right), Agent 3 (bottom, left) and Agent 4 (bottom, right).

[4] As mentioned in § "Approximate solutions", obtaining solutions of coupled Riccati equations arising in nonzero-sum differential games is not straight-forward, in general. In this numerical example, the solution of the coupled AREs has been obtained by numerically solving the finite-horizon equations (8), $i = 1,\dots,4$, and (9) backwards in time using the function 'ode45" in MATLAB. In this particular example, the resulting values of $P_i(0)$, for $i = 1,\dots,N+1$, converge to a solution of the AREs characterizing the solution of the differential game considered.

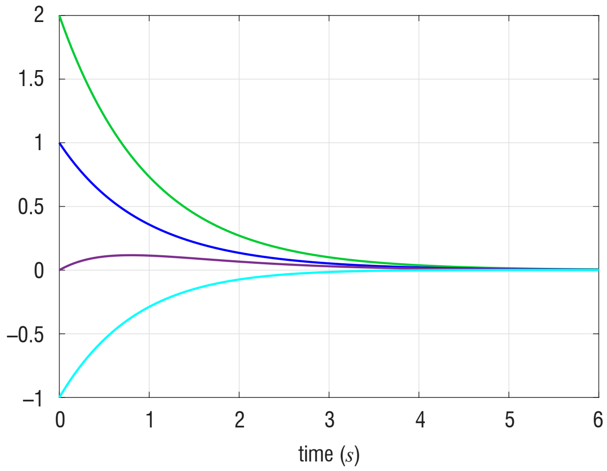


Figure 3 - The time histories of x_1 (blue line), x_2 (green line), x_3 (purple line) and x_4 (cyan line), when the disturbance and control inputs correspond to the Nash equilibrium solution of Problem 2.

The time histories of each individual state corresponding to the case in which $k = 1$, *i.e.*, when $\omega = \omega^*$ and $u_i = u_i^*$, $i = 1, \ldots, 4$, is shown in Figure 3.

Consider now the case in which the system starts in equilibrium[5], *i.e.*, $x_i(0) = 0$, for $i = 1, \ldots, 4$, and is subject to the disturbance

$$\omega = \begin{cases} -2 & \text{for } 0.5 < t < 1, \\ 20e^{-\frac{t}{2}}\sin(20t) & \text{for } 4 < t < 6, \\ 0 & \text{otherwise} \end{cases}$$

depicted in Figure 4. The resulting time histories of x_1 (top, left), x_2 (top, right), x_3 (bottom, left) and x_4 (bottom, right) are shown in Figure 5, whereas the time histories of the feedback control inputs u_1^* (top, left), u_2^* (top, right), u_3^* (bottom, left) and u_4^* (bottom, right) are shown in Figure 6. The time histories of the cost functionals (4) (top), for $i = 1, \ldots, 4$, and (5) (bottom) are shown in Figure 7. Notably $J_5 > 0$ at all times, which indicates that the robustness property (C2) of Problem 2 is satisfied.

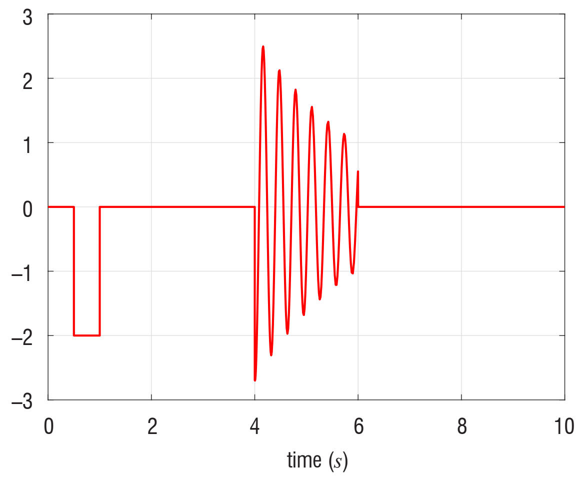


Figure 4 - Time history of the disturbance ω influencing the system.

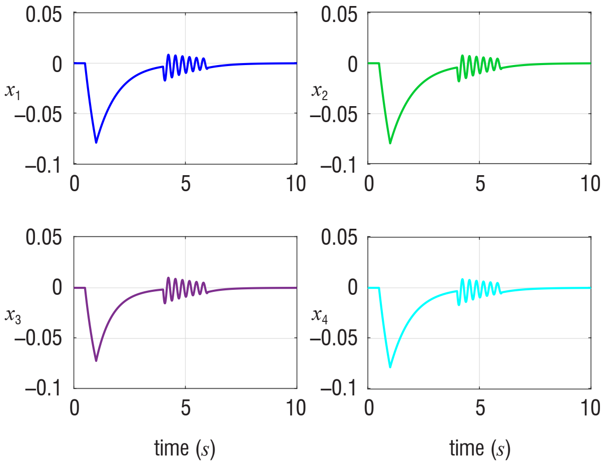


Figure 5 - Time histories of the states x_i, $i = 1, \ldots, 4$.

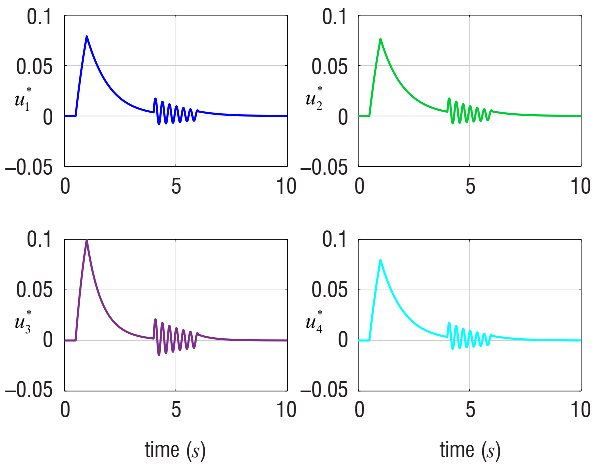


Figure 6 - Time histories of the control inputs u_i^*, $i = 1, \ldots, 4$, of each agent in response to the disturbance ω.

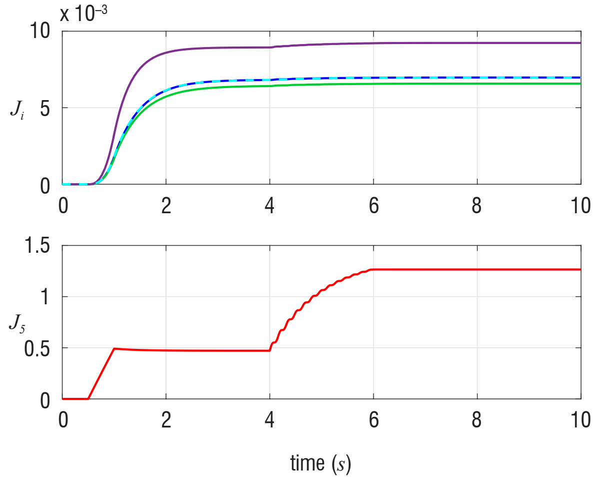


Figure 7 - Time histories of costs (4) (top), for $i = 1, \ldots, 4$, and (5) (bottom).

[5] Note that the worst-case disturbance $\omega^*(t) \equiv 0$ in this case.

Application to UAV formation flight

In this example, we consider four UAVs connected according to the same graph as in the previous example. The dynamics of the agents are modified to reflect the UAVs' behaviors and are described by the Euler-Lagrange system

$$M\ddot{q}_i + C\dot{q}_i = u_i + d \tag{25}$$

where $u_i \in \mathbb{R}^n$ is the control input of Agent i, $M \in \mathbb{R}^{n\times n}$ is the inertia matrix of Agent i, $C \in \mathbb{R}^{n\times n}$ is the matrix of the Coriolis and centripetal terms, and d is the additive external state perturbation (which is common to all agents). The values of M and C are considered identical for all agents. The state vector of each agent is defined as $(q_i, \dot{q}_i)$, which corresponds to the position and speed of the UAVs in some reference frame. In the following, we consider only the y and z variations of the positions, and assume that the component of the trajectory along the x axis is controlled separately and decoupled from the y and z evolution. The control objective is to drive the fleet to a desired target formation in some global reference frame $\mathcal{R}$. The target formation is represented via the relative coordinate vectors $r_{ij} = q_i - q_j$ between two agents i and j, and the target relative coordinate vector r_{ij}^* for all $(i,j) \in \mathcal{N}$. A target formation is defined by the set $\left\{r_{ij}^*, (i,j) \in \mathcal{N}\right\}$. Consider, without loss of generality, the first agent as a reference agent and introduce the target relative configuration vector $r^* = \left[r_{11}^{*T} \quad \dots \quad r_{1N}^{*T} \right]^T$. Any target relative configuration vector r_{ij}^* can be expressed as $r_{ij}^* = r_{1i}^* - r_{1j}^*$. The global formation problem can thus be expressed using the dynamic model

$$M\ddot{r}_{ij}^* + C\dot{r}_{ij}^* = u_i + d\ ,$$

for $i = 1,\dots,4$. The control laws that ensure convergence to the consensus at the origin can be sought by using a consensus error expressed as in (3). Since we are considering the same communication graph as in the previous example, the running cost for each agent is defined by the matrices Q_i , $i = 1,\dots,5$ presented in the previous subsection. Consider the case in which the matrices M and C are given by

$$M = \begin{bmatrix} 0.56 & -2.23 \\ -2.23 & 0.56 \end{bmatrix},$$

$$C = \begin{bmatrix} 1.40 & -1.76 \\ -1.76 & 2.99 \end{bmatrix}.$$

The target formation has the shape of a square whose center evolves along the x axis. The feedback control laws u_i^* that drive the fleet to the sought formation are obtained, as in the previous example, via the solution of the coupled AREs. The resulting trajectories and the time histories of the speed vectors for the four agents in the presence of the worst-case disturbance are depicted in Figures 8 and 9, respectively. To assess the robustness properties of the approach, the set of trajectories and speed evolutions using the feedback control laws with random initial state values and disturbances equal to k times the worst-case disturbance, with $0 < k \le 5$ randomly chosen, are presented in Figures 10 and 11, respectively. The target formations are reached in all cases while the speed values converge to the consensus speed.

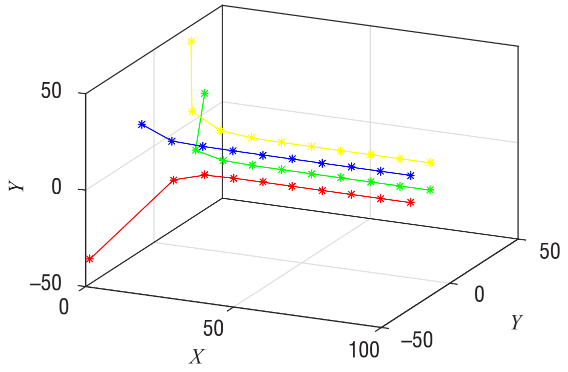


Figure 8 - Evolution of the four agents' trajectories obtained with the Nash equilibrium strategies.

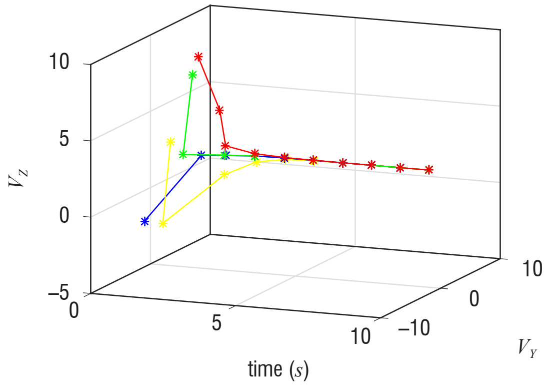


Figure 9 - Evolution of the four agents' speed values obtained with the Nash equilibrium solution.

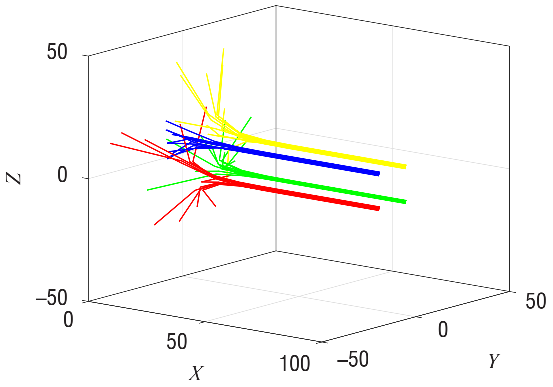


Figure 10 - Variations of the four agents' trajectories with random uncertainty.

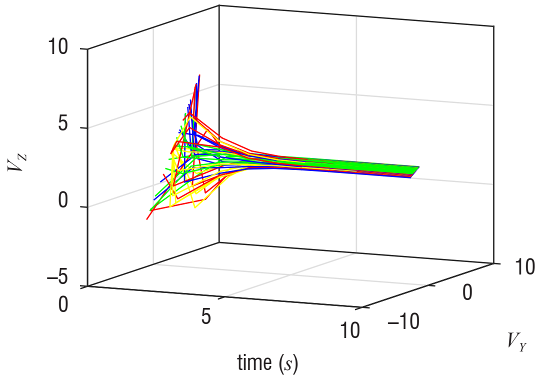


Figure 11 - Variations of the four agents' speed vectors with random uncertainty.

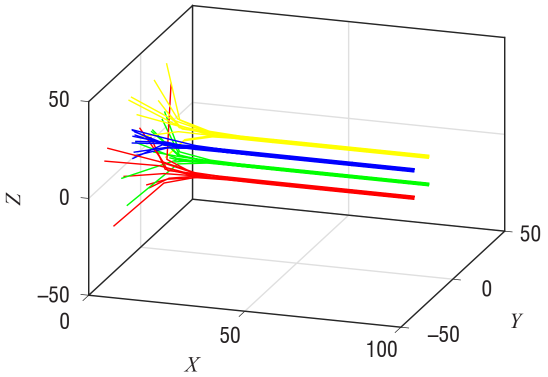


Figure 12 - Variations of the four agents' trajectories with perturbed dynamics.

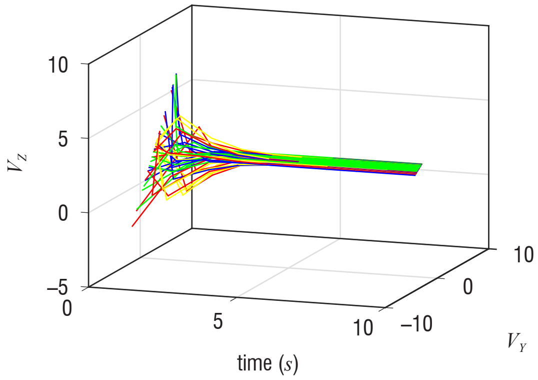


Figure 13 - Variations of the four agents' speed vectors with perturbed dynamics.

In order to evaluate the robustness of the approach to perturbations on the system dynamics, Matrices M and C have been replaced by perturbed matrices, where each element is perturbed with an additional uncertainty of at most 10% of the initial value. As illustrated in Figures 12 and 13, the formation still converges to the consensus.

Conclusion

The *robust consensus-seeking problem* is considered in this paper. Multi-agent systems in which each agent satisfies linear dynamics are considered, and the consensus problem is formulated as a multi-player nonzero-sum differential game. Exact solutions are provided for both finite-horizon and infinite-horizon problems, in terms of coupled Riccati equations. Motivated by the fact that coupled algebraic Riccati equations are, in general, difficult to solve, approximate solutions are provided for the latter. The results are demonstrated by means of two simulation studies. Directions for future research include the consideration of nonlinear systems. Moreover, it is of particular interest to consider distributed settings, in which the control inputs for each agent must be computed subject to communication and information constraints ■

References

[1] A. ASTOL, P. COLANERI - *Trading Robustness with Optimality in Nonlinear Control*. Automatica, 37(12), 1961-1969, 2001.

[2] T. BASAR - *A Counterexample in Linear-Quadratic Games: Existence of Nonlinear Nash Solutions.* Journal of Optimization Theory and Applications, 14(4), 425-430, 1974.

[3] T. BASAR - *On the Uniqueness of the Nash Solution in Linear-Quadratic Differential Games*. International Journal of Game Theory, 5(2-3), 65-90, 1976.

[4] T. BASAR, P. BERNHARD - *H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach.* Springer, 2 edition, 2008.

[5] T. BASAR, G. J. OLSDER - *Dynamic Noncooperative Game Theory.* Academic Press, 1999.

[6] D. BAUSO, T. MYLVAGANAM, A. ASTOLFI - *Crowd-Averse Robust Mean-Field Games: Approximation via State Space Extension*. IEEE Transactions on Automatic Control, 61(7), 1882-1894, 2016.

[7] D. P. BERTSEKAS - *Dynamic Programming and Optimal Control*. Volume 1. Athena scientific Belmont, MA, 2005.

[8] D. CAPPELLO, T. MYLVAGANAM - *Distributed Control of Multi-Agent Systems via Linear Quadratic Differential Games with Partial Information*. IEEE Conference on Decision and Control, 4565-4570, 2018.

[9] D. CAPPELLO, T. MYLVAGANAM - *A Game Theoretic Framework for Distributed Control of Multi-Agent Systems with Acyclic Communication Topologies*. IEEE Conference on Decision and Control, 2019.

[10] S. DJAIDJA, Q. WU, L. CHENG - *Stochastic Consensus of Single-Integrator Multi-Agent Systems Under Relative State-Dependent Measurement Noises and Time Delays*. International Journal of Robust and Nonlinear Control, 27(5), 860-872, 2017.

[11] M. EGERSTEDT, X. HU - *Formation Constrained Multi-Agent Control.* IEEE Transactions on Robotics and Automation, 17(6), 947-951, 2001.

[12] N. C. EKNELIGODA, W. W. WEAVER - *Optimal Transient Control of Microgrids Using a Game Theoretic Approach.* IEEE Energy Conversion Congress and Exposition, 935-942, 2011.

[13] J. ENGWERDA - *LQ Dynamic Optimization and Differential Games.* John Wiley & Sons, 2005.

[14] J. C. ENGWERDA - *A Numerical Algorithm to Calculate the Unique Feedback Nash Equilibrium in a Large Scalar LQ Differential Game*. Dynamic Games and Applications, 7(4), 635-656, 2017.

[15] G. FREILING, G. JANK, H. ABOU-KANDIL - *On Global Existence of Solutions to Coupled Matrix Riccati Equations in Closed-Loop Nash Games*. IEEE Transactions on Automatic Control, 41(2), 264-269, 1996.

[16] Q. JIAO, H. MODARES, S. XU, F. L. LEWIS, K. G. VAMVOUDAKIS - *Multi-Agent Zero-Sum Differential Graphical Games for Disturbance Rejection in Distributed Control.* Automatica, 69, 24-34, 2016.

[17] V. KAPILA, A. G. SPARKS, J. M. BUFFINGTON, Q. YAN - *Spacecraft Formation Flying: Dynamics and Control.* Journal of Guidance, Control, and Dynamics, 23(3), 561-564, 2000.

[18] D. LIMEBEER, B. ANDERSON, B. HENDEL - *A Nash Game Approach to Mixed H2/H∞ Control*. IEEE Transactions on Automatic Control, 39(1), 69-82, 1994.

[19] W. LIN - *Mixed H2/H∞ Control via State Feedback for Nonlinear Systems.* International Journal of Control, 64(5), 899-922. doi:10.1080/00207179608921664, 1996.

[20] M. MESBAHI, F. Y. HADAEGH - *Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching*. Journal of Guidance, Control, and Dynamics, 24(2), 369-377, 2001.

[21] T. MYLVAGANAM - *A Game Theoretic Approach to Distributed Control of Homogeneous Multi-Agent Systems*. IEEE 56th Annual Conference on Decision and Control, 6217-6222, 2017.

[22] T. MYLVAGANAM, A. ASTOLFI - *Control of Micro-Grids using a Differential Game Theoretic Framework.* Proceedings of the 54th IEEE Conference on Decision and Control, 5839-5844, 2015.

[23] T. MYLVAGANAM, A. ASTOLFI - *Dynamic Algorithms for Solving Coupled Algebraic Riccati Equations Arising in Mixed H2/H∞ Control for Scalar Linear Systems.* Proceedings of the 55th IEEE Conference on Decision and Control, 652-657, 2016a.

[24] T. MYLVAGANAM, M. SASSANO A. ASTOLFI - *Constructive ϵ-Nash Equilibria for Nonzero-Sum Differential Games*. IEEE Transactions on Automatic Control, 60(4), 950-965, 2015.

[25] T. MYLVAGANAM, A. ASTOLFI - *A Nash Game Approach to Mixed H2/H∞ Control for Input-Affine Non-Linear Systems*. 10th IFAC Symposium on Nonlinear Control Systems, volume 49, 1024-1029, 2016b.

[26] M. NAZARI, E. A. BUTCHER, T. YUCELEN, A. K. SANYAL - *Decentralized Consensus Control of a Rigid-Body Spacecraft Formation with Communication Delay*. Journal of Guidance, Control, and Dynamics, 39(4), 838-851, 2016.

[27] R. OLFATI-SABER, J. A. FAX, R. M. MURRAY - *Consensus and Cooperation in Networked Multi-Agent Systems*. Proceedings of the IEEE, 95(1), 215-233, 2007.

[28] G. PAPAVASSILOPOULOS, J. MEDANIC, J. J. CRUZ - *On The Existence of Nash Strategies and Solutions to Coupled Riccati Equations in Linear-Quadratic Games*. Journal of Optimization Theory and Applications, 28, 49-76, 1979.

[29] C. POSSIERI, M. SASSANO - *An Algebraic Geometry Approach for the Computation of All Linear Feedback Nash Equilibria in LQ Differential Games*. Proceedings of the 54th IEEE Conference on Decision and Control, 5197-5202, 2015.

[30] C. POSSIERI, M. SASSANO - *On Polynomial Feedback Nash Equilibria for Two-Player Scalar Differential Games*. Automatica, 74, 23-29, 2016.

[31] W. REN, R. W. BEARD - *Distributed Consensus in Multi-Vehicle Cooperative Control*. Springer, 2008.

[32] W. REN, R. W. BEARD, E. M. ATKINS - *A Survey of Consensus Problems in Multi-Agent Coordination.* Proceedings of the American Control Conference, 1859-1864, 2005.

[33] J. SEO, Y. KIM, S. KIM, A. TSOURDOS - *Consensus-Based Reconfigurable Controller Design for Unmanned Aerial Vehicle Formation Flight.* Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 226(7), 817-829, 2012.

[34] S. STANKOVIĆ, N. ILIĆ, Ž. DJUROVIĆ, M. STANKOVIĆ, K. H. JOHANSSON - *Consensus Based Overlapping Decentralized Fault Detection and Isolation*. Conference on Control and Fault-Tolerant Systems, 570-575, 2010.

[35] A. W. STARR, Y. C. HO - *Nonzero-Sum Differential Games*. Journal of Optimization Theory and Applications, 3(3), 184-206, 1969.

[36] M. TURPIN, N. MICHAEL, V. KUMAR - *Trajectory Design and Control for Aggressive Formation Flight with Quadrotors*. Autonomous Robots, 33(1-2), 143-156, 2012.

[37] R. VINTER - *Optimal Control.* Springer Science & Business Media, 2010.

[38] C. WANG, P. K. HADAEGH, K. F. Y. LAU - *Synchronized Formation Rotation and Attitude Control of Multiple Free-Flying Spacecraft*. Journal of Guidance, Control, and Dynamics, 22(1), 28-35, 1999.

[39] P. WANG, F. HADAEGH - *Coordination and Control of Multiple Microspacecraft Moving in Formation*. Journal of the Astronautical Sciences, 44, 1996.

[40] L. XIAO, S. BOYD, S. LALL - *A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus*. Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, 63-70, 2005.

[41] M. D. YAN, X. ZHU, X. X. ZHANG, Y. H. QU - *Consensus-Based Three-Dimensionalmulti-UAV Formation Control Strategy with High Precision.* Frontiers of Information Technology & Electronic Engineering, 18(7), 968-977, 2017.

[42] M. YE, B. D. O. ANDERSON, C. YU - *Bearing-Only Measurement Self-Localization, Velocity Consensus and Formation Control*. IEEE Transactions on Aerospace and Electronic Systems, 53(2), 575-586, 2017.

[43] X. ZHANG, H. DUAN - *Altitude Consensus Based 3D Flocking Control for Fixed-Wing Unmanned Aerial Vehicle Swarm Trajectory Tracking*. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 230(14), 2628-2638, 2016.

AUTHORS

Thulasi Mylvaganam was born in Bergen, Norway, in 1988. She received the M. Eng. degree in Electrical and Electronic Engineering from Imperial College London, UK, in 2010. She completed her Ph.D. degree in the Department of Electrical and Electronic Engineering, Imperial College London, UK, where she was a Research Associate from 2014-2016. From 2016 to 2017 she was a Research Fellow in the Department of Aeronautics, Imperial College London, UK, where she has been a Lecturer (Assistant Professor) since 2017. Her current research interests include nonlinear control design, optimal control, differential game theory and distributed control with applications to multi-agent systems. She is Associate Editor for the IEEE CSS Conference Editorial Board and member of the IFAC Technical Committee on Optimal Control.

Héléne Piet-Lahanier Hélène Piet-Lahanier graduated from SupAero (Toulouse) and obtained both her PhD in Physics and her *Habilitation à Diriger les Recherches* (HDR) from the University Paris XI, Orsay. She is currently Scientific Deputy of ONERA's Information Processing and Systems Department. Her research interests include modelling under uncertainty, bounded error estimation, cooperative guidance and event-triggered estimation with application to multi-agent systems. She is a member of the IFAC Technical Committee on Aerospace.