Runtime mission reconfiguration via decentralized and distributed architectures

Gabriella Fiore, Davide Di Ruscio, Patrizio Pelliccione

I. Introduction

There exists a huge range of applications based on Unmanned Aerial Vehicles (UAVs), both in the civilian and in the military domains. Currently, mission's specification, development, and execution are difficult tasks, thus they are reserved to users having a strong technical expertise. FLYAQ [1],[6] is a framework that enables non-expert users to define high-level missions for a network of UAVs, such as searching for an object in a certain geographical area, or taking pictures while patrolling an area. Once a high-level mission has been specified, the framework automatically generates a detailed flight plan for each UAV of the network. FLYAQ also allows users to specify no-fly zones, obstacles, etc., thus flight plans are generated according to these specifications, while also avoiding collisions among UAVs of the network. Currently, the framework only allows off-line mission definition by means of a centralized architecture. That is, based on a high-level user-defined mission, a Ground Control Station (GCS) sends a detailed flight plan to each UAV that accomplish its tasks only based on this information. For this reason, dedicated flight plans are conservative, in order to avoid vehicles' trajectories collision or predetermined obstacles.

The aim of our current research is to extend the functionalities of FLYAQ by implementing decentralized and distributed control architectures[1], where the high-level mission is concurrently carried out by the team of UAVs, rather than being executed by the GCS in a centralized manner, thus allowing to define and modify mission execution at run-time. In this paper we provide more details on the former approach.

Based on the requirements enforced by the considered applications (e.g., disaster prevention and management, homeland security), high-level missions must guarantee safety and efficiency of operation. As an example, on the one side, we can consider that predicting the future occurrence of critical situations allows performing actions on the system in order to prevent failures, malfunctioning, or abnormal behaviors. On the other side, predicting the forthcoming availability of functional resources allows reconfiguring the system in a more efficient way. Our objective is to integrate the FLYAQ functionalities with run-time mission reconfiguration, when forthcoming critical situations or available resources are predicted, rather than when they are detected (that is, when a failure already occurred). In particular, we want to characterize the possibility of predicting in advance the occurrence of critical situations with the objective of proactively reconfiguring the system.

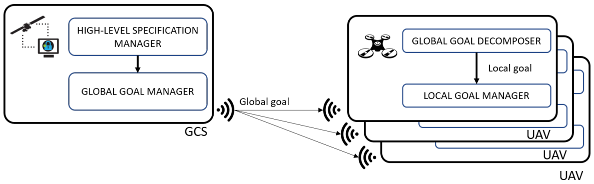

Fig. 1. Decentralized control architecture.

In Section II we describe the decentralized/distributed control architecture for UAVs. In Section III we describe the predictability property. Concluding remarks are offered in Section IV.

II. UAV Control Strategy

Typical scenarios considered in the implementation of the FLYAQ framework are: disaster prevention and management; homeland security; protection of critical infrastructures; networking and communications; and environmental protection [1]. Currently, coordination among UAVs is guaranteed by means of a centralized architecture where a GCS sends a flight plan to each UAV, assuring collision and obstacle avoidance. To prevent collisions between different trajectories, a geographical region is statically assigned to each UAV in an exclusive manner. This brings to a conservative solution, that is FLYAQ does not allow promptly reacting to undesired situations such as the loss of a UAV, moving obstacles, incomplete or inaccurate context specification, or the availability of new functional resources, and any other occurrence requiring to manage a dynamically varying environment.

The purpose of our current research is to endow FLYAQ with two different control architectures, namely: (i) decentralized, when the UAVs are not able to communicate with each other, but they have to agree in advance on the high-level mission to be carried out (see Fig. 1); (ii) distributed, when the UAVs are equipped with additional communication capabilities and they can share information, while carrying out a mission (see Fig. 2). To make a comparison with

G.Fiore, and D. Di Ruscio are with the Department of Information Engineering, Computer Science and Mathematics, Center of Excellence for Research DEWS, University of L'Aquila, 67100, L'Aquila, Italy, email: gabriella.fiore, davide.diruscio@univaq.it.

P. Pelliccione is with the Chalmers University of Technology, University of Gothenburg, Sweden, email: patrizio.pelliccione@cse.gu.se.

[1] We follow standard practice and we refer to a decentralized architecture to indicate the case where a global specification is concurrently performed by a collection of local controllers that are not allowed to communicate. In this case, local controllers have to agree in advance on the specification to enforce. Instead, in a distributed control architecture, local controllers are allowed to communicate (for example, by exchanging information with neighbors).

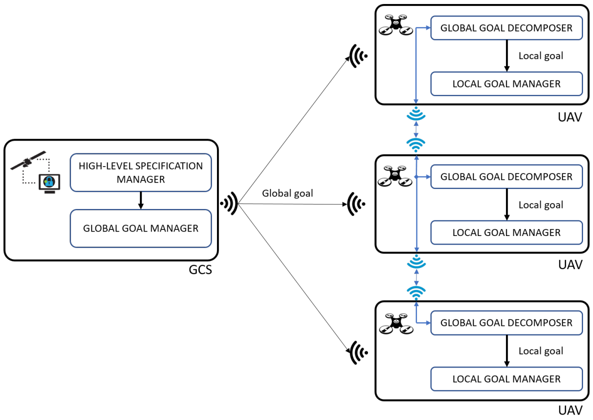

Fig. 2. Distributed control architecture.

the centralized architecture, in the decentralized strategy the high-level mission is sent to each UAV that, in turn, derives its own flight plan, based on the off-line knowledge of the environment, obstacles, the number of UAVs in the team, etc. This strategy is particularly useful when UAVs are not equipped with additional communication capabilities and they cannot exchange information while carrying out the mission. On the contrary, when UAVs are allowed to communicate, as represented in Fig. 2, a distributed control architecture can be implemented. The main advantage with respect to the decentralized case is that UAVs do not need to agree off-line on the high-level mission. The possibility of sharing information among UAVs enable to on-line handle local flight plans, to manage dynamical varying environments, device's failures, and mission's changes.

Depending on the requirements of the considered scenarios, the advantages of the decentralized and distributed architectures with respect to a centralized one are manifold. Indeed, by allowing UAVs to concurrently carrying out high-level missions, they provide greater flexibility and robustness with respect to external disturbances and failures, and they give the possibility of carrying out a mission with relaxed specifications even when the nominal initial conditions are violated.

We propose to exploit a decentralized symbolic control strategy (see [4], [5] for more details), based on formal methods and discrete abstractions of continuous systems. The main feature of this approach is that it relies on the discrete abstraction of a continuous system, where an abstract state corresponds to an aggregate of continuous states. Once a discrete abstraction has been derived, well developed methodologies for supervisory control of discrete event systems can be applied. Also, the choice of applying a symbolic control strategy is motivated by the fact that symbolic methods allow enforcing complex specifications expressed as regular languages or in linear temporal logic that are difficult to enforce by means of classical control theory techniques [5]. The main contribution of the strategy we propose is to complement the FLYAQ functionalities with run-time mission reconfiguration to enhance safety and efficiency properties, thus in the next section we introduce the notion of predictability in terms of both critical situations and forthcoming availability of functional resources.

III. Critical predictability for mission reconfiguration

As teams of UAVs usually operate in safety-critical environments with humans in-the-loop, one of the main challenge is to ensure safety of the overall system, and to increase its robustness with respect to degradations, failures, and any other condition affecting the stability of the operation. In this scenario predicting the future occurrence of particular states of interest is of paramount importance. This subset of states is called critical set and it may represent faulty states, unsafe operations or, more generally, any subset of states which is of particular interest from the system's behavior point of view. We refer to the predictability of both abnormal behaviors (to perform actions on the system in order to prevent critical situations such as failures) and forthcoming availability of functional resources (to reconfigure the system in a more efficient way).

Predictability has been studied for both continuous systems and Discrete Event Systems (DES), see e.g. [3], and references therein. As pointed out in [7], the problem of verifying the predictability property in a decentralized manner for networks of DES running synchronously has not been addressed, yet. Motivated by this need, in [2] the authors consider the decentralized predictability for networks of DES. A decentralized architecture for critical predictors of the network is proposed, in the sense that prediction is performed by a collection of local predictors, each associated with a DES in the network. Furthermore, by applying bisimulation theory, the original network is reduced to a smaller one with the positive effect of lowering complexity. In particular, equivalence classes on the set of DES are defined, based on the bisimulation relation, and a representative DES is associated with each equivalence class. Finally, the critical predictability property can be tested on the quotient network composed by representatives DES, that is smaller than the original one (when a bisimulation relation can be found between different DES). When the network is critically predictable, [2] proves that a network of critical predictors (each one associated with an FSM) is equivalent to a centralized predictor.

IV. Conclusions

The strategy we propose in this extended abstract to improve the FLYAQ functionalities is sound as it is based on: (i) results derived in [4] and [5] for decentralized control of networks of symbolic models approximating nonlinear systems; and (ii) results derived in [2] for predictability of networks of FSMs. We point out that the main advantage offered by the predictability property is to promptly and efficiently reconfigure mission control when a critical situation is predicted, both in terms of forthcoming failures of available resources. Despite the fact that here we focus our attention on team of UAVs, the approach we propose can be used for any multi-agent system (e.g., multi-robots system). Indeed, our future research will be devoted to the implementation of the proposed strategy, and to the evaluation of its effectiveness by simulation on real robots.

References

[1] D. Bozhinoski, D. D. Ruscio, I. Malavolta, P. Pelliccione, and M. Tivoli. Flyaq: Enabling non-expert users to specify and generate missions of autonomous multicopters. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 801–806, Nov 2015.

[2] G. Fiore, G. Pola, E. De Santis, M. D. Di Benedetto, D. Di Ruscio, and P. Pelliccione. Decentralized predictability of networks of Finite State Machines. Submitted, 2018.

[3] S. Genc and S. Lafortune. Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, 45(2):301 – 311, 2009.

[4] G. Pola, P. Pepe, and M. D. Di Benedetto. Symbolic models for networks of control systems. *IEEE Transactions on Automatic Control*, 61(11):3663–3668, Nov 2016.

[5] G. Pola, P. Pepe, and M. D. Di Benedetto. Decentralized supervisory control of networks of nonlinear control systems. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2017.

[6] D. D. Ruscio, I. Malavolta, P. Pelliccione, and M. Tivoli. Automatic generation of detailed flight plans from high-level mission descriptions. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, MODELS '16, pages 45–55, New York, NY, USA, 2016. ACM.

[7] X. Yin and S. Lafortune. Verification complexity of a class of observational properties for modular discrete events systems. *Automatica*, 83:199 – 205, 2017.