Cooperative UAV-UGV modeled by Petri Net Plans specification

Andrea Bertolaso and Masoume M. Raeissi and Alessandro Farinelli and Riccardo Muradore [1]

1 Problem Description: UAV/UGV Cooperative Landing Scenario

Aerial and ground vehicles working in corporation are crucial assets for several real world applications, ranging from search and rescue to logistics. Here we turn our attention to high level languages or team plans [3, 2, 4, 1] for modeling a cooperative scenario, where the UAV/UGV team should operate in tight cooperation to perform a joint task. In this context, by tight coordination we mean that robots must continuously synchronize their individual actions to successfully perform the joint task. This is because the joint task imposes execution constraints to a vehicles that might depend on the state of the other vehicle. For example, in our landing scenario, the UAV must know the intended future locations of the UGV to properly plan a trajectory so to smoothly land on the UGV.

In more detail we investigate the use of the Petri Net Plans (PNPs) specification [4] to specify the collaborative landing task. There are several benefits related to the use of the PNPs framework: first it provides a rich graphical representation that helps the designers to create plans with minimal effort, second the generated plans can be monitored during the execution, third PNPs support well-defined structures for handling tight coordination and on-line synchronization in multi robot systems.

In summary, our main contributions to the state of the art are as follow:

- we use an advanced framework for multi agent plan specification to design a complex cooperative behavior in multi robot systems. Specifically, we design an effective strategy for cooperative landing for our UAV/UGV system that is able to recover from unexpected situations (i.e., sudden deviation of the UGV from the planned trajectory). To the best of our knowledge this is the first application of a team-oriented plan specification framework to a complex cooperative control scenario such as cooperative landing.

- We evaluate our approach in a realistic simulation environment using state of the art tools for robot control and simulation. Specifically, we use ROS to connect and control the simulated platforms, and V-REP to simulate the two platforms and the environment.

2 Simulation and Evaluation

We model the cooperative UAV-UGV scenario described above by using PNPs which consists of three phases:

1. both the UGV and UAV are moving according to their specific and non-cooperative tasks;
2. the UAV approaches the UGV (*flyFar* action using the PNP terminology);
3. the UAV lands on the UGV (*flyClose* action using the PNP terminology).

In Phase 2 the UAV is using its sensing system (e.g. camera) to locate the UGV and plans the faster trajectory to approach the UGV. In this phase the UGV in not aware of the intention of the UAV and so it is continuing its task as in Phase 1. In Phase 3, the UAV is close to the UGV and information are exchanged between them: the UGV is getting aware of the intention of the UAV and so it decreases its velocity and sends to the UAV its planned trajectory to easier the landing. This means that the UGV is still pursuing its objective (e.g. patrolling an area) but in a slower way.

We used JARP for designing the petri net plan; however, any available graphical tool that supports pnml file format could be used. Actions' name and all external conditions have been defined in the plan.

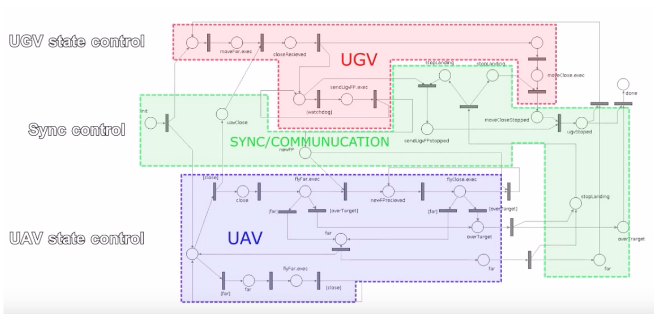


Figure 1. Petri Net plan created for modeling UAV/UGV cooperative landing task.

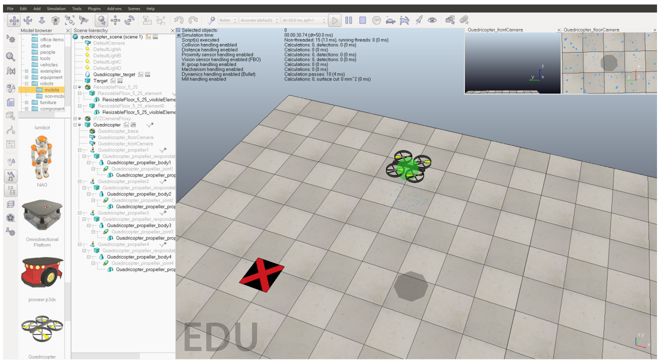
In PNPs actions represent robot behaviors, for example in our case the *flyFar* action represents the UAV flying towards the UGV constantly following its position. Conditions are external events and need to be checked at run time. The possibility to define conditions is a powerful feature that allows to enrich the plan behavior at run time. The simplified version of the petri net designed for our coordination problem is shown in figure 1. Three different colors are used to highlight the tasks of UGV, UAV and the synchronization part.

After designing the plan, we have to hand coding actions and conditions, and makes them available to the PNPros system which connects the PNP library with ROS. Actions will be implemented using the ROS Action-lib interface.

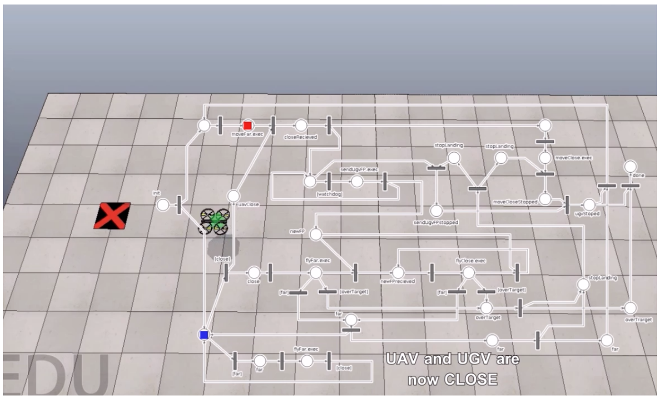
[1] Computer Science Department, University of Verona, Italy, email: andrea.bertolaso@studenti.univr.it, masoume.raeissi@univr.it, alessandro.farinelli@univr.it, riccardo.muradore@univr.it

The following software tools are integrated to implement and execute the above mentioned plan within the Robot Operating System (ROS) middleware:

- JARP: it is a graphical interface for creating the Petri net plan,
- PNP: it is a library that processes pnml file and executes the plan,
- PNPros: it is the bridge between the PNP library and ROS that allows the execution of PNP in ROS using the actionlib module. It can be used for implementing different actions and for defining the firing rules for transitions,
- V-REP: it is used for the visualizing and simulation of the environment,
- Reflexxes: it is a library that computes the UAV trajectories.



(a) UAV flies toward UGV



(b) UAV and UGV gets close

Figure 2. V-REP environment setup for simulating the cooperative landing task.

V-REP, a realistic simulation environment running on Linux operating system, is used for executing the developed plan. Communication with V-REP is possible through the ROS topics. When the simulation environment and the system that handles the plan are launched, the initial position of UAV and UGV is retrieved from V-REP via ROS topics. With these information the plan can start its progress by using PNP library. PNP library communicates with PNPros to

- start new actions (a thread is launched for each action),
- check external conditions based on the environmental knowledge that is available thanks to PNPros,
- interrupt a running action.

A video showing the complete execution of the plan can be seen at `https://goo.gl/hiZIKP`.

The video starts while the vehicles are set far from each other. Then it shows how the coordination can happen when UAV decides to land on UGV. At that point the UGV tries to decrease its speed meanwhile sends its future position to the UAV periodically. UAV must be informed of UGVs future position in order to follow the new location of UGV, for example consider a situation where the UAV is initiating the landing maneuver. If the UGV must suddenly change its current trajectory (i.e., due to a moving obstacle) the UAV should smoothly adapt its plan to recover from a possible failure. The whole system keeps on running until a final state in the Petri Net Plan is reached. Figure 2(a) shows a snapshot of the simulation environment while figure 2(b) displays the state when the UAV is flying toward the UGV.

Overall, the video illustrates that the coordination between the two vehicles is not a one-step synchronization action but it is a continuous behavior. The simulation results confirm that our proposed model is successfully works for such a tight coordination system.

3 CONCLUSION

The problem addressed in this work is a particular kind of collaboration between heterogeneous autonomous vehicles: the landing of an UAV on an UGV. We model the execution of this task by exploiting the power of the PNPs framework. The PNPs framework provides different structures for handling interruption and synchronization behavior which makes modeling and monitoring of the plans easier specifically when tight coordination among team members exist. Simulation results show that our system can properly monitor the joint mission carried out by the UAV/UGV team, hence confirming that the use of a formal planning language significantly helps in the design of such complex scenarios

REFERENCES

[1] Alessandro Farinelli, Nicoló Marchi, Masoume M. Raeissi, Nathan Brooks, and Paul Scerri, 'A mechanism for smoothly handling human interrupts in team oriented plans', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 377–385, Richland, SC, (2015). International Foundation for Autonomous Agents and Multiagent Systems.

[2] Gal A. Kaminka and Inna Frenkel, 'Flexible teamwork in behavior-based robots', in *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pp. 108–113, (2005).

[3] Milind Tambe, 'Towards flexible teamwork', *Journal of Artificial Intelligence Research*, 83–124, (1997).

[4] V.A. Ziparo, L. Iocchi, PedroU. Lima, D. Nardi, and P.F. Palamara, 'Petri net plans', *Autonomous Agents and Multi-Agent Systems*, **23**(3), 344–383, (2011).