Continuous Planning for the Control of an Autonomous Agile Satellite

Grégory Beaumet
Supaéro, ONERA, CNES
2, avenue Édouard Belin 31055 Toulouse cedex 4 FRANCE
gregory.beaumet@onera.fr

Abstract

Because of the complexity of the encountered problems, controlling autonomous satellites is an interesting field for the AI research community. This document introduces the current thesis about planning the activities of an agile autonomous Earth-observing satellite.

Application Domain

Mission

The application domain of the thesis is an ONERA-CNES project of development of a ground demonstrator of an autonomous satellite (AGATA project, (Charmeau & Bensana 2005)). In this project we consider an Earth-observing mission. A satellite on a heliosynchronous low circular orbit around the Earth aims to acquire images of specified areas on the Earth surface, and to download them to one or more ground mission centers.

Agile satellite

The satellite that we consider is an agile satellite, like the Pleiades satellites (Boussarie & Boissin 2006), able to operate freely and quickly along the three axes of rolling, lacing and pitching (Figure 1) thanks to a cluster of gyroscopic actuators.

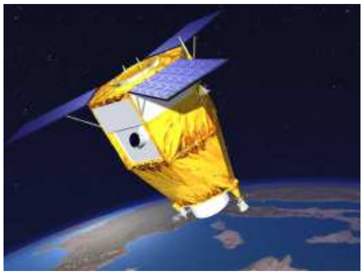

Figure 1: A Pleiades satellite

This satellite is equipped with (1) an optical high-resolution instrument to acquire images, (2) a cloud cover detection instrument, (3) a radio antenna allowing the satellite to download the observation data, and (4) solar generators and batteries producing and storing electric energy. These components are not mobile: they are fixed to the satellite. Finally, the satellite has a fixed size mass memory to save the detection and observation data.

Planning the activities

Currently, the activities of the Spot satellites are planned offline, as will be those of the Pleiades satellites: the ground mission center builds plans over a horizon of 24 hours and downloads them daily to the satellites. These plans are very precise: fixed schedule of activities with fixed starting times. They are executed without any possibility for replanning.

Spacecraft Autonomy

Potential advantages of autonomy

During a revolution period round the Earth, an Earth-observing satellite has limited visibility windows with the ground stations. Autonomy would allow the satellite to make decisions between two visibility windows in order to react to unforeseen events such as:

- **subsystem failure**. The autonomy allows the system to react immediatly if a failure arises during the execution of a task.
- **unexpected level of resources**. Some actions of the satellite have nondeterministic effects on the consumption or the production of the onboard resources: for example, it is impossible to foresee the quality of an image and its compression rate before its realization, and thus to know the memory space it will use. Autonomy would make the satellite able to make decisions by knowing the actual current state of the onboard resources.
- **unexpected cloud cover**. The detection instrument may detect a cloud cover different from that provided by the weather forecast, authorizing or preventing some observations. Because the detection can be performed by pointing the satellite 30 degrees ahead, it must decide autonomously within a few seconds wether to add or remove these observations from its actions plan.

State of the art

The EO-1 Autonomous Science Agent This software enables the Earth-Observing One (EO-1) spacecraft to autonomously detect and respond to science events occuring on the Earth. It is organized into a traditional three-layer architecture. At the highest level of abstraction, the Continuous Activity Scheduling Planning Execution and Replanning (CASPER) software is responsible for mission planning functions. CASPER uses a local search approach (Rabideau *et al.* 1999) to develop operations plans. It plans within limited CPU resources by using a hierarchical, continuous (Chien *et al.* 2000) planning paradigm. Rather than attempt to plan out an entire week of operations in a single batch timeslice, it utilizes a long-term, more abstract plan for the longest planning horizon (one week), and plans at a detailed level for the next day of operations. As time proceeds forward, it incrementally replans for the new observations that fall within this one-day horizon.

Non-agile satellite The work presented by S. Damiani (Damiani 2005; Damiani, Verfaillie, & Charmeau 2005) allowed us to design, implement and test successfully an autonomous decision mechanism onboard a non-agile satellite. It is supported by a permanently active planning module, reasoning on more and more complex problems to improve quality of the proposed decisions, using all the time it has at its disposal, but able to provide a realizable decision at any time, even if it is not necessarily optimal according to the principles of the anytime algorithms (Zilberstein 1996).

Application to an agile satellite The observation instrument of a non-agile Earth-watching satellite like Spot is permanently pointed under the satellite, and a mobile mirror in front of it allows it to observe ground areas laterally. The starting times of observations are thus fixed.

On the contrary an agile satellite is able to bring forward or delay the starting time of an observation by a simple change of its attitude; then the observations have starting time windows which relax planning but make the selection and scheduling of observations significantly more difficult, due to the larger search space for potential solutions (Figure 2).

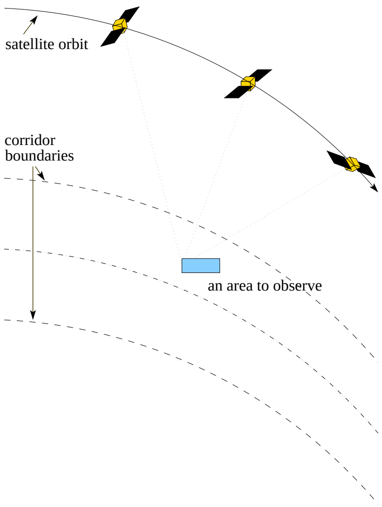

Figure 2: Three possible attitudes of an agile satellite for starting an observation

Some works (Lemaître *et al.* 2002) deal with the problem of offline selecting and scheduling observations of agile satellites. They present different methods which have been investigated in order to solve a simplified version of the complete problem: a greedy algorithm, a dynamic programming algorithm, a constraint programming approach and a local search method.

However, these works deal with the planning of the observations independently of the other activities of the satellite (cloud cover detection, data downloading...). But many activities of an agile satellite need to control its attitude which can be seen as a shared resource. For example, an observation cannot be executed in parallel of a cloud cover detection which requires an orientation of the satellite 30 degrees ahead. Thus it becomes necessary to plan together all the activities controlling the attitude of the satellite.

Contribution

Objective

This study aims to extend the work developed by S. Damiani (Damiani 2005) to an agile satellite: permanent planning task of all the activities of the satellite by using as well as possible the time available to reason.

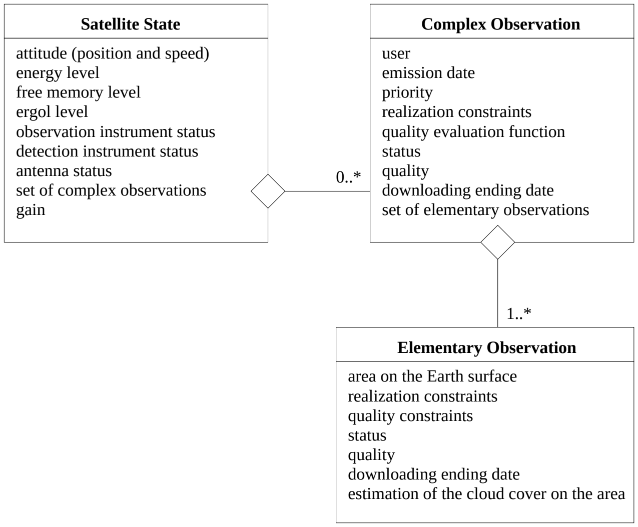

Figure 3: Model of the current state of the system

Achieved work

List of possible activities We distinguish two categories of activities realizable by the satellite: activities with controlled attitude trajectory during which the attitude of the satellite, in position and speed, is entirely determined, and activities with uncontrolled attitude trajectory which can be executed parallel with the other activities.

We listed seven activities with controlled attitude trajectory: the *observation* of an area on the Earth surface, the *detection* of the cloud cover in front of the satellite, the *recharge* of the batteries (to point the solar panels to the sun), the *downloading* of observation data (to point the satellite to the ground station), a *change in attitude*, an *orbital manoeuvre* (to correct the orbit of the satellite if necessary), a *geocentric pointing* (when the satellite "does nothing" or is in safety mode) and three activities with uncontrolled attitude trajectory: the *parallel downloading* of observation data, the *analysis* of the results of an observation (to evaluate the quality of an image, compress or delete the saved image), the *analysis* of the results of a detection (to evaluate the cloud cover).

Model The first step of our work consisted in modelling the decision problem by using the PDDL language and its extension to the durative actions (Fox & Long 2003): the current state of the system (Figure 3), the various actions realizable by the satellite, their preconditions, their effects (deterministic or not) on the state of the system and on the satisfaction of the objectives.

The figure 4 presents the model of the "observe" action using the PDDL language. The satellite starts watching the area $?o_i$ at the date $?t_s$.

```
(: durative-action observe
 : parameters (?o_i - observation ?t_s - date)
 : duration (= ?duration (observationDuration ?o_i ?t_s))
 : condition (and (at start (= (status ?o_i) notAcquired))
                  (at start (visible ?o_i))
                  (at start (= attitude (obsStartAttitude ?o_i ?t_s)))
                  (at start (= obsInstrStatus available))
                  (at start (not assignedAttitude))
                  (at start (≥ energy (energyConsum ?o_i ?t_s)))
                  (at start (≥ memory (memoryConsum ?o_i)))
                  (over all (visible ?o_i))
                  (over all (= obsInstrStatus used))
                  (over all (assignedAttitude))
                  (over all (≥ energy 0))
                  (over all (≥ memory 0))
                  (at end (= obsInstrStatus used))
                  (at end (assignedAttitude))
                  (at end (visible ?o_i))
                  (at end (≥ energy 0))
                  (at end (≥ memory 0))

 : effect (and (at start (decrease energy (energyConsum ?o_i ?t_s)))
               (at start (decrease memory (memoryConsum ?o_i)))
               (at start (assign obsInstrStatus used))
               (at start (assignedAttitude))
               (at end (assign attitude (obsEndAttitude ?o_i ?t_s)))
               (at end (assign (status ?o_i) acquired))
               (at end (assign obsInstrStatus available))
               (at end (not assignedAttitude))
               (at end (increase energy (energyProd ?o_i ?t_s)))
         )
)
```

Figure 4: Model of the "observe" action

Current work

To plan online, we need to estimate (1) the duration of each activity of the satellite, depending on its starting time and on the attitude profile of the satellite and (2) the production and the consumption of energy and memory for each activity of the satellite. For that, we try to compare two different methods : an analytical one using a simplified model of the satellite kinematics, and a learning one based on the use of neural networks to approximate the quantities of interest.

Future work

We plan to solve, initially offline, this problem of planning with a dynamic programming approach like this one used by S. Damiani (Damiani 2005), then with a local search method. A second step will consist in adapting the algorithms to a mode of anytime reasoning in order to be able to use them online.

References

Boussarie, E., and Boissin, B. 2006. Pléiades - Dual Optical System for Metric Resolution Observations. http://smsc.cnes.fr/PLEIADES/.

Charmeau, M.-C., and Bensana, E. 2005. AGATA: A Lab Bench Project for Spacecraft Autonomy. In *8th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-05)*.

Chien, S.; Knight, R.; Stechert, A.; Sherwood, R.; and G., R. 2000. Using iterative repair to improve responsiveness of planning and scheduling. In *Fifth International Conference on Artificial Intelligence Planning and Scheduling*.

Damiani, S.; Verfaillie, G.; and Charmeau, M.-C. 2005. An earth watching satellite constellation : How to manage a team of watching agents with limited communications. In *Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2005)*, volume 1, 455–462.

Damiani, S. 2005. *Gestion d'une constellation de satellites de surveillance de la Terre : autonomie et coordination*. Ph.D. Dissertation, Supaéro, Toulouse, France.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research (JAIR)* 20:61–124.

Lemaître, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.-M.; and Bataille, N. 2002. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology* 6:367–381.

Rabideau, G.; Knight, R.; Chien, S.; Fukunaga, A.; and Govindjee, A. 1999. Iterative repair planning for spacecraft operations in the aspen system. In *International Symposium on Artificial Intelligence Robotics and Automation in Space*.

Zilberstein, S. 1996. Using anytime algorithms in intelligent systems. *AI Magazine* 17(3):73–83.