Estimation of the Minimal Duration of an Attitude Change for an Autonomous Agile Earth-Observing Satellite

Grégory Beaumet[1], Gérard Verfaillie[1], and Marie-Claire Charmeau[2]

[1] ONERA
2 avenue Édouard Belin, BP 74025, 31055 Toulouse Cedex 4, France
{Gregory.Beaumet,Gerard.Verfaillie}@onera.fr
[2] CNES
18 avenue Édouard Belin, 31401 Toulouse Cedex 9, France
Marie-Claire.Charmeau@cnes.fr

Abstract. Most of the currently active Earth-observing satellites are entirely controlled from the ground: observation plans are regularly computed on the ground (typically each day for the next day), uploaded to the satellite using visibility windows, and then executed onboard as they stand. Because the possible presence of clouds is the main obstacle to optical observation, meteorological forecasts are taken into account when building these observation plans. However, this does not prevent most of the performed observations to be fruitless because of the unforeseen presence of clouds. To fix this problem, the possibility of equipping Earth-observing satellites with an extra instrument dedicated to the detection of the clouds in front of it, just before observation, is currently considered. But, in such conditions, decision upon the observations to be performed can be no longer made offline on the ground. It must be performed online onboard, because it must be performed at the last minute when detection information is available and because visibility windows between Earth-observing satellites and their control centers are short and rare. With agile Earth-observing satellites which are the next generation ones, decision-making upon observation requires the computing of an as short as possible attitude trajectory allowing the satellite to point to the right ground area within its visibility window. In this paper, we show the results of an experiment consisting in using a continuous constraint satisfaction problem solver (RealPaver) to compute such optimal trajectories online onboard.

1 The Problem

1.1 Applicative Context

Earth-observing satellites are placed on heliosynchronous low altitude circular orbits around the Earth. Most of them are equipped with optical observation instruments, with a mass memory able to record observation data, and with a high-rate antenna able to download it towards ground mission centers. When

C. Bessiere (Ed.): CP 2007, LNCS 4741, pp. 3–17, 2007.

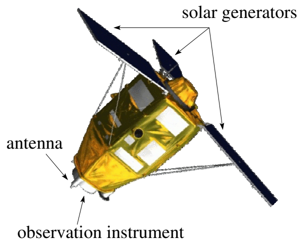
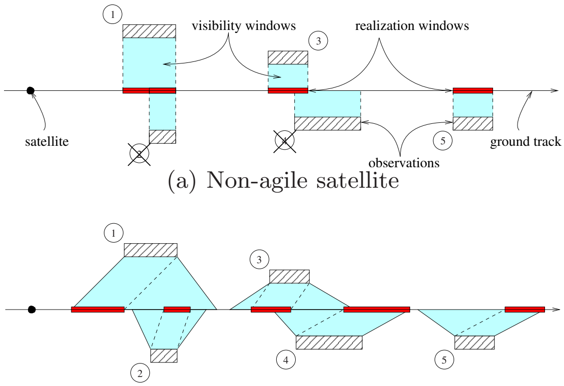


Fig. 1. A PLEIADES satellite

they are not agile, that is keeping pointing to the Earth center, as the current French SPOT satellites do, they can observe specific ground areas thanks to mirrors placed in front of their instruments and orientable along the roll axis. When they are agile, that is keeping controlling their attitude movement along the roll, pitch, and yaw axes thanks to reaction wheels or to gyroscopic actuators, as the future French PLEIADES satellites will do, this is the whole satellite which orientates itself to observe specific ground areas (see Figure 1; for more details, see http://smsc.cnes.fr/PLEIADES/).

The main advantage of *agility* is to offer more freedom in terms of observation. With non agile satellites, the realization window of an observation is fixed, because observation is only possible when the satellite flies over the target ground area. With agile ones, this realization window can be freely chosen within a wide visibility window, because observation is now possible before, when, or after the satellite flies over the target ground area. This freedom may allow more observations to be performed because observations which conflict with each other in the context of non agile satellites may no longer conflict in the context of agile ones. This is illustrated by Figure 2 which represents five candidate observations, from 1 to 5. With a non agile satellite (see Figure 2(a)), observations 1 and 2 conflict with each other, because their realization windows overlap. This is also the case with observations 3 and 4. As a result, it is only possible to perform three observations, for example observations 1, 3, and 5. With an agile satellite (see Figure 2(b)), all these conflicts can be resolved and the five observations can be performed.

Agile or not, these satellites are assumed to perform observations following observation requests emitted by users. With each request, are usually associated a ground area, a priority, and some observation conditions. For each candidate observation, it is then necessary to decide on whether, when, and how it will be performed. For almost all the currently active Earth-observing satellites, these decisions are made *on the ground* under the supervision of human operators in centralized mission centers. Typically, an observation *plan* is built each day for the next day, taking into account the current set of candidate observations (see [1] for a description of the problem in the context of a non agile satellite and [2] for a similar description in the context of an agile one). The resulting optimization problem may be huge and hard to be solved, even when using powerful computers and sophisticated algorithms running for tens of minutes.



(a) Non-agile satellite

(b) Agile satellite

Fig. 2. Agile *vs* non-agile satellite

When a plan has been produced, it is uploaded to the satellite using a visibility window and executed as it stands without any execution freedom. The only notable exception to this organizational scheme is the US EO-1 satellite which is equipped with onboard planning and replanning capabilities [3].

Unfortunately, optical observation is sensitive to clouds and it is very difficult to foresee the actual presence of clouds the next day over a given ground area. Despite the use of meteorological forecasts by the planning algorithms, many observations (until 80%) remain fruitless due to the presence of clouds. To fix this problem, engineers are currently considering the possibility of equipping Earth-observing satellites with an extra instrument dedicated to the detection of the clouds in front of it, just before observation becomes possible. But, if one wants to exploit information coming from the detection instrument, it is no longer possible to decide upon observations offline each day for the next day. Decisions must be made *online* at the last minute, when detection information is available. Moreover, because Earth-observing satellites are not often within the visibility of their control centers, decisions can no longer be made on the ground. They must be made *onboard* the satellite. The previously offline long-term optimization problem, using unreliable information about cloud presence, becomes an online short-term decision problem, using far more reliable information about cloud presence. One may hope that information reliability will compensate for a quick decision performed over a limited temporal horizon and that globally more fruitful observations will be performed.

This is such a scenario we are currently working on: an agile PLEIADES-like satellite equipped with a cloud detection instrument and with autonomous decision capabilities. This work can be seen as an extension to the case of agile satellites of a similar work performed in the context of non agile ones [4]. It is one of the pieces of the joint ONERA[1]-CNES[2] AGATA[3] project, which aims at developing techniques for improving spacecraft autonomy [5].

1.2 Planning Problem

The global planning problem consists in optimizing the mission feedback throughout the whole satellite life, that is the observations that are performed and delivered to users, taking into account their number and their priorities.

One practical way of approaching this objective consists in optimizing the mission feedback each time over a limited temporal horizon ahead, taking into account the available information about the satellite state (orbital position, attitude, available energy and memory . . .), the candidate observations, the meteorological forecasts, and the cloud detection, and in optimizing again each time the current temporal horizon ends or changes occur. In the particular case of an agile Earth-observing satellite, possible actions are:

– the *observation* of any ground area;
– the *downloading* of any observation data by pointing the satellite and thus its antenna to any ground station;
– the *detection* of clouds in front of the satellite by pointing the satellite 30 degrees ahead;
– the *recharge* of its batteries by orienting its solar generators towards the Sun;
– a *geocentric pointing* when it "has nothing to do" or must remain in a safety mode;
– an *orbital manoeuvre* when the drift of its orbital trajectory from the reference orbit becomes too important.

All these actions impose strong constraints on the satellite attitude and on its attitude speed. They are thus systematically in conflict with each other and must be executed *sequentially*, with a necessary *attitude movement* between two successive ones.

The basic decision problem is thus to decide before the current action ends which action to perform next. To make such a decision, one must be able to assess the feasibility and the effects of each candidate action, including the attitude movement necessary to start it.

In planning problems, assessing the feasibility and the effects of a candidate action is usually easy and what is difficult is to make the right choice among all the feasible candidates. In our problem, this assessment is itself difficult, because it requires reasoning on the attitude movement necessary to start the candidate action and then to perform it.

Considering only the attitude movement necessary to start a candidate action a and only temporal constraints enforcing that a must start within a given temporal

[1] ONERA: the French Aerospace Lab, http://www.onera.fr
[2] CNES: the French Space Agency, http://www.cnes.fr
[3] AGATA: Autonomy Generic Architecture, Tests and Applications, http://www.agata.fr

window w, one must be able to check whether or not a can start within w. If this is possible, it may be moreover useful to compute the minimal time t at which a can start within w. This is the planning subproblem we focused on.

1.3 Attitude Change Subproblem

This attitude change subproblem can be formulated as follows. Let be:

– an *initial state* specified by the attitude and the attitude speed at the end of the current action;
– a *goal state* specified by constraints on the time, the attitude, and the attitude speed at the beginning of the next action;
– *constraints on the attitude movement* such as maximal moments and torques in terms of attitude.

The decision problem consists in checking if a goal state is reachable. If the answer is positive, the associated optimization problem consists in computing the minimal time at which this is possible.

Constraints on the goal depend on the nature of the next action to be performed. For example, if the next action is a Sun pointing, the goal must be reached before the satellite be in eclipse, the goal attitude is fixed, and the associated speed is consequently null (see Section 3.2). If the next action is a data downloading towards any ground station, the goal must be reached before the end of the visibility window between the satellite and the station, and the goal attitude and attitude speed are functions of the time at which the goal is reached because the satellite is moving on its orbit and the Earth is rotating on itself. In fact, the problem can be seen as the problem of tracking a mobile target from a mobile engine. Things are similar if the next action is an observation of any ground area a, with only extra constraints on the goal attitude and attitude speed to allow the line of sensors of the observation instrument to scan a in the right direction at the right speed (see Section 3.3).

To simplify the problem, we decompose the attitude movement into *three parallel movements*, each one performed along one axis (roll, pitch, or yaw) using one actuator. Moreover, we enforce that each of these movements be decomposed into *three successive phases* (see Figure 3):

– a first phase with a *constant torque* T_1;
– a second phase with a *constant moment* L_2;
– a third phase with a *constant torque* T_3.

Such a strategy can be justified as follows. To get a given goal attitude as soon as possible, it is necessary to start with an acceleration until a maximum speed (first two phases). But, to get at the same time a given goal attitude speed, it may be necessary to end with a deceleration phase (third phase). Each of these phases may be empty. To get an intuition of the problem at hand, think that you are driving a car at a given speed on a highway and that you want to go as soon as possible behind the car of a friend who is driving at another speed.

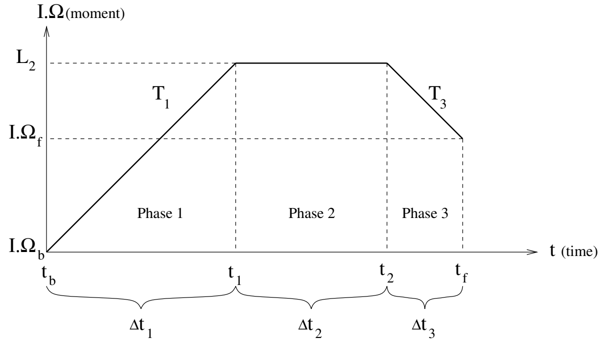


Fig. 3. The three successive attitude movement phases along one axis

The start time t_b of the first phase and the end time t_f of the last one must be obviously equal along the three axes. But in addition, in order to limit the disturbances due to the instantaneous acceleration/deceleration changes, we enforce that the phase commutation times t_1 and t_2 be equal along the three axes too.

The problem consists now in determining the three common *phase end times* t_1, t_2, and t_f, and, along each axis, the *torques* T_1 and T_3 in the first and third phases and the *moment* L_2 in the second one. This is this problem we tried to model as a continuous CSP problem and to solve using a continuous CSP solver.

2 Why Constraint Programming ?

2.1 Existing Methods

Numerical methods exist to solve this kind of problem. In particular, CNES has developed a library called MANIAC [6] which includes an algorithm dedicated to the computing of attitude changes for families of satellites whose attitude is controlled thanks to reaction wheels (MYRIADE, PROTEUS, ...). This algorithm allows the feasibility of an attitude change of a given duration between an initial attitude and a observation goal attitude to be checked. The minimal duration of such an attitude change is obtained thanks to an iterative call to this algorithm. In the context of the future PLEIADES satellites whose attitude is controlled thanks to gyroscopic actuators, CNES has developed a similar library called GOTlib. However, these methods have several drawbacks:

- they have been developed to be used *on the ground* in the context of the regular production of activity plans; they are time consuming and cannot be used in an online context as they stand;
- they consist in searching the values for which a function gets null thanks to *gradient-based* mechanisms such as Newton and Lagrange algorithms; they offer consequently no optimality guarantee in terms of attitude change duration;

– they use *no explicit model* of the problem to solve: initial and goal states, constraints on the attitude movement; this model is hidden in the algorithms.

2.2 Why Considering Constraint Programming ?

Interval analysis has been already used in robotics [7]. In the same direction, we explored the use of constraint programming to remedy the drawbacks of the existing methods. The main *a priori* advantages of such an approach are:

– the use of an *explicit model* of the problem, what is consistent with the model-based approach followed by the AGATA project at all the levels of the autonomous control of a spacecraft, for both situation tracking and decision;
– the ability to use the same model to answer *various requests*, by changing only the optimization criterion (see Section 3.5);
– the ability to get the *global optimum* and not only a local one, or at least a *bound on this optimum*, for example a lower bound on the minimal duration of an attitude change, allowing the decision algorithm to remove infeasible candidate actions.

3 How Constraint Programming ?

3.1 Kinematic Models

The first task is to build a simplified kinematic model of the various bodies which come into play (Sun, Earth, and satellite): movement of the Earth around the Sun and on itself, of the satellite orbit around the Earth, and of the satellite on its orbit and on itself.

Reference Frames. Let $R_\phi(\alpha)$, $R_\theta(\alpha)$, and $R_\psi(\alpha)$ be the rotation matrix of any angle α around respectively the first, the second, and the third base vector. Let $\boldsymbol{u}^{\boldsymbol{R_i}}$ be the vector $\boldsymbol{u}$ expressed in any reference frame R_i. When no reference frame is specified, the vector is assumed to be expressed in the inertial reference frame R_c: $\boldsymbol{u} = \boldsymbol{u}^{\boldsymbol{R_c}}$.

The chosen inertial reference frame is the geocentric equatorial reference frame $R_c = \{\boldsymbol{x_c}; \boldsymbol{y_c}; \boldsymbol{z_c}\}$, with $\boldsymbol{z_c}$ in the direction of the Earth inertial rotation vector and $\boldsymbol{x_c}$ pointing to the vernal equinox point[4] γ.

For any ground area to observe starting in point M, we define the reference frame $R_a = \{\boldsymbol{x_a}; \boldsymbol{y_a}; \boldsymbol{z_a}\}$ with $\boldsymbol{z_a}$ pointing to the zenith and $\boldsymbol{x_a}$ in the direction of the observation.

We define the satellite local orbital reference frame $R_{lo} = \{\boldsymbol{x_{lo}}; \boldsymbol{y_{lo}}; \boldsymbol{z_{lo}}\}$ with $\boldsymbol{z_{lo}}$ pointing to the zenith and $\boldsymbol{y_{lo}}$ in the direction of the kinetic moment of the satellite moving around the Earth.

[4] The vernal equinox point is one of the two points on the celestial sphere where the celestial equator intersects the ecliptic. It is defined as the position of the Sun on the celestial sphere at the vernal equinox time.

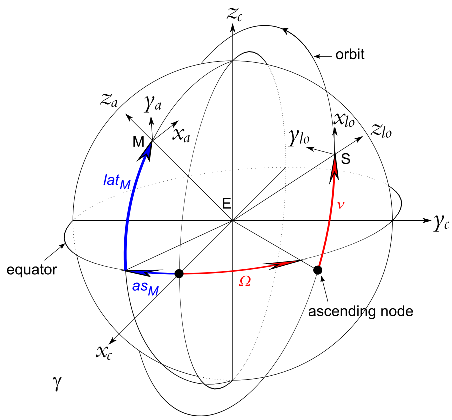


Fig. 4. Used reference frames

Finally, we define the satellite reference frame $R_s = \{\boldsymbol{x_s}; \boldsymbol{y_s}; \boldsymbol{z_s}\}$ with $\boldsymbol{z_s}$ in the opposite direction to the observation instrument direction and $\boldsymbol{y_s}$ in the direction of the line of sensors of the observation instrument. When the satellite points to the Earth center, R_s and R_{lo} are the same (see Figure 4 and see [8] for more details).

Movement of a Point at the Earth Surface. An ground area to observe is defined by a point M on the Earth surface and a course (the observation direction). The latitude lat_M of M is constant and the right ascension[5] $as_M(t)$ of M at any time t is defined by the following equation:

$$as_M(t) = as_{M_b} + \Omega_E(t - t_b)$$

where as_{M_b} is its right ascension at time t_b and $\Omega_E = \frac{2\pi}{86400}$ rd.s^{-1} is the Earth inertial rotation speed. The orientation of the ground area reference frame R_a is given by the transformation matrix from the inertial reference frame R_c to R_a:

$$M_{ca}(t) = R_\theta(-\frac{\pi}{2})R_\phi(\pi)R_\phi(as_M(t))R_\theta(lat_M)R_\psi(course)$$

Consequently, the location $\boldsymbol{EM}(t)$ of M in R_c at any time t is given by:

$$\boldsymbol{EM}(t) = M_{ca}(t)\begin{bmatrix} 0 \\ 0 \\ R_E \end{bmatrix}^{R_a} = \begin{bmatrix} R_E\cos(lat_M)\cos(as_M(t)) \\ R_E\cos(lat_M)\sin(as_M(t)) \\ R_E\sin(lat_M) \end{bmatrix}^{R_c}$$

where $R_E = 6378.13$ km is the Earth radius. Its velocity $\boldsymbol{V_M}(t)$ is given by:

$$\boldsymbol{V_M}(t) = \Omega_E \boldsymbol{z_c} \wedge \boldsymbol{EM}(t)$$

[5] The right ascension is the angular distance of a celestial body or point on the celestial sphere, measured eastward from the vernal equinox.

Movement of the Satellite Orbit around the Earth. The satellite circular orbit is characterized by its radius R_o and its inclination i. The right ascension $\Omega(t)$ of its ascending node[6] at any time t is defined by the following equation:

$$\Omega(t) = \Omega_b + \dot{\Omega}(t - t_b)$$

where Ω_b is the right ascension of its ascending node at time t_b and $\dot{\Omega}$ its constant rotation speed around $\boldsymbol{z_c}$.

Movement of the Satellite on its Orbit. The satellite location on its orbit at any time t is defined by its true anomaly[7] $\nu(t)$ defined by the following equation:

$$\nu(t) = \nu_b + \dot{\nu}(t - t_b)$$

where ν_b is its true anomaly at time t_b and $\dot{\nu}$ its mean motion. The orientation of the local orbital reference frame R_{lo} is given by the transformation matrix from the inertial reference frame R_c to R_{lo}:

$$M_{clo}(t) = R_\psi(\Omega(t))R_\phi(i)R_\psi(\frac{\pi}{2})R_\phi(\frac{\pi}{2})R_\theta(\nu(t))$$

Consequently, the location $\boldsymbol{ES}(t)$ of the satellite in R_c at any time t is given by the vector:

$$\boldsymbol{ES}(t) = M_{clo}(t) \begin{bmatrix} 0 \\ 0 \\ R_o \end{bmatrix}^{R_{lo}}$$

Its velocity $\boldsymbol{V_S}(t)$ is given by:

$$\boldsymbol{V_S}(t) = \boldsymbol{\Omega_{clo}} \wedge \boldsymbol{ES}(t) = (\dot{\Omega}\boldsymbol{z_c} + \dot{\nu}\boldsymbol{y_{lo}}) \wedge \boldsymbol{ES}(t)$$

Movement of the Satellite on itself. The attitude movement of the satellite is constrained by its inertial matrix I and the maximum moment and torque vectors L_{max} and T_{max}:

$$I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} = \begin{pmatrix} 850 & 0 & 0 \\ 0 & 850 & 0 \\ 0 & 0 & 750 \end{pmatrix} \text{ m}^2\text{.kg}$$

$$\begin{bmatrix} L_{x_{max}} \\ L_{y_{max}} \\ L_{z_{max}} \end{bmatrix} = \begin{bmatrix} 45 \\ 45 \\ 20 \end{bmatrix} \text{ N.m} \qquad \begin{bmatrix} T_{x_{max}} \\ T_{y_{max}} \\ T_{z_{max}} \end{bmatrix} = \begin{bmatrix} 7 \\ 7 \\ 6 \end{bmatrix} \text{ N.m}$$

[6] The orbit ascending node is the orbit point where the satellite crosses the equatorial plane headed from south to north.

[7] The satellite true anomaly is the angular distance, viewed from the Earth center and measured in the orbital plane from the perigee to the current location of the satellite.

3.2 Attitude Change with a Fixed Goal Attitude

We start by considering the simplest case, where the goal attitude is fixed. This happens when the candidate next action is a battery recharge with the solar generators oriented towards the Sun. Because of the huge distance between the Earth and the Sun and the relatively slow movement of the Earth around the Sun, we can indeed consider that the attitude A_{Sun} required by battery recharge remains constant during a non-eclipse period (more than half a revolution of the satellite around the Earth, which lasts about 100 minutes; see Figure 5).

As the satellite attitude movement has been decomposed into three parallel movements, each one along one axis, each other constrained by only the three common phase end times (see Section 1.3), we can consider the attitude movement along one axis.

Let α_b (resp. α_f) be the initial (resp. goal) attitude, with $\Delta\alpha = \alpha_f - \alpha_b$. Let Ω_b (resp. Ω_f) be the initial (resp. goal) attitude speed. Let t_b be the initial time and t_1 (resp. t_2 and t_f) be the end time of phase 1 (resp. 2 and 3) with $\Delta t_1 = t_1 - t_b$, $\Delta t_2 = t_2 - t_1$, and $\Delta t_3 = t_f - t_2$. Let T_1 (resp. T_3) be the constant torque during phase 1 (resp. 3) and L_2 be the constant moment during phase 2 (see Figure 3). The moment $I \cdot \Omega(t)$ at any time t is given by the following equations:

- during phase 1: $I \cdot \Omega(t) = I \cdot \Omega_b + T_1(t - t_b)$;
- during phase 2: $I \cdot \Omega(t) = L_2$
- during phase 3: $I \cdot \Omega(t) = I \cdot \Omega_f + T_3(t - t_f)$;

Moment continuity at times t_1 and t_2 enforces that $I \cdot \Omega_b + T_1(t_1 - t_b) = I \cdot \Omega_f + T_3(t_2 - t_f) = L_2$. This results in Equations 1 and 2:

$$T_1 = \frac{L_2 - I \cdot \Omega_b}{\Delta t_1} \tag{1}$$

$$T_3 = \frac{I \cdot \Omega_f - L_2}{\Delta t_3} \tag{2}$$

If we consider the attitude change $\Delta\alpha$ between t_b and t_f, we get:

$$I \cdot \Delta\alpha = \int_{t_b}^{t_f} I \cdot \Omega(t)\, dt = I \cdot \Omega_b \cdot \Delta t_1 + T_1 \cdot \frac{{\Delta t_1}^2}{2} + L_2 \cdot \Delta t_2 + I \cdot \Omega_f \cdot \Delta t_3 + T_3 \cdot \frac{{\Delta t_3}^2}{2}$$

Combined with equations 1 and 2, this results in Equation 3.

$$I \cdot \Delta\alpha = \frac{L_2 + I \cdot \Omega_b}{2} \Delta t_1 + L_2 \cdot \Delta t_2 + \frac{L_2 + I \cdot \Omega_f}{2} \Delta t_3 \tag{3}$$

The problem consists now in minimizing t_f with constraints given by Equations 1, 2, and 3, at which must be added the following inequalities:

$$t_b \leq t_1 \leq t_2 \leq t_f$$

$$-L_{max} \leq L_2 \leq L_{max}$$

$$-T_{max} \leq T_1, T_3 \leq T_{max}$$

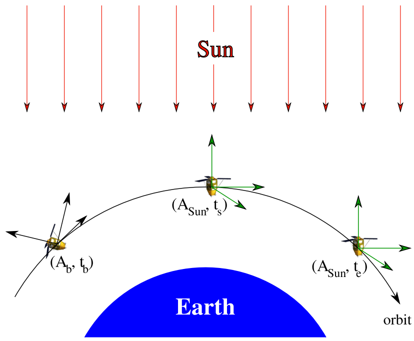


Fig. 5. Reaching a Sun pointing attitude

It is worth emphasizing that, because the decision variables are t_1, t_2, t_f, L_2, T_1, and T_3, the resulting problem is not linear.

3.3 Attitude Change with a Variable Goal Attitude

Except when the next candidate action is a battery recharge, the goal attitude and attitude speed are not fixed. They are variable and depend on the time t_f at which the attitude movement ends. This is the case when the candidate next action is an observation, a data downloading, a detection, a geocentric pointing, or an orbital manoeuvre.

We consider here the most constrained case when the candidate next action is the observation of any ground area a (see Figure 6 which shows how the goal attitude depends on the time at which observation starts, anywhere in the visibility window $[t_s; t_e]$ of a). All the constraints described in the previous section for the case of an attitude change with a fixed goal attitude remain valid, except that α_f and Ω_f are no longer constant. They are variable, but constrained due to the following requirements:

– the observation instrument must point to the starting point M of a:

$$\boldsymbol{z_s} = \frac{\boldsymbol{ES}(t_f) - \boldsymbol{EM}(t_f)}{||\boldsymbol{ES}(t_f) - \boldsymbol{EM}(t_f)||}$$

– the sensor line of the observation instrument must be orthogonal to the observation direction of a:

$$\boldsymbol{x_s} = \boldsymbol{y_a}(t_f) \wedge \boldsymbol{z_s}$$

So, the orientation of the satellite reference frame R_s at time t_f is given by the transformation matrix from the inertial reference frame R_c to R_s:

$$M_{cs} = \left(\boldsymbol{x_s} \; \boldsymbol{z_s} \wedge \boldsymbol{x_s} \; \boldsymbol{z_s} \right)$$

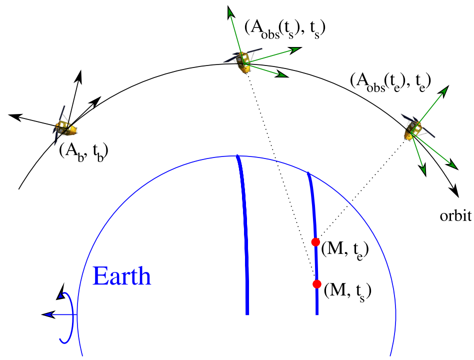


Fig. 6. Reaching an observation attitude

– the speed of the projection on the ground of the sensor line of the of the observation instrument must be equal to the sum of the ground area speed and of the required scanning observation speed v_{scan}:

$$\boldsymbol{V_S}(t_f) + \boldsymbol{\Omega_{cs}}(t_f) \wedge (\boldsymbol{EM}(t_f) - \boldsymbol{ES}(t_f)) = \boldsymbol{V_M}(t_f) + v_{scan}\boldsymbol{x_a}(t_f)$$

where $\boldsymbol{\Omega_{cs}}(t)$ is the rotation speed of the satellite reference frame R_s in the inertial reference frame R_c at any time t.

3.4 CSP Model

The problem can be cast into the following continuous CSP [9].

Variables representing the attitude movement:

$t_f \in [t_s; t_e]$	$T_{1x}, T_{3x} \in [-T_{maxx}; T_{maxx}]$	$L_{2x} \in [-L_{maxx}; L_{maxx}]$
$t_1, t_2 \in [t_b; t_e]$	$T_{1y}, T_{3y} \in [-T_{maxy}; T_{maxy}]$	$L_{2y} \in [-L_{maxy}; L_{maxy}]$
	$T_{1z}, T_{3z} \in [-T_{maxz}; T_{maxz}]$	$L_{2z} \in [-L_{maxz}; L_{maxz}]$

Variables representing the goal state at time t_f:

$M_{ca} \in \mathcal{M}_3(\mathbb{R})$	$\boldsymbol{x_a} \in [-1, 1]^3$	$\boldsymbol{y_a} \in [-1, 1]^3$
$M_{clo} \in \mathcal{M}_3(\mathbb{R})$	$\boldsymbol{\Omega_{cs}} \in]-\infty; +\infty[^3$	$\boldsymbol{\Omega_{cs}^{R_s}} \in]-\infty; +\infty[^3$
$M_{cs} \in \mathcal{M}_3(\mathbb{R})$	$\boldsymbol{x_s} \in [-1, 1]^3$	$\boldsymbol{z_s} \in [-1, 1]^3$
$\Delta\phi \in [-\pi; \pi]$	$\Delta\theta \in [-\pi; \pi]$	$\Delta\psi \in [-\pi; \pi]$

Constraints:

$t_b \leq t_1 \leq t_2 \leq t_f$ and Equations 1, 2, and 3 along the three axes

$$M_{ca} = M_{ca}(t_f) = R_\theta(-\tfrac{\pi}{2})R_\phi(\pi)R_\phi(as_M(t_f))R_\theta(lat_M)R_\psi(course)$$

$\boldsymbol{x_a} = M_{ca}\boldsymbol{x_c}$ $\quad\quad$ $\boldsymbol{y_a} = M_{ca}\boldsymbol{y_c}$

$$M_{clo} = M_{clo}(t_f) = R_\psi(\Omega(t_f))R_\phi(i)R_\psi(\tfrac{\pi}{2})R_\phi(\tfrac{\pi}{2})R_\theta(\nu(t_f))$$

$\boldsymbol{z_s} = \frac{\boldsymbol{ES}(t_f)-\boldsymbol{EM}(t_f)}{||\boldsymbol{ES}(t_f)-\boldsymbol{EM}(t_f)||}$ $\quad$ $\boldsymbol{x_s} = \boldsymbol{y_a} \wedge \boldsymbol{z_s}$

$M_{cs} = \ (\boldsymbol{x_s}\ \boldsymbol{z_s} \wedge \boldsymbol{x_s}\ \boldsymbol{z_s})$ $\quad$ $\boldsymbol{\Omega_{cs}} = M_{cs}\boldsymbol{\Omega_{cs}^{R_s}}$

$$\boldsymbol{V_S}(t_f) + \boldsymbol{\Omega_{cs}} \wedge (\boldsymbol{EM}(t_f) - \boldsymbol{ES}(t_f)) = \boldsymbol{V_M}(t_f) + v_{scan}\boldsymbol{x_a}$$

$$\Delta\phi = \arctan(\frac{(M_{cs})_{32}}{(M_{cs})_{33}}) \qquad \Delta\theta = -\arcsin((M_{cs})_{31}) \qquad \Delta\psi = \arctan(\frac{(M_{cs})_{21}}{(M_{cs})_{11}})$$

3.5 Resolution with RealPaver

We used the RealPaver continuous CSP solver [10] to compute the minimal duration of an attitude change. We present an example of computation where the satellite attitude is A_b at time $t_b = 0$ and the candidate action consists in observing point M ($lat_M = 40.1\,^\circ$, $as_{M_b} = 1.0$ rd, and *course* $= 0.5$ rd) with a null scanning speed ($v_{scan} = 0$ m.s^{-1}) within the visibility window $[t_s; t_e] = [0; 100]$ s.

The considered satellite orbit is characterized by its radius $R_o = 7204.8$ km, its inclination $i = 98.72\,^\circ$, its ascending node rotation speed $\dot{\Omega} = 1.99e^{-7}$ rd.s^{-1}, and its ascending node right ascension $\Omega_b = -2.23$ rd at time t_b. The satellite mean motion is $\dot{\nu} = 1.03e^{-3}$ rd.s^{-1} and its true anomaly at time t_b is $\nu_b = 2.3$ rd.

$$A_b = \{ \begin{bmatrix} \psi_b \\ \theta_b \\ \phi_b \end{bmatrix}, \begin{bmatrix} \dot{\psi}_b \\ \dot{\theta}_b \\ \dot{\phi}_b \end{bmatrix} \} = \{ \begin{bmatrix} 1.34 \\ -0.0682 \\ -0.254 \end{bmatrix} \text{ rd}, \begin{bmatrix} 0.0495 \\ 0.0133 \\ 0.0494 \end{bmatrix} \text{ rd.s}^{-1} \}$$

Figure 7 shows the result of the computation of a *lower bound* on the attitude change minimal duration ($\Delta T = t_f - t_b \approx 16.75$ s) obtained in 240 ms. From that, we just know that there is no attitude change whose duration is shorter than 16.75 seconds, but know neither the minimal duration, nor the associated attitude movement.

Figure 8 shows the result of the computation of an *upper bound* on the attitude change minimal duration ($\Delta T \approx 49.02$ s) obtained in 600 ms. This upper bound has been obtained by fixing the remaining degrees of freedom on the trajectory, more precisely by setting the torques T_1 and T_3 to the maximal acceleration and deceleration along the axis for which the angular movement to perform is the greatest, in this case the x axis, with $T_{1x} = -T_{x_{max}}$ and $T_{3x} = T_{x_{max}}$. From that, we get an attitude movement whose duration is an upper bound on the attitude change minimal duration.

But attempts to get the minimal duration and the associated attitude movement within a reasonable CPU time were unsuccessful. So, it remains to explore other ways, possibly more sensible, of fixing the remaining degrees of freedom on the trajectory.

Interestingly, it is possible to use the same model to optimize another criterion. If there are many observations to perform, minimizing the attitude change duration may be sensible. But, if there are only few observations, it may be more sensible to try and optimize their quality. One way of doing that consists in

```
OUTER BOX: HULL of 1 boxes
  tf            in [16.75 , 16.77]
  delta_psi     in [-1.5 , -1.499]
  delta_theta   in [-0.1763 , -0.1761]
  delta_phi     in [1.144 , 1.146]
  T1x           in [-7 , -0.1952]
  T3x           in [-7 , +7]
  T1y           in [-7 , +7]
  T3y           in [-7 , +7]
  T1z           in [-6 , -1.016]
  T3z           in [-6 , +6]
  L2x           in [-45 , +41.59]
  L2y           in [-45 , +45]
  L2z           in [-20 , +20]
  t1            in [2.84 , 16.77]
  t2            in [2.84 , 16.77]
  [...]

  precision: 90, elapsed time: 240 ms
```

Fig. 7. Lower bound computation

```
OUTER BOX: HULL of 1 boxes
  tf            in [49.02 , 49.04]
  delta_psi     in [-1.552 , -1.551]
  delta_theta   in [-0.3708 , -0.3704]
  delta_phi     in [1.207 , 1.209]
  T1x           = -7 **point**
  T3x           = 7 **point**
  T1y           in [-1.697 , -1.688]
  T3y           in [0.5936 , 0.6169]
  T1z           in [-1.398 , -1.382]
  T3z           in [-3.478 , -3.413]
  L2x           in [-45 , -44.91]
  L2y           in [-9.745 , -9.662]
  L2z           in [19.68 , 19.84]
  t1            in [12.43 , 12.45]
  t2            in [43.27 , 43.31]
  [...]

  precision: 59.6, elapsed time: 600 ms
```

Fig. 8. Upper bound computation

```
OUTER BOX: HULL of 1 boxes
  cos_a-1       in [1.036 , 1.038]
  tf            in [107.4 , 107.5]
  delta_psi     in [1.363 , 1.365]
  delta_theta   in [-0.7951 , -0.7948]
  delta_phi     in [1.349 , 1.351]
  T1x           in [-7 , +1.02]
  T3x           in [-7 , +7]
  T1y           in [-7 , +7]
  T3y           in [-7 , +7]
  T1z           in [-6 , -0.1586]
  T3z           in [-6 , +6]
  t1            in [2.84 , 107.5]
  t2            in [2.84 , 107.5]
  [...]

  precision: 105, elapsed time: 4,760 ms
```

Fig. 9. Observation angle optimization

minimizing the observation angle[8]. Figure 9 shows the result of the computation of a *lower bound* on the new criterion `cos_a-1` $= \frac{1}{<\boldsymbol{z_s}, \boldsymbol{z_{lo}}>}$ obtained in $4,760$ ms, with $\Delta T \approx 107.4$ s. However, we can observe that the CPU time is significantly greater than with the previous optimization criterion.

4 Added Value of Constraint Programming

The main lessons we draw from this work are the following ones:

- maybe, the main advantage of using constraint programming tools is to compel oneself to write an *explicit model* of the problem, which can be used as a reference for any algorithmic development; moreover, the same model can be used with different *optimization criteria*;

[8] The observation angle is defined as the angle between the base vectors z_s and z_{lo}.

- unlike gradient-based methods, interval analysis allows *bounds* to be computed; in our case, the lower bound on the minimal duration of an attitude change may allow infeasible candidate actions to be removed;
- unfortunately, the recursive domain splitting used by constraint programming tools does not allow *optimal solutions* to be produced within a reasonable CPU time; in practice, some heuristic choices are necessary to get solutions of reasonable quality.

References

1. Bensana, E., Lemaître, M., Verfaillie, G.: Benchmark Problems: Earth Observation Satellite Management. Constraints 4(3), 293–299 (1999)
2. Verfaillie, G., Lemaître, M.: Selecting and Scheduling Observations for Agile Satellites: Some Lessons from the Constraint Reasoning Community Point of View. In: Walsh, T. (ed.) CP 2001. LNCS, vol. 2239, pp. 670–684. Springer, Heidelberg (2001)
3. Chien, S., et al.: The EO-1 Autonomous Science Agent. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) Adaptive Agents and Multi-Agent Systems II. LNCS (LNAI), vol. 3394. Springer, Heidelberg (2005)
4. Damiani, S., Verfaillie, G., Charmeau, M.C.: Cooperating On-board and On the ground Decision Modules for the Management of an Earth Watching Constellation. In: Proc. of i-SAIRAS-05 (2005)
5. Charmeau, M.C., Bensana, E.: AGATA: A Lab Bench Project for Spacecraft Autonomy. In: Proc. of i-SAIRAS-05 (2005)
6. Parraud, P., Flipo, A., Jaubert, J., Lassalle-Balier, G.: Computing Smooth Attitude Guidance Laws for Homing Maneuvers. In: Proc. of the International Symposium on Space Technology and Science (2006)
7. Merlet, J.P.: Interval Analysis and Robotics (Invited Presentation). In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, p. 15. Springer, Heidelberg (2006)
8. Hoots, F.: Theory of the Motion of an Artificial Earth Satellite. Celestial Mechanics and Dynamical Astronomy (1979)
9. Granvilliers, L., Benhamou, F.: Continuous and Interval Constraints. In: Rossi, F., Beek, P.V., Walsh, T. (eds.) Handbook of Constraint Programming, pp. 571–603. Elsevier, Amsterdam (2006)
10. Granvilliers, L., Benhamou, F.: RealPaver: an Interval Solver using Constraint Satisfaction Techniques. ACM Trans. Math. Softw. 32(1), 138–156 (2006)