Pleiades Programming: towards a biggest reactivity

Lachiver Jean-Michel

Mission and Programmation Department, CNES
18 av. E. Belin, 31401 TOULOUSE, FRANCE.

E-mail: jean-michel.lachiver@cnes.fr

Abstract

The definition and the conception of Earth observation satellite systems were essentially guided for a long time by the search for the biggest precision of the "images", that means their resolution or their localization. Even if it remains true even today, it is a constituent which takes henceforth more and more importance: the system reactivity. We mean by system reactivity the capacity to give to the user the freshest possible datum (minimization of acquisition delay) in the shortest possible time (minimization of the age of the information). It is one of major requirements which chaired over the development both of the chronology system but also in the programming function of the new Pleiades system successor of the SPOT system. This article begins with explaining and presenting the elements which contribute to the reactivity of an Earth observation system. It will describe then the adopted solutions, in the programming function of the Pleiades system, to meet the needs at best system reactivity in particular as regards the acquisition delay of the images while keeping a system with a large capacity, that means which is designed to acquire a significant number of images. We shall approach successively the Synchronous Programming, the Very Urgent Programming and to finish the Direct Tasking Programming.

1. Introduction

We have moved, in some years, in the era of information, which means everybody can reach, in a relatively easy way, the knowledge of what happens everywhere without the globe in a faster and faster way. And the demand doesn't stop increasing.

The systems of observation of the Earth by satellite do not break the rule. Not only, we ask them for a more and more precise information, in term of resolution or localization, but we ask them henceforth to be the most reactive possible, that is to reduce, at most, the delay between the image request expressed to the system and the availability of this image.

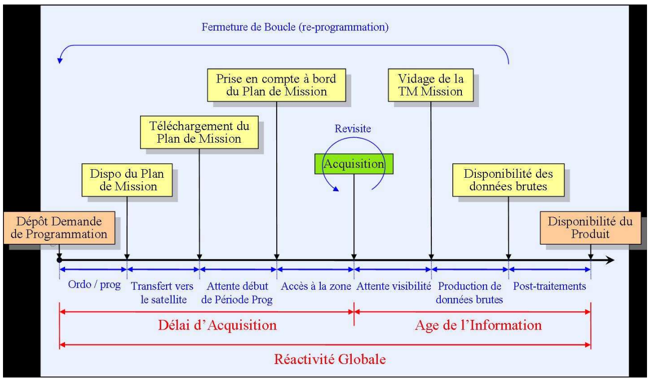
It is with this double objective that was conceived the new Earth observation Pleiades system, successor of the SPOT system.

The reactivity depends, mostly, on the way the programming of the satellites of the system is realized. When we speak here about programming, we evoke the ground function allowing elaborating the satellite Mission Plans from the requests expressed by the users and including, for a given period of time, the sequences of acquisition of the corresponding images and their downloading on the image receiving stations of the system distributed on the surface of the Earth. These Mission Plans are sent to satellites by uploading through a network of Command/Control stations distributed too on the surface of the Earth.

After having defined in detail which are the various contributors of the reactivity of such a system, this article will describe the way the objective of better reactivity was treated in the Pleiades programming function in particular the constraints which were led by the consideration of such a requirement. We shall approach successively the Synchronous Programming, the Very Urgent Programming and finally the Direct Tasking Programming.

2. The Reactivity

The reactivity of a system is measured by estimating the duration which separates the date of filing the user request and the date of availability of the product, delivered by the system and corresponding to this request.



Global Reactivity

The drawing given above details the various durations which contribute in this performance. The reactivity decomposes at first into two durations which are:

- The acquisition delay: this characterizes the duration between the deposit of the request and the acquisition of the corresponding image by the satellite sensor,
- The age of the information: this characterizes the duration between the acquisition of the image and the provision of the corresponding product to the end user.

We are going to try now to decompose these two main durations in elementary durations by describing each time the elements which contribute to these durations.

2.1 The Acquisition delay

2.1.1 The Programming Delay

The first duration which contributes to the acquisition delay is the duration of consideration or integration of the programming request expressed by user, in the next Mission Plan.

To do that, several operations are necessary.

The **first operation** consists in analyzing the feasibility of the request by the system. According to the zone of geographical interest of the request, it is a question at first of cutting it geographically in compatible units of the instrumental swath of the satellite sensor, then to verify that it exists for every programming unit, at least an orbit, and a satellite, from which the acquisition is possible. According to the constraints of realization of the request, he can also be necessary to verify that the cinematic capacities of the satellites of the system allow the realization of the request. This treatment allows at the end to know in particular, which will be the next opportunity offered by the system to realize the request.

This operation ends in the recording of the user request in the system and in its programming database. We shall note here that the performances of the tools to access to the system and to analyse the requests are a first contributor in the system reactivity.

We can already see that the system reactivity is largely conditioned by the capacity of this one to reach such or such geographical zone of the Earth and with which frequency. Indeed, we saw that the request analysis consisted in knowing the next opportunity of accessing the request. This one depends obviously on what we call the system revisit capacity. Typically, for a system as Pleiades for example which acquires images in the field of the visible optics, the images are systematically acquired in the same conditions of solar illumination what is simply resumed by the fact that a zone on the ground can be acquired at best only once a day and at a specific hour. The result is: if the user is interested in the acquisition of a zone on the ground which has just been flown over by one or several satellites, he will anyway have to wait for the next day before obtaining an image of this zone.

The **second operation** consists in generating or in calculating the Mission Plan of the satellite which will be in charge to acquire one or several images which will allow answering to the user request then to download them on one of the system image receiving stations.

We are going to look here at "how" we realize this operation. We shall be interested in "when" in the following chapters as we shall evoke the various possible programming on Pleiades. It is necessary to specify here that the Pleiades programming is completely made in the ground segment. Satellites have no decision-making power and execute what the ground segment asks them to make.

In term of duration, we can easily imagine that the programming delay, that is realization of a Mission Plan will been directly bound and proportional on the temporal horizon on which we make the Mission Plan. On Pleiades, the typical horizon on which we work is 24 hours.

To make a satellite Mission Plan, it is necessary to collect at first all the requests which are going to be candidate in the programming. We speak here about census or about inventory. This first stage will be all the longer to lead that the programming horizon will be long and that the databases of programming will be loaded with. Furthermore, in a shared system as Pleiades, which serves several contributing countries, the programming being made in a centralized way, is added to the census duration, the transmission duration from the user centres towards the programming centre through ground networks.

Once the data are available in the programming centre, the satellite Mission Plan is generated. It consists in choosing the best sequence of images acquisitions (and their downloading) according to a criterion of optimization which takes into account in particular the relative importance of every request, by implementing, if necessary, resource sharing protocols between the various user entities and by respecting certain number of constraints of satellites using as, for example: their cinematic agility, their capacity to store in mass memory the acquired images, the downloading flow of the images towards the image receiving stations, thermal constraints, electric power constraints, limitations in duration of use of the sensors, etc. We speak here about Mission Plan elaborated on a horizon of 24 hours and to exploit at most the acquisition capacities of satellites, that is containing some hundreds of acquisitions.

It is easy to imagine that the optimization of a Mission Plan is time consuming and it's the case all the more that the volume of input data is important. For example, for Pleiades, to make a Mission Plan containing at the end, around 500 acquisitions, we have around ten times more candidates in input. This operation is thus an important contributor of the programming delay and it is often necessary to find the right compromise between an optimal Mission Plan and a quickly elaborated Mission Plan.

2.1.2 The Uploading Delay

The second duration which contributes to the acquisition delay is the uploading delay. It represents the duration between the moment when the Mission Plan is available in the ground segment and the moment when it is available aboard the satellite ready to be executed by the flight software.

This delay is almost completely bound at the moments during which the satellite is in visibility of a Command/Control station either directly, or possibly via a relay satellite (we can neglect here time of ground transfer between the programming centre and the stations). In the case of Pleiades, there is no relay satellite. Thus, it is the network of the Command/Control stations which conditions this delay. We shall notice that it is an element on which it is not always easy to act. Generally, the stations network is an existing input and is more often treated as a constraint than as a parameter of optimization when we try to improve the reactivity of a system.

2.1.3 The Waiting Delay before starting Mission Plan on board

The third duration which contributes to the acquisition delay is the waiting delay before starting Mission Plan on board. It represents the duration between the moment when the Mission Plan is available on board and the moment when it can be effectively activated on board.

At first sight, we could think that this duration could be null easily or at least unimportant. In fact, it is not the case because it depends in particular on where the Command/Control stations are localized towards the moments when it is possible to modify the programming on board. Indeed, it is not always possible to modify immediately the programming (at least, it is still the case on Pleiades). It is necessary, to do that, to guarantee the integrity of the satellite, to make it when the satellite is said in "rest state", that is: it has no mission activity. And it is generally possible only on portions of orbit where either no acquisition is made, or no downloading or even attitude movements asked by the ground segment.

This element often leads, in the idea to make a system reactive, to impose a priori moments in the day when the satellite will be in rest state which will be so many possible resumption points for a fast update of the programming on board.

2.1.4 The Access Delay to the acquisition zone

The last duration which contributes to the acquisition delay is the access delay to the acquisition zone. It represents the duration between the beginning of the Mission Plan on board and the flying by the satellite over the zone where images have to be acquired.

Obviously, this delay can be long all the more as the horizon on which is established the Mission Plan will be long (if the zone to be acquired is at the end of the horizon for example) or more exactly than the frequency of renewal the Mission Plan on board will be low.

To improve this point and reduce this duration, it is necessary to adapt the renewal frequency of the Mission Plan on board and consequently the frequency of the contacts with the satellite.

2.2 The Age of the Information

Once image acquired, it is henceforth a question of delivering it to the user in a fastest way so that this information is the freshest possible when it will be exploited.

2.2.1 Downloading Waiting Delay

The first delay which contributes to the age of the information is the downloading waiting delay. It represents the duration between the acquisition and the moment when the corresponding image can be downloaded on the image receiving station required by the user.

This delay depends naturally on the image receiving station network defined for the system even if we imagine that if the user specifies a station in particular so that his image is downloaded, it will have no other choice than to wait that the satellite flies over the station to make the downloading.

Indeed, to improve this delay and finally make banal the station which will receive the image, we can lean also on networks ground to forward the image to the processing centre which in fine will produce the image. So the transport vector of the image towards the user will not be any more the satellite itself but the network ground by making the hypothesis that this one will have a flow being enough for conveying the image in a speed superior to what than can make the satellite. A positive aspect of this solution of transfer by the ground lies in the fact that she allows to limit the size of the on board mass memory but also flows of downloading of the telemetry because the images are quickly downloaded on local stations so avoiding having to store them and to transport them up to a central station which would represent then a bottleneck for the reception of the images.

The reduction of this delay thus crosses by the optimization of the system architecture with a good distribution of the image receiving stations network combined with a robust transmission ground network.

2.2.2 The Unspatializing Delay

The second delay which contributes to the age of the information is the unspatializing delay that is the necessary time to decode the image telemetry and to get back only the useful part by having deleting all the information used for its transfer from the satellite towards the image receiving station.

The reduction of this delay rests essentially on the speed of these treatments. We are here in the field of the optimization of treatment software.

2.2.3 The Production Delay

This last contributor at the age of the information represents the duration of production of the final image intended for the user.

Here still, as for the unspatializing delay, the reduction of this delay rests on the speed of treatments.

3. The Pleiades Programming

The first part of this article allowed to present and to explain in detail all the elements or the factors which contribute to the system reactivity, in this second part we are going to show how the Pleiades system tried to define a programming function aiming to have the best possible reactivity while keeping the biggest image acquisition capacity. In these conditions, we shall see that this notion of reactivity will be estimated in an average way or not according to the type of envisaged programming.

We shall note that in the case of the Pleiades system, it is an operational system, by opposition to a system demonstrator, who aims at serving significant number of users or costumers with an almost permanent continuity of service. It leads to strong constraints on the programming function so that satellites have permanently the most complete possible Mission Plan.

We describe after three different ways to program Pleiades satellites by explaining, each time, to what extent this programming allows to improve the system reactivity essentially from the point of view of the acquisition delay.

3.1 The Synchronous Programming

We speak about Synchronous Programming, by opposition to an Asynchronous Programming, when this last one is governed by a chronology fixed a priori and does not depend on outer events which can activate it.

In an operational system, it is this type of programming that it has to be privileging to facilitate the organization and the progress of the operations led in the ground segment by the persons asked to realize or to monitor this programming.

Thus, to improve the reactivity consists in reducing the acquisition delay. Satellites flying over the same zone at best once a day, we improve this delay by making more frequent the programming.

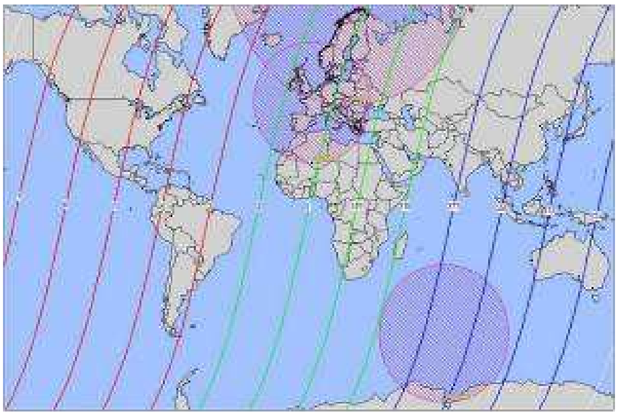
Indeed, let us imagine that a user, whose function is to monitor the natural risks, is interested in a seismic zone, located for example in Italy. Satellites being set on a local time of observation of 10:30 am, Italy is daily flown over at this hour. Now, if the same user wishes to program images of Italy only during earthquakes, according to the frequency with which satellites are programmed, his acquisition delay will be more or less long. If satellites are programmed once a day, in the evening for example for the next day, and if the earthquake occurs in the middle of the night, it will be impossible to him to program images for the following morning. He will have to wait more than 24 hours to obtain his first image. If, on the contrary, satellites are programmed several times by days, it increases the chances to be able to acquire the images more quickly and at the end to be more reactive.

It is the choice which was made on Pleiades. While SPOT satellites are programmed once a day, Pleiades satellites (there are two satellites in the Pleiades constellation) are programmed three times a day. That means that the Mission Plan of each satellite is worked out again in the ground segment and renewed on board so much time.

3.1.1 Three Times a Day

Why three times a day and not four or five or even once by orbit, that means fourteen or fifteen times a day?

This result arises from a compromise between the search for a good average reactivity on the main zones of interest of the system users, from constraints linked to the Command/Control stations network allowing reaching satellites for the uploading and the renewal of the Mission Plans and the feasibility of the operations in the ground segment.



Programming Periods Map

The map above shows, for a given day, tracks on the ground of the orbits for a Pleiades satellite (for the part of the orbits when image acquisitions are possible). While the satellite circulates on its polar orbit, the Earth turns, so allowing reaching all the emerging lands. Orbits are described east westward. The changes of colour of orbits indicate the moments when the Mission Plan is worked out again and reloaded on board.

We shall call, afterward, Programming Period, the duration which separates every moment of reloading the Mission Plan on board.

Thus, we notice that the reprogramming is made just before flying over three main zones of interest of the users that are Asia, Europe and Africa then finally Americas. Every continent is entered with a programming updated by the last user requests deposited into the system. It is nevertheless necessary to understand that it is first flown over zones which will benefit from the best reactivity while zones most on the West will have to wait for the passage of the satellite to be acquired, so increasing the acquisition delay.

If the choice of the frequency has been three programming a day, the reasons are multiple:

- The performance of acquisition delay on the zones of interest, of the order of 8 hours on average has been judged sufficient,
- The Command/Control station network, constituted) by the stations of Toulouse (France), Kiruna (Sweden) and Kerguélen (France) did not easily allow to increase this frequency, due to the fact that he did not offer an access to satellites in every orbit, and that it was not intended, for economic reasons, to complete it with other stations,
- As, it would have been easy to win on the access time at the zone, as it was more difficult to win over the programming delay, the order of height of which is rather close to the orbit duration. Indeed, to elaborate a Mission Plan on a horizon of 24 hours for a satellite possessing the capacity to acquire several hundreds of images over this period takes time (around 40 minutes) because we try to obtain the most complete possible and the most optimal possible Mission Plan. Add to it the interest to realize, in a coordinated way, the Mission Plans of both satellites at the same time, to improve still a little more the use of the satellite resource.

3.1.2 A flexible Chronology

If the current choice is a programming renewed three times a day, the system is not frozen.

Indeed, to be able to modify this chronology during the operational life of the system, this one was defined through a parameter setting which allows defining the dates of programming renewal. We can move thus easily the moments when the programming is renewed but also numbers of wished renewals. It is, so, very easy to switch from a chronology with three programming a day, synchronized with the flying over the main continents, to a chronology with four or five programming a day synchronized with different zones of interest although the Command/Control stations network offers a reasonable waiting delay before starting Mission Plan on board.

This flexibility is allowed, in particular, because when programming, the satellite Mission Plan will impose on it to be in a "rest state" at the moments defined for the renewal of the programming on board. A rest state is defined as a state when the satellite "makes nothing", that means it does not acquire images, it does not download images and it is in an attitude of waiting for programming. The satellite being in waiting for programming, in these moments of transition, it is possible to upload a new Mission Plan and so to update its programming.

We shall note that it is necessary to choose judiciously the moments when we decide to update the programming on board. Indeed, to ask the satellite to be in a rest state is not without impact on its nearby activity. It needs certain time to reach this state (due to movements in attitude to join a canonical position or still in procedures for stopping its instruments) and it also needs certain time to restart from a rest state to become again operational (for inverse reasons). For the nominal Pleiades chronology, for two dates of programming renewal, we chose the passage over the equator by night (no imaging, no downloading) and for the third; we chose the flying over the North Atlantic before entering the American south continent where, even there, we don't make either imaging or downloading.

3.1.3 A Sliding Programming Horizon

As Pleiades is an operational system, it is important that satellites are permanently (or as often as possible) fuelled by a current Mission Plan. By renewing the programming three times a day, and consequently by uploading this programming towards satellites with the same frequency, we increase the risks of "missing" an uploading and having a satellite which has "nothing to do".

Furthermore, this risk is increased by the fact that being in search for the best reactivity, we upload the Mission Plan at the latest, that is on the last visibility station which precedes the starting up of the Mission Plan on board. We so go without the capacity to reload the Mission Plan in case of failure.

To enhance the reliability to have, at any time, a Mission Plan aboard satellites, the programming is realized three times a day on a sliding horizon. This horizon is nominally fixed to a nearby duration of 24 hours. In every case, the beginning of the horizon is synchronized on the beginning of the next Programming Period to program and the end of the horizon is synchronized at the end of the third Programming Period.

A Mission Plan is thus worked out on a horizon which duration is three consecutive Programming Periods: the first Programming Period is said nominal and two others are said backup. So, every satellite has autonomy of 24 hours programming. If we miss the uploading which has to replace the Mission Plan on board from the first backup Programming Period, it is this one which will be executed while waiting for the next uploading. At the end, we compensate for the not redundancy of the uploading means by Mission Plans elaborated on a sliding horizon.

3.1.4 What effects on programming constraints management?

We are going to describe in this chapter, which effects can have such a frequency of programming on the management of the programming constraints.

Indeed, an Earth observation system as Pleiades has a double periodic functioning:

- The first periodicity is relative to the day. At the end of the day, satellites fly over again the same zones,
- The second evident periodicity is relative to the duration of an orbit.

As soon as the generation of a Mission Plan has to share a period in two, an orbit or a day, it raises problems of continuity which are exposed below.

To illustrate this subject, we shall set for every period, an example of constraint which it was necessary to manage.

Resource Sharing

The first difficulty concerns the management of the resource sharing between the different user entities of the system. This sharing was the object of an initial agreement which stipulates that every entity has a daily right of use which is defined in number of images. The sharing protocols, which are integrated into the programming function, are in charge of assuring the respect for this agreement. But in the case of a programming with a sliding horizon, how to respect this notion of daily rights?

The solution consisted in defining the notion of Reference Period. It allows to specify the horizon on which will be managed and checked the respect for the sharing agreements. This, being expressed in number of images a day, the duration of one Reference Period thus corresponds approximately at 24 hours. The way to synchronize this Reference Period with regard to the chronology remained to be defined, that means: when begins and when finishes one Reference Period. We have chosen to adopt a definition close to the day definition. A Reference Period begins with the Programming Period which contains Asia (which the beginning time is close to 0:00 GMT) and finishes with the Programming Period which contains the American continent (and which thus finishes around 24:00 GMT).

When we work out a Mission Plan on a horizon of 24 hours for example, two cases can appear:

- Either this horizon is synchronized with one Reference Period and in that case, it is enough to apply the daily rights,
- Or this horizon is shifted and is between two Reference Periods. In this case, the programming has to take into account, for the first Reference Period, with it remains to consume with regard to the daily rights and, for the second Reference Period, it has to work on a projected part of the daily rights to avoid to consume all the rights.

Thus, between two consecutive programming, it is necessary to manage a state of daily rights consumption follow up. The problem still complicates a little since it is required to set up a regulation of daily rights from a Reference Period to next one.

The limitation of the imaging instrument duration on an orbit

We saw, before, that we had defined for Pleiades; the Programming Periods by trying to place the transitions between these periods during moments when the satellite has "nothing to do". In two cases, this transition was placed as the satellite crosses the equator on the rising part of its orbit (Ascending Knot) and the third transition is located over the North Atlantic.

Now, it turns out that a lot of constraints of use of the satellite are expressed on duration of an orbit. It is the case, for example, for the maximum duration of use of imaging instrument. This constraint is bound, in particular, to the thermal of instruments.

The change of orbit being made, by definition, at the Ascending Knot, it is necessary to define, for this type of constraint, and in the case of a Programming Period transition during an orbit, an a priori distribution of this constraint on both parts of the orbit. Then, when we shall chain two consecutive programming, it will be necessary to assure that we use well in entrance of the second activation, the result of the previous activation to guarantee the respect for the constraint on the whole orbit.

3.2 The Very Urgent Programming

The previous chapter showed that Pleiades Synchronous Programming allows guaranteeing a good average reactivity for all the users programming requests. It is however possible to make much better in exceptional cases. For that purpose, we use the Very Urgent Programming which allows, for a particular request, having priority on all the others, to interrupt the Synchronous Programming momentarily to realize, in a asynchronous way, a new Mission Plan which will be uploaded, as quickly as possible, to the satellite to replace the current Mission Plan. This new Mission Plan will contain, at least, the acquisition and the "as soon as possible" downloading of this very priority request.

We indeed understand that this possibility gas to be used parsimoniously for several reasons:

- It implies an intervention of the operators in the ground segment while the Synchronous Programming can, most of time be made in a automatic way,
- It "breaks" the Synchronous Programming and thus harms in the efficiency of the system by forbidding for a moment quite other programming,
- An the end, it is not certain that to insert a Very Urgent Programming between two Synchronous Programming

will improve significantly the reactivity since Synchronous Programming is already itself enough frequent.

3.2.1 Find the Programming Resumption Point

When we decide to realize a Very Urgent Programming, the first thing to be made is to determine the resumption point from which we are going to elaborate the new Mission Point and additionally the horizon on which this new Mission Plan will be worked out.

We have seen that in conformance with the Synchronous Programming, we placed programming resumption point (that is a satellite rest state) at each Programming Period transition. It is a minimum. In fact, in supplement of these mandatory resumption points, we add systematically in each Mission Plan at least a resumption point by orbit by placing it, if possible, at the change of orbit (at the passage at the Ascending Knot). It allows, in particular, to guarantee that we shall know how to, at least, renew the programming at every orbit if necessary. To finish, the satellite activity being what it is: images are acquired only on the emerged lands and the downloading only on the image receiving stations, the satellite is finally naturally in a rest state often the day and thus there is a lot of what we can call fortuitous resumption points.

Once the urgent programming request identified by its zone and the hour at which it will be possible to fly over this zone at the earliest, it is a question of going back in time and finding the first resumption point from which we can renew the programming on board. Then, knowing this resumption point, it is necessary to determine the Command/Control station which will allow uploading the new Mission Plan what will give the date for which, this Mission Plan will have to be made at the latest. Finally, it is necessary to compare this date with the current date and to see if it still has time to realize the Mission Plan considering the necessary durations for its realization.

3.2.2 Realize the Mission Plan

For the realization of the Very Urgent Mission Plan, considering that it must be made with a limited duration, it was necessary to set up a specific programming based on an iterative process.

The Mission Plan realized in the Synchronous Programming is built in several steps. We have at first four main steps which are linked in sequence:

- We realize the Image Acquisition Plan,
- Once this sequence known, in particular from the satellite kinematics during imaging, we complete this kinematics with all the intermediate guidance, to have at the end, the complete attitude of the satellite during all the duration of the Mission Plan,
- We realize then the Image Telemetry Downloading Plan,
- And to finish, we work out the Mission Plan verifying it according to some final satellite constraints.

Because the satellite resource is shared between different entities, the application of the sharing protocol implies to build the Image Acquisition Plan in several steps. Each step corresponds to a priority of access to the resource: every new step coming to enrich the Image Acquisition Plan obtained in the previous step.

Knowing it, for the realization of the Very Urgent Mission Plan, we proceed in the following way with, at the end of each step, the availability of a complete Mission Plan ready to be uploaded to the satellite in the hypothesis where the following step would not succeed within the given time:

- We realize a first Mission Plan limited, for the Image Acquisition Plan, to the very urgent acquisition. If at the end of this step, it still remains time,
- We realize a second Mission plan which, this time, will contain in its Image Acquisition Plan the urgent acquisition and all the images which had been retained in the Synchronous Programming in conformance with the first priority step of the Image Acquisition Plan,
- And so on until be capable of regenerating completely the total Image Acquisition Plan which had been made in the Synchronous Programming.

At the end, in the best case, we are able to upload a new Mission Plan, enriched by the very urgent request (eventually in place of a less priority request). In the worst case, this new Mission Plan will be limited to the very urgent request. But in both cases, we shall have reduced in a important way the acquisition delay of the very urgent request.

3.3 The Direct Tasking Programming

Previous both chapters described two different ways to realize a new Mission Plan with the objective to make the system reactive. In these two solutions, the new Mission Plan which is uploaded, replace completely the one which is present on board from a resumption point and on a variable horizon.

There is, in fact, an intermediate solution which allows improving locally the system reactivity without being obliged to regenerate totally the Mission Plan but by limiting this updating to an orbit slot. It is the Direct Tasking Programming.

The principle is the following:

- 24 hours before the flying over a zone, we define and we reserve an orbit slot, which we shall call afterward Direct Tasking slot, during which we wish to use the Direct Tasking Programming,
- When we realize the Synchronous Programming on a 24 hours horizon, we take into account this reserved slot, so that we make sure that we can, at the last moment, modify, by an additional uploading, the part of the

Mission Plan located inside the slot to replace it by a more up to date Mission plan. To do it, the part of the Mission Plan which is inside the slot is bounded by markers: on at the beginning and one at the end. In the Synchronous Programming, inside the Direct Tasking slot, we work out Mission Plan said "by default",

- Just before flying over the zone corresponding to the Direct Tasking slot, we realize a new Mission Plan limited to the duration of the slot, and included between the markers, we upload it so that it comes to replace the already present Mission Plan on board on the horizon of the Direct Tasking slot. If we do not succeed in uploading it, it is the "by default" Mission Plan calculated in the Synchronous Programming which is executed.

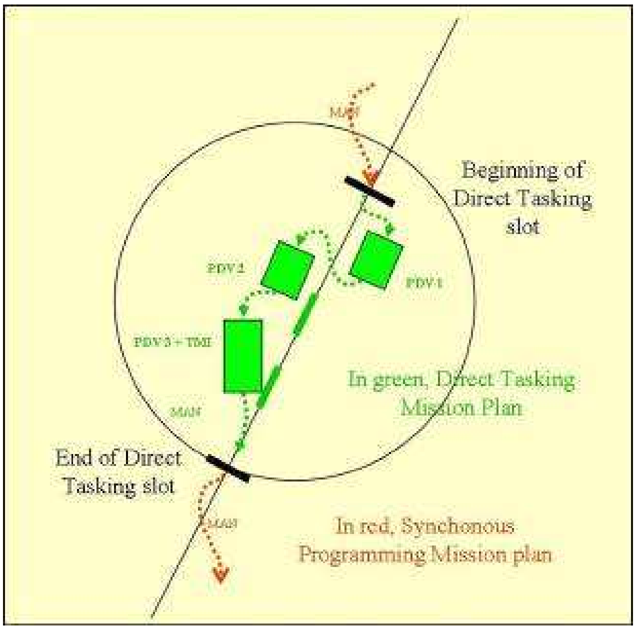
Said like that, it could be simple but actually, it is far from being the case, because the objective is to be able to allow this Direct Tasking Programming on a slot (which duration is lower than 10 minutes) without damaging nor unoptimizing the programming which is made on the same orbit but outside slot and more globally on the totality of the Mission Plan horizon.

We remind ourselves, in particular, that a simple way to renew the programming on board is to start from a resumption point which corresponds to a moment when the satellite is in a rest state. We also saw that to impose a resumption point requires before and after this point a satellite stopping of the activities, and it takes a significant duration (several minutes). Not to damage the nearby programming, it was thus necessary to find an another way to do, in particular towards the guidance of the satellite, to avoid it passing by a neutral canonical position at the beginning and at the end of the slot, synonym for waste of time and thus of unoptimization.

3.3.1 Guarantee the Attitude Continuity

In the particular case of Pleiades, a very agile satellite, which pointing towards zones to be acquired is made thanks to the changes of attitude, to guarantee the continuity of this attitude between the programming made outside the Direct Tasking Slot and the Direct Tasking Programming, while preserving the acquisition capacity, we chose to place at the beginning and at the end of the slot, particular profiles in attitude called Direct Tasking pivot.

In the following drawing, the black stripes bound the Direct Tasking slot. Under each stripes, we place a so called image acquisition which will then allow guaranteeing agile kinematics manoeuvres with the previous and following image acquisition. We speak about so called image acquisition because we impose a kinematics profile which looks like an image acquisition (with a nadir pointing) but we do not realize the corresponding recording of the image, only the attitude interests us.



Direct Tasking Mission Plan

These two pivots are calculated during the Synchronous Programming. They are then taken back in the Direct Tasking Programming to guarantee that this one is linked with Synchronous Programming in a kinematics way.

3.3.2 Realize an Independent Programming

As far as the Direct Tasking Programming comes to fit in one Synchronous Programming, it must be autonomous and have no influence on the programming outside the slot. It means, in particular, that all the programming which is made in the slot must be "finished" before "going out" of the slot.

It is true, in particular, concerning the downloading of acquisitions realized in the slot. If an image is acquired in the slot, whether it is in conformance with Synchronous Programming in which we calculate a "by default" Mission Plan or in the Direct Tasking Programming, we cannot download it except the slot itself for evident reasons. Indeed, if in Synchronous Programming, this image was downloaded outside the slot, we could not modify it any more in the Direct Tasking Programming (delete it or move it) because it would be necessary to guarantee, in conformance with the Direct Tasking Programming to respect its downloading which cannot be any more updated.

It is the reason why, Direct Tasking slots are defined in association with image receiving stations and more particularly as all or any part of a slot of satellite visibility seen by the station to be able to download the images acquired in conformance with the Direct Tasking Programming. The images acquired locally in Direct Tasking slot are also downloaded locally.

In fact, we can see a Direct Tasking Slot as a way, for a given user having an image receiving station, to reserve the satellite resource for an exclusive usage which he can exploit at the last moment according to his last programming request but also according to the last cloud forecasts he will have, so making his programming most effective possible from this point of view.

3.3.3 Respect the Programming Constraints

The Direct Tasking Programming is made in two phases: a first phase which takes place during the Synchronous Programming and in which, we work out, inside a Direct Tasking slot, a "by default" Mission Plan. Then the second phase in which we actually work out the final Mission Plan which will be uploaded at the last moment.

It means, among other things, that it is completely possible to modify totally the Mission Plan of the Direct Tasking slot between the Synchronous Programming and the Direct Tasking Programming. It is thus necessary to guarantee whether it is possible, in particular towards the respect for the programming constraints which are estimated and controlled themselves either on a complete orbit or on a whole day as we saw it in a previous chapter.

This point is treated and solved by defining and managing constraints allocations for each Direct Tasking slot. These allocations concern notably:

- The duration of the instrument using in the slot. The Synchronous Programming first then the Direct Tasking programming in a second time will have to respect this allocation in order to guarantee that the consumption made inside the slot added to the consumption made outside the slot will respect the constraint of the orbit. This allocation depends on the slot duration of course but also on the slot location. In the hypothesis that every orbit have the same constraint, we will allocate different duration to a slot depending on if it is located on an very busy orbit or not,
- The volume to reserve in the on board mass memory at the beginning of the slot in order to be able to acquire, store and download the images during the slot. Here too, this allocation must be adjusted with respect to the slot duration to avoid limiting this resource for images acquired outside the slot.

The other aspect of this problem of management and respect for the constraints concerns the constraints of tendency which are estimated on day duration. It is, in particular, the case for the respect of the satellite energy budget that allows guaranteeing that the satellite will have the sufficient electrical energy to execute the Mission Plan. For that purpose, during the Synchronous Programming, we realize a budget using margins and extremist conditions in the sense that it is made with the hypothesis of a maximum use of the resource in every Direct Tasking slot according to the allocations described before. So, a Mission Plan realized by the Direct Tasking Programming can be only less energy-consuming and will not put in danger the satellite.

3.3.4 Upload the Direct Tasking Mission Plan

We said previously that the Direct Tasking Mission Plan resulting from the Direct Tasking Programming was uploaded "at the last moment". It is effectively the case. To reduce as much as possible the acquisition delay, and in particular the waiting delay before starting Mission Plan on board, we chose to define image receiving stations which had in more the capacity of Command/Control, that is the capacity to upload a Mission Plan.

Consequently, as soon as the satellite is in visibility of the station for which we defined a Direct Tasking slot, we upload the Direct Tasking Mission Plan (the operation is rather short) and we can start it so very quickly after the end of the uploading.

4. Conclusions

This article allowed describing the different solutions developed for Pleiades system, in order to reach what we can be called a "good reactivity", or at least an improved reactivity compared to SPOT system, and also to explain, what were the points to be treated and solved concerning the programming and working out of the satellite Mission Plans.

Choices that have been done result in a trade off between reactivity and acquisition capability. This means that while improving the system reactivity, we tried to maintain as much as possible the ability of satellites to acquire daily a huge number of images, considering less penalizing solutions toward system performance. Synchronous Programming and Direct Tasking Programming fit perfectly to this compromise. Very Urgent Programming is to be used carefully, and only when Synchronous Programming fails for critical situations, as it does not fully reach the initial objective.

Now, it is possible to further improve the system reactivity, working on uploading delay, and waiting time before start of programming on board, which are directly linked to visibility of satellites through the Command/Control station network, but also working on the age of the information, which depends on the image receiving stations network. No doubt that, future systems will focus on this improvement.