Automated Planning for Earth Observation Spacecraft under Attitude Dynamical Constraints

Johannes Löhr, Johannes Aldinger,
Albert-Ludwigs-Universität Freiburg, Germany

Stefan Winkler and Georg Willich
Astrium GmbH Satellites Friedrichshafen, Germany

Keywords: autonomy, automated planning, agile earth observation

Agile Earth observation missions continuously require a large amount of planning during the spacecraft's observations. Beside priorities of the observation sites, especially the agility constraints of the satellite are important to be taken into account during the planning process. This is due to the body-fixed instrument's line of sight, requiring the whole satellite to point to the observation sites while scanning. Scanning a sequence of observation sites leads to complex slew maneuvers which must not exceed the satellite's actuator capacities, attitude constraints or maximum angular rates. Additionally, the regions of interest may change over time, making it necessary to adapt and optimize the observation sequence continuously. An automated process is required to efficiently handle this task. We present a planning algorithm to sequence an arbitrarily distributed set of observation patches to a feasible observation plan, considering priority criteria of the observation sites and agility constraints of the satellite.

1 Introduction

Earth observation is an important field in space applications. In agile observation missions the observation is done with instruments rigidly mounted on the satellite's body. The resolution of the images can be increased by larger sensors or the use of instruments with smaller field of view yielding a reduced coverage of the Earth's surface during each revolution of the satellite in its orbit. In such missions it is aspired to scan designated observation sites rather than to achieve global coverage of the Earth's surface as discussed by Lamaitre et al. (2000). This requires the instrument or the whole satellite to slew all axes like in the Pleiades, Eros or WorldView missions (Jacobsen, 2005). Instead of taking pictures with matrix sensors we focus on instruments which continuously scan an observation site using time delayed integration (Wong et al., 1992). Therefore, a specific relative motion of the instrument's line of sight with respect to the observation site has to be realized. The result is an observation strip on the Earth's surface which we call a *patch*. An observation scenario is defined by a large set of patches arbitrarily distributed on Earth with predefined priorities which shall reflect the customer's need.

Our goal is to find suitable sequences of observation patches yielding a feasible sequence of slew maneuvers, considering the satellite's orbital motion, its attitude and angular rate as well as its torque capability in realistic scenarios. Such a sequence is called a *plan* in the following. Instrument alignment and the required scan velocity pose additional constraints. The feasibility of slews between two successive scans depends on the satellite's attitude, angular rate and position and is varying in time. Any decision to scan a certain patch at a certain time may affect the feasibility of future scan maneuvers. This makes the problem difficult to solve in case of larger sets of observation patches.

Previous work on planning for agile missions (Aldinger and Löhr, 2013) is based on heuristic search algorithms with semantic attachments (Dornhege et al., 2012), using simplified dynamics and a successive validation to get feasible plans. Side effects of the simplification are sporadic infeasible maneuvers which lead to replanning. An issue of the heuristic search approach is mainly the justification of found plans which show a good quality in terms of number of planned patches but do not offer an explanation in case of neglected patches. Therefore, we developed an algorithm which considers the priority of the patches and is not based on a simplified dynamic, since it directly uses the functionality of the ASTRIUM ASSET toolbox (Barschke et al., 2012), which calculates the whole satellite guidance including optimized slew maneuvers for a given patch sequence. The proposed planning algorithm neglects observable patches only, if their observation inhibits the scan of patches with higher priority.

The remainder of this paper is structured as follows. After a brief problem definition we sketch the functionality of the ASSET toolbox. Then, the planning algorithm is described and exemplary simulation results of a random scenario are presented. Finally, we conclude and identify future work.

2 Problem Definition

Desired observations of regions on Earth define an arbitrarily distributed set of n observation patches $\mathcal{O}_{\mathrm{rnd}} = \langle o_1, o_2, \ldots, o_n \rangle$ to be scanned by the satellite. Each patch considered in this paper is a triple $o = \langle c, v, p \rangle$ consisting of a set of coordinates c, a scan velocity v which is required by the sensor and its priority p. The priority is unique for each patch and we obtain it by combining a priority level with the timestamp of its request. This reduces long delays for older scan requests of the same priority level. It is unwanted to scan patches of lower priorities to the cost of patches with higher priority. Despite this additional constraint the distinct priority ordering for patches heavily reduces the search space compared to an utility based approach[1] as used by Aldinger and Löhr (2013).

3 The ASSET Toolbox

The functionality of the ASSET toolbox is described by Barschke et al. (2012). For a *given* sequenced `scenario` $s = \langle o_1, o_2, \ldots, o_m \rangle$ of m patches and `parameters` defining the satellite's properties, the instrument, orbit and mission duration it returns either a feasible guidance profile or an error in case of an infeasible slew between two patches. An *infeasible* slew maneuver exceeds either the actuator capabilities or the maximum angular rates, requires an infeasible attitude of the satellite or the patch is simply not acquirable by the satellite due to visibility constraints. The torque profile is subject to optimization in ASSET as well. It can be optimized with respect to *time*, *energy*, or a *smooth transition* in order to generate torque profiles avoiding stimulation of flexible modes of the satellite. The time optimized slews are particularly interesting for planning, since in general a maximum number of scans within a given time window is desired. Therefore, we selected this optimization mode in the `preferences` of ASSET as well as a coarse optimization step size in order to minimize the tool's runtime.

4 Planning of Earth Observation

Our contribution is to reformulate the Earth observation task into a planning problem which is solvable by the proposed planning algorithm. The combinatorial space of connecting all or a subset of patches to a *sequence* is very high, as demonstrated in Section 4.1. Therefore, we have a two step process for finding suitable sequences. First of all, we reduce the complexity of the planning problem by cancelling out as many obviously infeasible sequences as possible, see Section 4.2. Secondly, we search within the remaining sequences for a feasible one using the planning algorithm described in Section 4.3. It generates plans that respect the priority of the patches and fulfil anytime[2] properties to flexibly adjust the runtime.

[1] In an utility (e.g. profit) based approach scanning multiple patches of lower priority can become favorable over scanning one patch of higher priority.

[2] The anytime property allows the user to interrupt the execution of the algorithm at any time and to obtain the best plan found so far.

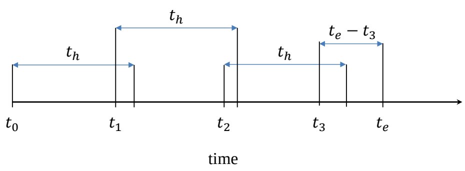

Figure 1: Planning horizons.

4.1 Complexity

Generally, the number of sequences to connect n patches with respect to feasibility and priority grows exponentially with n. Every patch can be connected to all other patches, while the length of the sequence is unknown. That leads to

$$P = \sum_{i=1}^{n} n!$$

possible sequences. It is obvious that it is impossible to try out every patch sequence for larger n. We assume that there are thousands of scan requests for patches distributed over Earth. It is also obvious that most of the possible sequences are not feasible to be scanned by an agile satellite. Pure geometric visibility constraints like maximum deflection from the nadir pointing as well as dynamical constraints derived from actuator capabilities delimit the admissible set of successive patch observations. Even when considering only the theoretically visible set of patches, arbitrary sequencing of patches will most likely violate the dynamical capabilities of the satellite. Therefore, we try to cancel out as many infeasible sequences as possible in a preprocessing phase.

4.2 Preprocessing

Preprocessing consists of multiple steps that all reduce the complexity of the Earth observation task to obtain a manageable planning problem. The most important aspect of the complexity reduction is the concept of a receding horizon. Here, we plan only over a fragment of the mission time, since scan decisions usually only have short-term effects on future decisions. In the visibility analysis we select among all patches $\mathcal{O}_{\text{rnd}}$ a subset $\mathcal{O}_{\text{vis}} \subseteq \mathcal{O}_{\text{rnd}}$ of patches visible within the current planning horizon. In the overlap analysis we match the amount of patches $\mathcal{O}_{\text{plan}} \subseteq \mathcal{O}_{\text{vis}}$ to the agility of the satellite. In the transition analysis we generate a matrix M

Algorithm 1: Planning Horizon

```
1  def planSlew(O_rnd, t_0, t_h, t_end, x_0):
2      plan ← ⟨⟩
3      t_i ← t_0
4      while t_i < t_end do
5          O_vis ← chkVisibility(O_rnd, t_i, t_h)
6          O_plan ← overlapAnalysis(O_vis)
7          M ← transitionAnalysis(O_plan)
8          s* ← findSequence(O_plan, M, x_i)
9          plan ← append(plan, s*)
10         if isEmpty(s*) then
11             t_i ← t_i + t_h/2
12             x_i ← Nadir pointed state at t_i
13         else
14             t_i ← scanSeqEndTime(s*)
15             x_i ← satellite state at t_i
16     return plan
```

containing all infeasible transitions between two patches with respect to time constraints.

Planning Horizon Algorithm 1 illustrates the receding horizon planning concept. We start with an initially empty plan (line 2). Instead of planning over the whole mission time $t \in [t_0, t_{\text{end}}]$, we plan over short time horizons t_h that split the mission in manageable fragments. The current time point t_i which is initially t_0 (line 3) is successively progressed until the end of the mission t_{end} is reached (line 4). The number of manageable patches subjected to planning within the current horizon $t \in [t_i, t_i + t_h]$ is narrowed down during the preprocessing steps (line 5-7) which will be explained in more detail in the upcoming paragraphs. Once a feasible sequence $s^\star$ for the planning horizon is found (line 8), we connect it to the previous sequence (line 9). In the unlikely case that no patches were scheduled in the interval (line 10) we progress the time for half an interval (line 11) and move the satellite into a nadir pointed attitude. Otherwise, the next planning horizon starts at $t_{i+1} \in [t_i, (t_i + t_h)]$ (line 14) which is chosen as the time point at which the scan maneuver of sequence $s^\star$ ends. To better link the horizon fragments and to avoid important patches to be omitted, a small number of the last scheduled patches is removed from the end of $s^\star$ before calculating the scan end time. Even-

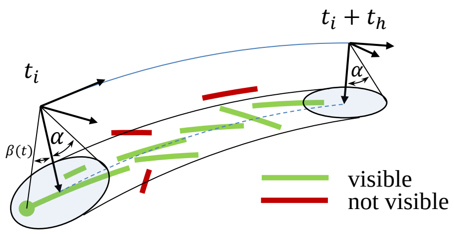

Figure 2: Patch visibility in planning horizon.

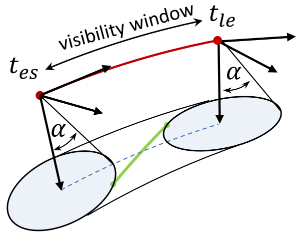

Figure 4: Visibility window for observation.

tually, we use the last time interval $t \in [t_i, t_{\text{end}}]$, see Figure 1. The initial state of the satellite at the start of each horizon is given by $x_i = \langle pos, att, rate \rangle$ with orbit position, attitude and angular rate at time t_i (line 15) which is also part of the ASSET output.

Visibility Analysis The visibility analysis is the part (line 5 in Algorithm 1) of the preprocessing, where patches that are visible in the current horizon are extracted. In a first step it is analyzed which patch is visible from an orbit position using the maximum angle α of the instruments line of sight with respect to the Nadir axis, as depicted in Figure 2. A point of the patch is visible if $\beta(t) \leq \alpha$ at an arbitrary time $t \in [t_i, t_i + t_h]$. Only patches that fulfil this visibility constraint for all points of the patch are part of the visible set $\mathcal{O}_{\text{vis}} \subseteq \mathcal{O}_{\text{rnd}}$.

Overlap Analysis The amount of patches, which are visible within the planning horizon can be arbitrarily high, while the amount of patches which can be scanned by the satellite depends on its agility. The task of this analysis (line 6 of Algorithm 1) is to reduce the amount of patches to a manageable number, which means to reject patches which are unlikely to be observed due to observation of patches of higher priority in the same visibility window. The visibility window of a patch begins at the earliest time t_{es} where the scan can be started and ends at the last time t_{le} where the scan has to be ended, see Figure 4. The visibility window is determined by simple geometric calculations for all $o \in \mathcal{O}_{\text{vis}}$. We generate a set $\mathcal{O}_{\text{plan}} \subseteq \mathcal{O}_{\text{vis}}$ by subsequent adding patches of the highest priority to $\mathcal{O}_{\text{plan}}$ as long as for all times $t \in [t_i, t_i + t_h]$ the number of visible patches is lower or equal than $N_{\max}$. This rejects patches of low priority, if many patches of higher priority are visible at the same time. Furthermore, we include patches of low priority, if only few patches of higher priority are visible at the same time. An example for $N_{\max}$ is depicted in Figure 3.

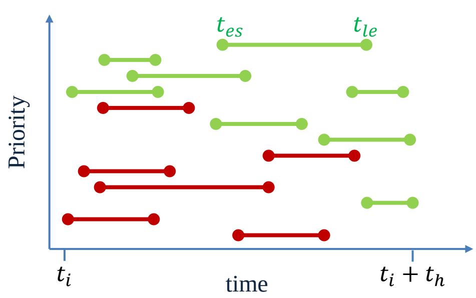

Figure 3: The subset $\mathcal{O}_{\text{plan}}$ in green of the visible set $\mathcal{O}_{\text{vis}}$ with $N_{\max} = 3$.

Transition Analysis In the transition analysis (line 7 in Algorithm 1) we identify infeasible transitions between two patches that violate simple time constraints. Analogous to the earliest start and the latest end defining the visibility window, the latest point in time t_{ls} to start the scan of a patch and the earliest time t_{ee} to end the scan of the patch can be identified, see Figure 5. A transition from patch $i \in [1, m]$ to patch $j \in [1, m]$ is only feasible if $(t_{\text{ls}}(i) - t_{\text{ee}}(j)) > \Delta t$, where Δt corresponds to a minimum slew time. This constraint is checked and stored in the transition matrix $M(i, j)$ for all transitions $i \neq j$ between all patches in $\mathcal{O}_{\text{plan}}$.

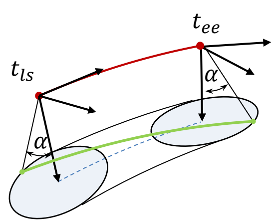

Figure 5: Latest start and earliest end of a scan.

Algorithm 2: Planning Algorithm

1 **def** ***findSequence(*** $\mathcal{O}_{plan}, M, x_i$ ***)*** **:**
2 $\quad s^\star \leftarrow \langle\rangle$
3 $\quad \mathcal{S}_{\text{open}} \leftarrow \{s^\star\}$
4 $\quad ig \leftarrow 0$
5 $\quad$ **while true do**
6 $\quad\quad \mathcal{S}_{\text{feas}} \leftarrow \{\ \}$
7 $\quad\quad$ **while** $isNotEmpty(\mathcal{S}_{open})$ **do**
8 $\quad\quad\quad s \leftarrow removeLongestSeq(\mathcal{S}_{\text{open}})$
9 $\quad\quad\quad$ **if** $assetCheck(s, x_i)$ **then**
10 $\quad\quad\quad\quad \mathcal{S}_{\text{feas}} = \mathcal{S}_{\text{feas}} \cup s$
11 $\quad\quad\quad\quad s^\star \leftarrow longestSeq(\mathcal{S}_{\text{feas}})$
12 $\quad\quad\quad\quad \mathcal{C} \leftarrow children(s, ig, \mathcal{O}_{\text{plan}}, M)$
13 $\quad\quad\quad\quad \mathcal{S}_{\text{open}} \leftarrow \mathcal{S}_{\text{open}} \cup \mathcal{C}$
14 $\quad\quad\quad\quad$ **if** $|s^\star| + ig = |\mathcal{O}_{plan}|$ **then**
15 $\quad\quad\quad\quad\quad$ **return** $s^\star$
16 $\quad\quad \mathcal{S}_{\text{open}} \leftarrow longestSequences(\mathcal{S}_{\text{feas}})$
17 $\quad\quad ig \leftarrow ig + 1$

Algorithm 3: Expand Algorithm

1 **def** ***children(*** $s, ig, \mathcal{O}_{plan}, M$ ***)*** **:**
2 $\quad nextPrio \leftarrow |s| + ig + 1$
3 $\quad p \leftarrow getPatch(\mathcal{O}_{\text{plan}}, nextPrio)$
4 $\quad$ **for** $0 \le pos \le |s|$ **do**
5 $\quad\quad s' \leftarrow \langle s(0 : pos), p, s(pos + 1 : |s|)\rangle$
6 $\quad\quad$ **if** $fastCheck(s', M)$ **then**
7 $\quad\quad\quad \mathcal{C} \leftarrow \mathcal{C} \cup s'$
8 $\quad$ **return** $\mathcal{C}$

4.3 Planning Algorithm

In this section we describe the planning algorithm, Algorithm 2, which is used to find a feasible sequence $s^\star = \langle o^\star_1, o^\star_2, \dots, o^\star_k\rangle, k \le m$ of m patches in $\mathcal{O}_{\text{plan}}$ within the planning horizon.

We start the algorithm with the initialization of an empty plan $s^\star$ (line 2) and with an open list $\mathcal{S}_{\text{open}}$ containing the empty plan (line 3). Furthermore, we have an initially empty counter ig of unobservable patches (line 4). A loop (line 5) increases the number of ignored patches (line 17) in each iteration. In each iteration, we maintain an initially empty set of feasibility checked sequences $\mathcal{S}_{\text{feas}}$ (line 6). As long as there are sequences in $\mathcal{S}_{\text{open}}$ (line 7) we select and remove the *longest* sequence s (line 8) and check it with the Asset toolbox[3] (line 9). If the patch sequence is observable by the satellite and its attitude dynamical constraints are satisfied we add it to the feasible set $\mathcal{S}_{\text{feas}}$ (line 10). The best known sequence $s^\star$ corresponds to the longest sequence in $\mathcal{S}_{\text{feas}}$ (line 11). The feasible partial plan s is used to generate new sequences called children using Algorithm 3 (line 12).

Algorithm 3 starts with determining the priority of the next patch (line 2). The patches that were considered before are either contained in the sequence s or they are ignored in which case ig was increased. The priority of the next patch is thus obtained by summing the number of patches in s, ig and 1. The corresponding patch p is then chosen from $\mathcal{O}_{\text{plan}}$ (line 3). The new patch can be inserted at each of the $|s| + 1$ many positions pos of sequence s which is done iteratively (line 4). Each successor sequence s' (line 5) is evaluated by a *fastCheck* (line 6) $M(s'_i, s'_{i+m})\ \forall i \in [1, |s'| - 1], \forall m \in [1, |s'| - i]$, which succeeds if no infeasible transitions regarding to M are contained in s'. Only then, s' is inserted into the set of children $\mathcal{C}$ (line 7). The set of children that comply to the transition analysis are then returned (line 8).

The children returned from the subroutine are then added to the $\mathcal{S}_{\text{open}}$ list (line 13 in Algorithm 2). In case that the length of the best sequence plus the number of ignored patches equals the number of patches of the planning problem $\mathcal{O}_{\text{plan}}$ (line 14), we found an optimal sequence which ends the algorithm (line 15). Since we always select the longest sequence from $\mathcal{S}_{\text{open}}$, we find plans of good quality (plans including a large number of patches) quite fast.

In case that no way exists to extend the current sequence, the open list $\mathcal{S}_{\text{open}}$ will run empty. A patch must be ignored, and we continue the algorithm with the longest feasible sequence which is stored in $\mathcal{S}_{\text{feas}}$ (line 16). Of course all other sequences in $\mathcal{S}_{\text{feas}}$ are feasible, too, but they omit at least one patch of higher priority compared to the longest sequences. The new open list contains only sequences with infeasible children. Therefore the last patch is not observable and ig is increased by one (line 17).

The algorithm can be stopped at any time to return the best sequence $s^\star$ found so far.

[3] The empty plan is always feasible, since the satellite has no patch to observe

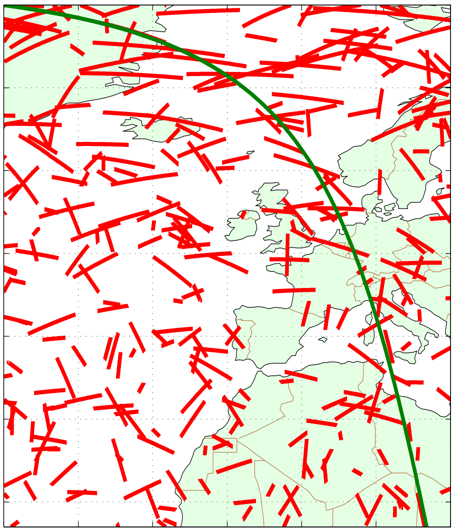

Figure 6: Random scenario with view over Europe. Observation patches are depicted as red stripes while the ground track of the satellite shown in green.

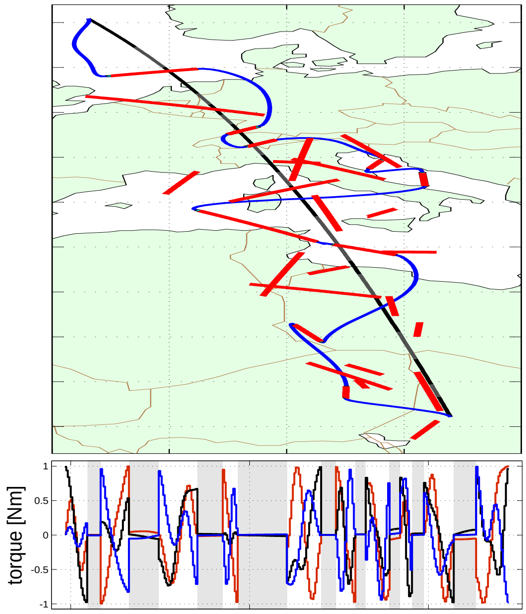

Figure 7: Optimal sequence for the first planning horizon with visible patches in red, ground track in black and intersection between line of sight and Earth's surface in blue. The ground track is shown in grey while scanning. The torque demand is within the specified limits of 1 Nm in each axis.

5 Exemplary Results

Random Scenario We generated a random Earth observation scenario to test the planning algorithm. It consists of 5000 randomly distributed observation patches of length from 100 km to 600 km and arbitrary orientation, see Figure 6. The scenario is highly over-specified and the goal is to find a plan with duration of one orbit period which includes as many patches as possible considering the priority conditions discussed before. We considered a satellite with a moment of inertia of 500 kgm^2 in each axis and a maximum available torque of 1 Nm. The body-fixed instrument provides scans of acceptable quality if both the roll angle and the pitch angle are below 35° with respect to Nadir pointing. The maximally acceptable angular rates of the satellite are 10 $\frac{deg}{s}$ in each axis.

One Planning Horizon In order to reduce the complexity of the planning task, we chose a moving planning horizon of 10% of the orbit period, see Figure 7.

There are still $m = 29$ patches visible in $\mathcal{O}_{vis}$ of the first horizon which yield an enormous amount of possible sequences, see section 4.1. The optimal plan contains nine patches, the solution was found after 15 minutes. The number of visible patches does not necessarily correspond to the number of theoretically aquirable patches. Therefore, a lot of unnecessary sequences are tested. The overlap analysis reduces this number. Table 1 shows the trade-off between runtime and plan quality $|s^\star|$ depending on the maximum number N_{max} of overlapping patches. Choosing $N_{max} = 7$ already leads to an optimal result with significantly reduced runtime.

Connecting Planning Horizons For generating plans that are longer than the planning horizon, we move the horizon continuously as described in Algorithm 1. Simply linking the

Table 1: Reduction a realistic number of acquirable patches in 10% planning horizon.

N_{max}	$\|\mathcal{O}_{plan}\|/\|\mathcal{O}_{vis}\|$	Runtime	$\|s^\star\|$
1	3/29	15 s	2
2	5/29	30 s	4
3	8/29	2 min	6
4	10/29	3 min	7
5	12/29	4 min	7
6	14/29	8 min	8
7	17/29	9 min	9
8	19/29	10 min	9
∞	29/29	15 min	9

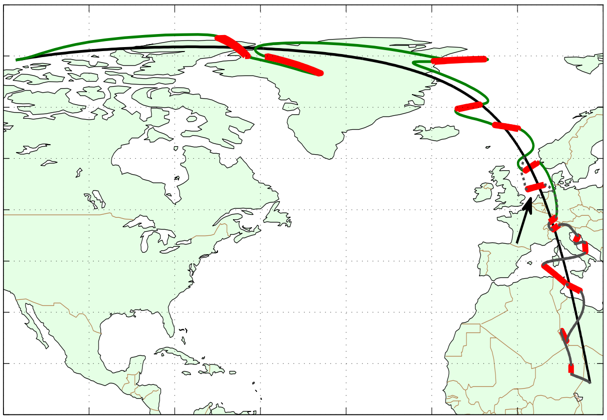

Figure 8: Connection of two planning horizons. The committed part of the line of sight is shown as grey line while the rejected part is dashed. The omitted patch is marked by an arrow. The new horizon starts at the end of the last committed patch with green line of sight.

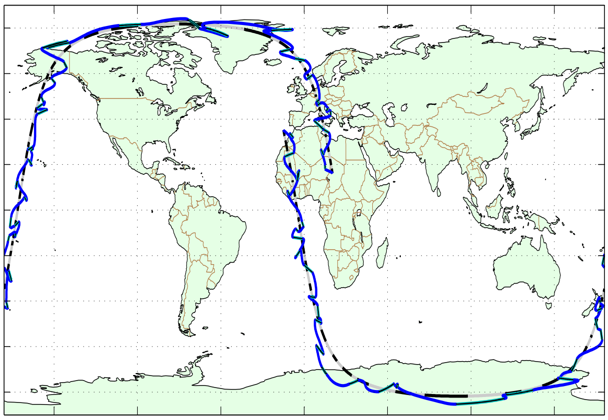

Figure 9: The final plan over one orbit. It corresponds to the guidance profile of the satellite over one orbit connecting 61 observation sites with the line of sight of the instrument on Earth, shown in blue. The ground track of the satellite is shown in black during slews and in grey while scanning.

planning horizon fragments leads to plans of inferior quality, since the choice of patches at the end of the previous horizon influences the plan for the next horizon. Therefore, we start the new horizon earlier in the time interval of the previous one. In order to reduce this horizon linking problem, it is reasonable to omit patches at the end of the found sequence and to commit only to the remaining sub-sequence. It turns out that already omitting one patch reduces the linking effect significantly, as depicted in Figure 8.

Mission Planning When planning over longer mission times, it is important to find a good balance between runtime and *plan quality*, which can be estimated by the number of planned patches. Therefore, we have to find a suitable planning horizon containing a sufficient number of patches scaled by N_{max}. The number of patches should correspond to the agility of the satellite. This is the case if we can find an optimal value for N_{max} which does not significantly increase the number of planned patches during the mission time.

Table 2 shows that an increasing number of patches in $\mathcal{O}_{plan}$ increases the search space and the runtime heavily while the number of patches converges to the maximum number patches that can be scanned during one orbit at $N_{max} = 8$.

Up next, we investigate the anytime property of the planning algorithm which is important to generate plans for long missions in reasonable time. The planning algorithm is designed to find plans of good quality very early. Therefore we can force the algorithm to stop at a certain runtime to decrease the runtime as much as possible. It is important to point out that we may loose plan quality but we can guarantee that only patches of lower priority in each horizon are omitted since the timeout forces the algorithm to stop adding patches of lower priority compared to the already planned set of patches at a certain time.

Table 3 shows the anytime properties of the algorithm. We can reduce the runtime of the algorithm by 52% using a timeout of 120 s with a loss of only 10% plan quality. A result for one orbit, $N_{max} = 8$, a planning horizon of 10% and no timeout is shown in Figure 9. The run-

Table 2: Runtime of the planning algorithm varying $\mathcal{O}_{plan}$ scaled by N_{max} over mission time of one orbit.

N_{max}	horizon	patches	runtime
1	10% of Orbit	32	9 min
2	10% of Orbit	41	10 min
3	10% of Orbit	52	19 min
4	10% of Orbit	54	26 min
5	10% of Orbit	56	34 min
6	10% of Orbit	58	41 min
7	10% of Orbit	60	47 min
8	10% of Orbit	61	54 min
∞	10% of Orbit	61	155 min

Table 3: Anytime properties for a 10% horizon with $N_{max} = 8$ for the mission time of one orbit.

timeout	patches	runtime
10 s	32	7 min
60 s	45	17 min
120 s	55	26 min
240 s	58	44 min
∞	61	54 min

time[4] of the algorithm is 54 min. The number of patches scanned during the orbit period is 61.

6 Conclusion

We presented a planning algorithm for agile Earth observation missions. Starting point is the ASSET toolbox which calculates the attitude guidance and related torque profiles for a given patch sequence. This allows to check whether given kinematic or dynamical constraints are satisfied or not. This information is used to iteratively generate a *feasible* observation plan with respect to the priority of the operation sites. The complexity of the scenario is shown to be intractable for scenarios with a large number of observation sites. Considering visibility and time constraints over a short planning horizon, we can generate plan segments of good quality, depending on the planner parametrization. Generally, we can state that there is a trade-off between runtime of the algorithm and the number of patches contained in the final plan. While planning over a time horizon does not affect the plan quality significantly, the overlap analysis is an important instrument to adjust the number of patches subjected to planning to the agility of the satellite. Since patches of high priority are considered first in the presented algorithm, the plan shows good quality after short runtime already. Therefore, we can generate plans for larger mission times in reasonable time using a timeout for planning of each planning horizon.

7 Future Work

Subject of future work is an optimization of the ASSET toolbox with respect to runtime. Faster feasibility checks without simplifying the dynamics of the satellite will directly increase the runtime of the planning algorithm. Furthermore we want to include the stereo observation of patches from different angles. Another aspect is the data management. Planning of memory resources and dumping possibilities during ground station visibility pose additional constraints. In case of optical observation instruments there are also day and night periods to be considered as well as short term weather forecasts.

[4] All simulations are performed on a 64bit system with CPU of 3.3GHz and 4GB of memory.

Nomenclature

$\langle\rangle$ .... empty sequence
$\{\}$ .... empty set
s .... sequence of observation patches
$\mathcal{O}$ .... set of observation patches
$|\mathcal{O}|, |s|$ .... number of elements in $\mathcal{O}, s$

References

Aldinger, J. and Löhr, J. (2013). Planning for Agile Earth Observation Satellites. In Proceedings of the ICAPS-2013 Workshop on Planning in Continuous Domains (PCD 13).

Barschke, M. F., Levenhagen, J., Reggio, D. and Roberts, P. C. E. (2012). Design and Analysis of Manoeuvre Scenarios for Highly Agile Satellites. In Proceedings of the 62. German Aerospace Congress.

Dornhege, C., Eyerich, P., Keller, T., Trüg, S., Brenner, M. and Nebel, B. (2012). Semantic attachments for domain-independent planning systems. In Towards Service Robots for Everyday Environments.

Jacobsen, K. (2005). High Resolution Satellite Imaging Systems-an Overview. Photogrammetrie Fernerkundung Geoinformation *2005*.

Lamaitre, M., Verfaillie, G., Frank, J., Lachiver, J.-M. and Bataille, N. (2000). How to Manage the new Generation of Agile Earth Observation Satellites. In Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space.

Wong, H.-S., Yao, Y. L. and Schlig, E. S. (1992). TDI charge-coupled devices: design and applications. IBM Journal of research and development *36*.