# Planning for Agile Earth Observation Satellites

**Johannes Aldinger and Johannes Löhr**
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Georges-Köhler-Allee 52
79110 Freiburg, Germany
{aldinger, loehr}@informatik.uni-freiburg.de

### Abstract

Agile Earth observation satellites are satellites orbiting Earth with the purpose to gather information of the Earth's surface by slewing the satellite toward regions of interest. Constraints arise not only from dynamical and kinematic aspects of the satellite and its sensors. Regions of interest change over time and bad weather can conceal important observation targets. This results in a constant need to replan the satellite's tasks and raises the desire to automatize this planning process. We consider the Earth observation problem with the help of the module extension of the numerical planning system Temporal Fast Downward. Complex satellite slew maneuvers are calculated within modules, while the planner selects and schedules the regions to be scanned. First results encourage deeper research in this area so that forthcoming satellite space missions can draw on automated planning to improve the performance of agile Earth observation tasks.

## Introduction

We are interested in the feasibility of automated planning techniques in the context of Earth observation scenarios. The task of Earth observation missions is to scan regions of interest, straight stripes referred to as *patches*, during the flyover.

A task in the context of an Earth observation mission is to select and to schedule a sequence of observation patches. Determining the sequence of patches is a rather simple discrete planning problem for current automated planning systems. However, complex numerical calculations have to be performed to determine the slew maneuver of the satellite to approach and scan a patch. The feasibility of the discrete plan tightly depends on the continuous aspects since it has to consider the satellite's orbital motion, its attitude and angular rate as well as its torque capability in realistic scenarios. Instrument alignment and 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 position in orbit may affect the feasibility of future scan maneuvers. This makes the problem difficult to solve in case of larger sets of patch observations. Nevertheless, it allows to decouple the deterministic planning task, form the numeric calculations which makes Earth observation an appealing task for modular planning systems e.g. Temporal Fast Downward with Modules (TFD/M) (Dornhege et al. 2009).

In recent years, potent automated planning systems emerged from the planning community. Often, these planners are tested on IPC benchmark domains. While these benchmarks are well suited to determine strengths and weaknesses of different planning systems, the potential for industrial applications has not been fully exploited yet. We aim to solve realistic problems relevant to industrial needs. Other examples of successful industrial applications include work of Penna et al. (2010) and Fox, Long, and Magazzeni (2011). Thereby, weaknesses of current planning systems are discovered, so that future research can support the applicability of real world problems.

We found the planning system TFD/M to be suitable for our purpose. In TFD, numerical calculations, in our case slew maneuvers of the satellite, are outsourced into modules, while the basic planning problem, the selection and scheduling of the patches, is performed by TFD.

## Basics

In this section we define the Earth observation task. Afterwards, we will show how to solve the problem by an automated planning system. We contemplate over automated planning systems in general at first before considering the TFD/M planning system.

### Earth observation

Earth observation missions are an important topic in aerospace where the goal is to scan the Earth's surface with the help of satellites. Earth observation applications include among others geodesy, cartography, climatology and weather forecast. Depending on the desired application, different sensors from radar over infrared to visual sensors are used. The patches of interest are straight stripes that have to be scanned at a constant scan velocity which depends on the sensors used. The satellite has only restricted storage capacities and has to dump collected data to a ground station from time to time. The regions of interest change over time, and weather influences the visibility of interesting patches. Earth observation tasks vary in the agility of sensors which usually depends on the sensor's weight. Some instruments can be aligned to regions of interest without altering the attitude of the satellite. *Agile* Earth observation tasks considered in this paper carry heavy instruments which are firmly fixed to the satellite. Patches are scanned by slewing the satellite's line of site towards the regions of interest.

As of now, human experts identify reasonable and feasible maneuvers by hand. The identified maneuvers are then optimized and verified by potent physics simulation tools, before they are transmitted to the satellite.

## Modeling a Planning Problem

To solve a real world planning problem with an automatized planning system, it is necessary to model it first. Formally, a planning task is defined as tuple $\langle \mathcal{V}, s_0, s_\star, \mathcal{O} \rangle$. The set of variables $\mathcal{V}$ contains Boolean variables with domain $\{\top, \bot\}$ as well as numeric variables with domain $\mathrm{dom}(v) \subseteq \mathbb{Q}$ for $v \in \mathcal{V}$. The *states* of the planning problem are assignments of all variables $v \in \mathcal{V}$ to a value in their domain and $s_0$ is the initial state. The goal states $s_\star$ are defined by a partial assignment over some of the logical variables. The set of operators $\mathcal{O}$ contains operator triples $\langle C, E, c \rangle$ consisting of preconditions $C$, effects $E$ and an action cost $c$.

Driven by the International Planning Competition IPC, the predominant language to describe planning tasks is the Planing Domain Description Language (PDDL) (Ghallab et al. 1998). The interface for "semantic attachments" in PDDL (Dornhege et al. 2009) allows the addition of three types of modules to the planning domain: `conditionchecker` modules, `cost` modules and `effect` modules. `Conditionchecker` modules evaluate to logical (Boolean) variables that occur in the precondition of a planning operator. Similarly, `cost` modules represent numeric variables. When `cost` modules occur in the precondition of a planning operator, the numeric value is compared to another numerical statement with a comparison operator $\{<, \leq, =, \geq, >\}$ while it can also be used directly to determine the action cost. Finally, `effect` modules modify a set of variables in the planner state. The modified variables can be either logical or numeric variables.

## Temporal Fast Downward with Modules

To solve a continuous planning problem such as the Earth observation problem, a planning system capable of dealing with numeric variables is required. Even though numeric domains can be successfully solved with a numeric planner (Löhr et al. 2012) it is favorable to outsource numeric complexity into modules.

An extension of the fast downward (FD) planning system (Helmert 2006) to allow for temporal and numerical aspects (TFD) has been proposed by Eyerich, Mattmüller, and Röger (2009). TFD extends Fast Downwards Context Enhanced Additive Heuristic to numerical variables. While TFD supports numerical state variables, heuristic estimates in numerical rich domains are coarse. To handle complex numerical processes, the calculation should be separated from the logical planning task. The modules extension TFD/M (Dornhege et al. 2009) allows to access "semantic attachments", modules that outsource the numerical

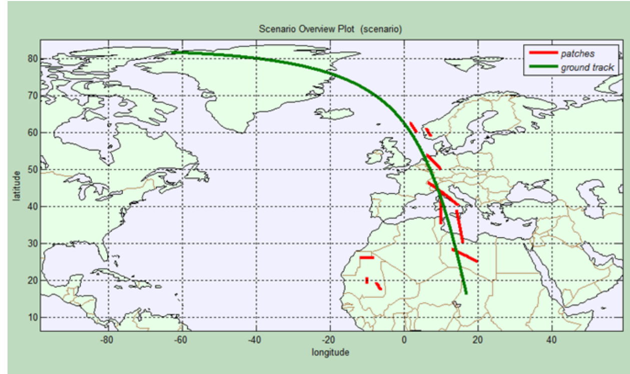


Figure 1: Earth observation scenario with subtrack of the satellite and observation sites to be scanned.

calculations from the logical planning level. TFD/M interleaves the causal planning problem "what to do" with the numerical task "how to achieve it". Neither a top-down nor a bottom-up approach can satisfy the interdependency between low level calculations and high level plan structure, and we therefore rely on an interleaved approach. A classical top-down decomposition of the planning task solves the problem on an abstract symbolic domain, and then refines that symbolic plan. In the case of Earth observation the planner would first schedule the sequence of patches to be scanned, while the maneuvers to follow this sequence would then be calculated in a refinement step. The drawback of a top-down approach is, that high-level solutions can be incorrect or pose contingencies for low-level planners. Even if the maneuvers are feasible, the resulting plan is unlikely to be good. The opposite approach, a bottom-up decomposition, precomputes *all* refined solutions, so that a higher level symbolic planner can then draw on the lower level plans. While the resulting plans are usually optimal, precomputing all low level solutions requires excessive memory and runtime. In continuous settings there are infinitely many low level plans. In the Earth observation scenario, all possible maneuvers would have to be precomputed which is intractable even for coarse discretizations. The semantic attachments of TFD evaluate the decomposition of a symbolic action on demand and can thus involve the interdependency between high level symbolic actions and low level numeric calculations. This allows TFD/M to solve an Earth observation task as presented in the next section.

## Planning the Earth observation task

We described the Earth observation scenario problem, and a tool to solve it: TFD/M. When solving Earth observation problems with TFD/M, the continuous world has to be discretized. Then, we can exploit the strength of modern planning systems: the selection and scheduling of the good actions.

Our general framework is a three step process. At first, we precompile the real world problem into a planning task, at second solve this planning task with TFD/M and finally verify the planned results with a physics simulator. The pre-

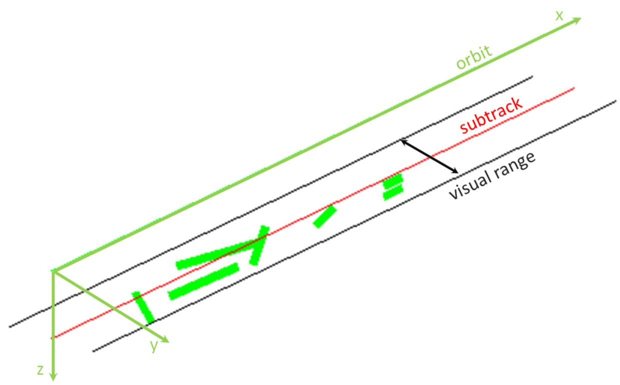


Figure 2: Planning Problem obtained from the Earth observation scenario of Figure 1 after preprocessing.

processing step reduces some of the numerical complexity from the planning problem. We consider the subtrack obtained by projecting the satellite perpendicular to the earth surface and peel off a stripe of the Earth's surface following the subtrack. The width of the peeled off stripe includes all patches that are within the satellite's visual range determined by its maximal angular deflection. For an example, consider the Earth observation scenario in Figure 1. The ground track of the satellite is depicted by a green line. The patches to be scanned are depicted in red. Some of the patches in north west Africa are out of range of the satellite's sensors. The planning problem that arises after preprocessing is depicted in Figure 2. The green patches correspond to the observation sites in Figure 1. In the planning problem we omit Earth's curvature and treat the surface of the precompiled problem as long plane with a satellite flying over it on an orbit depicted in olive green (Figure 2). If the problem horizon entails multiple satellite orbits, the same "patch" can be visible from different orbit positions. This results in multiple instances of the same patch in the precompiled stripe, usually in different orientation. To distinguish such patches, we use the term *observation site* for a site on Earth that has to be scanned, and use *patch* for a concrete instance observed from the current orbit.

Scanning patches corresponds to achieving soft goals because it is not always possible to scan all patches in the planning problem. Following Keyder and Geffner (2009) we introduce an action to `ignore` an observation site, which results in a high penalty cost. By modifying the penalty for ignoring a patch, the observation sites can be given different priorities. The goal of the planning problem is to *deal* with all observation sites, which can be done by either scanning a patch, or by actively ignoring it. While scanning occurs at a cost depending on the optimization criterion of the planning problem (e.g. available time or energy consumption) ignoring an observation site occurs at a much larger penalty cost.

The state variables $\mathcal{V}$ of the Earth observation planning problem contain logical variables as well as numerical ones. Some "variables" can not change their value during the planning process and we refer to them as *constants*. We use the common notion of *fluents* for variables that can be manipulated by the planning operators. The state of the Earth observation scenario contains Boolean constants (e.g. `(belongsto ?patch ?osite)` describing that a patch belongs to an observation site) Boolean fluents (e.g. `(dealt ?osite)` describing that an observation site has been processed) numerical constants describing satellite parameters (e.g. `(roll-max ?sat)` describing the maximal roll angle of the satellite) as well as numerical fluents (e.g. `(roll-angle ?sat)` describing the current roll angle of the satellite).

In the initial state $s_0$, no observation site has been *dealt* with, the numerical fluents describe the satellite's current orbit position, attitude and angular rates. The numeric constants model the attitude constraints of the satellite such as maximal angle deflection and maximal angular rates. The goal states $s_\star$ of the planning task are all states, where all observation sites have been *dealt* with. The planning operators $\mathcal{O}$ are `scan` to scan a patch and `ignore` to ignore an observation site.

## Discretization

To model the Earth observation scenario, each state of the Earth observation planning problem describes a "snapshot" of the continuous world. Usually the satellite has just scanned a patch and the numeric state fluents describe the attitude and rate of the satellite in this position with line of sight toward the end of the patch. Discrete planning decisions are made between these snapshot states. The available actions at such a state are to either scan or to ignore one of the remaining patches. While ignoring a patch results in a discrete successor state only altering Boolean fluents, determining the successor snapshot state of the planner after applying a `scan` action is not obvious. As the world is continuous, deciding for the next patch to scan could result in infinitely many possible successor states, since it is possible to scan a patch from different positions in the orbit. To commit to one discrete successor state after deciding for a patch to scan, we make the following assumption:

**Assumption 1.** *It is always best to scan a patch as soon as possible.*

We assume that it is always best to scan the chosen patch as soon as possible, thus leaving wider scope for future actions. This assumption implies that it is more important to scan many patches than to scan them with a good image quality which is usually obtained when the angular deflection of the satellite's line of sight is minimal. It is not obvious how to calculate the earliest possible satellite state to scan the patch.

In the following we will show how this "earliest possible" satellite state at the start of the scan maneuver can be estimated. We will first look at the extreme positions and omit the constraints posed from other patches. Then we will propose a method based on interval nesting to determine the earliest possible orbit position to start scanning the selected patch. At first we consider the case, where the patch to scan is far away from the satellite's current orbit position. The

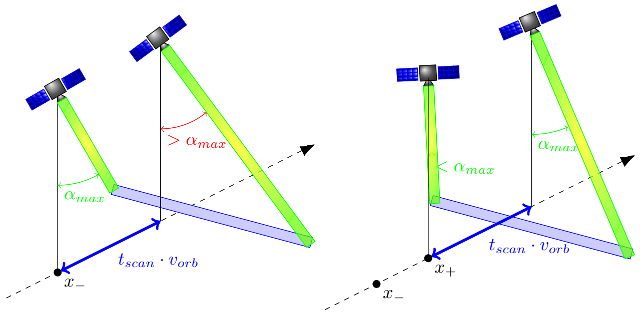


Figure 3: Satellite states to determine the earliest possible position to scan the patch

earliest possible approach configuration of a satellite is obtained by deflecting the satellite as far possible. The maximal deflection is limited by the maximal angle under which the sensor can operate.

An example is illustrated in Figure 3. The subtrack of the satellite is depicted by the dashed line. The left satellite in the left graphic depicts the state of the satellite at the earliest orbit position $x_-$ to scan the patch. There, the satellite is deflected with the maximal angle $\alpha_{max}$ towards the patch. However, the attitude dynamic constraints of the satellite could be violated by scanning the patch starting form $x_-$. After scanning, the satellite would be in the infeasible state depicted on the right side of the left graphic in Figure 3. In this case, the state of the satellite when exiting the patch is critical to approach the patch as early as possible. The right satellite in the graphic on the right depicts the earliest possible satellite state to finish scanning the patch. This state is reached, if the scan started at orbit position $x_+$. The orbit position $x_+$ can be calculated because the scan time needed for scanning the patch $t_{scan}$ and the orbital velocity $v_{orb}$ are known. Depending on the scanning speed and the orientation of the patch relative to the satellite's subtrack position, either $x_-$ or $x_+$ can be the critical earliest possible orbit positions to scan the patch. The satellite state $s_{first}$ is the state adopted at the earliest possible orbit position $max(x_-, x_+)$.

Analogously the last possible orbit position to scan the patch can be computed by minimizing over the latest attitude under which the satellite can start or end a scan, where the satellite adopts state $s_{last}$. All feasible maneuvers to scan the patch are in the interval between the orbit positions at $s_{first}$ and $s_{last}$. In real planning problems with multiple patches it is not likely that all patches can be scanned as early as $s_{first}$. Instead, scanning a patch should start as soon as possible from the satellite's current state. Unfortunately, neither the satellite's orbit position nor its attitude after executing this best slew maneuver are known in advance. The principal problem of finding the earliest orbit position to start the next scan builds a non-linear equation system for which no closed form solutions methods are known to us. We therefore approximate the satellite state with the help of interval nesting.

The flow chart in Figure 4 illustrates the mathematical calculations needed inside a `scan` planning operator, which

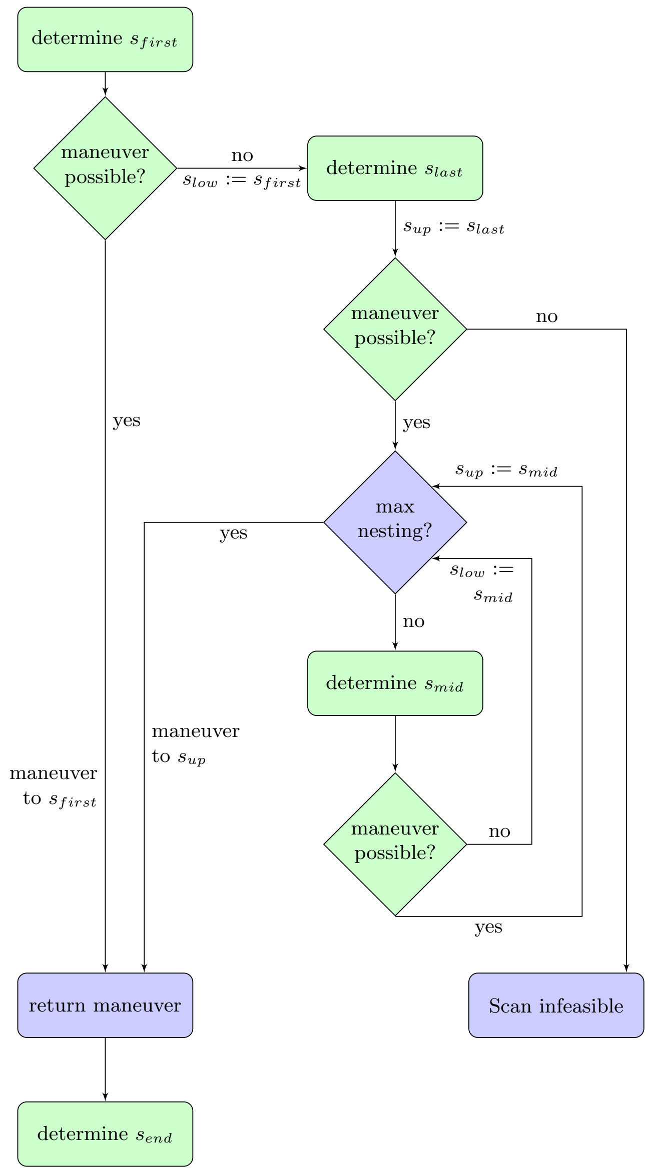


Figure 4: Inside a `scan-patch` planning operator. Green boxes can be calculated by Function 1 while green diamonds are calculated by Function 2.

approximates the earliest orbit position to scan the patch as well as the corresponding slew maneuver. Two types of mathematical calculations have to be performed frequently in an Earth observation planning task:

**Function 1.** *Determine the satellite's state (attitude and angular rates) from a given orbit position when pointing towards a patch, with angular rates satisfying the vectorial velocity for scanning the patch.*

**Function 2.** *Determine the feasibility of a maneuver given two satellite states.*

We will take a deeper look at the mathematical calculation of Function 1 and Function 2 in the next section and assume for now that both functions can be computed efficiently. The green boxes in the flow chart in Figure 4 are all instances of Function 1 while the green diamonds can be calculated with Function 2. The orbit position after the optimal slew maneuver from the current state of the satellite to the best approach state lies in the interval between the orbit position at $s_{first}$ and $s_{last}$. If a maneuver from the current satellite state $s_{current}$ to the earliest possible scan configuration is possible, this maneuver from $s_{current}$ to $s_{first}$ is optimal, and can be returned. After scanning the patch the satellite adopts state $s_{end}$, with an orbit position depending on orbital velocity and the scanning time, both of which are given in the domain description. The deliberations made for $s_{first}$ and depicted in Figure 3 ensure that $s_{end}$ is valid. If the maneuver from $s_{current}$ to $s_{first}$ is infeasible, it is used as lower bound $s_{low}$ of an interval nesting, and $s_{last}$ is calculated as latest possible orbit position to scan the patch. If the maneuver to $s_{last}$ is infeasible, the patch can not be scanned at all. Otherwise, the satellite's state of a time optimal maneuver is found between the unreachable lower bound $s_{low}$ and the reachable but not time optimal upper bound satellite state $s_{up} = s_{last}$. Unless the maximal nesting depth is exceeded, we determine the satellite state $s_{mid}$ in the middle of the boundaries with the help of Function 1, and check if the maneuver from the current satellite configuration to the middle configuration is feasible with the help of Function 2. If the maneuver is feasible, a better upper bound has been found, while the $s_{mid}$ is used as lower bound if the time slew time is exceeded.

In our implementation we limit the depth of the interval nesting to 10 which offers a good trade-off between run-time (less than 1 ms) of the operation and precision (approximately 10 m deviation). We note that more sophisticated interval nesting methods to determine the orbit position of the most promising middle configuration could be used.

While we calculate the satellite's attitude and angular rates $s_{end}$ after scanning, we do not check if the maneuver from $s_{up}$ to $s_{end}$ is feasible with our method described in Function 2.

**Assumption 2.** *When entry and exit configuration are feasible, the whole scan maneuver is.*

The maneuver of the satellite during a scan is given by kinematical constraints and not considered here in this paper.

As mentioned earlier, the extension of the PDDL planning language allows planning operators to contain three types of modules: `conditionchecker` modules, `cost` modules and `effect` modules. In our implementation, scanning a patch is decoupled into two actions `approach-patch` and `scan-patch`. This is mostly done for technical reasons, since it is easier to determine the satellite state after each operator execution during planning which makes it easier to verify the feasibility of the plan during post processing. Within these two operations, we use five modules, which all calculate parts of the flow chart of Figure 4. To avoid the recalculation of the same function, a database stores all calculations performed by Function 1 and Function 2. The time intensive interval nesting is therefore only computed once for each configuration.

The modules executed by the `approach-patch` operator all follow the flow chart diagram (Figure 4) and basically compute the same thing. A `conditionchecker`-module `approach-patch-possible` tests, if approaching a patch is possible, a `cost`-module `approach-time` determines the maneuver time to approach the patch, and finally an `effect`-module `approach-effect` modifies the planning state and sets the planning variables of the satellite to $s_{up}$. The modules used in the `scan-patch` operator are a rather simple module `scan-time` that calculates the time needed for scanning. We do not need a `conditionchecker` module, since Assumption 2 ensures that the feasibility of each scan is already checked by approaching the patch. The `scan-effect` module obviously sets the planner state of the variables concerning the satellite to $s_{end}$. Additionally the planning operator sets the corresponding observation site `dealt` variable to true.

## Satellite Attitude Dynamics

In the previous section we have identified two mathematical functions calculating the attitude and angular rates of the satellite and the feasibility of the slew maneuvers. Both have to be calculated frequently within the modules of the planner. Function 1 consists of two parts: determining the *attitude* of the satellite when pointing towards the targeted patch, and the *angular rates* of the satellite that is necessary to scan the patch. Afterwards we will present the calculations for Function 2 that checks the feasibility of the slew maneuver.

**Coordinate Systems** The Earth observation scenario is preprocessed from the Earth spherical coordinates (see Figure 1) to a flat cartesian coordinate system along the subtrack of the satellite (see Figure 2). The x-axis points in flight direction, the z-axis points towards center of Earth (Nadir) and the y-axis completes the right hand coordinate system. The center of the coordinate system lies within the center of mass of the satellite at time $t_0$. This Earth fixed planning frame is notated as N-frame. The satellite's body-fixed frame is referred to as B-frame, where the z-axis is assumed to be coaxial with the instruments line of sight. A vector $\mathbf{x}$ in N-frame is notated as $\mathbf{x}^N$, in body fixed coordinates as $\mathbf{x}^B$, respectively and its norm is denoted as $\|\mathbf{x}^N\|$. The components of a vector are identified by stating the respective axis in subscript e.g. $\mathbf{p}^N = [p_x^N, p_y^N, p_z^N]$.

**Attitude Determination** We define the attitude of the satellite's fixed body-frame, the B-frame by its Euler angles with respect to the N-frame. The roll angle $\phi$ is defined by a rotation of the satellite around the x-axis and the roll rate is referred to as $\dot{\phi}$. The pitch angle $\theta$ and pitch rate $\dot{\theta}$ are defined around the y-axis and the yaw angle $\psi$ and yaw rate $\dot{\psi}$ are defined around the z-axis respectively. Vectors defined in the N-frame are transformed into the B-frame by the direct cosine rotation matrix

$$
DCM^{BN} = DCM_\theta \; DCM_\phi \; DCM_\psi, \text{where}
$$
$$
DCM_\theta = \begin{pmatrix} \cos -\theta & 0 & \sin -\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos -\theta \end{pmatrix}
$$
$$
DCM_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos -\phi & \sin\phi \\ 0 & \sin -\phi & \cos -\phi \end{pmatrix} \quad (1)
$$
$$
DCM_\psi = \begin{pmatrix} \cos -\psi & \sin\psi & 0 \\ \sin -\psi & \cos -\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.
$$

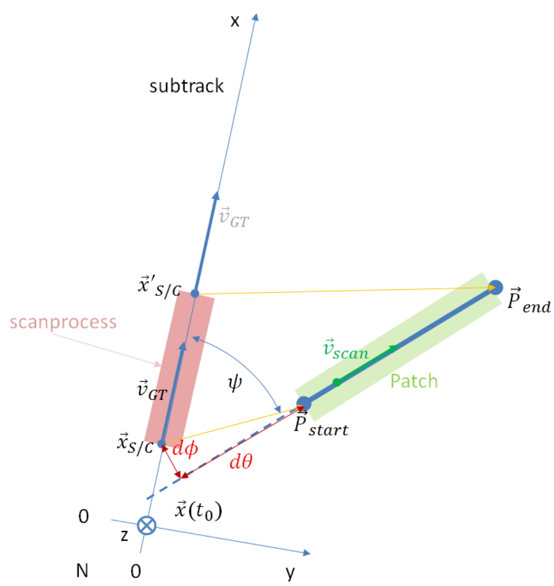


Figure 5: Geometry of an exemplary patch position

The patch to be observed is specified by a start coordinate $\mathbf{P}^N_{start}$ and an end coordinate $\mathbf{P}^N_{end}$ and has to be scanned with a constant scan velocity $v_{scan}$. The direction of the patch is given by

$$
\mathbf{p}^N = \mathbf{P}^N_{end} - \mathbf{P}^N_{start}
$$

with length $l = \|\mathbf{p}^N\|$ which leads to the scan time

$$
t_{scan} = \frac{l}{v_{scan}}.
$$

To scan a patch from an orbit position $\mathbf{x}^N_{S/C}$, the instrument's line of sight has to point to the start position of the patch, which corresponds to a specific attitude $DCM^{BN}$ of the B-frame with respect to the N-frame. Additionally the angular rate of the satellite $\boldsymbol{\omega}_{S/C} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ is specified by the scan velocity of the patch. The attitude and angular rate of the satellite after scanning a patch can be obtained similarly. Here the satellite has also a specified attitude $DCM^{BN}$ and angular rate $\boldsymbol{\omega}_{S/C}$ depending on the new orbit position at $\mathbf{x}'^N_{S/C} = \mathbf{x}^N_{S/C} + \mathbf{v}^N_{GT}\, t_{scan}$ and the end position of the patch. Both, attitude and angular rate are functions of $\mathbf{x}^N_{S/C}$ and patch point $\mathbf{P}^N$ to be aimed at. In order to calculate the attitude $DCM^{BN}$ we sequence rotations around the yaw axis, the roll axis and the pitch axis. The satellite yaws with angle $\psi$, as depicted in Figure 5.

$$
\psi = \arctan \frac{\mathbf{p}^N_y}{\mathbf{p}^N_x}
$$

The line of sight vector $\mathbf{r}^N = [\mathbf{P}^N - \mathbf{x}^N_{S/C}]$ points from the spacecraft to a start point or to an end point of a patch. This is firstly rotated to a auxiliary frame $H$ with angle $\psi$

$$
\mathbf{r}^H = DCM^{HN}_\Psi \mathbf{r}^N
$$

which yields the roll angle

$$
\phi = \arctan(\frac{d\phi}{h})
$$

and the pitch angle

$$
\theta = \arctan(\frac{d\theta}{\sqrt{h^2 + d\phi^2}}),
$$

where $d\phi = r^H_y$ and $d\theta = r^H_x$. The direct cosine matrix, that defines the attitude of the satellite pointing the instrument to a start or end point of a patch, can finally be computed by Equation 1.

**Angular Rate Determination** The required scan velocity of the line of sight at the patch

$$
\mathbf{v}^N_{scan} = v_{scan} \frac{\mathbf{p}^N}{\|\mathbf{p}^N\|}
$$

is generated by the subtrack velocity $\mathbf{v}^N_{GT}$ and a compensating velocity induced by a rotation of the satellite

$$
\mathbf{v}^N_{comp} = \mathbf{v}^N_{scan} - \mathbf{v}^N_{GT}.
$$

Using the compensation velocity transformed to the B-frame

$$
\mathbf{v}^B_{comp} = DCM^{BN} \mathbf{v}^N_{comp}
$$

and the line of sight vector in body frame

$$
\mathbf{r}^B = DCM^{BN} \mathbf{r}^N
$$

the angular rate in body frame $\boldsymbol{\omega}^B$ can be implicitly obtained by

$$
\mathbf{v}^B_{comp} = \boldsymbol{\omega}^B \times \mathbf{r}^B
$$

The compensation velocity is a cross product of the satellite rotation and the line of sight vector

$$
\mathbf{v}^B_{comp} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ \|\mathbf{r}\| \end{bmatrix} = \begin{bmatrix} \omega_y \|\mathbf{r}\| \\ -\omega_x \|\mathbf{r}\| \\ 0 \end{bmatrix}
$$

which leads to $\omega^B_y = \frac{\mathbf{v}^B_{comp.x}}{\|\mathbf{r}\|}$ and $\omega^B_x = -\frac{\mathbf{v}^B_{comp.y}}{\|\mathbf{r}\|}$. The angular rate in N-frame finally is given by

$$
\boldsymbol{\omega}^N = DCM^{BN} \boldsymbol{\omega}^B.
$$

**Slew Feasibility** We are interested to check the feasibility of a slew maneuver between two arbitrary patches meaning a maneuver that slews the satellite from an arbitrary initial attitude and angular rate to a desired attitude and angular rate. Ideally the necessary torque profile in B-frame is analysed. This is computationally expensive and exceeds the acceptable runtime of modules by far. Therefore we introduce conservative assumptions to be able to quickly check the feasibility of a slew.

**Note on Maximum Torques** Torques $\mathbf{T}$ acting on a rigid body and the resulting angular rates $\boldsymbol{\omega}$ in body fixed coordinates are connected by the well known Euler equation

$$\mathbf{T}^B = J^B \dot{\boldsymbol{\omega}}^B + \boldsymbol{\omega}^B \times J^B \boldsymbol{\omega}^B, \tag{2}$$

where $J$ is the (here diagonal) inertia matrix of the satellite. During the slew the coupling of the axes is compensated by nonlinear feedback control such that

$$\mathbf{T}^B = \mathbf{T}^B_C + \mathbf{T}^B_{max}, \text{ where} \tag{3}$$

$$\mathbf{T}^B_C = \boldsymbol{\omega}^B \times J^B \boldsymbol{\omega}^B. \tag{4}$$

This allows to assess the maximum angular acceleration which can be realized during a slew around an axis $i$ of the B-frame

$$\dot{\omega}^B_{max,i} = \frac{T^B_{max,i}}{J_{ii}}. \tag{5}$$

**Slew Feasibility** The calculation of the slew maneuver is done in the N-frame to reduce the computational cost. However, the torques have physically to be generated in the body fixed frame. Therefore we use a conservative upper border for the maximum allowable angular acceleration in N-frame

$$\dot{\omega}^N_{max} = \min_i \sqrt{\frac{(\dot{\omega}^B_{max,i})^2}{3}} \tag{6}$$

such that the resulting torques in the body fixed B-frame cannot be exceeded. We use a steplike angular acceleration profile with duration $\Delta t$ see Figure 6.

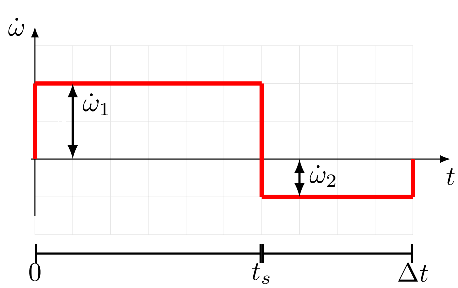


Figure 6: Generic steplike torque profile around one axis of the N-frame

The slew is calculated around each axis separately. Starting with the angular rate $\omega(t_0)^N$ at the beginning of the slew and the initial Euler angle $\alpha(t_0)$ corresponding to $\phi(t_0)$, $\theta(t_0)$ or $\psi(t_0)$ and the desired angular rate $\omega(t_0 + \Delta t)$ and angle $\alpha(t_0 + \Delta t)$, respectively, the necessary change of the states in each axis is given by

$$\Delta\alpha = \alpha(t_0 + \Delta t) - \alpha(t_0)$$
$$\Delta\omega = \omega^N(t_0 + \Delta t) - \omega^N(t_0).$$

Integration yields

$$\Delta\omega = \int_0^{\Delta t} \dot{\omega}^N(t)\, dt = \dot{\omega}_1\, t_s + \dot{\omega}_2\, (\Delta t - t_s), \tag{7}$$

corresponding to Figure 6. We denote $t_s$ as switching time in between both angular accelerations. The change in the angle is

$$\begin{aligned}\Delta\alpha = &\iint_0^{\Delta t} \dot{\omega}^N(t)\, dt\, dt \\ =&\frac{1}{2}\dot{\omega}_1 t_s^2 + \omega(t_0) t_s + \frac{1}{2}\dot{\omega}_2(\Delta t - t_s)^2 \\ &+ (\omega(t_0) + \dot{\omega}_1 t_s)(\Delta t - t_s).\end{aligned} \tag{8}$$

Equation 7 can be converted to

$$\dot{\omega}_2 = \frac{\Delta\omega - \dot{\omega}_1 t_s}{\Delta t - t_s}$$

while Equations 7 and 8 yield

$$t_s = \frac{\Delta\alpha - \frac{1}{2}\Delta\omega\Delta t - \omega_0 \Delta t}{\frac{1}{2}\dot{\omega}_1 \Delta t - \frac{1}{2}\Delta\omega}. \tag{9}$$

Equation 9 solely depends on the unknown initial angular acceleration $\dot{\omega}_1$. We set

$$\dot{\omega}_1 := \pm\dot{\omega}^N_{max}$$

under the assumption that it is always reasonable to begin a slew with maximum angular acceleration in order to have a maximum scope for the choice of $t_s$ and $\dot{\omega}_2$. The correct sign of $\dot{\omega}_1$ can be found by evaluation of $t_s$.

$$\dot{\omega}_1 = \begin{cases} \dot{\omega}^N_{max}, & t_s > 0 \\ -\dot{\omega}^N_{max}, & t_s < 0 \end{cases}$$

Equation 9 is evaluated again, if necessary. If $t_s$ or $\dot{\omega}_2$ holds one of the following equations

$$t_s < 0$$
$$t_s > \Delta t$$
$$|\dot{\omega}_2| > \dot{\omega}^N_{max}$$

for any axis of the N-frame, the maneuver is infeasible[1].

## Experimental Results

To test the feasibility of our approach, we implemented the precompiled planning problem from Figure 2 modeling the Earth observation scenario from Figure 1 in PDDL and solved it with TFD/M. Figure 7 shows a visualization of the

[1] It is worth to mention that the commanded maneuver in B-frame could be feasible anyhow due to the conservatism induced in Equation 6. This is part of future optimization.

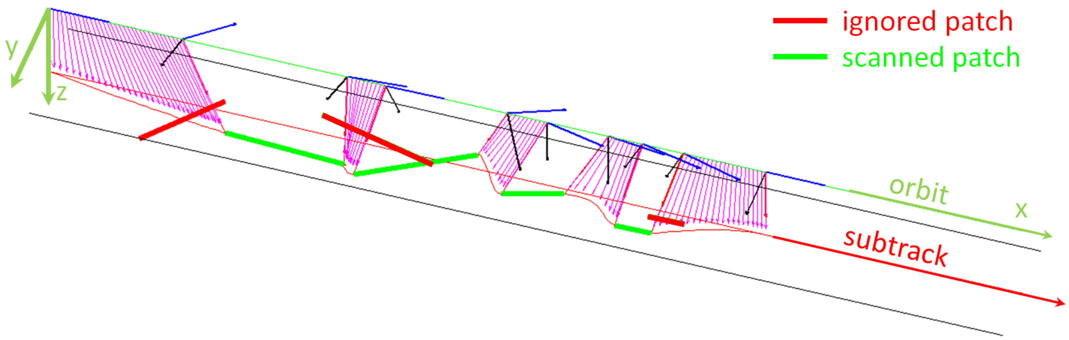


Figure 7: Earth observation scenario with subtrack of the satellite and observation sites to be scanned. The magenta colored arrows depict the line of sight of the instrument during the slew maneuver.

intermediate states of the satellite extracted from the resulting plan. The satellite's attitude is depicted by the current body fixed frame in blue black and magenta.

The satellite slews towards the first patch to scan it at its earliest possible state. The start of the scan maneuver of all other patches is constrained by the attitude of the satellite after scanning the previous patch, so all other maneuvers have to be calculated by interval nesting.

The resulting plan happens to be optimal for the tested planning instance given our assumptions and the satellite parameters used. The leftmost patch cannot be scanned because the satellite's state to start the scan maneuver is infeasible. In additional experiments we investigated the influence of increasing the maximal angular rates of the satellite in the planning problem. In this case the slew maneuver from the last patch to the ignored rightmost patch becomes feasible. With more angular scope also the first patch is in range and TFD/M finds a plan scanning six of the patches.

Although our approach is promising and seems to work well in practice, optimality cannot be guaranteed, even regarding the inaccuracies of the model such as plain Earth surface, circular orbits, etc. The interval nesting approach is only iterated to a certain depth so that each scan action is started at a minimally later orbit position than theoretically possible. Now it is easy to construct a problem that will not be solved optimally by our approach by adding a new patch that is reachable by a slew maneuver from the orbit position after scanning the previous patch from the theoretical earliest orbit position but not from the orbit position found by interval nesting. Completeness can be achieved with the trivial plan, but in scenarios where all patches could be scanned it is not guaranteed that TFD/M will unnecessarily ignore some observation sites with the analogous argument as for optimality.

## Conclusion and Future Work

We have presented an agile Earth observation task and an automated planning system capable of solving it. Preliminary experiments show the feasibility of our approach. We will continue our research in more complex problems.

The automated planning system TFD/M can be successfully applied to our Earth observation scenario. Nevertheless we believe that better planning systems could be developed for larger Earth observation tasks. As modeled, the Earth observation problem does not involve temporal concurrency, and the planning problem is even serialized artificially with the help of Boolean “idle” variables. While numerical variables are required for the Earth observation scenario, the temporal aspect of TFD is not. Many other real world problems do not require temporal concurrency but have to deal with complex numerical calculations in the same way as TFD/M handles semantic attachments. A serial numerical planner would hav to solve a problem with a smaller branching factor of sequential actions instead of time stamped states which offers the potential for better heuristics. These heuristics could either be more informed or faster to compute which increases the performance of a planning system and therefore the industrial applicability.

The planning problem can be formalized with the semantic attachments extension of the planning language PDDL. However, it is strongly connected with the syntax in the modules, and the interface between modules (implemented in the programming language C++) and planning language has to be maintained by hand. An object oriented planning language would be better suited to setup the problem. A promising approach that improves the communication between modules and planning problem is the Object Oriented Planning Language OPL (Hertle 2011).

Satellites in an Earth observation mission have only limited storage capacities and have to dump collected data to a ground station from time to time. We intend to model this data handling in future implementations. Further improvements include the handling of the satellite's orbit which was modeled to be circular, while it is somewhat elliptical in reality. Considering the deviations from a circular orbit does not change the general structure of our model and is beneficial to better approximate the satellite's real behavior.

## Acknowledgments

This work was supported by the German Aerospace Center (DLR) as part of the Kontiplan project (50 RA 1221).

## References


Dornhege, C.; Eyerich, P.; Keller, T.; Trüg, S.; Brenner, M.; and Nebel, B. 2009. Semantic Attachments for Domain-independent Planning Systems. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 09)*.

Eyerich, P.; Mattmüller, R.; and Röger, G. 2009. Using the Context-enhanced Additive Heuristic for Temporal and Numeric Planning. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 09)*, 130–137. AAAI Press.

Fox, M.; Long, D.; and Magazzeni, D. 2011. Automatic Construction of Efficient Multiple Battery Usage Policies. In *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS 2011)*.

Ghallab, M.; Isi, C. K.; Penberthy, S.; Smith, D. E.; Sun, Y.; and Weld, D. 1998. PDDL - The Planning Domain Definition Language. Technical report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research* 26:191–246.

Hertle, A. 2011. Design and Implementation of an Object-Oriented Planning Language. Master's thesis, Albert-Ludwigs-University, Freiburg.

Keyder, E., and Geffner, H. 2009. Soft Goals can be Compiled Away. *Journal of Artificial Intelligence Research (JAIR)* 36:547–556.

Löhr, J.; Eyerich, P.; Keller, T.; and Nebel, B. 2012. A Planning Based Framework for Controlling Hybrid Systems. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*.

Penna, G. D.; Intrigila, B.; Magazzeni, D.; and Mercorio, F. 2010. Planning for autonomous planetary vehicles. In *27th Congress of the International Council of the Aeronautical Sciences (ICAS 2010)*, 131–136.