AUTOMATED TARGET PLANNING FOR FUSE USING THE SOVA ALGORITHM

Scott Heatwole, R. James Lanzi
NASA Goddard Space Flight Center's Wallops Flight Facility

Thomas Civeit, Humberto Calvani, Jeffrey W. Kruk, Anatoly Suchkov
Johns Hopkins University

1.0 Introduction

The SOVA algorithm was originally developed under the Resilient Systems and Operations Project of the Engineering for Complex Systems Program from NASA's Aerospace Technology Enterprise as a conceptual framework to support real-time autonomous system mission and contingency management. The algorithm and its software implementation were formulated for generic application to autonomous flight vehicle systems, and its efficacy was demonstrated by simulation within the problem domain of Unmanned Aerial Vehicle autonomous flight management.

The approach itself is based upon the precept that autonomous decision making for a very complex system can be made tractable by distillation of the system state to a manageable set of strategic objectives (e.g. maintain power margin, maintain mission timeline, and et cetera), which if attended to, will result in a favorable outcome. From any given starting point, the attainability of the end-states resulting from a set of candidate decisions is assessed by propagating a system model forward in time while qualitatively mapping simulated states into margins on strategic objectives using fuzzy inference systems. The expected return value of each candidate decision is evaluated as the product of the assigned value of the end-state with the assessed attainability of the end-state. The candidate decision yielding the highest expected return value is selected for implementation; thus, the approach provides a software framework for intelligent autonomous risk management. The name adopted for the technique incorporates its essential elements: Strategic Objective Valuation and Attainability (SOVA). Maximum value of the approach is realized for systems where human intervention is unavailable in the timeframe within which critical control decisions must be made.

The Far Ultraviolet Spectroscopic Explorer (FUSE) satellite, launched in 1999, has been collecting science data for eight years.[1] At its beginning of life, FUSE had six gyros in two IRUs and four reaction wheels. Over time through various failures, the satellite has been left with one reaction wheel on the vehicle skew axis and two gyros. To remain operational, a control scheme has been implemented using the magnetic torque rods and the remaining momentum wheel.[2] As a consequence, there are attitude regions where there is insufficient torque authority to overcome environmental disturbances (e.g. gravity gradient torques). The situation is further complicated by the fact that these attitude regions shift inertially with time as the spacecraft moves through earth's magnetic field during the course of its orbit.

Under these conditions, the burden of planning targets and target-to-target slew maneuvers has increased significantly since the beginning of the mission.[3] Individual targets must be selected so that the magnetic field remains roughly aligned with the skew wheel axis to provide enough control authority to the other two orthogonal axes. If the field moves too far away from the skew axis, the lack of control authority allows environmental torques to pull the satellite away from the target and can potentially cause it to tumble. Slew maneuver planning must factor the stability of targets at the beginning and end, and the torque authority at all points along the slew. Due to the time varying magnetic field geometry relative to any two inertial targets, small modifications in slew maneuver timing can make large differences in the achievability of a maneuver.

Target and slew maneuver planning must be done with an eye to keep stored momentum in the skew-wheel as low as possible. The reaction wheel has a saturation limit of +/- 21 Nms where there is little to no control torque availability above this level. If this limit is exceeded, the spacecraft may lose control as excess momentum is transferred to the body. Operationally the wheel is kept to +/- 14 Nms. Thus, target and maneuver plans must be set up in such a fashion so that margin is maintained.

The end result of all of this is that one week's worth of science targets requires roughly 3 to 5 working days to plan. To help reduce the burden on the ground support team, an autonomous planning tool has been developed based on the Strategic Objective Valuation and Attainability algorithm. For the FUSE science-target management problem domain, there was no requirement (or desire) for a fully autonomous planning capability. However, the existing SOVA software provides a convenient structure for intelligently automating planning tasks. Existing corporate experiential knowledge is incorporated into the planner by using the strategic objective mechanism, and there are facilities within the planner to enable the utilization of target feasibility data generated planning aid software already in operational use. The SOVA mission planning software was ported into a stand-alone computer program that could be executed for automated off-line target sequence generation which can be used by the FUSE mission planners to build a mission planning schedule. A separate application was developed that takes target priority data, and planning information generated by existing analysis programs and incorporates them into planning configuration scripts utilized by the SOVA planning application. This tool shows ability to produce viable target sequences. However, it is still unable to produce plans of the same quality as the FUSE mission planners.

2.0 Representation of Tasks

The SOVA algorithm makes use of hierarchical decomposition for the organization, machine representation, storage, and execution of tasks. Any task (e.g. “Acquire Target G035-03-01”) may be decomposed into a number of sub-tasks that may be still further decomposed into other tasks. At the bottom of the task decomposition are fundamental primitive tasks which are used to set system parameters and wait for continuation conditions. Figure 1 gives a concrete example of the decomposition of a typical task.

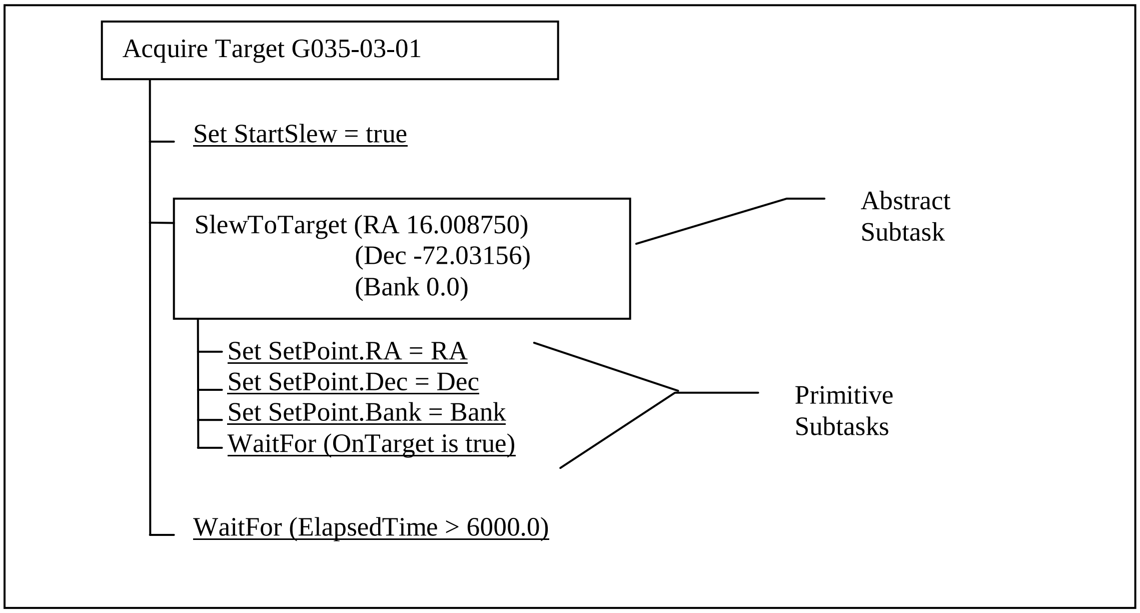


Figure 1. Hierarchal Decomposition of an “Acquire Target” Task

The planning software uses the SOVA algorithm to assemble individual tasks into a consistent plan that achieves a desired objective. To accomplish this, individual tasks may be equipped with a number of attributes to support planning and execution functions.

***Arguments* -** A task may take a list of variable arguments that are set by a parent task.

Preconditions – A task may be configured with a set of logical preconditions that must be true prior to attempting to execute the task. The preconditions are used by the planner to filter a list of candidate tasks when formulating the ‘next step’ of a plan.

Value – A task may be assigned an ‘inherent worth’ or ‘value’ that is used by the planner to provide some level of relative weight/priority when considering a list of candidate tasks to attempt.

Applicable Strategic Objectives – A task may be assigned a list of qualitative strategic objectives that apply during the execution of the task and any sub-tasks. The strategic objectives are utilized by the SOVA planner as a means of estimating the attainability of the task. The expected return value of the task is estimated as a product of the attainability of the task with its assigned value. It will be shown how the planner achieves online risk management by selecting the plan with the highest expected return value.

3.0 Strategic Objectives and the Assessment of Attainability

One of the foundational principles of the SOVA technique rests in emulating the observed real-time risk management performed by a skilled human operator when conducting a task. It is hypothesized that the human accomplishes on-the-fly risk assessment by knowing, qualitatively monitoring, and responding to a relatively small set of ‘critical items’ dubbed Strategic Objectives. For the FUSE mission planning problem, such strategic objectives include: Maintain momentum margin and avoid loss of control.

Due to the inherently qualitative nature of the process, a fuzzy inference system [4] is utilized to compute a figure of merit indicating how much margin one has for each strategic objective. A second fuzzy inference system is used to map computed strategic margin into an attainability status assessment which further depends upon the pre-assigned criticality of the specific strategic objective.

Thus each strategic objective is accompanied by a user configurable fuzzy inference system that relates state variables to an assessment of strategic margin (positive, none or negative) for that objective. Furthermore, the attainability status of the strategic objective is computed in accordance with a rule base that looks very much like a risk assessment code matrix:

Table 1. Strategic Objective Attainability Status versus Strategic Margin and Criticality

	Margin		
Criticality	**Negative**	**None**	**Positive**
Low	Marginal	Attainable	Attainable
Medium	Unattainable	Marginal	Attainable
High	Unattainable	Unattainable	Attainable

This process for assessing a status rating for a particular strategic objective is shown schematically in Figure 2.

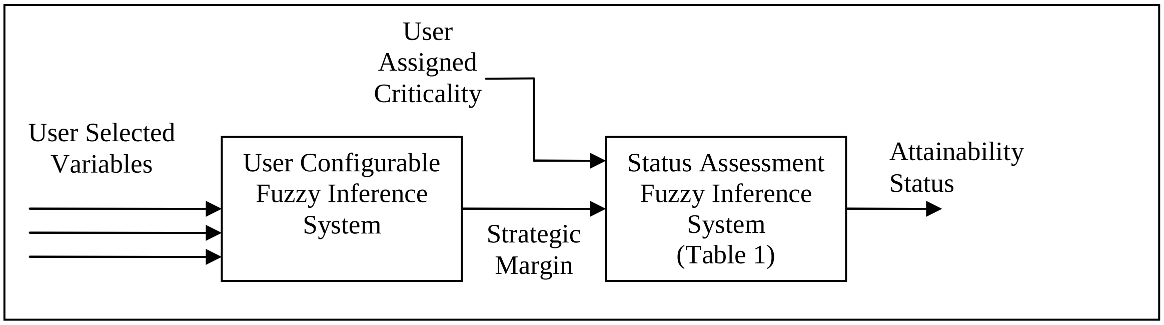

User Assigned Criticality
User Selected Variables
User Configurable Fuzzy Inference System
Strategic Margin
Status Assessment Fuzzy Inference System (Table 1)
Attainability Status

Figure 2. Strategic Objective Status Rating Calculation Process

Cubic S, PI, and Z functions are used to map real valued Strategic Margin, Criticality, and Status values into membership within the sets denoted by adjectival descriptors given in Table 1.

The Strategic Objective Attainability Status rating takes a value between 0 and 1 which can be interpreted as a conditional probability. Specifically, it is interpreted as the probability of successfully accomplishing the task given the conditions associated with the strategic objective under analysis.

At any given time, multiple strategic objectives are evaluated in parallel. Each abstract task may be assigned a list of strategic objectives which apply during its execution, and the execution of any child tasks. Thus the overall attainability of the task is computed as the product of the individual status ratings for all of the strategic objectives that are applicable during the execution of the phase of flight under consideration:

$$Overall\ \ Attainability = \prod_{All\ Objectives} Status(Objective) \tag{1}$$

Equation (1) represents the end product of a methodology that uses the Strategic Objective device as a means of mapping user selected state values at an instant in time to an overall attainability rating that is only valid at that instant. Its equivalent in human terms would be the intuitive status assessment that a vehicle operator makes upon a single glance at the dashboard. In the subsequent sections, it will be shown how this methodology can be used to project attainability values over multiple phases of a mission.

4.0 Assessment of Expected Return Value

In order to integrate the SOVA techniques with a flight planner, a number of elements are required including: a means of forecasting flight conditions that will result during execution of a candidate task, a means of taking time dependent attainability data over the projected duration of the task and converting into a single attainability figure of merit, and a means of computing a total expected return value for a plan based upon the integrated path value and the assessed attainability of each task in the plan.

For the mission planning application, a six-degree-of-freedom simulation is used as the plant model to propagate the state of the spacecraft in response to actions driven by a candidate abstract task. The simulated spacecraft state response feeds the running attainability calculations discussed in the previous section. A 4th order fixed time step Runge-Kutta integrator is used to perform the simulation.

There are a number of ways that one could conceive of converting the time-dependent attainability data computed over the duration of the execution of a task into a single attainability figure of merit required to assess the expected return value from the task. In the SOVA algorithm, the minimum value of overall attainability status occurring over the duration of the task is taken as the attainability associated with the task:

$$Task\ Attainability = \inf_{t \in Task\ Duration} \left(\prod_{i}^{Objectives} Status_i(t) \right) \qquad (2)$$

If the task attainability is interpreted as the probability of successful completion, then the expected return value from the task can be computed as the product of the assigned value and attainability:

$$E[Task\,Value] = (Task\ Attainability) \times (Task\ Assigned\,Value) \qquad (3)$$

To maintain consistency with Equations (2) and (3), one simply notes that a plan formed from a sequence of abstract tasks is simply an abstract task in its own right. Thus, Equation (2) is used with the minimum attainability assessed over the entire duration of the task, and Equation (3) is used with the Assigned Task Value taken as the sum of the assigned values of the constituent tasks.

5.0 Planning Algorithm

A simple total-order situation-space progression planning algorithm has been implemented with the SOVA adaptations described in previous sections. Given the current state, status of the current and completed tasks, and a list of candidate tasks to perform, the planner executes a tree search for candidate task sequences (plans) that yield a goal state (e.g. end of a week). The plan that yields the greatest expected path value is selected for execution. The selection of candidate tasks to expand during the plan search is governed by the expected return value (product of attainability and path value) of each task in the search queue.

Figure 3 shows a simplified illustration of the operation of the planner. Three data structures are integral to the execution of the search: the Candidate Task List, the Search Tree, and the Search Queue. The Candidate Task List is populated with abstract tasks at start up via a configuration file load. Each candidate task is assigned a value corresponding to its overall priority, and a list of preconditions that can be evaluated based upon the state of the simulated system.

The Search Tree is populated during the execution of the search algorithm. Each Search Tree node holds a pointer to a candidate task operator. Additionally, integrated path value, attainability, and expected path value are maintained for each node to aid in the search process. A valid plan can be formed by tracing a path from any goal node at the bottom of the tree to the root node at the top. The Search Queue is simply a list of pointers to unexpanded fringe nodes on the Search Tree. The SOVA adaptations determine the order in which the nodes on the Search Queue will be explored. A description of the search algorithm is given below:

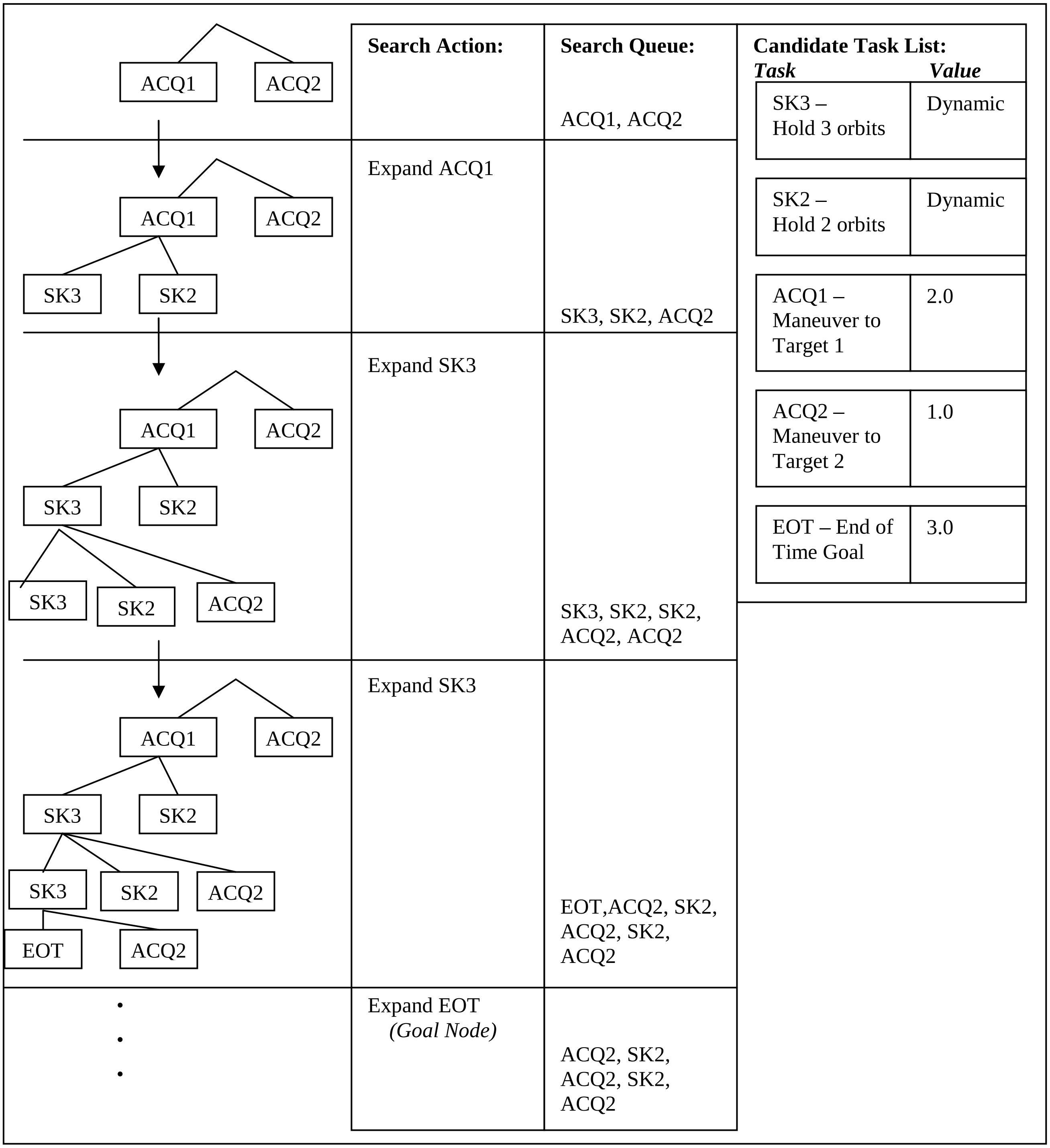
Plan Search Algorithm with SOVA Adaptations:

1. Load Search Tree with a node representing the Task in Work. Also add this node to the Search Queue.

2. Iterate through the Candidate Task List. For each task in the list, add it to the top layer of the Search Tree and to the Search Queue if its preconditions are met by the present state of the simulated state variables.
3. While the Search Queue is not empty, do the following:
    a. Extract the fringe node from the Search Queue with the highest value
    b. Push the operator referenced by the fringe node onto the Plan Stack.
    c. Execute a simulation, starting from the current state based on the end state of the parent node and evaluate the applicable Strategic Objective margins and Attainability as discussed in Sections 4 and 5. (Abort the simulation if the Attainability Assessment falls below an assigned threshold value.)
    d. Set the Attainability and Expected Value data elements of the associated fringe node based upon the results of the simulation.
    e. Check to determine whether or not a Goal State has been achieved:
        i. If a Goal State has been achieved, add the fringe node to the Goal Node queue
        ii. If a Goal State has NOT been achieved, but the Attainability of the fringe node is above the assigned threshold value, then iterate through the Candidate Task List, adding children nodes to the fringe node and to the Search Queue if their preconditions are met by the end state of the simulated state variables.
4. The Goal Node with the highest Expected Value is selected. A plan is formed by iterating up the Search Tree from the Best Goal Node, placing the referenced task operators onto the Plan Stack.

For the FUSE off-line planning problem, the algorithm above is used with two modifications. Once a single goal node is found the planner will start over and search for a new path. To ensure that the planner does not simply find a repeat solution, a small random value is added to the value of each task as it is added to the search queue. Practically, this will result in multiple unique plans in a relatively short period of time.

Since the SOVA heuristic utilized comprises a measure of the value of the entire path, the proposed planning algorithm is in fact a variant of the A* search. As such, this algorithm inherits its completeness, and optimality characteristics from the A* algorithm. On the downside, it also shares its complexity characteristics with the A* search, thus the number of nodes within the goal contour search space (i.e. time and memory required for a complete search) is expected to be exponential in the length of the solution [5]. Prospects for sub-exponential search space growth are hinged upon the merits of the heuristic and the efficiencies gained with attainability based pruning.



Search Action:	Search Queue:
	ACQ1, ACQ2
Expand ACQ1	SK3, SK2, ACQ2
Expand SK3	SK3, SK2, SK2, ACQ2, ACQ2
Expand SK3	EOT,ACQ2, SK2, ACQ2, SK2, ACQ2
Expand EOT *(Goal Node)*	ACQ2, SK2, ACQ2, SK2, ACQ2

Candidate Task List:

Task	***Value***
SK3 – Hold 3 orbits	Dynamic
SK2 – Hold 2 orbits	Dynamic
ACQ1 – Maneuver to Target 1	2.0
ACQ2 – Maneuver to Target 2	1.0
EOT – End of Time Goal	3.0

Figure 3. Illustration of Partial Tree Search Conducted by Planner

6.0 Configuring SOVA

The SOVA planner was applied to the FUSE target scheduling problem domain by establishing a hierarchy of tasks which accomplish the spacecraft maneuvering and station-keeping, selecting a small set of suitable strategic objectives which ensure safe spacecraft operation during the conduct of the tasks, and integrating the planning software with a 6-DOF dynamic model of the spacecraft.

6.1 Task Definitions

In applying SOVA to the FUSE problem, target acquisition and target station-keeping or hold maneuvers were hierarchically decomposed in such a manner that enabled them to be scripted for the planning software. Functionally, there are four major tasks utilized by the SOVA planner to schedule a week of targets:

1) AcquireTarget<Target Name>
2) HoldFor<Orbit Count>
3) GoalNode
4) EmptyTask

In the SOVA planning script, seven AcquireTarget tasks are defined for each science target provided in a prioritized list corresponding to seven bank angles (0°, ±8°, ±16°, ±24°) applied at the target right ascension and declination. Within each AcquireTarget task are the sub-tasks required to slew the spacecraft to the target and subsequently settle out any residual oscillations (See Figure 1). The value assigned to the AcquireTarget task is based upon a priority value set by the short term planning tool used by the FUSE team. There are five preconditions associated with an AcquireTarget which are used by the planner to screen the candidate task list when deciding the next task to add on a decision tree. The SOVA planner may select an AcquireTarget task for a specific science target if:

1) The spacecraft is not already holding on the target.
2) The time needed for science on the target is less than science time spent on the target already. (This allows for split observations without over scheduling)
3) The current simulation time is ahead of the end date/time set by the user
4) The current simulation time is inside a window of time where the target is stable. These so called visibility windows are computed by a planning aid application already utilized by the FUSE mission planners and allows the number of possible targets at each point in the plan to be greatly reduced, thus speeding up the program.
5) The spacecraft has been holding at the current target for at least the minimum amount of time. (Prevents the spacecraft from bouncing from one target to the next without holding to collect science)

In the SOVA planning script for FUSE, there are HoldFor tasks defined for durations of one-half, one, two and three orbits. The HoldFor tasks are placed by the planner after target acquisition tasks to accumulate science time on the targets. The value of the task is based on the length of time the hold is for (longer = higher value). There are three basic preconditions used by the planner to screen whether or not to add the task to a plan:

1) The HoldFor task cannot be the first task scheduled (a slew must take place first from the initial conditions)
2) The current simulation time must be ahead of the end time set by the user.
3) The time left in the current visibility window must greater than the duration of the hold task. (prevents trying to hold during unstable periods)

A fourth precondition is applied to the HoldFor tasks of duration less than three orbits. For these station-keeping tasks, the spacecraft must already have accrued 3-orbits on target before tacking on a lesser duration HoldFor task onto the plan. This prevents the spacecraft from bouncing between targets with little science time. The three orbit minimum is generally the minimum time spent on a target and was determined by observations made by the FUSE planning team.

The GoalNode task is a dummy task that permits the SOVA planner to close out a plan. It possesses a single precondition that the current simulation time must be past the end time specified by the user (typically 7 days). It sets an internal flag to inform the planner that a goal node has been found and to output the plan.

The EmptyTask construct was established to permit skeleton plans to be assembled by the SOVA planner for situations where it is impossible to schedule a full week of targets without exceeding one or more constraints on the strategic objectives. The EmptyTask causes the model to jump forward in time by a prescribed duration and reinitialize the attitude and momentum at the advanced point in the orbit. The EmptyTask possesses a single precondition which allows it to be incorporated into a plan if an excessive period of real time has elapsed without an advancement of the depth of the decision tree. This prevents the planning software from getting stuck, makes for more efficient use of planning computation resources, and generally increases the likelihood that the planner will produce a weeklong schedule even if it has gaps. If a promising plan has EmptyTask elements, mission planners can manually insert suitable filler targets to eliminate the gaps.

The various AcquireTarget, HoldFor, EmptyTask and GoalNode tasks are assembled by SOVA planner to build a valid target sequence to maximize expected science return over a specified duration.

6.2 Strategic Objectives

Three qualitative strategic objectives were devised for the FUSE problem domain to ensure that the spacecraft will remain within a safe operating state while maneuvering between and pointing to the various candidate targets. Each strategic objective discussed below utilizes a single-input/single-output fuzzy inference system to assess margin.

1. **Maintain Momentum Margin**

   This objective is monitored and enforced during all plan tasks. It is established as a rule to prevent saturation of the skew reaction wheel. A fuzzy inference system is used to establish a quantitative relationship between the skew wheel angular momentum and the status of the objective. The fuzzy inference system itself contains 5 fuzzy rules which map the wheel momentum into a fuzzy status value for the objective:

   if *fzWheelMomentum* **is** *Into-Neg-Saturation* **=>** *fzStrategic-Margin* **is** *Negative*
   if *fzWheelMomentum* **is** *Near-Neg-Saturation* **=>** *fzStrategic-Margin* **is** *None*
   if *fzWheelMomentum* **is** *In-Normal-Range* **=>** *fzStrategic-Margin* **is** *Positive*
   if *fzWheelMomentum* **is** *Near-Pos-Saturation* **=>** *fzStrategic-Margin* **is** *None*
   if *fzWheelMomentum* **is** *Into-Pos-Saturation* **=>** *fzStrategic-Margin* **is** *Negative*

2. **Maintain Control Authority during Target Station-Keeping Holds**

   This objective is monitored and enforced while executing Empty and HoldFor tasks. A fuzzy inference system is used to establish a relationship between pointing error and the status of this objective. Generally speaking, the inference system is set up to flag a normal state as long as the pointing error is less than 5-degrees. Even for the best targets, there will be periods during an orbit when positive control authority will be lost resulting in the spacecraft drifting off the target. So long as the excursions are kept within 5-degrees, enough satisfactory science time on the target can be accumulated.

3. **Maintain Control Authority during Slew Maneuvers**

   This objective is monitored and enforced while executing AcquireTarget tasks. Similar to the control authority objective during station-keeping, a fuzzy inference system establishes a relationship between pointing error along the slew maneuver and the status of this objective. It is recognized that the spacecraft may experience brief periods of negative torque authority over the course of a slew maneuver. The objective is set up to flag a normal state as long as the attitude of the spacecraft is within 15-degrees of the commanded slew profile.

6.3 FUSE Model

A 6DOF rigid body model of the FUSE spacecraft was developed with the interfaces required by the SOVA planner to propagate the effects of attempting a target acquisition or station keeping maneuver in the presence of environmental disturbance torques.

A NORAD two-line-element set and initial quaternion are used to initialize the satellite's position, velocity and attitude at a user specified epoch. A 4th order Runge-Kutta integrator is used to integrate the equations of motion.

The simulation includes models of gravity gradient and residual magnetic dipole disturbance torques. The IGRF magnetic field model[6] is used to compute applied torque resulting from the spacecraft dipole. Torques from solar pressure and atmospheric drag are neglected. Applied torque from the actuators is modeled using idealized reaction wheel and magnetic torque bar models including momentum (20 Nms) and dipole (138 A-m^2) saturation limits.

Idealized spacecraft axis resolved inertial attitude, inertial rate, and magnetic field solutions are used to drive the so-called spacecraft medium control law included in the FUSE model. This control law consists of an inner velocity loop control law with feed-foward of gravity gradient and reaction moments due to the skew wheel. The spacecraft least-squares torque distribution algorithm is included in the model to allocate the commanded torques from the control law to the torque bars and skew wheel. The slew generation algorithm used by the spacecraft to determine the ramp-coast-ramp profile of a target acquisition maneuver is also included in the spacecraft model.

The reduced fidelity model allows the simulations to execute as swiftly as possible with some sacrifice in accuracy. The primary differences observed between the simulation model utilized by the SOVA planner and the Orbital Sciences Corporation's Hybrid Dynamic Simulator (HDS) have stemmed from small differences in attitude estimate. During verification runs for maneuvers taking place where the magnetic field vector is nearly orthogonal to the skew wheel axis, differences in the attitude error have resulted in fairly substantial divergences in momentum profiles. It was determined that the sensor noise model, incorporated in the HDS, was one of the primary sources of this discrepancy. A small amount of sensor noise results in small errors in the attitude estimate or in the real attitude which can lead to control torque requests that are difficult to realize when the magnetic field is nearly orthogonal to the skew wheel. This can result in large, unpredictable jumps in wheel momentum. The solution was improved management of the on-board momentum unloading algorithm. This algorithm applied unloading torque to the wheel when the magnetic filed vector was greater than 80 degrees away from the skew axis. The need to apply magnetic torque simultaneously for control caused many of these apparent opportunities for unloading to be ineffective. Once unloading events were actively scheduled to occur only at the best opportunities, the net efficiency of the unloading algorithm increased significantly and we obtained much better correlation between the simulators and the spacecraft performance.

7.0 FUSE Integration and Results

Planning of science targets for the FUSE spacecraft is a time intensive effort, taking approximately two working days to develop a schedule for a week's worth of targets. A FUSE mission planner begins a scheduling effort with a prioritized list of targets for a specific week. The first-pass planning tool estimates the pointing performance of the spacecraft at each target over the week. For each target, the tool calculates momentum buildup rate as a function of time over the week, and generates target "visibility windows" for times when the magnetic field vector allows for stable pointing. Once this is accomplished, the planner uses this data to manually assemble a schedule by placing high priority targets in time slots that are stable first and organizing other targets around them so that the momentum buildup in the skew wheel is managed over time by shifting between positive and negative momentum targets. The planner uses a simulation tool during the plan assembly to iterate on the momentum timeline and the station-keeping error timeline for each target in the proposed schedule.

To augment the manual scheduling process, two software tools were developed for the FUSE mission planning team for utilizing the SOVA algorithm. The first application, TaskGen, takes target data which includes visibility windows for different roll angles and outputs planning configuration scripts used by the SOVA planning application. The SOVA planning tool takes inputs of the strategic objectives, tasks, and initial conditions of the spacecraft from the user and generates possible plans. The SOVA planning application can typically produce approximately 25 diverse candidate plans from a mission planning set of 80 prioritized targets in less than 12 hours on a capable desktop computer. The first candidate plan is typically found within 30 – 60 minutes. It is envisioned that the tool could be used at the beginning of a planning cycle by running it in batch mode overnight to generate a set of candidate plans. The amount of time a particular mission planning set takes to produce plans depends on the number and visibility of the targets provided.

Mission Planning Set 615 for the week beginning March 7, 2007 is used as an example. In this set, there were 74 targets of which there were 9 high valued science targets, 40 low value science targets, 23 filler targets and 2 calibration targets. Using the SOVA Planner, 31 candidate plans of 7 days in length were produced in 15 hours of runtime on a Dell Latitude D810 laptop with a 1.73GHz Pentium-M processor and 1GB RAM. Of these, there were fours target schedules in which over 45% of the plan-duration was spent on the high valued science targets. The average duty cycle on high valued science targets was 36%, with a minimum of 23% and a maximum of 49%.

Figure 4 shows the instruction set for the target sequence with the highest ratio of science time. The high valued science targets are those with D/E/F/G in the beginning of the name. This schedule has an EmptyTask at 436540.0 seconds and resumed six orbits later at 472541.0 seconds. The break could be easily filled by the planner with a filler target. To make this a working schedule, the EmptyTask would have to be filled with a filler target to bridge the gap and still maintain momentum margin. The plan incorporates 6 of the 9 high valued science targets including a split observation of G938-10-02.

```
(AddToPlan FIRSTSLEWOFFSET )                  // StartTime = 0.00
(AddToPlan ACQUIRE_TARGET_U104-14-06_ANGLE:24 ) // StartTime = 901.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 7202.00
(AddToPlan ACQUIRE_TARGET_G921-09-02_ANGLE:16 ) // StartTime = 25203.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 31504.00
(AddToPlan ACQUIRE_TARGET_S605-27-03_ANGLE:-24 )// StartTime = 49505.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 55806.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 73807.00
(AddToPlan ACQUIRE_TARGET_E989-04-01_ANGLE:24 ) // StartTime = 91808.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 98109.00
(AddToPlan ACQUIRE_TARGET_S605-53-01_ANGLE:-16 )// StartTime = 116110.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 122411.00
(AddToPlan HOLDFORONEORBIT )                  // StartTime = 140412.00
(AddToPlan HOLDFORHALFORBIT )                         // StartTime = 146413.00
(AddToPlan ACQUIRE_TARGET_S405-65-03_ANGLE:8 )  // StartTime = 149414.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 155715.00
(AddToPlan ACQUIRE_TARGET_G923-01-07_ANGLE:16 ) // StartTime = 173716.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 180017.00
(AddToPlan HOLDFORHALFORBIT )                         // StartTime = 198018.00
(AddToPlan HOLDFORONEORBIT )                  // StartTime = 201019.00
(AddToPlan ACQUIRE_TARGET_U104-11-03_ANGLE:16 ) // StartTime = 207020.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 213321.00
(AddToPlan HOLDFORONEORBIT )                  // StartTime = 231322.00
(AddToPlan HOLDFORTWOORBIT )                  // StartTime = 237323.00
(AddToPlan ACQUIRE_TARGET_G938-10-02_ANGLE:24 ) // StartTime = 249324.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 255625.00
(AddToPlan HOLDFORONEORBIT )                  // StartTime = 273626.00
(AddToPlan ACQUIRE_TARGET_U104-13-01_ANGLE:8 )  // StartTime = 279627.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 285928.00
(AddToPlan ACQUIRE_TARGET_G938-10-02_ANGLE:24 ) // StartTime = 303929.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 310230.00
(AddToPlan HOLDFORTWOORBIT )                  // StartTime = 328231.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 340232.00
(AddToPlan HOLDFORHALFORBIT )                         // StartTime = 358233.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 361234.00
(AddToPlan HOLDFORHALFORBIT )                         // StartTime = 379235.00
(AddToPlan HOLDFORTWOORBIT )                  // StartTime = 382236.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 394237.00
(AddToPlan ACQUIRE_TARGET_E848-12-03_ANGLE:24 ) // StartTime = 412238.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 418539.00
(AddToPlan EMPTYTASK )                                // StartTime = 436540.00
(AddToPlan ACQUIRE_TARGET_G921-07-02_ANGLE:0 )  // StartTime = 472541.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 478842.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 496843.00
(AddToPlan HOLDFORHALFORBIT )                         // StartTime = 514844.00
(AddToPlan HOLDFORTWOORBIT )                  // StartTime = 517845.00
(AddToPlan ACQUIRE_TARGET_U103-84-01_ANGLE:24 ) // StartTime = 529846.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 536147.00
(AddToPlan HOLDFORONEORBIT )                  // StartTime = 554148.00
(AddToPlan ACQUIRE_TARGET_M105-28-06_ANGLE:24 ) // StartTime = 560149.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 566450.00
(AddToPlan HOLDFORONEORBIT )                  // StartTime = 584451.00
(AddToPlan ACQUIRE_TARGET_U103-71-03_ANGLE:16 ) // StartTime = 590452.00
(AddToPlan HOLDFORTHREEORBIT )                        // StartTime = 596753.00
(AddToPlan GOALNODE )                         // StartTime = 614754.00
```

Figure 4. Target schedule output by SOVA Planning Tool for MPS 615

Figure 5 shows the momentum timeline for the plan with the highest science ratio of 49%. It is maintained well within the acceptable ±14 Nms bounds set in the strategic objective.

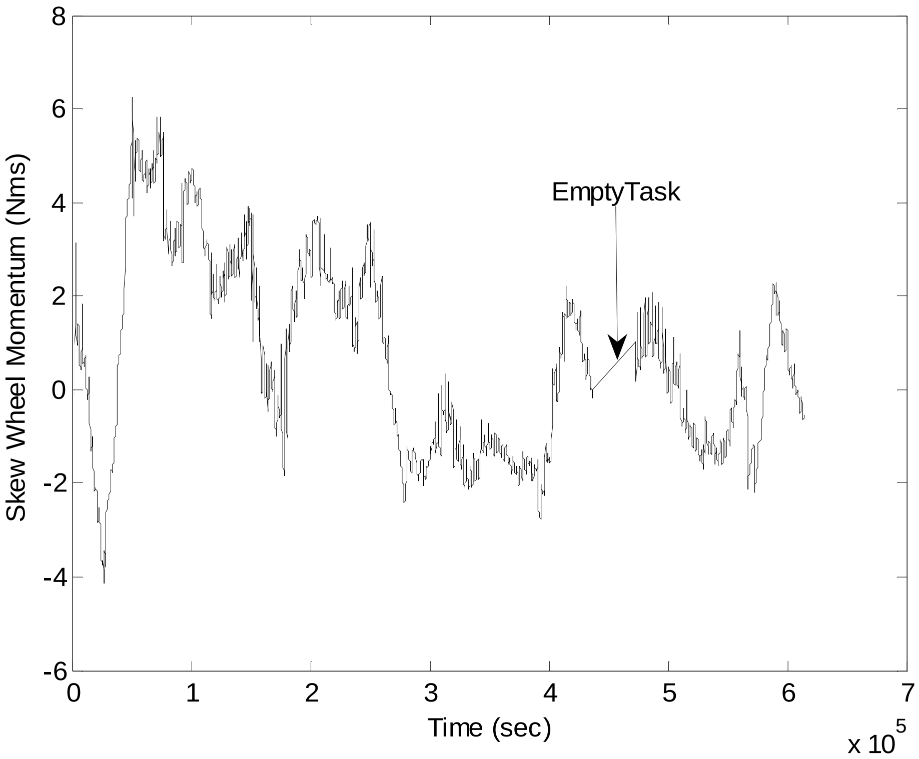


Figure 5. Plot of momentum timeline for sample plan output for MPS 615

The actual plan for MPS 615 produced by the FUSE mission planning team and delivered to the satellite control center contained 66% science efficiency as compared to the 49% maximum found using SOVA. Besides the finding a target schedule, building a mission planning set includes selecting offset fields, performing guide star selections, satisfying data volume constraints and fulfilling all other planning constraints.

Another mission planning set used to test the SOVA algorithm was MPS 605, corresponding to a week beginning on December 27, 2006. This MPS has 50 targets of which there were 8 high priority science targets. Running the SOVA planner for 20 hours on a Dell Inspiron 6000 with a 1.6 GHz Pentium-M processor and 1GB of RAM, 16 target schedules were produced. The science efficiencies for these plans range from 9.5% to 26.8% with an average of 20.6%. Most plans had 3 high priority science targets with several having only 2. The mission plan delivered by the FUSE mission planning team to the satellite control center had ~70% science efficiency and incorporated 7 of the 8 high priority science targets.

A comparison of MPS 605 and 615 shows that the input targets and their respective visibilities have a direct affect on the output plan science efficiencies and the time it takes to produce a candidate schedule. The MPS 615 run averaged ~2 schedules per hour as compared to the MPS 605 run’s 0.8 schedules per hour. MPS 605 has average visibility for the target set where MPS 615 has good visibility for the targets. Good visibility makes the SOVA planner faster since there is a larger candidate set of targets at each step making it less likely to get stuck near a momentum limit. In general, the worse the visibility of a target set, the lower the science efficiency of the output plans and longer it takes to produce them. Figure 6 shows a plot of the science efficiencies for both MPS 605 and 615. It is interesting to note the wide spread of the efficiencies for plans found for the MPS 615 target set. The good visibility allowed many different possible paths to be tried by the planner. However, the average visibility for MPS 605 resulted in a much tighter spread in efficiency values since there were fewer choices which caused the planner to not be able to follow as many different paths which would have resulted in finding more desirable schedules.

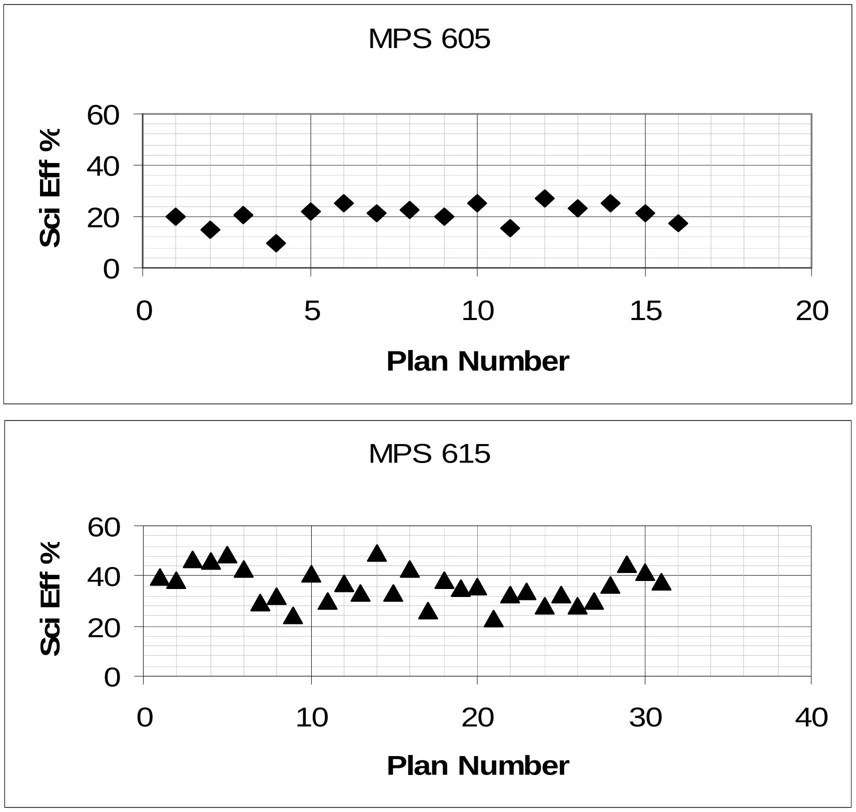


Figure 6. Plot of the Science Efficiency for each plan output by the SOVA planner for MPS 605 and 615

The difficulty of a planning set presents challenges to both the human planner and the SOVA planning tool. Over time, the human planners have come up with tools to compensate for difficult planning periods. One method has been to add filler targets that have long periods of stable pointing with a favorable momentum buildup. This allows for increasing science efficiency since momentum buildup can be controlled by adding filler targets if no science target suits. This technique does not quite translate to the SOVA mechanism. SOVA can only use targets that are generated apriori and does not have the intelligence to add targets. There are a number of filler targets included in the set but an eye must be kept to total number of targets since it is impractical for computation reasons to provide the planner with a high density of filler targets to optimize momentum management for a given schedule.

The inability of the SOVA planner to exceed the ability of the human planner appears to be a limitation. While the SOVA planner produces target sequences that are physically realizable, it is unable to find the optimum sequence to maximize science time in a reasonable amount of time. This is mostly due to the fact that the SOVA algorithm is minimally intelligent and uses a brute force style of search to find the optimum. To find the optimum is impracticable due to the large number of targets and length of time.

A human planner would start by placing the high priority targets first in sections of time that looked suitable and filling in around with low priority and filler targets; whereas, the SOVA planner starts trying to place the highest priority target at each step until it works. This is not optimal and will cause the planner to be in a position where it cannot place high priority science targets in the timeline when they become visible because it has followed a path that does not have the momentum margin for the target at that time.

8.0 Conclusion and other applications

A software planning tool has been developed for the FUSE mission planners to assist in the creation of desirable target maneuver sequences attainable within the control authority of the disabled spacecraft. The SOVA algorithm provides a means of automatically assembling a diverse set of available target scheduling options for consideration by the mission planners. However, the quality of the target sequences produced by the SOVA algorithm does not seem to be able to come near to the human planner.

The SOVA planner's current level of average science efficiency is roughly on par with that demonstrated by the human planners during the initial phase of one wheel operations. During the development of the SOVA planning tool, the human planners have developed innovative techniques and tools that allow them to achieve the much higher level of science efficiency. Likewise, the time to produce a full mission plan (including targets sequence, selecting offset fields, performing guide star selections, satisfying data volume constraints, etc..) has gone from 5 working days to produce a schedule of 7-days length to 5 working days for an experienced planner to produce a 14 day schedule.

The SOVA algorithm does have potential application where manpower and not science quality is of primary concern. The automatic generation of a large number of feasible plans could help to reduce the manpower needed for mission planning when only a physically realizable sequence of targets is needed. Future work on this technique would be to generate a large number (several hundred) of plans and see if there are any outliers with science efficiencies close to the human planner's capability.

Beyond FUSE mission planning, the SOVA algorithm can be used for some problem domains where complex dynamic problems require a solution of simulating various different combinations of tasks with little or no human involvement.

9.0 References

[1] Moos, W.H. et al, “Overview of the Far Ultraviolet Spectroscopic Explorer Mission”, Astrophysical Journal, 538, L1-L6, 2000.

[2] Sahnow, D.J. et al, “Operations with the new FUSE Observatory: three axis control with one reaction wheel”, Proc. SPIE, 6266, 2006.

[3] Aberman, A.F. et al, “FUSE Planning and Scheduling Under One Wheel Attitude Control”, Proceedings of IWPSS-05, 5th International Workshop on Planning and Scheduling for Space, STScI. Baltimore Maryland. 2006.

[4] Wang, Li-Xin, “A Course in Fuzzy Systems and Control”, Prentice Hall, 1997.

[5] S. Russell, P. Norvig, “Artificial Intelligence – A Modern Approach”, Prentice Hall, 1995.

[6] 10th Generation International Geomagnetic Reference Field, http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html.