

DRAFT

PRE-PRINT 2003 AIAA GUIDANCE, NAVIGATION & CONTROL
CONFERENCE, AUGUST 11-14, 2003, AUSTIN, TEXAS

On-board Autonomous Attitude Maneuver Planning for Planetary Spacecraft using Genetic Algorithms

Richard P. Kornfeld
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., Pasadena, CA, 91109
richard.p.kornfeld@jpl.nasa.gov
818-354-3103

Abstract

A key enabling technology that leads to greater spacecraft autonomy is the capability to autonomously and optimally slew the spacecraft from and to different attitudes while operating under a number of celestial and dynamic constraints. The task of finding an attitude trajectory that meets all the constraints is a formidable one, in particular for orbiting or fly-by spacecraft where the constraints and initial and final conditions are of time-varying nature. This paper presents an approach for attitude path planning that makes full use of a priori constraint knowledge and is computationally tractable enough to be executed on-board a spacecraft. The approach is based on incorporating the constraints into a cost function and using a Genetic Algorithm to iteratively search for and optimize the solution. This results in a directed random search that explores a large part of the solution space while maintaining the knowledge of good solutions from iteration to iteration. A solution obtained this way may be used 'as is' or as an initial solution to initialize additional deterministic optimization algorithms. A number of example simulations are presented including the case examples of a generic Europa Orbiter spacecraft in cruise as well as in orbit around Europa. The search times are typically on the order of minutes, thus demonstrating the viability of the presented approach. The results are applicable to all future deep space missions where greater spacecraft autonomy is required. In addition, onboard autonomous attitude planning greatly facilitates navigation and science observation planning, benefiting thus all missions to planet Earth as well.

1. Introduction

1.1. The Nature of the Problem

When planning a spacecraft slew maneuver, great care has to be taken in order to protect sensitive science or stellar reference (i.e. star tracker) instruments from direct exposure to the sun and other bright bodies, as well as to meet a variety of other requirements during the spacecraft turn. The types of constraints typically encountered are [2]:

- **geometric constraints:**
angular separation between body vector x and celestial vector y shall never be less than δ (e.g. star tracker, science instrument boresight)

- **timed constraints:**
angular separation between body vector x and celestial vector y shall never be less than δ for a time period greater than T (e.g. power, thermal constraints)
- **dynamic constraints:**
spacecraft turn rates and accelerations shall be smaller than ω_{max} and a_{max} , respectively (e.g. limited star tracking capability or control authority)

To account for geometric and timed constraints, the position of the sun and other bright bodies with respect to the spacecraft have to be included in the attitude maneuver determination. For a spacecraft in inter-planetary cruise, the positions of the sun and relevant bright bodies can normally be assumed as *time-fixed* for the duration of the spacecraft slew maneuver. Consequently, the geometric and timed constraints are *static* in nature. Similarly, the boundary conditions, i.e. the initial and final spacecraft attitudes, are most likely *not a function of time*. On the other hand, for spacecraft in planetary orbit or on a flyby trajectory, the geometric and timed constraints are frequently *time-varying*, since the position of the sun and other bright bodies are constantly changing with respect to the spacecraft fixed inertial coordinate frame, as the spacecraft progresses along its trajectory. In addition, the initial and final attitudes of the spacecraft slew maneuver are most likely *time-dependent*.

Figure 1 shows the different types of constraints drawn on a celestial sphere whose orientation is inertially fixed. The spacecraft is at the center of the sphere and its attitude trajectory (i.e. the trajectory of one of its body vectors x) is shown along with the time-varying and time-fixed constraints.

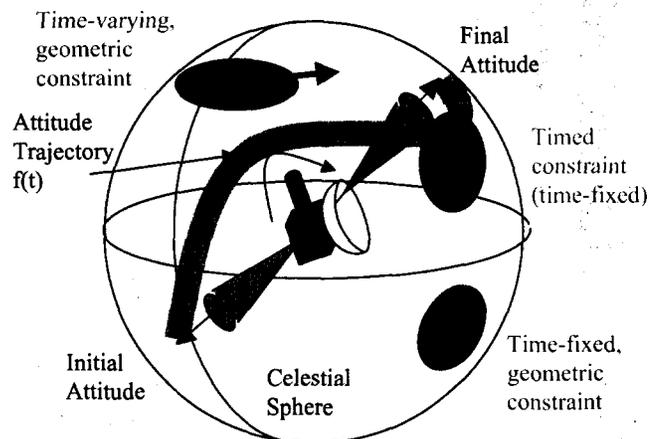


Figure 1: Types of Constraints Drawn on a Celestial Sphere

Determining a constraint free attitude maneuver is closely related to the task of navigating a robot in the presence of moving obstacles and robot dynamics. The latter is usually referred to as *kinodynamic planning*, and has been the object of considerable interest in the recent past. Consequently, concepts developed for autonomous robot

motion planning are directly applicable to the spacecraft attitude maneuver planning problem. However, even the simple problem of navigating a kinematic robot in a known environment with polyhedral obstacles has been proven to be computationally hard [3,4]. Even though *complete*[‡] algorithms are available, these cannot be used for real-time path planning in many real-world applications [3]. When the dynamics of the vehicle are also considered, there is strong evidence that the computational complexity of a complete algorithm will grow exponentially fast in the number of dimensions of the *state space*.

Consequently, determining an attitude trajectory that meets all the time-varying constraints is a formidable task. Because of this difficulty, attitude maneuver determination has traditionally been done on the ground, with only a few spacecraft partially addressing this task in flight.

The Topex Autonomous Maneuver Experiment (TAME) was a first step in implementing an autonomous attitude planner for an earth orbiting spacecraft [3]. The TAME algorithms, however, constituted an extension of algorithms that were originally formulated for the simple case of an interplanetary scenario where celestial constraints are *fixed* in time. The algorithms were based on a simple and undirected trial-and-error search with no guarantee whatsoever that a feasible solution is approached. While the TAME implementation was feasible and reasonable for this particular application, it didn't incorporate the time varying nature of the problem *per se* and therefore lacked the computational efficiency and scalability necessary to attack more complex problems.

The Cassini spacecraft uses a constraint monitor (CM) [4] to autonomously check the commanded attitude trajectory for any constraint violation. Typically, the commanded attitude maneuver is first carefully designed on the ground before it is up-linked to the spacecraft. If a violation is nevertheless detected, a constraint avoidance function is invoked that computes an alternate attitude trajectory. The algorithm hereby commands the spacecraft to 'circumnavigate' the violated constraint. However, due to its short replanning time-horizon, the algorithm has no knowledge of whether any other constraints are being violated along its replanned trajectory. While the CM has been shown to successfully perform in most cases, it does not incorporate the time-varying constraints into the actual re-planning task proactively, and thus lacks computational efficiency and may expend a large amount of fuel before achieving the goal attitude. Similarly, the Remote Agent Experiment on the DS-1 spacecraft [1] used a simple attitude commander with no constraints avoidance planning capability and a Cassini-type Constraint Monitor to autonomously turn the spacecraft, and thus did not address the complexity of time-varying scenarios.

While the search for the optimal solution is an inherent part of maneuver planning, it is not the primary purpose in the current context. As stated above, complete algorithms (i.e. algorithms that are guaranteed to find the optimal solution whenever one exists or notify otherwise) may exist, but these cannot be used for real-time path planning since their

[‡] A complete algorithm is guaranteed to find a solution whenever one exists, and otherwise to notify that there exists none.

execution time typically is of exponential order. Thus, in order to implement an on-board attitude planning capability, the notion of completeness has to be abandoned. Consequently, several heuristic methods have been developed. Some of the more impressive results have been obtained using potential field methods [4,7]. Such methods are attractive, since the heuristic potential field function guiding the search for a path can be easily adapted to the specific problem to be solved. However, the main disadvantage of this class of planners is the presence of local minima in the potential field which may prevent the algorithm from reaching the desired goal attitude. Computing potential fields with no local minima (so-called navigation functions) is practically intractable, except for trivial examples. Other global optimization techniques have been applied, but the curse of dimensionality in non-convex optimization always leads to a computational complexity that prevents the practical implementation of deterministic planners for non-trivial applications.

1.2. Proposed Approach for EO Attitude Planner

For the implementation discussed in this paper, a different approach has been considered. Unlike other approaches, where optimality of the solution is emphasized, the current approach adopts the desire to obtain a *feasible* solution in a *reasonable* amount time as its driving consideration. To this end, the approach is based on the following two strategies:

- Trade-off of optimality with computational complexity
- Use of randomized search techniques

These strategies are explained next.

Optimality versus Complexity Trade-off

Optimality is traded with computational complexity: that is, a number of simplifications are introduced to reduce the problem to be computationally tractable at the expense of potentially obtaining a sub-optimal solution. Simplifying steps may include:

- Omission of non-relevant spacecraft dynamics
- 'Customization' of the problem: e.g. by defining a limited, convex attitude maneuver space and/or a limited set of possible attitude turns¹
- Over-constraining the problem to account for uncertainties

While the maneuver planning problem should be simplified to the greatest extent possible, care has to be taken to maintain the validity of the model and consequently of the solutions found.

Randomized Search Techniques

In the current approach, randomized search techniques are used to search for and, if found, optimize attitude maneuver solution. The notion of *randomness* as a part of the maneuver planning search strategy is hereby introduced. A number of path planners

¹ There is a trade-off between up-front customization of the attitude maneuver planning problem and the level of flexibility maintained.

relying on random search or probabilistic methods have been developed in recent years. They include Rapidly Exploring Random Trees [8] and Probabilistic Roadmap Planners [9]. More recently, a Hybrid Control Architecture [10] has been developed that proved to be very efficient. However, for this study, a random search based path planning method relying on a Genetic Algorithm (GA) was chosen. The GA based approach has a number of beneficial attributes making it uniquely suited for the path planning application:

- A GA allows for a *directed random search* of a large solution space. It utilizes a 'survival of the fittest strategy' to search in the vicinity of good solutions for better ones. At the same time, randomly induced 'mutations' ensure that a large solution space is searched.
- GAs have been shown to perform successfully in many real-world applications
- GAs are easy and fast to implement e.g. by using available COTS toolboxes.
- GAs are highly parallelizable. While the current implementation is executed on a standard CPU in a sequential manner, future implementations are likely to rely on dedicated or application specific highly parallel processors
- GAs are well suited for discrete events such as thruster firings.

GAs have been previously studied for the problem of Flight Path Optimization for Mars Precision Landing [11]. However, for the latter application, the GA was used as an off-line optimization tool and its execution time required usually tens of hours.

To apply the GA based approach in the current context, the problem at hand is first converted from a constraint into an unconstrained optimization problem. This is achieved e.g. by incorporating the attitude maneuver constraints into a cost function that subsequently is minimized using the GA based search algorithm. The GA based approach is outlined in the next section in more detail.

Finally, it is important to note that a feasible solution obtained this way may be used 'as is' or, if further optimization is desired, as an initial solution for additional deterministic or random optimization algorithms.

Section 2 discusses the Genetic Algorithm (GA) based implementation pursued in this study in more detail. Section 3 examines a number of case examples and evaluates certain characteristics of the GA approach. Section 4, finally, summarizes the assessment and provides conclusions.

2. Implementation of a GA Based Autonomous Attitude Maneuver Planner

This section discusses the implementation of the Genetic Algorithm based Autonomous Attitude Maneuver Planner. The underlying assumptions are discussed first. Next, the cost function that is to be minimized is defined. The cost function penalizes any constraint violations and, consequently, turns the constraint into an unconstrained optimization problem. Next, the encoding of the free parameters, which define the

dimensions of the search space is explained. Last, the basic functioning of a Genetic Algorithm is shortly explained.

2.1. Simplifications and Assumptions

The following assumptions and simplifications are being made for this study:

- Instantaneous turn rate changes are assumed. That is, the time to accelerate the spacecraft to its maximum allowable turn rate is much smaller than the spacecraft coasting time for a typical turn maneuver ($t_{acc} \ll t_{maneuver}$). As a consequence, spacecraft dynamics are neglected and only spacecraft kinematics are considered. This simplification eliminates the need to integrate the spacecraft dynamic equations of motions and reduces computationally complexity. Instead, the spacecraft kinematics only are propagated forward. This assumption is valid for most of today's spacecraft that are equipped with a reaction control system (RCS).
- Fixed initial and final attitudes (Θ_i, Θ_f), as well as a fixed initial time t_i are assumed. Only the final time t_f is assumed to be free. However, the concepts developed in this study can easily be extended to include free initial and final attitudes and free initial time.
- The planning problem is constraint to a maneuver that is composed of two slews. This simplification reduces the search space and thus computational complexity. However, the concepts developed in this study can easily be extended to multi-turn maneuver with three, four and more turns.
- A time-parameterized ephemeris model of celestial objects is assumed to be available. That is, spacecraft-to-sun vector and all relevant spacecraft-to-celestial object vectors are assumed to be known as a function of time. In particular, to simplify constraint evaluation, it is assumed that the motion of celestial constraint vectors can be approximated by instantaneous turn rates that are valid for the duration of a typical spacecraft turn. In most cases, the turn rates of celestial constraints are only significant for planetary bodies in close vicinity to the spacecraft, where the celestial constraints changes its direction during the spacecraft slew. For spacecraft in interplanetary cruise the turn rate of celestial constraint vectors is typically negligible.

2.2. Cost Function

The degree that a given candidate attitude maneuver satisfies the different constraints is measured by a cost function. The cost function is composed of four components each contributing to the overall cost of a given candidate solution. The components are a) cost due to the violation of geometric constraints, b) cost due to the violation of timed constraints, c) fuel cost and d) maneuver time cost. These components are explained in subsequent sections.

To evaluate the cost function, define the vector $[\omega, \Delta t]$ to describe a particular spacecraft slew maneuver. The direction and magnitude of ω define the rotation axis and the turn rate, respectively, and the time interval Δt the duration of the spacecraft slew. A

particular candidate attitude maneuver is based on two slews and is thus defined by a pair $[\omega_1, \Delta t_1]$ and $[\omega_2, \Delta t_2]$. The first slew rotates the spacecraft from its given initial attitude Θ_o (at time t_o) to an intermediate attitude Θ_{interm} (at time $t_{\text{interm}} = t_o + \Delta t_1$) and the second slew rotates it to its final attitude Θ_f (at time $t_f = t_{\text{interm}} + \Delta t_2$).

2.2.1. Geometric Constraints

A particular spacecraft slew $[\omega, \Delta t]$ is penalized if it causes the angular separation between a protected boresight and a celestial object to be less than a minimum required separation angle. Thus, the j -th body vector \mathbf{b}_j and the i -th celestial vector \mathbf{c}_i together define a pair $(\mathbf{b}_j, \mathbf{c}_i)$ of constraint vectors whose separation angle $\phi_{ij}(t)$ has to be larger than the required minimum separation angle Φ_{ij_min} throughout the slew. If $\phi_{ij}(t)$ ($0 < t < \Delta t$) is smaller than Φ_{ij_min} , the penalty is calculated based on the smallest angular distance encountered during the slew. In instances where the minimum angular separation angle is either at the beginning ($t = 0$) or at the end ($t = \Delta t$) of the slew, the minimum angular distance is set to the angular distance at the end of the slew. Hence, the minimum angular distance ϕ_{ij}^* between a constraint pair $(\mathbf{b}_j, \mathbf{c}_i)$ for a given slew maneuver $[\omega, \Delta t]$ is defined as

$$\phi_{ij}^* = \phi_{ij}^*(\mathbf{C}_i, \mathbf{v}_i, \mathbf{b}_j, \omega, \Delta t) = \begin{cases} \min \phi_{ij}(t) & \text{if } \exists \\ 0 < t < \Delta t \\ \phi_{ij}(\Delta t) & \text{else} \end{cases} \quad (1)$$

where \mathbf{v}_i denotes any instantaneous rotation rate of the celestial vector \mathbf{c}_i on the celestial sphere around the spacecraft. In this expression, \mathbf{b}_j and \mathbf{c}_i are expressed in an inertially fixed reference frame and correspond to the body and celestial vectors at the beginning of the slew.

For the inter-planetary case, the constraint pair $(\mathbf{b}_j, \mathbf{c}_i)$ is time-fixed (i.e. $\mathbf{v}_i = 0$) and ϕ_{ij}^* can be computed in closed-form solution. This allows for a computationally efficient evaluation. The algorithm to determine the closest approach angle for the time-fixed case involves solving the First Paden-Kahen Subproblem. For the more general, time-varying case (i.e. $\mathbf{v}_i \neq 0$), experienced e.g. in planetary orbits, ϕ_{ij}^* has to be calculated through an iterative search. The algorithm to determine the closest approach angle for this case requires solving a function minimization. Due to its iterative nature, the algorithm requires more execution time than the one for the time fixed case. These algorithms are beyond the scope of this paper.

In order to combine the cost of different geometric constraints (as well as of other types of constraints) into a single total cost, each contribution has first to be normalized (or weighted). For a constraint pair $(\mathbf{b}_j, \mathbf{c}_i)$ with a required minimum angular separation Φ_{ij_min} or larger and an actual minimum angular distance ϕ_{ij}^* , the cost contribution is normalized by a weighting function F_1 given by:

$$F_1 = F_1(\phi_{ij}^*, \Phi_{ij_min}) = \begin{cases} -\log(\phi_{ij}^* / \Phi_{ij_min}) & \text{if } \phi_{ij}^* < \Phi_{ij_min} \\ 0 & \text{else} \end{cases} \quad (2)$$

That is, if the minimum angular distance ϕ_{ij}^* is equal to or larger than Φ_{ij_min} , no cost is incurred. Otherwise, a logarithmic cost is applied. The weighting function F_1 is shown in Figure 7(a). Finally, for a two-slew maneuver, the total cost C_{GC} due to geometric constraints violations is given by the sum of the cost contributions of all applicable constraint pairs (\mathbf{b}_j , \mathbf{c}_i) for both slews, hence

$$C_{GC} = \sum_{\text{slew}=1}^2 \sum_{i=1}^{\text{\#celest. constr. sights}} \sum_{j=1}^{\text{\#bore-sights}} F_1(\phi_{ij}^*, \Phi_{ij_min}) \quad (3)$$

(i,j) are
constraint pair

Note that not all combination of boresight and constraint vectors necessarily constitute a constraint pair.

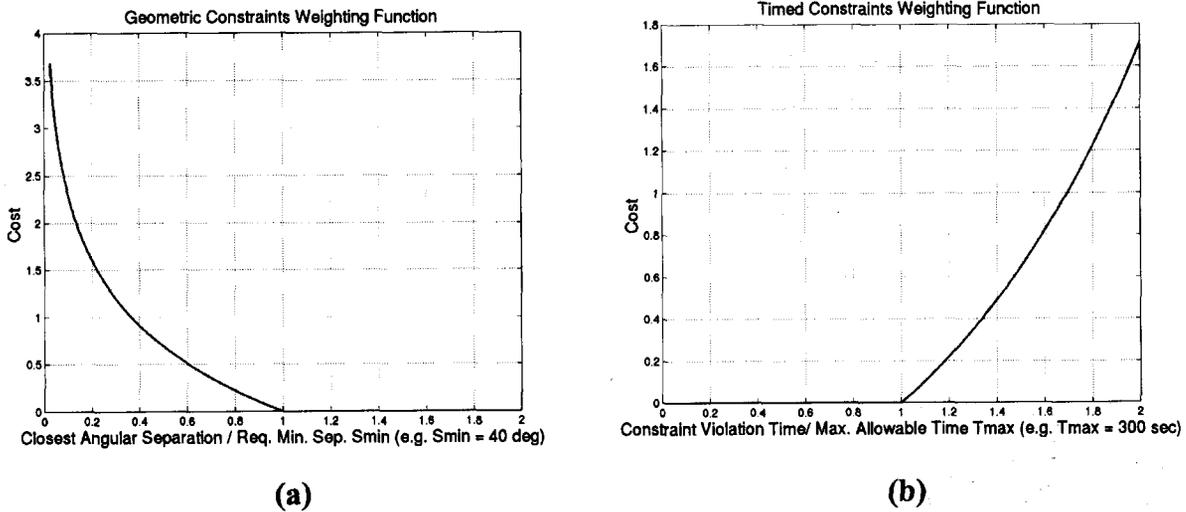


Figure 2: Weighting Functions for (a) Geometric Constraints (b) Timed Constraints

2.2.2. Timed Constraints

Timed constraints require the angular separation $\phi_{ij}(t)$ between body vector \mathbf{b}_j and celestial vector \mathbf{c}_i not to be less than Φ_{ij_min} for a time period greater than T_{ij_max} . In order to enforce this constraint, a candidate maneuver solution incurs a cost if it causes a protected body vector \mathbf{b}_j to stay longer in the 'forbidden cone' of half angle Φ_{ij_min} around the celestial vector \mathbf{c}_i . Slews that cause \mathbf{b}_j to be in the forbidden cone for less than T_{max} or not to cross the cone at all do not incur any cost. Hence, define a function $T_{ij}(\)$ for the time that \mathbf{b}_j spends in the forbidden cone as

$$T_{ij} = T_{ij}(\mathbf{c}_i, \mathbf{v}_i, \mathbf{b}_j, \boldsymbol{\omega}, \Delta t, \Phi_{ij_min}) = \begin{cases} \text{time intervall for which } \phi_{ij}(t) < \Phi_{ij_min} & \text{if } \exists \\ 0 & \text{else} \end{cases} \quad (4)$$

where \mathbf{v}_i denotes any instantaneous rotation rate of the celestial vector \mathbf{c}_i on the celestial sphere around the spacecraft. \mathbf{b}_j and \mathbf{c}_i are expressed in an inertially fixed reference frame and correspond to the body and celestial vectors at the beginning of the slew.

As in the case for geometric constraints, for the interplanetary case where \mathbf{b}_j and \mathbf{c}_i are time-fixed, T_{ij} can be computed in closed-form solution allowing for a efficient evaluation. The algorithm to determine the constraint violation time for the time-fixed case involves solving the Second Paden-Kahen Subproblem and is beyond the scope of this paper. For the more general time-varying case, T_{ij} has to be calculated by iterative search. The algorithm for the time-varying case was not implemented for this research but can easily be adapted from the time-varying geometric constraints.

The weighting function F_2 is chosen as

$$F_2 = F_2(T_{ij}, T_{ij_max}) = \begin{cases} \exp\left[\frac{(T_{ij} - T_{ij_max})}{T_{ij_max}}\right] - 1 & \text{if } T_{ij} > T_{ij_max} \\ 0 & \text{else} \end{cases} \quad (5)$$

and is shown in Figure 7(b). If T_{ij} is larger than T_{ij_max} the cost increases exponentially, while for $T_{ij} < T_{ij_max}$ no cost is incurred. Finally, for a two-slew maneuver, the total cost C_{TC} due to timed constraint violations is given by the sum of the cost contributions of all applicable constraint pairs ($\mathbf{b}_j, \mathbf{c}_i$) for both slews, hence

$$C_{TC} = \sum_{\text{slew}=1}^2 \sum_{\substack{i=1 \\ \text{(i,j) are} \\ \text{constraint pair}}}^{\substack{\# \text{ celest.} \\ \text{constr.}}} \sum_{j=1}^{\substack{\# \text{ bore-} \\ \text{sights}}} F_2(T_{ij}, T_{ij_max}) \quad (6)$$

Note that in order to simplify this implementation, each constraint pair is considered and penalized separately. However, for e.g. a radiator boresight to be protected in the space environment, the total constraint violation time of all applicable celestial constraints together may have to be considered instead. Similarly, the time a boresight vector spends outside any constraints may be considered as a "cool-off time" and may account against the time spent within the constraints.

2.2.3. Fuel, Maximum Turn Rate and Total Maneuver Time

Spacecraft turn rate in- or decreases are related to the amount of time thrusters (with a given thrust) fire. The firing time, in turn, determines the amount of fuel expended. Penalizing spacecraft turn rate sets thus an upper bound on the fuel usage for a particular maneuver.

In addition, penalizing spacecraft turn rate (or fuel) leads to smaller turn rates and thus longer maneuver durations. If, as in the current case, total maneuver time is penalized too, the search for solution is guided by set of competing criteria and a trade-off between maneuver turn rate (fuel) and maneuver time has to be found.

In order to focus the search on maneuver solutions that are primarily free of constraint violation, the importance of fuel and time optimality is somewhat de-emphasized.² This is accomplished by choosing an acceptable per-axis turn rate and turn duration, specified by ω_{\max} and T_{\max} , within which no penalty is incurred. However, a maneuver solution that exceeded ω_{\max} and T_{\max} is penalized as shown below.

Given a slew maneuver ω and a maximum allowable per-axis turn rate ω_{\max} (assumed equal for all axes), the weighting function F_3 for $\omega = [\omega_1, \omega_2, \omega_3]$ is given as

$$F_3 = F_3(\omega, \omega_{\max}) = \sum_{i=1}^3 \begin{cases} \exp[(\omega_i - \omega_{\max}) / \omega_{\max}] - 1 & \text{if } \omega_i > \omega_{\max} \\ 0 & \text{else} \end{cases} \quad (7)$$

The weighting function is similar in form to the one shown in Figure 7(b). For a maneuver solution consisting of $[\omega_1, \Delta t_1]$ and $[\omega_2, \Delta t_2]$ the contributions of both slews have to be added. For simplicity, it is hereby assumed that the spacecraft comes to a rest at the intermediate attitude between the two slew maneuvers. Hence, the total turn rate related cost is given by

$$C_{\text{Fuel}} = \sum_{\text{slew}=1}^2 F_3(\omega_{\text{slew}}, \omega_{\max}) \quad (8)$$

The total maneuver time is

$$T_{\text{total}} = \Delta t_1 + \Delta t_2 \quad (9)$$

and the associated cost and weighting function are defined as

$$C_T = F_4(T_{\text{total}}, T_{\max}) = \begin{cases} \exp[(T_{\text{total}} - T_{\max}) / T_{\max}] - 1 & \text{if } T_{\text{total}} > T_{\max} \\ 0 & \text{else} \end{cases} \quad (10)$$

The weighting function is similar in form to the one shown in Figure 7(b). For this study, the maximum allowable turn rate ω_{\max} , is chosen to be 0.1 deg/sec per axis as specified by EO AACS requirements [13]. This is driven by the requirement to keep the star tracker in lock during a turn. T_{\max} is arbitrarily chosen to be 2700 seconds allowing e.g. for a full 180 deg and subsequent 90 deg turn without incurring any penalty.

² I.e. a solution with a slight geometric constraint violation but a better fuel performance within the allowable range is not rated better than a solution with no geometric constraint violation and a worse, but still allowable, fuel performance.

2.2.4. Total Cost

The total cost of a particular maneuver solution $[\omega_1, \Delta t_1]$ and $[\omega_2, \Delta t_2]$ is thus given as

$$\text{Cost} = C_{GC} + C_{TC} + C_{Fuel} + C_T \quad (11)$$

2.3. Encoding

In order to apply a Genetic Algorithm (GA), a candidate maneuver solution has to be encoded into a *chromosome* representation. For the application described in this study, the chromosomes are encoded as a single-level binary string using a traditional Gray binary encoding. The use of Gray coding has been advocated as a method to overcome the hidden representational bias in conventional binary representation, since for Gray codes, the Hamming distance between adjacent values is constant [14]. As outlined in previous sections, the following parameters govern the search for a feasible maneuver solution and therefore influence the encoding:

- Two-slew turn: $[\omega_1, \Delta t_1]$ and $[\omega_2, \Delta t_2]$
- Fixed initial time t_0
- Free final time $t_f = \Delta t_1 + \Delta t_2 + t_0$
- Fixed initial and final attitude: Θ_o, Θ_f

where t_0 is assumed to be zero throughout this study.

Figure 8(a) shows the corresponding encoding. As shown in the figure, the only free parameters defining the entire maneuver are $\omega_1, \Delta t_1$ and Δt_2 . Starting from a fixed attitude Θ_o , the slew $[\omega_1, \Delta t_1]$ defines an intermediate attitude Θ_{interm} achieved at time t_{interm} . With a given intermediate and final attitude, and the free slew time Δt_2 , the rotation vector ω_2 can be calculated using

$$\mathbf{q}_2 = \mathbf{q}_f \otimes \mathbf{q}_{\text{interm}}^* \quad (12)$$

where $\mathbf{q}_2, \mathbf{q}_f, \mathbf{q}_{\text{interm}}$ denote the quaternion vectors corresponding to the respective attitude rotations, \otimes denotes quaternion multiplication, and $*$ denotes quaternion transpose. The determination of ω_2 from \mathbf{q}_2 is then straightforward. Since ω_2 is obtained indirectly, it might exceed the maximum allowable turn rate ω_{max} per axis. If this is the case, the magnitude of ω_2 has to be adjusted resulting in a modified Δt_2 .

A 6-bit resolution, covering a parameter interval of $[0, 0.1]$ deg/sec for the per-axis turn rate, and a 8-bit resolution for a slew time range of $[0, 1800]$ seconds seems appropriate for this application. This leads to a chromosome length of 34 bit for a particular maneuver solution.

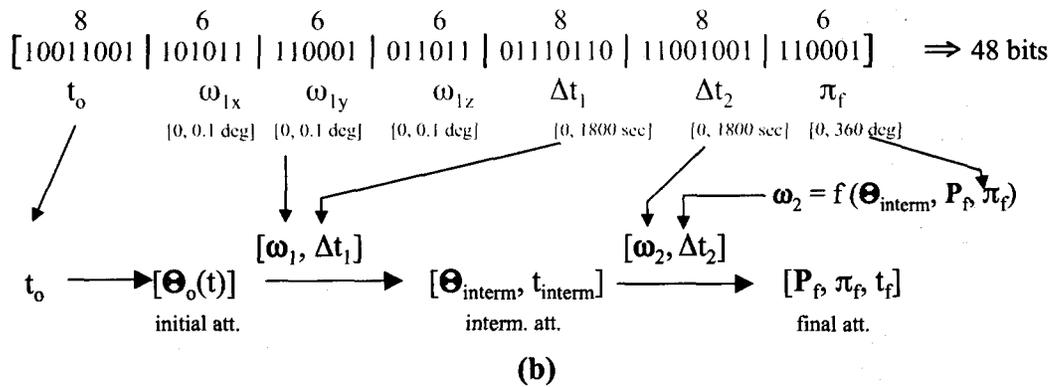
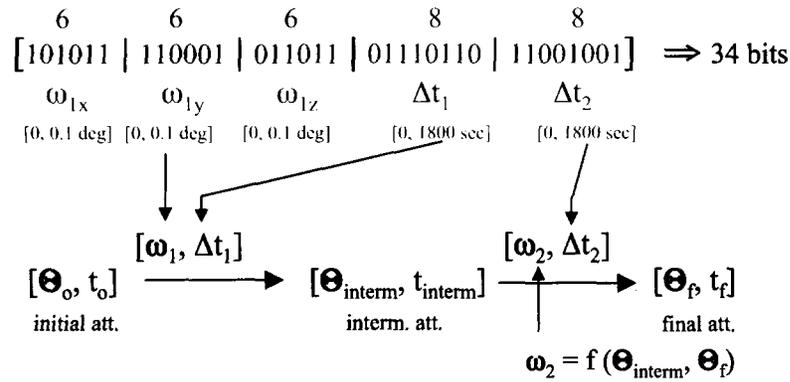


Figure 3: Encoding for (a) case examples (b) more general case

Figure 8(b) shows an example encoding for a more general case, namely:

- Free initial time: t_o
- Initial attitude as a function of t_o : $\Theta_o = F(t_o)$
- Fixed final pointing vector with free roll angle: P_f , free π_f

In this case, additional 8-bit encoding for t_o (in the range of [0, 1800] seconds) and a 6-bit encoding for the free angle π_f (range of [0 360 deg]), yield a total chromosome length of 48 bits. As explained in the next section in greater detail, a larger chromosome length corresponds to a larger search space and thus increases search time. This case is not pursued any further.

2.4. Genetic Algorithm

This section gives a brief outline of the basic Genetic Algorithm (GA). A detailed account is beyond the scope of this study, but can be found in many textbook on GAs, such as [14,15]. A GA consists of the following basic steps:

- 1) [Start] Generate random population of n chromosomes (i.e randomly selected solutions for the problem) of length l ($l = 34$ bit).

- 2) **[Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population using a cost function and ranking criteria.
3. **[Stopping Criteria]** Test whether stopping criteria is achieved? If yes: go to Step 6 and return the best solution in the current population. If no, continue with Step 4.
4. **[New population]** Create a new population by repeating the following steps until the new population is complete.
 - 4.1 **[Selection]** Select two parent chromosomes from the population according to their fitness (the higher the fitness, the higher the chance, P_{select} , to be selected)
 - 4.2 **[Crossover]** With a crossover probability, P_{cross} , crossover the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
 - 4.3 **[Mutation]** With a mutation probability, P_{mutate} , mutate new offspring at each locus (position in chromosome).
 - 4.4 **[Accepting]** Place new offspring in a new population
5. **[Loop]** Go to step 2
6. **[Stop]**

A number of implementation issues have to be considered:

- The best solutions found so far are inserted into the new population, in order to avoid losing them through the crossover and mutation operation (Elitism and Generation Gap). The number of chromosomes allowed to propagate is determined through the *generation gap* parameter.
- Stopping Criteria: In this implementation, the stopping criteria is composed of two conditions: maximum number of generations and maximum acceptable cost. If either the maximum number of generation or a preset maximum cost threshold is reached, the search will be terminated. Imposing a maximal number of generations is necessary in order to exit the search after a given time. The appropriate maximum number of generations has to be determined such that, on the one hand, the search does not end prematurely (i.e while the cost is still decreasing from generation to generation), and, on the other hand, that it does not keep searching once no improvement in cost has been made over large number of generations. The latter conditions corresponds to a case where the algorithm found a local minimum. Rather than trying to escape from the local minimum, it may sometimes be more efficient to terminate the current search and start over using a new set of chromosomes. The second condition, maximum cost threshold, ensures that the search is not prolonged by looking for a degree of optimality that may be not achievable or may not be necessary. That is, in many instances, solutions with a (small, but) non-zero cost may still be acceptable.

3. Case Examples

The GA has been implemented in MATLAB using the GA toolbox available from [14]. The toolbox is based on Matlab .m-files. The optimization was executed on a Sparc Ultra 80 (450 Mhz Sun UltraSparc-II CPU) with 1 GB Memory that was shared with one other user.

This section presents four case examples of the GA based attitude planner. Section 5.1 presents the *time-fixed* case of Europa Orbiter during cruise to Jupiter. Section 5.2 shows a pre-fabricated (and rather pathological) *time-fixed* case to more heavily exercise the GA algorithm. Section 5.3 presents the *time-varying* example of EO in Europa Orbit (EO²). Finally, Section 5.4 presents a pre-fabricated, time-varying case of rather pathological nature to explore the limits of the algorithms for a *time-varying* case. To fully exercise the algorithm and gain more insight into its performance, the EO spacecraft constraint configuration is adapted to the respective case example and thus varies from example to example.³

3.1. Europa Orbiter During Cruise

3.1.1. Setup

The following example discusses the attitude planning problem for EO during cruise phase about midway between Mars and Jupiter. A number of celestial constraints are assumed active during this phase, in particular Mars, Earth, Jupiter and foremost the Sun. The positions of the celestial objects (celestial vectors c_i) were computed with SOAP (Satellite Orbit Analysis Program) using the EO trajectory, current at the time of the study, for the date of June 1, 2004. Since the spacecraft is in interplanetary cruise, the celestial objects have no noticeable motion in the time interval typical for spacecraft turns, and, thus, the problem can be considered of time-fixed nature.

A number of spacecraft protected boresights (body vectors b_j) have been assumed for this example. The body vectors b_j , the celestial vectors c_i , the applicable constraint pairs as well as the constraint type are tabulated in Table 1, with the celestial constraint vector expressed in right ascension (RA) and declination (DEC).

Celestial Constraint Vectors				
Name	Label	Position Vector (J2000)	Angular Radius ⁴	
Sun	c1	RA = 2 , DEC = 0	1 deg	
Jupiter	c2	RA = 162 , DEC = 0	1 deg	
Earth	c3	RA = 338 , DEC = 0	1 deg	
Mars	c4	RA = 44 , DEC = 0	1 deg	
Body Constraint Vectors				

³ At the time this study was conducted, actual science instruments, thermal and other protected boresights have not been fully identified yet. All of the constraint configurations used in the case examples do not correspond to the constraint configuration of the actual Europa Orbiter mission.

⁴ For extended bodies, the angular radius of celestial object is used. If the object is a point source only, a minimum angular radius of 1 deg is assumed.

Name	Label	Position Vector (SC fixed)	Field of View	Constraint Pair with	Constraint Type
SRU	b1	[-1 0 0]	40 deg	c1, c2, c3, c4	Geometric
Science	b2	[1 0 0]	40 deg	c1, c2, c3, c4	Geometric
Radiator 1	b3	[-0.707 0.707 0]	25 deg	c1, c2, c3, c4	Geometric
Radiator 2	b4	[-0.707 0.707 0]	25 deg	c1, c2, c3, c4	Geometric
SRU2	b5	[0 0 -1]	25 deg	c1, c2, c3, c4	Geometric
Science2	b6	[0 0 1]	15 deg	c1, c2, c3, c4	Geometric

Table 1: Celestial and Body Constraints for Case 1

Each body and celestial vector form a constraint pair, as indicated in Table 1. The required separation angle is the sum of the angular radius of the celestial object and the field of view of the body constraint. The initial and final attitude quaternions for this example are given as:

$$\text{initial attitude: } \mathbf{q}_i = [0.5 \ -0.5 \ 0.5 \ 0.5]$$

$$\text{final attitude: } \mathbf{q}_f = [0.419 \ 0 \ -0.908 \ 0]$$

The case example is setup such that no single rotation can lead the spacecraft from its initial position to its final position without violating any constraints. Thus, the search algorithm is required to find a two-slew solution. The constraints in Table 1 and initial and final attitude quaternions are coded in an initialization file that is loaded at the beginning of the optimization.

Table 2 shows the Genetic Algorithm (GA) settings used for this example. The stopping criteria and the maximum number of generations were established based on a number of trial runs. The stopping criteria was further refined based on post-simulation evaluation. The Generation Gap and the Crossover and Mutation probabilities are recommended default values [14].

Parameter	Label	Value
# of Chromosomes	NIND	30
Chromosome Length	L	34 bits
Max. # of Generations	MAXGEN	50
Generation Gap	GGAP	0.9
Crossover Probability	Pcross	0.7
Mutation Probability	Pmut	0.7/L
Stopping Criteria		Cost Function < 8 * 1e-5

Table 2: GA Settings for Case 1

3.1.2. Results

Figure 10 depicts the lowest cost achieved for a given generation over the course of the optimization. In the first generation, i.e. the generation that is randomly selected at the outset of the optimization, the best solution has a cost of approximately 2.6. After one generation, the cost decreased to 1.1. With the cost decreasing in a stepwise fashion, the stopping criteria is reached after 37 iterations (i.e. in the 38th generation). The optimization was executed several times. This particular run required 227 sec and represents a slower than usual convergence for this problem.

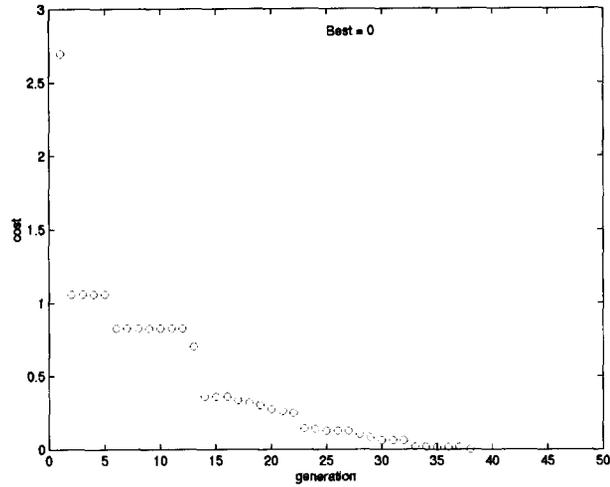


Figure 4: Lowest Cost versus Generation

Figure 11 shows the maneuver solution drawn on a celestial sphere whose orientation is inertially fixed. The spacecraft is at the center of the sphere and the trajectory of the body vector constraint cones is shown. As can be seen, the spacecraft orients itself from its initial to its final attitude using a two-slew maneuver turn without a violation of any geometric constraints. The two slew maneuver solution is given by:

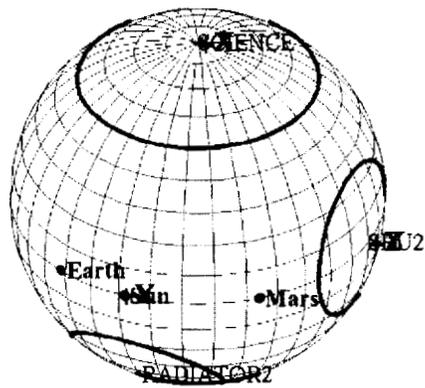
$$\omega_1 = 1e-3 * [-0.9142 \ -0.4156 \ -0.6926] \text{ rad/sec}$$

$$\Delta t_1 = 1349 \text{ sec}$$

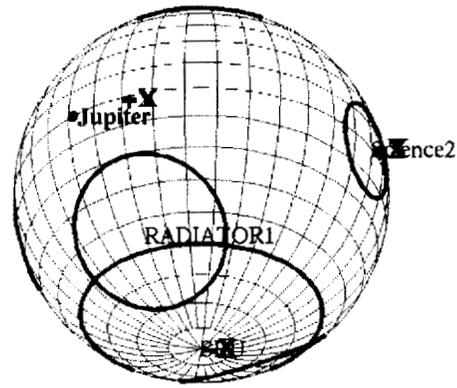
$$\omega_2 = 1e-3 * [-1.681 \ 0.0518 \ 1.2004] \text{ rad/sec}$$

$$\Delta t_2 = 904 \text{ sec}$$

The total turn time is 2253 seconds.

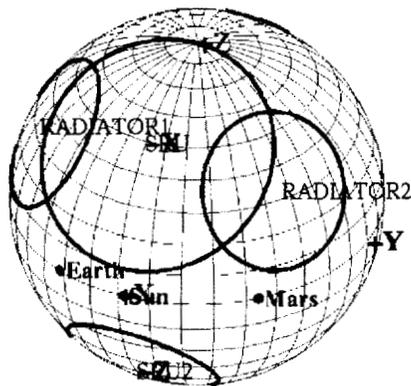


Maneuver Time: 0

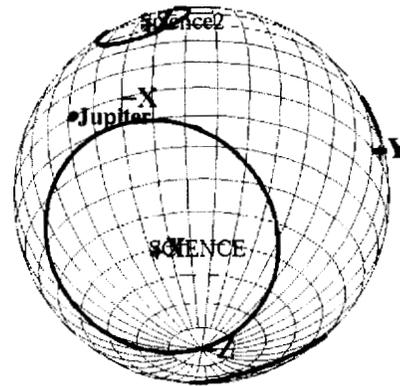


Maneuver Time: 0

(a)



Maneuver Time: 2253



Maneuver Time: 2253

(b)

Figure 5: Case 1 Solution (a) Initial Attitude (b) Final Attitude

3.2. A Pathological, Time-Fixed Case

The purpose of this example is to exercise and demonstrate the GA based maneuver planner on a pathological (i.e. worst), time-fixed case. The following example is based on a pre-fabricated 'worst case' with limited, but known optimal solutions.

3.2.1. Setup

The case example is constructed such that two body constraints vectors (\mathbf{b}_1 and \mathbf{b}_2) are "trapped" inside two "clusters" of celestial constraints ($\mathbf{c}_1 - \mathbf{c}_4$ and $\mathbf{c}_5 - \mathbf{c}_8$). No single rotation leads the spacecraft from its initial position to its commanded final position without violating any constraints. Instead, only a limited range of two-slew maneuvers result in a trajectory free of constraint violations, as shown below. This example does not correspond to case a real spacecraft would encounter. Table 3 show the body and celestial constraints for this example with the celestial constraint vector expressed in right ascension (RA) and declination (DEC). Figure 13 shows the constraints drawn on an unit sphere around the spacecraft.

Celestial Constraint Vectors					
Name	Label	Position Vector (J2000)	Angular Radius*		
C1	c1	RA = 0 , DEC = 30	10 deg		
C2	c2	RA = 30 , DEC = 0	10 deg		
C3	c3	RA = 0 , DEC = -30	10 deg		
C4	c4	RA = -30 , DEC = 0	10 deg		
C5	c5	RA = 40 , DEC = -60	10 deg		
C6	c6	RA = 140 , DEC = -60	10 deg		
C7	c7	RA = -45 , DEC = -45	30 deg		
C8	c8	RA = -135 , DEC = -45	30 deg		
Body Constraint Vectors					
Name	Label	Position Vector (SC fixed)	Field of View	Constraint Pair with	Constraint Type
SRU	b1	[0 0 -1]	10 deg	c1 - c8	Geometric
Science	b2	[1 0 0]	10 deg	c1 - c8	Geometric

Table 3: Celestial and Body Constraints for Case 2

The initial and final attitude quaternions for this example are given as:

$$\text{initial attitude: } \mathbf{q}_i = [0 \ 0 \ 0 \ 1]$$

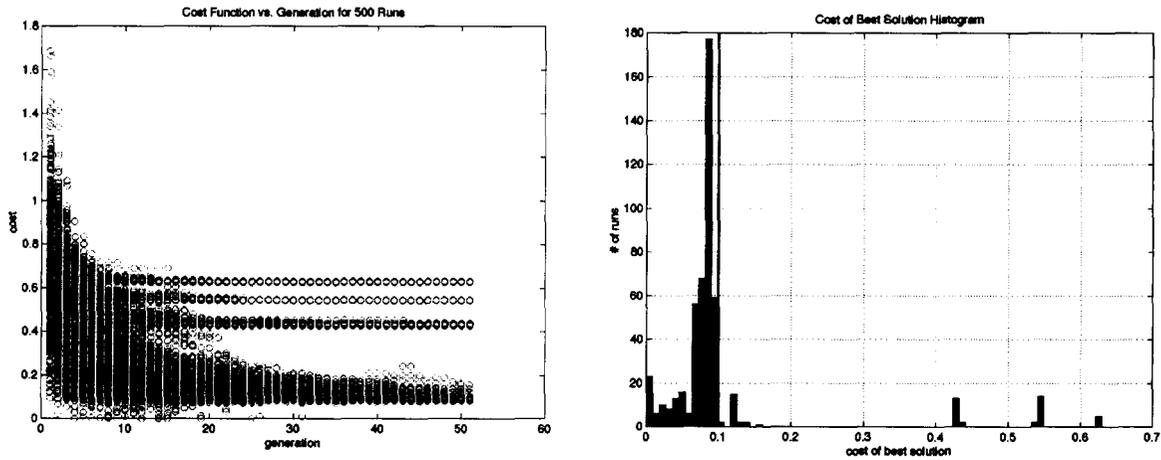
$$\text{final attitude: } \mathbf{q}_f = [0 \ -0.9239 \ -0.3827 \ 0]$$

The constraints in Table 3 and initial and final attitude quaternions are coded in an initialization file that is loaded at the beginning of the optimization.

The optimization was run 500 times to obtain quantitative insights into how the algorithm performed in this case. Each run consisted of a different randomly selected initial population of 70 chromosomes that was optimized over 50 generations. The GA settings used for this test are given in Table 4. The stopping criteria was established based on a number of trial runs.

Parameter	Label	Value
# of Runs		500
# of Chromosomes	NIND	70
Chromosome Length	L	34 bits
Max. # of Generations	MAXGEN	50
Generation Gap	GGAP	0.9
Crossover Probability	Pcross	0.7
Mutation Probability	Pmut	0.7/L
Stopping Criteria		Cost Function < 0.08

Table 4 GA Settings for Case 2



(a) (b)
Figure 6: (a) Lowest Cost versus Generation for 500 Runs (b) Histogram of Lowest Cost

3.2.2. Results

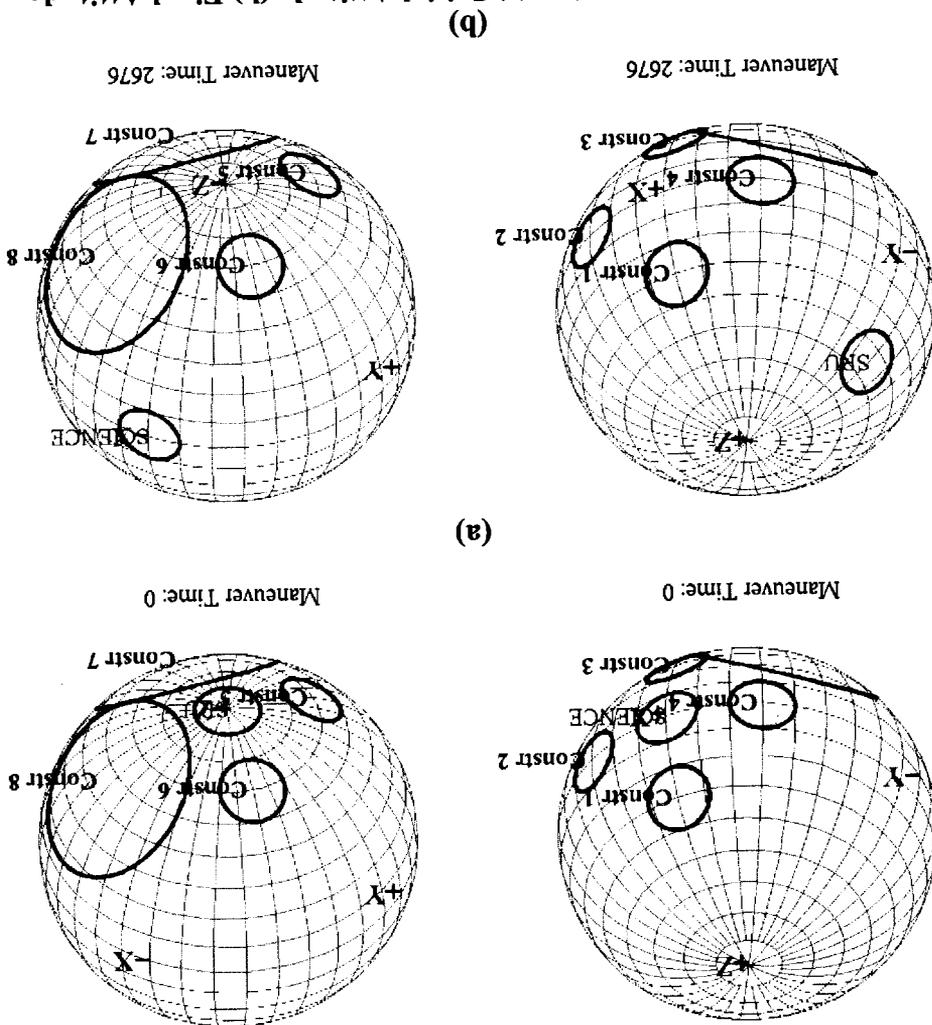
Figure 12(a) shows the lowest cost achieved for a given generation over the course of 50 generations for all 500 runs. While all runs start out at different initial costs in the first generation, as expected from a randomly selected population, they all converge to one of four final costs. Three "local" minima with final costs of approximately 0.42-0.43, 0.53-0.54, and 0.62-0.63, and a final "global" minimum region between 0 and 0.13 are clearly observable in this Figure. Figure 13 shows the corresponding solutions.

Figure 12(b) shows a histogram of the best solutions for all 500 runs with the acceptance threshold at the 0.1 cost level indicated on the chart. As can be seen, the majority of runs generate an acceptable solution with costs less than or equal to 0.1. In fact, 443 runs (i.e. 88.6% of the runs) found acceptable solutions. Besides the total number of runs with acceptable solutions, their distribution among the 500 runs has to be considered as well. This is important, since, in reality, an attitude planner on a spacecraft may initiate a new search using different initial chromosomes, once it hits a local minimum solution with unacceptable performance. Thus, the "fishing" for solutions using multiple tries (i.e.

The total turn time is 2666 seconds.

The two slew maneuver solution for the ideal solution in Figure 15(a) is given by:
 $\omega_1 = 1e-3 * [0.8588 - 0.0831 - 0.0277]$ rad/sec
 $\Delta t_1 = 911$ sec
 $\omega_2 = 1e-3 * [0.0048 - 1.2357 - 1.2326]$ rad/sec
 $\Delta t_2 = 1756$ sec

Figure 7: Case 2 Solution (a) Initial Attitude (b) Final Attitude



(a)

(b)

different genetic material) is an inherent part of the GA based search process. For the current example, in 41 cases it took two runs to obtain an acceptable solution, in 5 cases three runs, and in 2 cases four runs. Thus, the worst case for this pathological example required the maneuver planner to re-initialize and search four times to find an acceptable solution. The average computation time per run on the Sparc Ultra 80 using non-optimized Matlab .m files was 3.8 minutes with a standard deviation of 2.2 minutes. Consequently, for this example, the worst case required computation time on the order of 15 minutes.

3.3. Europa Orbiter in Europa Orbit (EO²)

The following example illustrates the attitude maneuver planning algorithm for EO during its mission in Europa orbit where some of the celestial constraint vectors are *time-varying*. Unlike the previous two, the following example demonstrates thus the behavior of the algorithm for a problem of time-varying nature.

3.3.1. Setup

The positions of the celestial objects (celestial vectors c_i) and their movement over time were computed with SOAP using the EO trajectory, current at the time of the study, for the date of September 22, 2008. A number of celestial constraints are active, in particular Europa, Ganymede, Callisto, Io, Jupiter and the Sun. Only Europa has significant motion during the time interval of a typical spacecraft turn and is thus treated as a time-varying constraint. Its motion can be described by an angular rate vector. Jupiter and the other planets exhibit limited motion and can easily be converted into time-fixed constraints by purposely increasing their angular radius by half of the longitudinal motion encountered during the turn interval (for this case, most of the planetary motion is exhibited in the longitudinal direction). That is, a larger constraint cone encompassing the motion of the actual constraint is assumed during the solution search. This conversion results in a significant reduction of execution time since the time-fixed constraints can be evaluated in closed form as explained in the previous section. A reduced number of protected spacecraft boresights has been assumed for this example. The body and celestial constraints are tabulated in Table 5.

Celestial Constraint Vectors						
Name	Label	J2000 Position Unit Vector (**)	Angular Radius (*)	Angular Rate Vector	Constraint Type	
Sun	c1	RA = 113.8 deg DEC = 0.3 deg	0.05 deg		Time-fixed	
Jupiter	c2	RA = 225.7 deg DEC = -1.7 deg	7.7 deg (actual 6.2 deg)		Converted to Time-fixed	
Europa	c3	RA = 299.4 deg DEC = 12.9 deg	62.5 deg	[0.366, 0.357, 0.567] * 1e-3,	Time-varying	
Callisto	c4	RA = 87.3 deg DEC = 1.9 deg	0.23 deg (actual 0.1 deg)		Converted to Time-fixed	
Ganymede	c5	RA = 25.7 deg DEC = 2.0 deg	0.65 deg (act. 0.37 deg)		Converted to Time-fixed	
Io	c6	RA = 231.4 deg DEC = -1.9 deg	2.3 deg (actual 0.1)		Converted to Time-fixed	
Body Constraint Vectors						
Name	Label	Position Vector (SC)	Field of View	Constraint Pair	Constraint Type	

		fixed)		with	
SRU	b1	[0 0 -1]	20 deg	c1 - c6	Geometric
Science	b2	[-1 0 0]	20 deg	c1 - c6	Geometric
Radiator 1	b3	[0.707 0.707 0]	20 deg	c1 - c6	Geometric

Table 5 Celestial and Body Constraints for Case 3

- * Angular radius is increased to account for limited planetary motion during spacecraft turn. The true angular rate is indicated in parenthesis.
- ** For this case, most of the planetary motion is exhibited in the longitudinal direction. Table 5 indicates thus right ascension in the middle and declination at the start of the turn interval.

The required separation angle is the sum of the angular radius of the celestial object and the field of view of the body constraint. As in the previous cases, the following example is setup such that no single rotation can lead the spacecraft from its initial position to its final position without violating any constraints. Thus, the search algorithm is required to find a two-slew solution. The initial and final attitudes are fixed and given:

$$\text{initial attitude: } \mathbf{q}_i = [0.327 \ -0.627 \ 0.327 \ 0.627]$$

$$\text{final attitude: } \mathbf{q}_f = [0.653 \ -0.271 \ -0.653 \ -0.271]$$

Table 6 shows the Genetic Algorithm (GA) settings used for this example. The stopping criteria and the maximum number of generations were established based on a number of trial runs.

Parameter	Label	Value
# of Chromosomes	NIND	40
Chromosome Length	L	34 bits
Max. # of Generations	MAXGEN	50
Generation Gap	GGAP	0.9
Crossover Probability	Pcross	0.7
Mutation Probability	Pmut	0.7/L
Stopping Criteria		Cost Function < 8 * 1e-5

Table 6: GA Settings for Case 3

3.3.2. Results

Figure 14 depicts the lowest cost achieved for a given generation over the course of the optimization. In the first generation, i.e. the generation that is randomly selected at the outset of the optimization, the best solution has a cost of approximately 0.93. After 16 generations, a zero cost solutions is found. This particular run required 230 sec and represents typical convergence for this problem.

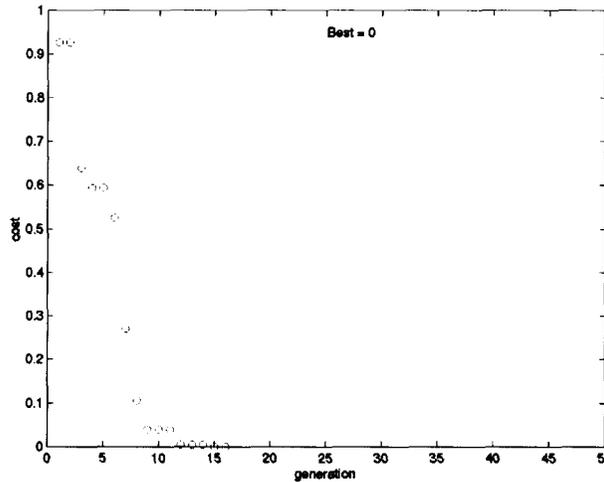


Figure 8: Lowest Cost versus Generation

Figure 15 shows the maneuver solution drawn on a celestial sphere whose orientation is inertially fixed. The spacecraft is at the center of the sphere and the trajectory of the body vector constraint cones is shown. As can be seen, the constraint cone due to Europa is moving from its initial position on the western hemisphere counter-clockwise to the southern hemisphere. At the same time, the spacecraft orients itself from its initial to its final attitude using a two-slew maneuver turn without a violation of any geometric constraints. The two slew maneuver solution is given by:

$$\omega_1 = 1e-3 * [1.0804 \ -0.8034 \ 0.1939] \text{ rad/sec}$$

$$\Delta t_1 = 1715 \text{ sec}$$

$$\omega_2 = 1e-3 * [0.9054 \ 0.3925 \ 1.4396] \text{ rad/sec}$$

$$\Delta t_2 = 943 \text{ sec}$$

The total turn time is 2658 seconds.

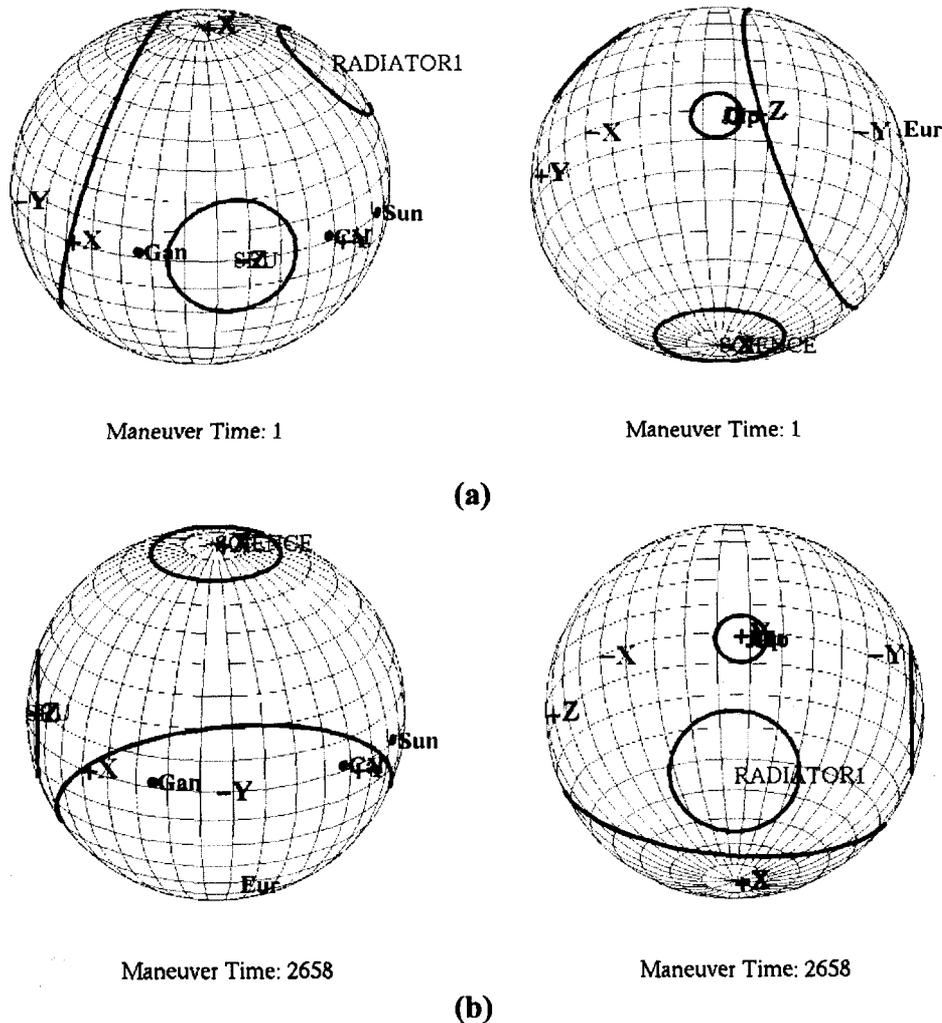


Figure 9: Case 3 Solution (a) Initial Attitude (b) Final Attitude

3.4. A Pathological Time-Varying Case

The following example is a pre-fabricated pathological case for time-varying constraint avoidance and serves to exercise and demonstrate the GA based maneuver planner.

3.4.1. Setup

For this case example, two body constraint vectors (b_2 , b_3) are initially positioned near the "North" and one (b_1) at the "South" pole. The three body constraints vectors have to cross the equator (from North to South and vice versa) where four rotating celestial constraints ($c_1 - c_4$) effectively limit the passage to four moving "gates" (rotation about J2000 z-axis). No single rotation leads the spacecraft from its initial position to its commanded final position without violating any constraints. Similar to the previous pathological case, this example does not correspond to case a real spacecraft would encounter. Table 7 show the body and celestial constraints for this example with the celestial constraint vector expressed in right ascension (RA) and declination (DEC). Figure 17 shows the constraints drawn on an unit sphere around the spacecraft.

Celestial Constraint Vectors					
Name	Label	Position Vector (J2000)	Angular Radius*	Angular Rate Vector	Constraint Type
Constr 1	c1	RA = 0 , DEC = 0	20 deg	$0.1 \cdot \pi / 180 \cdot [0 \ 0 \ 1]$ rad/sec	Time-varying
Constr 2	c2	RA = 90 , DEC = 0	20 deg	$0.1 \cdot \pi / 180 \cdot [0 \ 0 \ 1]$ rad/sec	Time-varying
Constr 3	c3	RA = 180 , DEC = 0	20 deg	$0.1 \cdot \pi / 180 \cdot [0 \ 0 \ 1]$ rad/sec	Time-varying
Constr 4	c4	RA = 270 , DEC = 0	20 deg	$0.1 \cdot \pi / 180 \cdot [0 \ 0 \ 1]$ rad/sec	Time-varying
Body Constraint Vectors					
Name	Label	Position Vector (SC fixed)	Field of View	Constraint Pair with	Constraint Type
SRU	b1	[0 -1 0]	10 deg	c1 - c4	Geometric
Science	b2	[0 1 0]	10 deg	c1 - c4	Geometric
Radiator1	b3	[0.707 0.707 0]	10 deg	c1 - c4	Geometric

Table 7: Celestial and Body Constraints for Example 4

The initial and final attitude quaternions for this example are given as:

$$\text{initial attitude: } \mathbf{q}_i = [0.7071 \ 0 \ 0 \ 0.7071]$$

$$\text{final attitude: } \mathbf{q}_f = [-0.7071 \ 0 \ 0 \ 0.7071]$$

The optimization was run 250 times to obtain quantitative insights into how the algorithm performed in this case. Each run consisted of a different randomly selected initial population of 40 chromosomes that was optimized over 25 generations. The GA settings used for this test are given in Table 8. The number of chromosomes, maximum number of generations and the stopping criteria were established based on a number of trial runs.

Parameter	Label	Value
# of Runs		250
# of Chromosomes	NIND	40
Chromosome Length	L	34 bits
Max. # of Generations	MAXGEN	25
Generation Gap	GGAP	0.9
Crossover Probability	Pcross	0.7
Mutation Probability	Pmut	0.7/L
Stopping Criteria		Cost Function < 0.0008

Table 8 GA Settings for Case Example 4

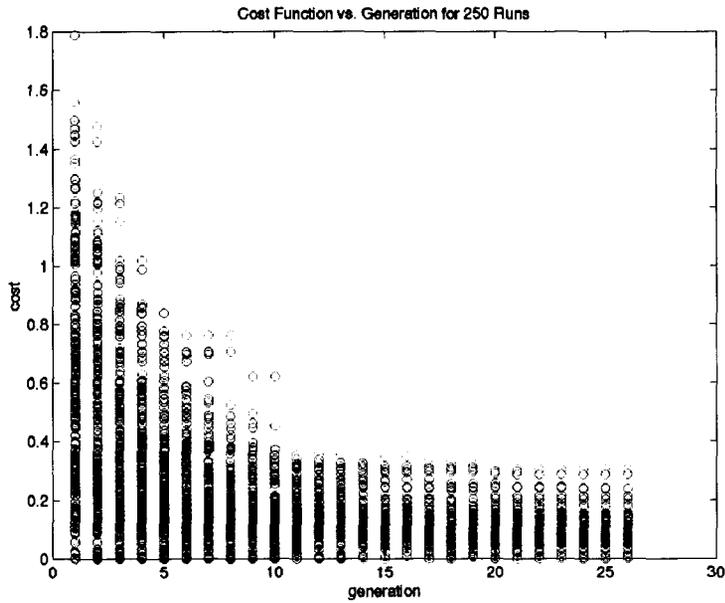
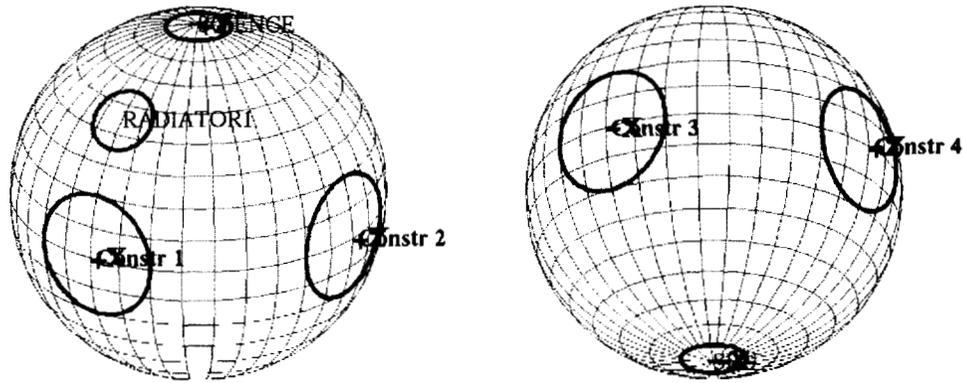


Figure 10: Lowest Cost versus Generation for 250 Runs

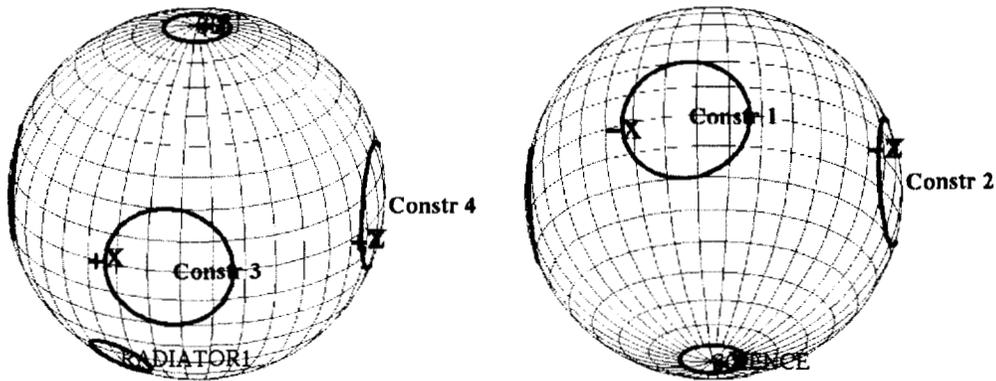
3.4.2. Results

Figure 16 shows the lowest cost achieved for a given generation over the course of 50 generations for all 250 runs. While all runs start out at different initial costs in the first generation, as expected from a randomly selected population, they all reduce the cost as the iteration progresses. Unlike Example 2, where the solutions converged to three local minima and 1 global maximum, here the costs converge to a final range of 0-0.3. Post-simulation analysis revealed that solutions with a cost of up to 0.075 provided an acceptable degree of constraint avoidance. Figure 17 shows the ideal, zero-cost solution.



Maneuver Time: 1

(a)



Maneuver Time: 2062

(b)

Figure 11: Case 4 Solution (a) Initial Attitude (b) Final Attitude

The two slew maneuver solution for the ideal solution in Figure 21(a) is given by:

$$\omega_1 = 1e-3 * [1.1358 -1.4683 0.1385] \text{ rad/sec}$$

$$\Delta t_1 = 1038 \text{ sec}$$

$$\omega_2 = 1e-3 * [1.3414 0.1445 1.5313] \text{ rad/sec}$$

$$\Delta t_2 = 1024 \text{ sec}$$

The total turn time is 2062 seconds.

Figure 18 shows a histogram of the best solutions for all 250 runs with the acceptance threshold at the 0.075 cost level indicated on the chart. As can be seen, the majority of runs generate an acceptable solution with costs less than or equal to 0.075. In 171 runs (i.e. 68.4 % of the runs) the algorithm found acceptable solutions. Besides the total number of runs with acceptable solutions, their distribution among the 250 runs has to be considered as well. This is important, since, in reality, an attitude planner on a spacecraft may initiate a new search using different initial chromosomes, once it hits a local minimum solution with unacceptable performance. For the current example, in 44 cases it

took two runs to obtain an acceptable solution, in 17 cases three runs. Thus, the worst case for this pathological example required the maneuver planner to re-initialize and search three times to find an acceptable solution. The average computation time per run on the Sparc Ultra 80 using non-optimized Matlab .m files was 8.3 minutes with a standard deviation of 5.8 minutes. Consequently, for this example, the worst case required computation time on the order of 25 minutes.

The larger computation time for this example (as compared to Example 2) is expected since, for the *time-varying* case, the closest angular distance has to be determined using iterative search methods. These are more time-consuming than evaluating the closed-form solutions available for the time fixed-case. However, it is important to note that this case example constitutes a worst-case example that is rarely encountered, if at all, in a real mission environment. For this example, twelve time-varying constraint pairs (i.e. 4 moving celestial object x 3 spacecraft boresights) had to be evaluated while in an actual Europa orbit environment only one or two celestial bodies show significant motion.

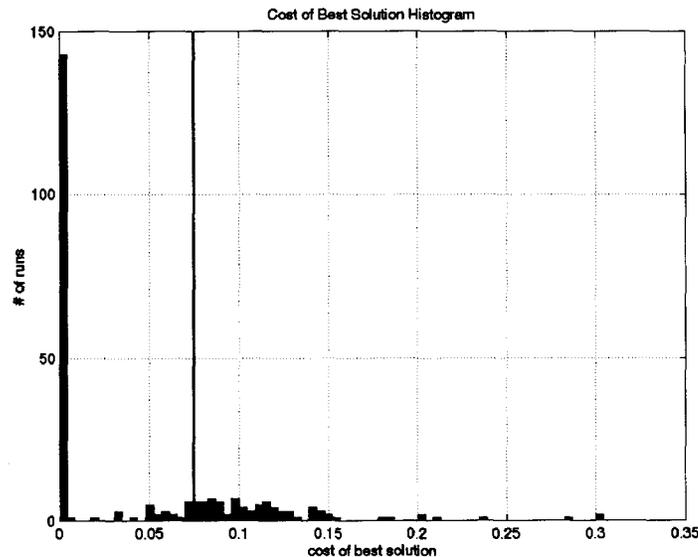


Figure 12: Histogram of Lowest Cost

4. Summary and Conclusions

This study investigated the feasibility of an autonomous attitude planner for the Europa Orbiter. A number of previous and existing maneuver planning implementations were reviewed and a novel approach to the attitude maneuver planning was formulated. It relied on converting the constraint path planning problem into an unconstrained problem by incorporating the constraints into a cost function and using a Genetic Algorithm based random search technique to search for a feasible solution. Four examples illustrated the application of this algorithm for time-fixed and time-varying constraints. Two of the examples were chosen to represent worst-case scenarios not likely to be encountered in a real mission. In all of the examples, feasible solutions were found with computation times in the order of minutes.

Based on the insights gained through this study, the following conclusions can be drawn:

- Even a relatively simple GA based attitude planner provided acceptable solutions for a number of difficult cases.
- Computation times obtained were on the order of minutes for non-optimized code (.m files). Faster performance might be expected in a real-time system environment (depending on the time allocated).
- An On-board autonomous attitude planner is definitely feasible, provided that
 - optimality is traded with computational tractability (e.g. through simplifications and customization)
 - the notion of 'completeness' is abandoned and the notion of 'randomness' is accepted (e.g. by using extensive simulations to gain confidence)
- An attitude planner in conjunction with a Cassini/DS-1 type constraint monitor greatly increases safe, efficient and autonomous attitude maneuver planning capability.
- A number of alternative attitude planner or optimization schemes are available and should be studied (e.g. [10]).
- The capability of current GA based attitude planner can readily be increased by including:
 - Free initial time
 - Fixed final pointing
 - Spacecraft dynamics for Reaction Wheel Control Mode
 - Multi-slew turn maneuvers

These extensions involve more degrees of freedom and require thus longer chromosomes or, equivalently, an optimization over a larger search space.

Acknowledgement

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

- [1] Ludwinski, J., "Europa Orbiter - Level 3 Flight Path Requirements", to be released.
- [2] Singh, G. et al., "A Constraint Monitor Algorithm for the Cassini Spacecraft", AIAA Guidance, Navigation and Control Conference, 1997 New Orleans, LA.
- [3] Canny, J.F., "The Complexity of Robot Motion Planning", MIT Press, Cambridge, MA, 1988.

- [4] Latombe, J.C., " Robot Motion Planning", Kluwer Academic Publishers, Boston, MA, 1991.
- [5] Kia, T. et al., "Topex/Poseidon Autonomous Maneuver Experiment (TAME) Design and Implementation", 20th Annual AAS Guidance and Control Conference, 1997, Breckenridge, CO.
- [6] Lisman, S. et al., "Autonomous Guidance and Control for the New Millennium DS-1 Spacecraft", AIAA Guidance, Navigation and Control Conference, 1996 San Diego, CA.
- [7] Barraquand, J. et al, "Numerical Potential Field Techniques for Robot Path Planning", IEEE Transaction on Systems, Man and Cybernetics, Vol. 22, No.2, March/April 1992.
- [8] LaValle, S.M. et al., "Rapidly Exploring Random Trees: A New Tool for Path Planning", Technical Report 98-11, Iowa State University, Ames, IA, October 1998.
- [9] Kavraki, L.E. et al., " Probabilistic Roadmaps for path Planning in High-Dimensional Configuration Spaces", IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, pp. 566 - 580, 1996.
- [10] Frazzoli, E. et al., " Real-Time Motion Planning for Agile Autonomous Vehicles", AIAA Conf. on Guidance, Navigation and Control, Denver, CO, 2000.
- [11] Bayard, D.S. et al., "Genetic Algorithms for Spacecraft Autonomy - Flight Path Optimization for Mars Precision Landing", JPL D-13900, Vol 6, October 1996.
- [12] Singh, G., "Cassini Inertial Vector Propagator Algorithm", Cassini Control Analysis Book, JPL IOM 3455-97-068, 1997.
- [13] OPSP ACS HW Requirements Peer Review, December 1999.
- [14] Chipperfield, A. et. al., "Genetic Algorithm Toolbox, User's Guide", Version 1.2, Dept. Automatic Control and Systems Engineering, University of Sheffield.
- [15] Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley Publishing Company, January 1989.
- [16] Barhen, J. et al., "TRUST: A Deterministic Algorithm for Global Optimization", Science, Vol 276, No.5315, May 1997.
- [17] Vanelli, C.A., "Autonomous Reorientation of a Maneuver-Limited Spacecraft Under Simple Pointing Constraints", Thesis, California Institute of Technology, May 1997.