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

Richard P. Kornfeld

Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, CA, 91109

AIAA-2003-5784
AIAA Guidance Navigation & Control Conference
Austin, TX, August 14, 2003

Content

- The Nature of the Problem
- Existing Approaches
- Approach to Path Planning
- Why GAs ?
- Cost Function Formulation
- Encoding
- Case Example I: Europa Orbiter during Cruise
- Case Example II: Europa Orbiter in Europa Orbit
- Summary & Conclusions
- References

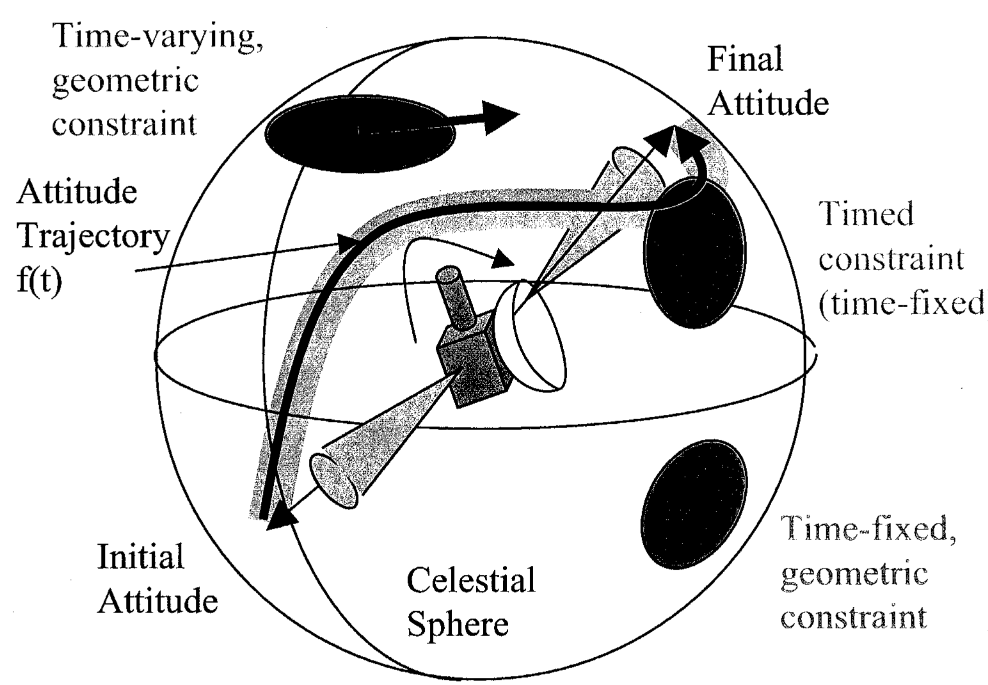
The Nature of the Problem

- For a spacecraft in interplanetary cruise:
  - **time-fixed** boundary conditions: initial and final attitude are not a function of time
  - **time-fixed** constraints: constraints are static
- For a spacecraft in orbit or during flyby:
  - **time-varying** boundary conditions: initial and final attitude are a function of time
  - **time-varying** constraints: constraints change as a function of time
- Types of constraints
  - **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)



The Nature of the Problem

- Robotics Heritage:
    - 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 (*kinodynamic planning*).

    ⇒ Concepts and approaches developed for autonomous robot motion planning are applicable to the spacecraft attitude maneuver planning problem.

- Computational Complexity:
    - Even the simple problem of navigating a kinematic robot in a known environment with polyhedral obstacles has been proven to be computationally hard [1,2].
    - 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*.

⇒ Autonomous path planning with moving obstacles is generally considered to be a computationally hard problem (robotics heritage).

– Even though *complete* algorithms are available, these cannot be used for real-time path planning in many real-world applications [1].

[1]Canny, J.F., "The Complexity of Robot Motion Planning", MIT Press, Cambridge, MA, 1988.
[2]Latombe, J.C., " Robot Motion Planning", Kluwer Academic Publishers, Boston, MA, 1991.

Existing Approaches

- Topex Autonomous Maneuver Experiment (TAME):
  - Iterative walk-around (undirected trial-and-error based) with two turns
  - Not suited for moving constraints
- Cassini Constraint Monitor:
  - The Cassini Constraint Monitor (CMT) *autonomously checks* the commanded attitude trajectory for any constraint violation and, if necessary, *modifies* commanded turn profile to be compliant.

  ⇒ CMT has limited planning horizon (4 seconds) and acts thus *retro-actively*.
- The DS-1 spacecraft used an Attitude Planning Expert (APE) to autonomously turn the spacecraft. It was an off-line tool (i.e. non-real time) to determine the feasibility of a turn. It consists of
  - a simple attitude commander with no constraints avoidance planning capability and
  - a Cassini-type Constraint Monitor

Approach for On-Board Attitude Maneuver Planner

⇒ Primary objective is to obtain a *feasible* solution in a *reasonable* time

⇒ Reduce the problem to be computationally tractable!

⇒ Trade-off between optimality and computational complexity

- Simplify the problem to the greatest extent possible while maintaining the validity of the model:
  - e.g omit non-relevant spacecraft dynamics
- 'Customize' the problem:
  - e.g. define limited, convex attitude maneuver space (e.g. DS-1)
  - e.g. define limited set of possible attitude turns

  ⇒ Trade-off between up-front customization and level of flexibility
- Over-constrain the problem to account for uncertainties
- Incorporate constraints into cost-function to convert it from a constraint to an unconstraint optimization problem
- Using (global) random search techniques, search for and, if found, optimize attitude trajectories that minimize the cost-function

⇒ Use solution 'as is' or as an initial condition to initialize additional deterministic optimization algorithm

Why Genetic Algorithm (GA) ?

Current motion planning approaches can be divided in:

- Heuristic Methods
  - Variational methods
  - Potential field methods
- Enumerative Methods
  - Exhaustive search
  - Dynamic programming
- Random Search or Probabilistic Methods
  - **Genetic Algorithms** →
  - Simulated Annealing
  - Rapidly Exploring Random Trees
  - Probabilistic Roadmap Planner.
- Use a hybrid approach based on a combination of the above

- Directed random search of a large solution space
- Easy and fast to implement
- Constraints on control and state-variables can be incorporated readily
- Many successful applications
- Suited for discrete events such as thruster firings
- Highly parallelizable

Simplifications for this Study

⇒ Used EO spacecraft and EO ACS requirements as a baseline for this study.

- Considered only spacecraft in RCS control mode
- Considered only spacecraft kinematics and neglected spacecraft dynamics:
  - ⇒ Assumed instantaneous turn rate changes ($t_{manuever} >> t_{acceleration}$)
  - ⇒ For EO thruster configuration, this introduces a position error of approximately 1 - 2 deg
- Assumed perfectly balanced spacecraft and thruster firing executions

- Assumed fixed initial time and free final time
- Assumed fixed initial and final attitude
- Constraint the problem to two turns per maneuver

- Assumed emphemeris of celestial objects are parametrized as a function of time

Elements of Cost Function

- **Geometric Constraints:**

$$C_{GC} = \sum_{slew=1}^{2} \sum_{\substack{(i,j)\in \\ \#celest.\ constr.}} \sum_{\substack{\{constraint\ pair\} \\ \#bore\ sights}} F_1(\phi_{ij}^{*}(), \Phi_{ij})$$

$$\phi_{ij}^{*}() = \phi_{ij}^{*}(\mathbf{c}_i, \mathbf{\grave{u}}_i, \mathbf{b}_j, \mathbf{\grave{u}}_{slew}, t_{slew})$$

$$= \begin{cases} \min\limits_{t_o<t<t_f} \phi_{ij}(t) & \text{if } \exists \\ \phi_{ij}(t_f) & \text{else} \end{cases}$$

- **Timed Constraints:**

$$C_{TC} = \sum_{slew=1}^{2} \sum_{\substack{(i,j)\in \\ \#celest.\ constr.}} \sum_{\substack{\{constraint\ pair\} \\ \#bore\ sights}} F_2(T_{ij}(), T_{ij\_max})$$

$$T_{ij}() = T_{ij}(\mathbf{c}_i, \mathbf{\grave{u}}_i, \mathbf{b}_j, \mathbf{\grave{u}}_{slew}, t_{slew}, \Phi_{ij})$$

$$= \begin{cases} \text{time interval } \Delta t \text{ for which } \phi_{ij}(t) < \Phi_{ij} \text{ if } \exists \\ 0 \quad \text{else} \end{cases}$$

- **Fuel:**

$$C_{Fuel} = \sum_{slew=1}^{2} F_3(\boldsymbol{\omega}_{slew}, \boldsymbol{\omega}_{max})$$

- **Total Maneuver Time:**

$$C_{MT} = F_4\left(\sum_{slew=1}^{2} t_{slew}, T_{max}\right)$$

- C_{GC}: cost function for geometric constraints
- $F_{()}$: weighting functions
- $\phi_{ij}(t)$: angular distance between constraint pair (i,j)
- Φ_{ij} : minimum required angular distance for c.p. (i,j)
- $\mathbf{c}_j$: celestial constraints vector
- $\boldsymbol{\omega}_i$: rotation vector of celestial constraint $\mathbf{c}_i$
- $\mathbf{b}_i$: spacecraft boresight
- $\boldsymbol{\omega}_{slew}$: spacecraft rotation vector
- t_{slew} : rotation duration
- T_{ij_max} : maximum allowable constraint violation time for c.p. (i,j)
- $\boldsymbol{\omega}_{max}$: maximum allowable turn rate
- T_{max} : maximum allowable maneuver time

- **Total Cost:**

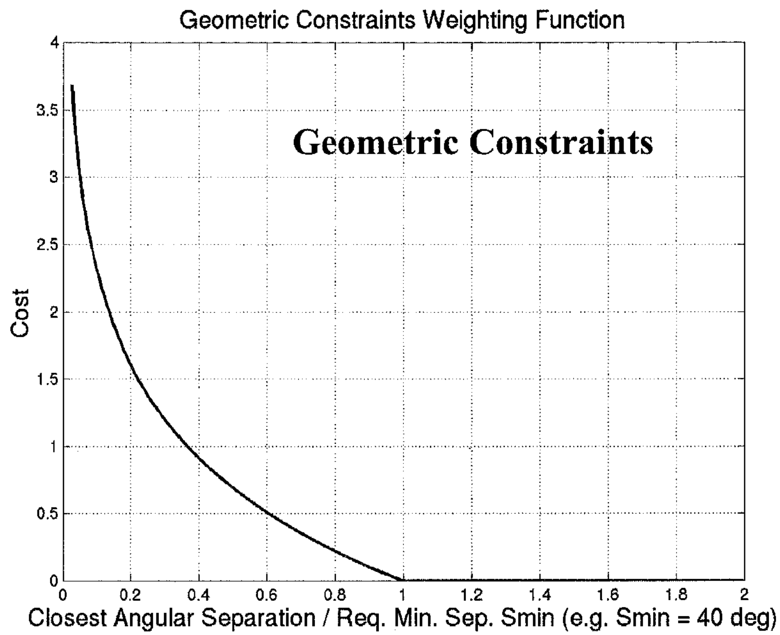
$$Cost = C_{GC} + C_{TC} + C_{Fuel} + C_{MT}$$

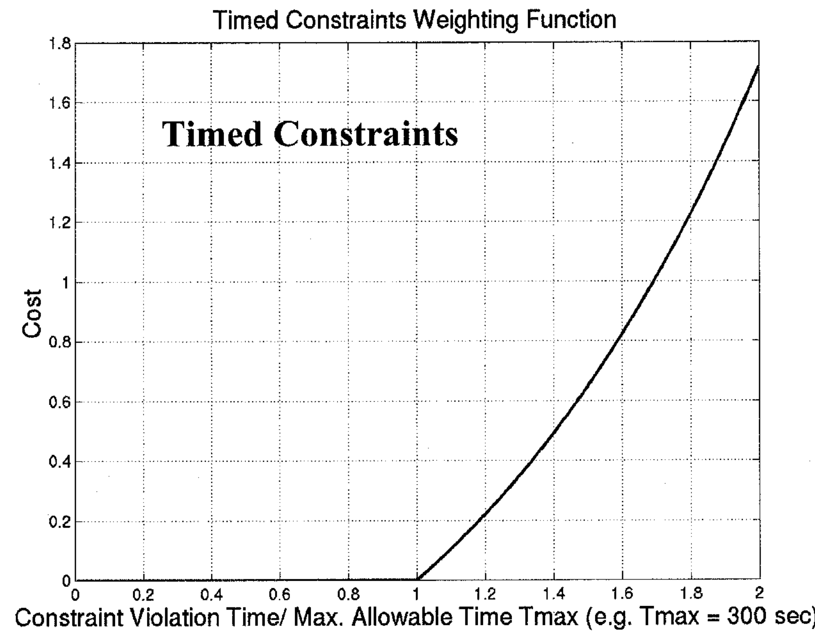
⇒ Time-fixed case: $\phi_{ij}^{*}()$ and $T_{ij}()$ calculated in **closed-form solution**

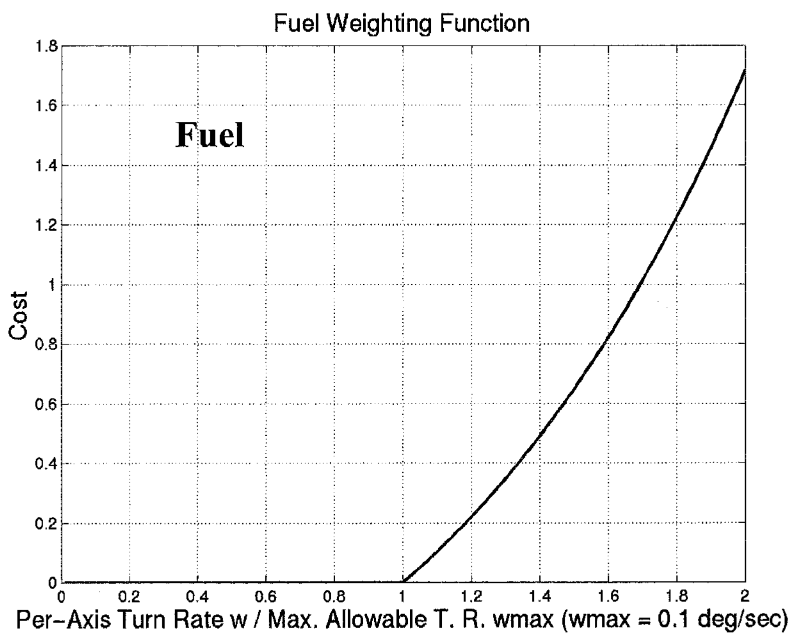
⇒ Time-varying case: $\phi_{ij}(t)$ parametrized as a function of time (t); $\phi_{ij}^{*}()$ and $T_{ij}(t)$ (*) calculated by **iterative search**.

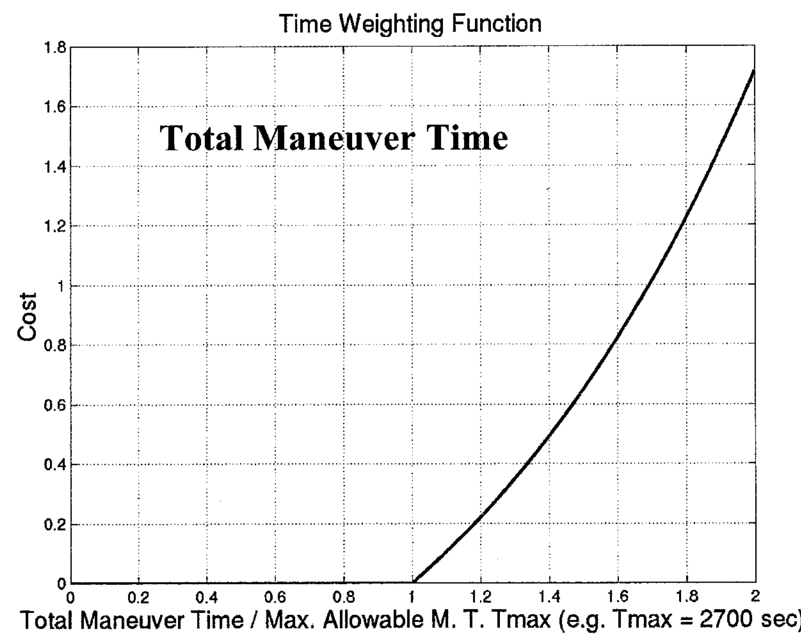
⇒ (*) not implemented

Weighting Functions









Encoding

- For this presentation:
  - Two-slew turn: $[\omega_1, t_1], [\omega_2, t_2]$
  - Fixed initial time: $t_o = 0$
  - Free final time: $t_f = t_1 + t_2$
  - Fixed initial and final attitude: $\Theta_{o,}, \Theta_f$
- Encoded as single-level binary string/chromosome using Gray Coding:

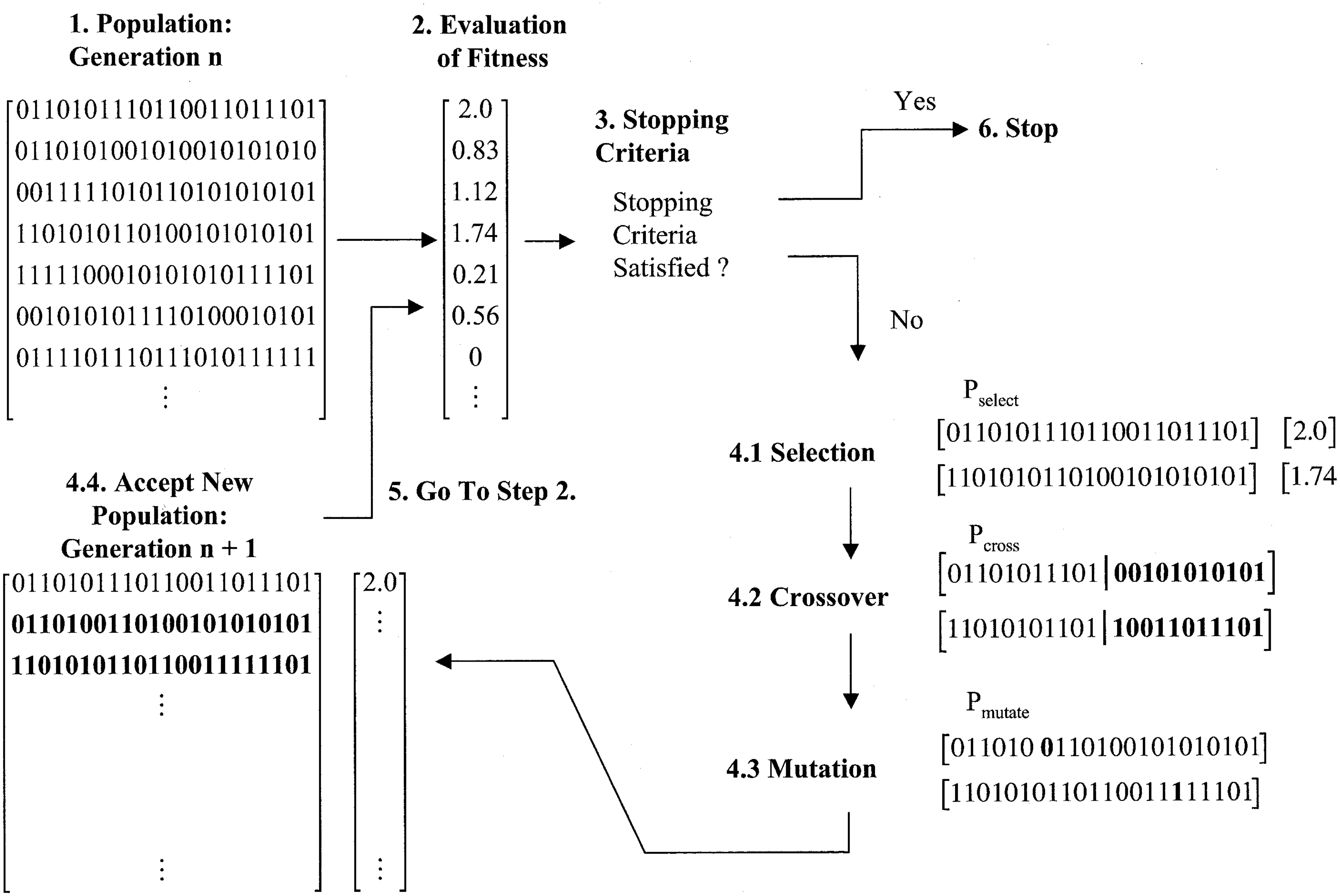
6	6	6	8	8
101011	110001	011011	01110110	11001001
ω_{1x}	ω_{1y}	ω_{1z}	Δt_1	Δt_2
[0, 0.1 deg]	[0, 0.1 deg]	[0, 0.1 deg]	[0, 1800 sec]	[0, 1800 sec]

$\Rightarrow$ 34 bits

$$\underset{\text{initial att.}}{[\Theta_o, t_o]} \xrightarrow{[\omega_1, \Delta t_1]} \underset{\text{interm. att.}}{[\Theta_{interm}, t_{interm}]} \xrightarrow{[\omega_2, \Delta t_2]} \underset{\text{final att.}}{[\Theta_f, t_f]}$$

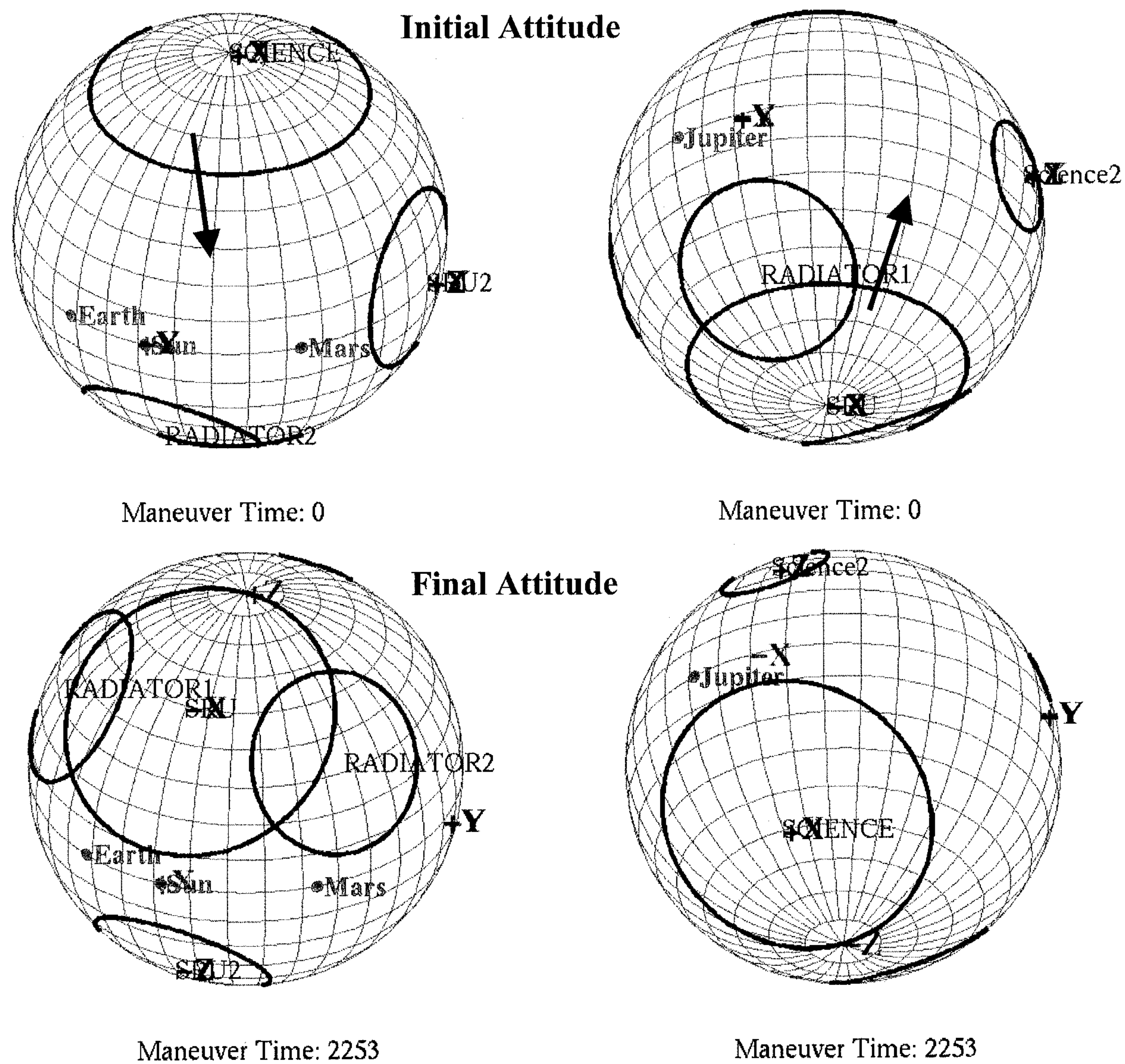
$$\omega_2 = f(\Theta_{interm}, \Theta_f)$$

Outline of the Basic Genetic Algorithm



Example I: EO During Cruise

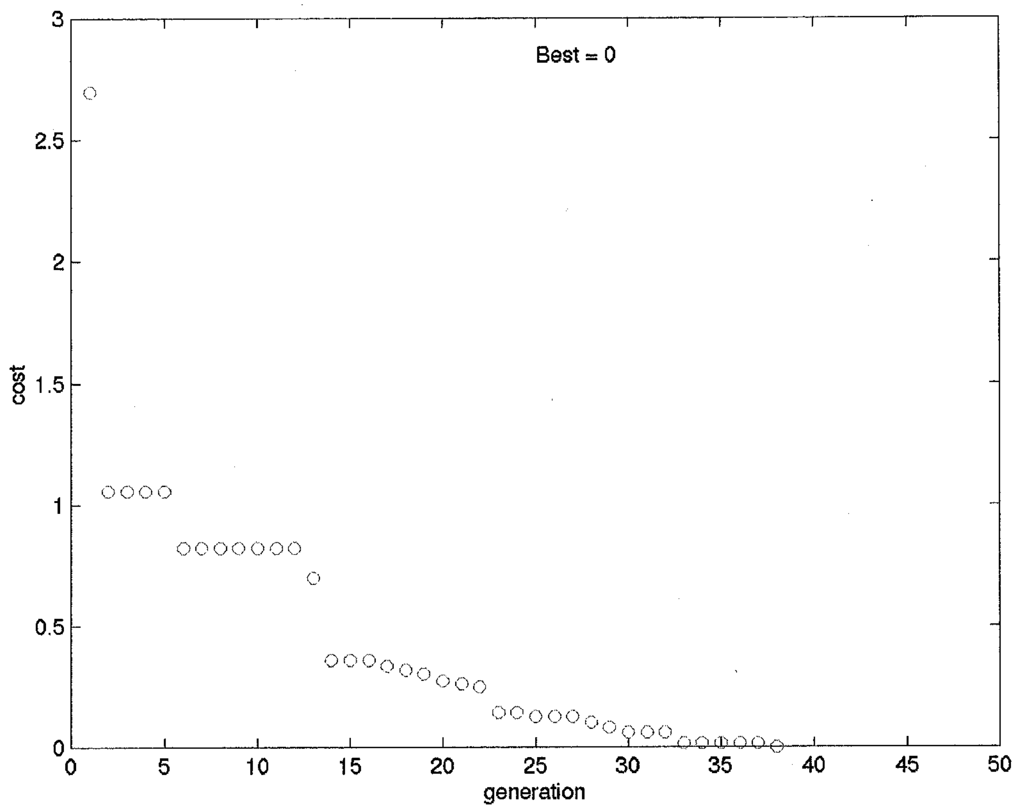
- Europa Orbiter during cruise (2004/06/01)
  ⇒ Celestial objects time-fixed
- Fixed initial and final attitude
- Fixed initial time and free final time
- Protected boresights:
  - SRU: FOV 40 deg
  - Science: FOV 40 deg
  - Radiator1: FOV 25 deg
  - Radiator 2: FOV 25 deg
  - SRU2: FOV 25 deg
  - Science2: FOV 15 deg
- Two slew maneuvers
- Rates < 0.1 deg/s per axis



Example I: EO During Cruise

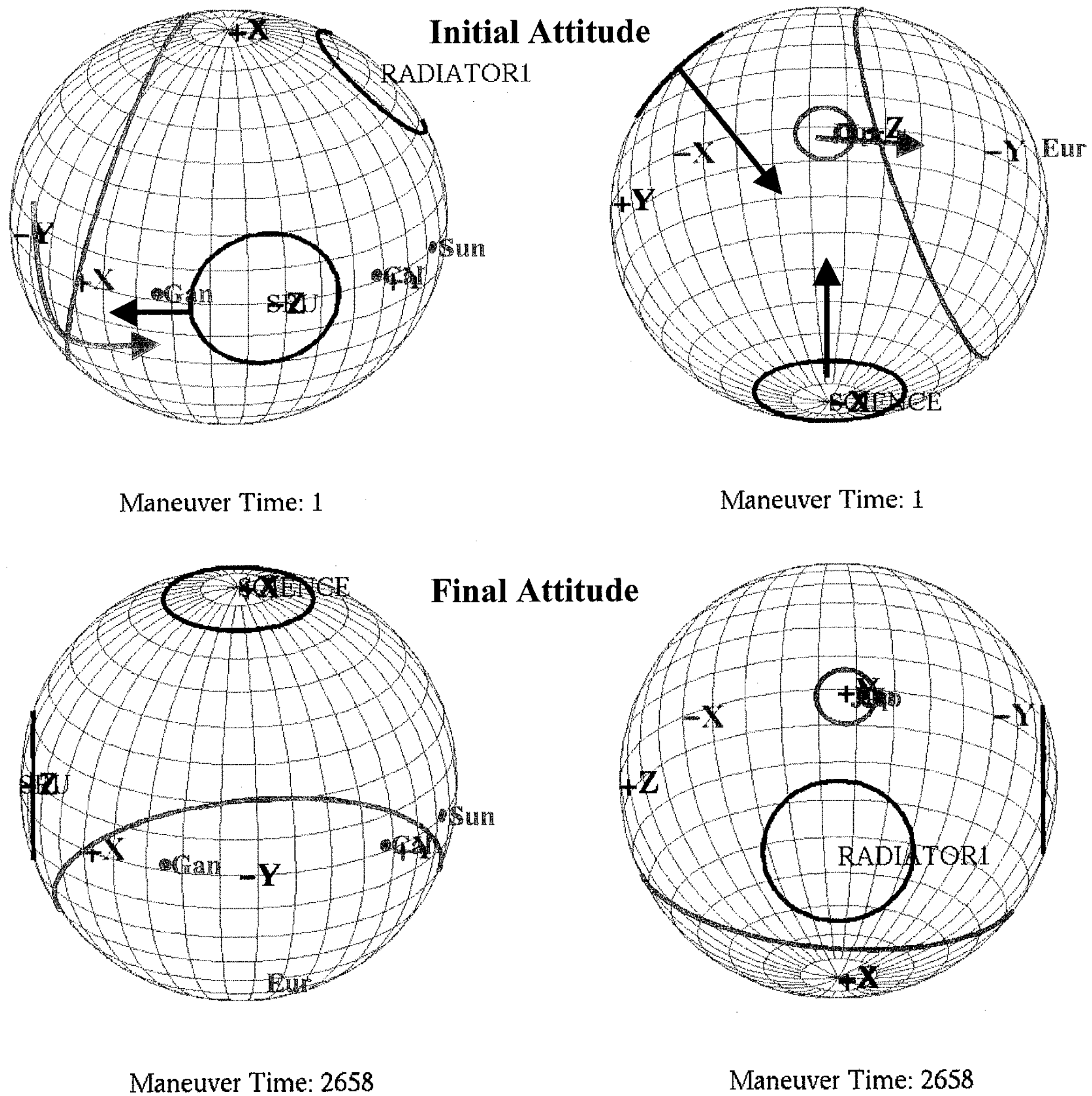
- GA Settings:
  - # of chromosomes: 30
  - max. # of generations: 50
  - chromosome length: L = 34 bit
  - Generation Gap = 0.9
  - Pcross = 0.7
  - Pmut = 0.7/L
  - Stopping criteria:
    cost function < 8*1e-5
- Execution Time:
  227 sec

Cost Function vs. Generation



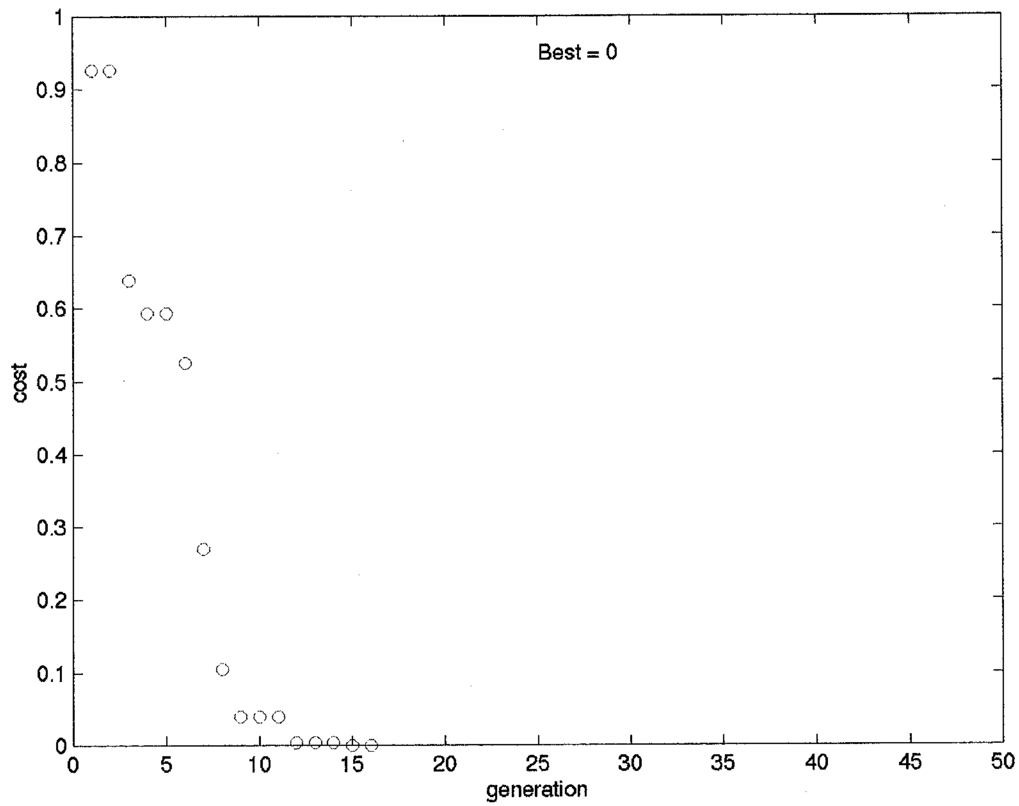
Example II: EO in Europa Orbit (EO^2)

- Europa Orbiter in Europa Orbit (2008/09/22)
  ⇒ Celestial objects **time-varying**
- Fixed initial and final attitude
- Fixed initial time and free final time
- Protected boresights:
  - SRU: FOV 20 deg
  - Science: FOV 20 deg
  - Radiator1: FOV 20 deg
- Two slew maneuvers
- Rates < 0.1 deg/s per axis



Example II: EO in Europa Orbit (EO^2)

- GA Settings:
  - # of chromosomes: 40
  - max. # of generations: 50
  - chromosome length: L = 34 bit
  - Generation Gap = 0.9
  - Pcross = 0.7
  - Pmut = 0.7/L
  - Stopping criteria:
    cost function < 8*1e-5
- Execution Time:
  230 sec.



Summary and Conclusions

- 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 expected when optimized)

⇒ An On-board autonomous attitude planner is definitely feasible, provided that we

- trade computational tractability with optimality (e.g.through simplifications and customization)
- abandon the notion of 'completeness', but instead get comfortable with the notion of 'randomness' (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

References

1. Singh, G. et al., "A Constraint Monitor Algorithm for the Cassini Spacecraft", AIAA Guidance, Navigation and Control Conference, 1997 New Orleans, LA.
2. Canny, J.F., "The Complexity of Robot Motion Planning", MIT Press, Cambridge, MA, 1988.
3. Latombe, J.C., " Robot Motion Planning", Kluwer Academic Publishers, Boston, MA, 1991.
4. Kia, T. et al., "Topex/Poseidon Autonomous Maneuver Experiment (TAME) Design and Implementation", 20th Annual AAS Guidance and Control Conference, 1997, Breckenridge, CO.
5. 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.
6. 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.
7. 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.
8. 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.
9. Frazzoli, E. et al., " Real-Time Motion Planning for Agile Autonomous Vehicles", AIAA Conf. on Guidance, Navigation and Control, Denver, CO, 2000.
10. Bayard, D.S. et al., "Genetic Algorithms for Spacecraft Autonomy - Flight Path Optimization for Mars Precision Landing", JPL D-13900, Vol 6, October 1996.
11. Vanelli, C.A., "Autonomous Reorientation of a Maneuver-Limited Spacecraft Under Simple Pointing Constraints", Thesis, California Institute of Technology, May 1997.
12. Chipperfield, A. et. al., "Genetic Algorithm Toolbox, User's Guide", Version 1.2, Dept. Automatic Control and Systems Engineering, University of Sheffield.
13. Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley Publishing Company, January 1989.