Multi-spacecraft Trajectory Optimization and Control Using Genetic Algorithm Techniques*

Seereeram, Li, Ravichandran, Mehra
Scientific Systems Company
500 West Cummings Park, Ste 3000
Woburn, MA 01801
(781) 933-5355
sanjeev@ssci.com

Robert Smith
Intelligent Comp. Sys. Ctr
3 Coldharbour Lane
Frenchay, Bristol, UK
44 (0) 117 965 6261
robert.smith@uwe.ac.uk

Randal Beard
444 Clyde Building
Brigham Young Univ.
Provo, Utah 84602
(801) 378-8392
beard@ee.byu.edu

Abstract— This paper presents an approach for multi-spacecraft trajectory planning, optimization and control. Maneuver planning as a global optimization problem is solved using Genetic Algorithms (GA). Methods were devised to reduce the dimensionality of the decision space, yet retain adequate generality of maneuver possibilities. A compact formulation based on thruster switching-times was used for generic point-to-point spacecraft maneuvers. Optimal control is implicitly satisfied by "bang-coast-bang" actuation schemes. Maneuver profiles, including line-of-sight and orthogonal collision avoidance, were developed. A GA optimizer selects the optimal parameter set for each scenario. Simulation case studies were performed for 2, 3 and 5-spacecraft formation initialization tasks. Objective criteria used in the evaluation function included: endpoint errors; collision avoidance; path lengths; maneuvering times; fuel usage and equalization. In all cases, a nominal GA computed feasible trajectories. Objective criteria trade-offs were demonstrated by selective weighting. Ongoing work includes multi-objective optimization of multiple spacecraft trajectories using Niched-Pareto Genetic Algorithms.

Table of Contents

1. Introduction
2. Multi-Spacecraft Motion Planning
3. Genetic Algorithms
4. Multi-Spacecraft Trajectory Optimization
5. Sample Results
6. Mission and Task Planning Using GAs
7. Conclusions

1. Introduction

Multiple-spacecraft missions, under either centralized or coordinated control, are rapidly emerging as a principal component of current and future planned space missions. Using relatively low-cost "micro-spacecraft" architectures, they offer enhanced mission robustness (eliminating single-point failures) with reduced overall mission costs. Programs such as NASA's ST-3 Separated Spacecraft Interferometer and the TPF (Planet Finder) missions, highlight the immediate need for effective multi-spacecraft control and autonomy technologies. Formation Flying introduces new challenges in control technology in order to meet desired specifications for successful mission completion.

This paper addresses the multiple spacecraft trajectory optimization and control problem by using Genetics-based (evolutionary) techniques for solving the implicit global trajectory optimization problem. Initial study used Genetic Algorithms to find and tabulate sets of control policies for various constellation initialization problems. These strategies can be used to form a closed-loop feedback policy which dynamically optimizes the next applicable control in the presence of dynamic uncertainties and changes in model and environment information. By using genetics-based techniques, families of control policies can be encoded and maintained during closed-loop operation as "ordered populations", from which the feedback controller can select optimal strategies according to its current state. By genetic exploration of the global space of control policies, disparate solutions can be analyzed, and alternate feedback strategies made dynamically available to meet sudden or gradual changes in environment variables.

Various encoding schemes were investigated for embedding the control of multiple spacecraft as an optimization decision variable. Because of the high-dimensionality of the multiple spacecraft control problem, methods were devised to reduce the size of the decision space, yet retain the full generality of maneuver possibilities. The main idea used was to avoid optimization of temporally-discretized actuator control variables - which leads to a computationally intractable problem. Instead, formulations based on switching-times were used to cast the deci-

*0-7803-5846-5/00/$10.00 c 2000 IEEE

sion space in terms of control switches - generically a low number for initialization-type maneuvers. This formulation is directly applicable to thrusters, and generically applicable to other saturation-type actuators. A critical advantage of this approach is that the basic objectives of optimal control are implicitly satisfied by selection of these "bang-coast-bang" actuation schemes.

Several maneuver profile components were developed, including line-of-sight and orthogonal collision avoidance components. A GA optimizer selects the optimal parameter set for each scenario. Simulation case studies were performed for 2, 3 and 5-spacecraft formation initialization tasks. Objective criteria used in the evaluation function included: position/orientation endpoint errors; collision avoidance; minimization of path lengths; minimization of maneuvering times; minimization of fuel usage; and equalization of fuel loads across multiple spacecraft. In all cases, a nominal application of the GA method resulted in feasible trajectories for all spacecraft. Various trade-offs among the objective criteria were demonstrated by selective weighting of their respective penalty functions.

2. Multi-Spacecraft Motion Planning and Control

Several control scenarios for multi-spacecraft constellations can be identified, including: initialization, precision station-keeping, reorientation, rotation, expansion/contraction and maneuvering. Several of these scenarios consider constellations with fixed inter-spacecraft relative locations, collectively called constellation maneuvering, and have been investigated in current research [1]. Various approaches have been applied to the constellation maneuvering problems, ranging from decentralized control of each micro-spacecraft [2] coordinated via inter-spacecraft communications, to constellation control based on the concept of a virtual rigid body, interfaced with control laws for formation keeping and relative attitude alignment [3].

In contrast, the constellation initialization problem has received comparatively little attention. There are several instances when a constellation must be initialized - for example: (1) immediately after the spacecraft are deployed from a launch vehicle, (2) when one spacecraft in the constellation experiences catastrophic failure, and (3) when the overall task of the fleet of spacecraft dramatically changes.

Several considerations complicate the spacecraft initialization problem: (1) The constellation may be a heterogeneous collection. (2) The spacecraft must position themselves without colliding with other spacecraft. (3) Each spacecraft will be equipped with sensors that detect the position of other spacecraft in its near vicinity, but that sensor will be limited in its range. Hence it may not be possible for any given spacecraft to know the relative, or inertial positions of all the other spacecraft. (4) Spacecraft may not be able to communicate with each other. (5) Spacecraft life is limited by fuel.

In its most general form, multi-spacecraft trajectory optimization and control leads to an extremely difficult dynamic programming problem. Practical issues such as collision avoidance and minimum fuel usage result in highly nonlinear, multi-modal and ill-conditioned global optimization problems, which cannot be solved by analytic techniques (such as gradient descent or calculus of variations-based optimal control). Given the potential difficulties of computing collision-free paths for N spacecraft in a dynamic environment, it is unreasonable to expect to find general solutions for motion planning and trajectory generation of N spacecraft. Instead, methodologies applied to multiple spacecraft maneuvering within a common workspace should incorporate facets of both optimal control of actuator resources together with generic collision avoidance and constraint-satisfaction considerations.

Autonomous Control Architecture

For autonomous multiple spacecraft trajectory optimization, we propose a hybrid architecture (Figure 1), in

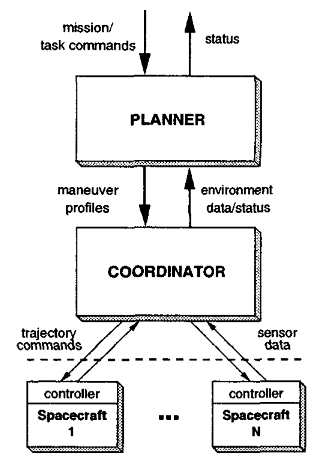

Figure 1: Supervisory Planning/Control Architecture

which global (mission and task level) supervisory planning is performed in a centralized manner, but local execution of each agent's command set is performed in distributed fashion. Off-line pre-planning of coordinated maneuvers will be done by the *planner*, resulting in open-loop instructions for each spacecraft to follow. During real-time maneuver execution, open-loop instructions are sequenced, scheduled and transmitted by the *coordinator* to each spacecraft's local closed-loop controllers. Open-loop instructions may comprise kinematic (positions/velocities), dynamic (forces/torques) and/or direct actuation (thruster switching) commands as appropriate to the task. Real-time control of each spacecraft in accordance with open-loop commands is performed by each spacecraft (de-centralized) under the coordination of the supervisory levels. The *coordinator* adds in timing information to the high-level instructions to produce synchronous trajectory commands to be followed. During execution, the *planner* monitors the evolution of the formation by interpretation of sensory and environment feedback, and instructs modifications or gross alterations to the maneuver(s) as updated environment and spacecraft interaction information becomes available.

Our current research is developing a comprehensive autonomous trajectory optimizer for multiple spacecraft. It includes open-loop trajectory optimization, and closed-loop control with trajectory modifications to account for dynamic uncertainties and/or unforeseen events. In this paper, we report on off-line global trajectory optimization using GA techniques, typically performed by the supervisory *planner* module. Closed-loop tracking of the off-line trajectories, together with real-time (pro-active or reactive) trajectory modifications will be developed in follow-on research. In the remainder of this paper, we present details on the Genetic Algorithm technique used for global optimization, and indicate the direction of current research into developing an autonomous multiple spacecraft trajectory optimizer.

Global Optimization

For space-based applications, globally efficient solutions for motion planning and control have the highest priority, given the limited on-board fuel, power, processing and control authority available. It is clear that finding solutions to the multi-spacecraft trajectory planning problem requires the use of an optimization-based strategy. However, it is difficult, if not infeasible, to find analytic solutions to the constrained trajectory optimization problem. Additionally, multiple objectives mean that the problem may not have unique solutions which simultaneously optimize all criteria. Our research has demonstrated that even for low numbers of spacecraft ($\sim$3), the topology of a typical objective space is multi-modal and possibly discontinuous. To address this problem, we propose the use of Genetic Algorithms (GA) as a suitable global optimizer.

3. Genetic Algorithms

The field of global optimization has generated significant interest in the past decade, mainly because of the importance of finding global optima as well as the difficulty of the problem in general. Genetic Algorithms are a class of search procedures whose mechanics are based on those of natural genetics. These algorithms represent a new approach to search and optimization that avoid most of the problems encountered by conventional approaches. No preconditions are applied to the behavior of the systems modeled, and solutions can be generated for broad sets of problems. GAs are global searches, with less sensitivity to the initial point of the search than calculus-based approaches. As a result, GAs can potentially locate effective solutions which are beyond the reach of conventional methods.

Although the evolutionary analogy on which GAs are based has an intuitive appeal, GAs are also supported by a substantial body of theoretical understanding. Key observations of GA theory are: (1) GA operators lead to the propagation of building blocks of genetic code that correlate to high fitness throughout the population. (2) The GA yields efficient, implicitly parallel evaluation of many building blocks, through the GA's population basis. (3) The GA yields near-optimal exploitation of stochastically obtained information on building block fitness [5].

Implementing a genetic algorithm requires the design of three components: **representation**, **fitness**, and **reproduction**. The most basic form of GA starts with a population of individuals (simple data structures, often bit strings), each representing a potential problem solution, or a portion of a potential problem solution. Some function, procedure, or simulation is used to evaluate the utility (called fitness) for each individual of this population. Then, a sequence of new populations is generated and evaluated using a set of simple, genetically inspired operators. These operators fit into three categories: *selection* (survival of the fittest individuals), *recombination* (mating of two or more individuals), and *mutation* (small, random changes to new individuals).

Tuning a genetic algorithm involves the selection of a number of parameters that specify the size of the population, and rates of crossover and mutation, methods that specify initialization, parent selection, and replacement. A cursory glance may leave the impression that GAs are *simple* to implement. Conceptually, the description of a GA is indeed simple. However, it is often difficult to select GA representations and tuning parameters in order for the resulting solutions to be useful.

For practical applications, the choice of *representation* is a crucial one. Relative efficiency of the optimization procedure relies on representing decision variables in a compact form, without losing generality and/or overly restricting the search space. Representational compactness (short bit-strings) is desirable because of its direct influence on the efficiency of the search procedure - the global optimizer is better equipped to rapidly explore disparate and disjoint regions of the decision space. However, sufficient discretization must be maintained if the decision variables are to retain generality and develop high-quality solutions (low errors). This trade-off must be explored in the context of each individual GA application to find adequate representation(s).

4. Multiple Spacecraft Trajectory Optimization

For our initial study, spacecraft are modeled by rigid body Newton/Euler six degree-of-freedom dynamics with fixed centers of mass. Given the emphasis on small (10 – 50 kg) spacecraft in many proposed missions, this is not an unrealistic assumption for first principles' analysis. In this study, cases are presented for two, three and five-spacecraft initialization tasks, each fully six degree-of-freedom actuated using on-off type actuators (eg. gas-jet RCS thrusters, PPTs, etc.). For spacecraft trajectory control, decision variables correspond to the control actuation applied to all the spacecraft in the constellation. It is imperative to encode possible and desirable maneuvers with as few parameters as possible. *A priori* system and engineering knowledge can provide parameterizations of quantitatively (and qualitatively) "good" control schemes, in a compact form.

Maneuver Parameterization

For point-to-point maneuvers, open-loop optimal trajectory control using gas-jet thrusters can be characterized in terms of a "bang-coast-bang" switching control. For the constellation initialization, we assume that each spacecraft's maneuver can be decomposed into one (or more) point-to-point maneuvers. Thrusters are assumed to provide fixed magnitude forces/torques. Typically, formation initialization involves re-positioning and/or re-orientation of several spacecraft from a given initial configuration to a specified goal configuration. Without loss of generality, we consider the problem of coordinated maneuvering of several free-flying spacecraft with specified starting and ending positions/orientations. We assume that each spacecraft can provide independent forces/torques along each principal axis. Arbitrary thrust vectors are scaled by the orthogonal components available from body-attached thrusters with respect to inertial coordinates. Each spacecraft has an assigned spherical collision radius, which defines its "collision" zone for gross maneuvering.

For this class of problems, the optimal path for each spacecraft (in isolation) is a straight-line, or line-of-sight (LOS), trajectory from start to goal, using a "bang-coast-bang" control profile for on-off thrusters. However, to provide for path modifications an orthogonal complement, the Collision Avoidance (CA), component is added to the LOS trajectory (Figure 2). Collision avoidance is achieved by describing a *bounding sphere* around each spacecraft, sufficiently large to contain the spacecraft at arbitrary orientation. In this manner, collision avoidance maneuvers can be generated by translational motion planning of the N bounding spheres, resulting in

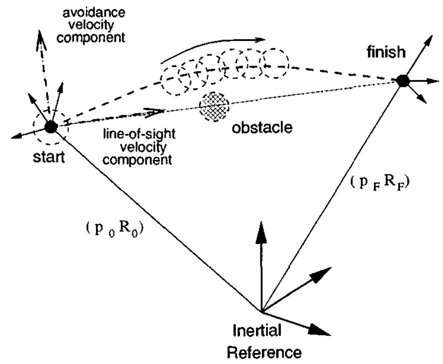

Figure 2: Line-Of-Sight and Collision-Avoidance Path Components

significant simplifications in the planning algorithm's implementation - collision checking is now reduced to the calculation of the Euclidean distance between any two bounding spheres. For the proposed miniature spacecraft missions, this represents a practical representational convenience, considerably reducing the geometric computations required for motion planning algorithms.

In the following, we rotate each spacecraft from initial to final orientation independently along the attitude geodesic on the orientation manifold $\mathcal{SO}(3)$, while the spacecraft is in (translational) flight from starting to final locations. For open-loop trajectory planning, switching times are directly related to the LOS coast speeds v_c and CA avoidance speeds v_q (parametric speeds along and perpendicular to the LOS path direction shown in Figure 2). For any given task, coast speeds will be limited by the maximum attainable velocities in the LOS directions (for minimum-time path traversal), and the minimum velocity which is consistent with reaching the goal position in a finite time T (minimum-fuel path traversal). Any coast speed v_c within these limits, can then be used for a given task, subject to collision avoidance and other constraints. Collision avoidance speeds v_q are limited by the maximum achievable force in the avoidance direction. For each spacecraft, v_c and v_q variable limits are pre-computed from actuator properties and spacecraft orientations. Figure 3 illustrates the basic force, velocity and displacement profiles along the LOS and CA directions respectively.

Objective Functions

Given a LOS and CA trajectory parameterizations above, the entire open-loop motion of each spacecraft is completely specified by two parameters v_c and v_q. Trajectory optimization for N spacecraft is represented in terms of $2N$ optimization parameters. For typical constellation initialization problems, this parameterization affords a rich set of trajectory variations, sufficient for many representative point-to-point motion planning

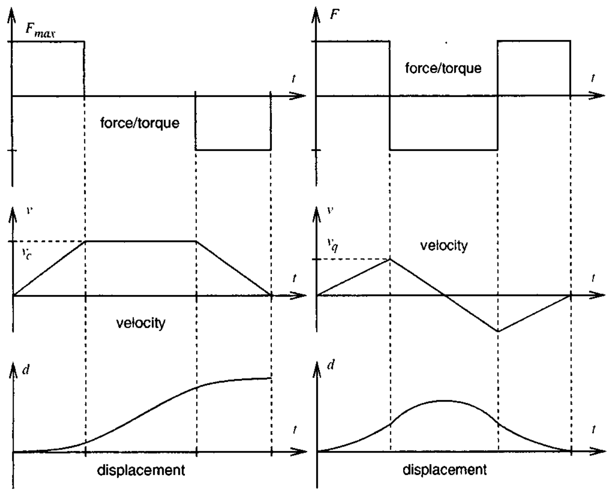

Figure 3: Line-Of-Sight and Collision-Avoidance Path Components

tasks. Importantly, the optimization variable grows linearly with N, resulting in a computationally efficient procedure for solution of N-spacecraft trajectory optimization.

For the N-spacecraft problem, our Phase I investigation considered trajectory optimization subject to the following requirements: (1) Collision free, (2) Shortest path, (3) Minimum time, (4) Minimum fuel usage, and (5) Uniform fuel distribution at the end of the maneuver. Note that some requirements may conflict with each other - eg. a collision-free path may not be the shortest (Euclidean distance) possible, and a trajectory that uses minimum-fuel may result in highly non-uniform fuel distribution.

The objective J used in the GA optimization problem is:

$$\begin{aligned} J &= \sum_{i \neq j} J_c(i,j) + w_p J_p + w_T T_{max} \\ &\quad + w_f \sum_{i=1}^{N} (f_i(t_0) - f_i(t_f)) \\ &\quad + w_d \sum_{i=1}^{N} \frac{f_i(t_f)}{\sum_{i=1}^{N} f_i(t_f)} \log \frac{f_i(t_f)}{\sum_{i=1}^{N} f_i(t_f)} \end{aligned} \quad (1)$$

where (i,j) are spacecraft indices for an N-constellation.

Collision Penalty , $J_c(i,j)$, is assigned to any candidate solution which contains potential collisions. The entire set of N trajectories is checked pairwise for collisions.

Path Length Penalty This term represents a minimum distance cost:

$$J_p = \sum_{i=1}^{N} (l(\vec{x}_i) - r_i) \quad (2)$$

where $l(\vec{x}_i)$ is the Euclidean length of actual trajectory $\vec{x}_i$, and r_i is the length of the line of sight (LOS) trajectory from the starting point to the end point of the trajectory. Note that $J_p = 0$ if and only if the spacecraft travels along its LOS trajectory (ie. $v_q = 0$).

Execution Time Penalty The third term represents a minimum time cost:

$$T_{max} = \max_i T_i \quad (3)$$

where T_i is the time taken by the i^{th} spacecraft to reach its goal position.

Fuel Consumption Penalty In (1), $f_i(t)$ is the amount of fuel contained on the i^{th} spacecraft at time t. Its dynamics is assumed to be

$$\dot{f}_i = \begin{cases} -\gamma(|F_{xi}| + |F_{yi}| + |F_{zi}|); & f_i(t) > 0 \\ 0; & \text{otherwise,} \end{cases} \quad (4)$$

where γ is a positive constant, and F_{xi}, F_{yi}, F_{zi} are the forces exerted by the three thrusters along the x, y, z axes of the body frame of the spacecraft.

Fuel Distribution Penalty The fourth term in (1) represent the costs associated with overall fuel usage *and* uniform fuel distribution after the maneuver respectively. This term is motivated by the negative entropy of a probability distribution, which is minimum for a uniform distribution.

Objective weights $w_p, w_T, w_f, w_d > 0$ determine the relative importance of each factor in a particular scenario/task.

5. Sample Results

In each of the cases below, the global optimization problem was formulated and a family of candidate solutions found using the GENESIS [6] GA software implementation. Model parameters are listed below:

Spacecraft Parameter	Value
Mass	10 kg
Moment of inertia	1 Nm
Bounding Sphere Diameter	3 m
Maximum turn rate	10 deg/s
Maximum thrust	0.01 N

Typical GA algorithm settings were used to facilitate easy comparisons. A small number of function evaluations (`Total Trials` = 1000 $\sim$ 5000) was used in order to ensure that solutions were found with a limited amount of computation. `Population Sizes` ($50 \sim 100$) are chosen to be larger than the number of bits per structure. `Structure Length` ($28 \sim 42$)is comprised of $2 \times N \times b$, where N is the number of spacecraft, and b is the number of bits used to represent the range of each decision variable. The `Crossover` (0.75) and `Mutation` (0.001) rates are generic GA settings for general-purpose optimization. They can be varied according to the desired trade-off between *exploration* of the decision space vs. *exploitation* (or convergence) to any neighboring optima. Relatively little was done to tweak GA software settings.

Case 1: Constellation Re-orientation

This scenario represents a typical three-spacecraft constellation reorientation. Three spacecraft are modeled, representing a combiner (diameter 5 m), and two collectors (diameter 3 m), with specified initial orientation. The constellation is required to point in specific directions to collect interferometry data. In this scenario we consider two consecutive multi-spacecraft maneuvers. The first maneuver starts at $t_0 = 0$, and represents an initial cluster of all three spacecraft executing a separation to the first formation orientation. The second maneuver follows at $t_1 = 400 \ldots t_2 = 1200$, and represents a re-pointing of the formation to the opposite direction. For the new formation, the combiner spacecraft must be moved to the opposing side of the formation, while the flanking collectors swap places.

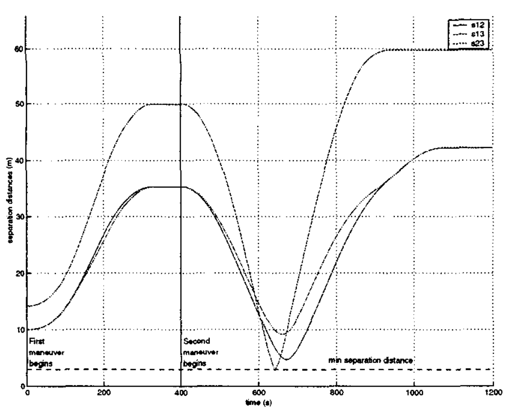

Figure 4: Case 1: Separation distances.

The overall maneuvers are collision-free, and meet the task specifications desired. Figure 4 shows the separation distances between all spacecraft pairs for the entire time history. It can be seen that the separation distances between all pairs of spacecraft remain greater than their safe distance. Snapshots of an animation sequence of the second formation re-orientation maneuver are shown in Figure 6.

This simulation requires all three spacecraft to cross

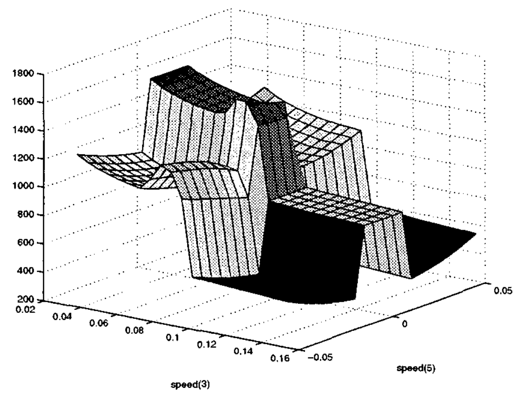

Figure 5: Case 1: Variation of $J(x)$ vs. decision variables (x_2, x_5).

paths during a re-configuration maneuver. No *a priori* information is provided about which units should avoid each other - although this can be influenced by the choice of optimization penalty weights for each unit. The "best" solution returned by the initial GA application is shown. Note, however, that other solutions returned may have differing roles for each spacecraft along the way. This is reflected in the fact that the topology of the objective function for multiple spacecraft and multiple criteria turns out to be highly multimodal (see Figure 5 for an example of $J(x)$) and possibly discontinuous. Clearly, any multi-variable optimization technique which progresses using gradient information, and/or hill-climbing techniques will rapidly converge to the nearest local minimum. This multi-modality of the objective function space almost requires a covering technique which can generate diverse and distributed candidate search vectors, in order to effectively explore the majority of the decision space.

Case 2: Fuel Equalization

In Case 2, we require a fuel usage pattern resulting in equalization of the final fuel loads for both spacecraft. Each spacecraft starts out with an uneven fuel load. (Figure 8). The objective weights are adjusted to achieve fuel equalization. The lowest-cost solution generated by a preliminary GENESIS GA application is:

$$v = \begin{bmatrix} 0.2783 \\ 0.0998 \\ 0.0256 \\ 0.0185 \end{bmatrix} \begin{array}{c} \text{Spacecraft 1 LOS coast speed} \\ \text{Spacecraft 2 LOS coast speed} \\ \text{Spacecraft 1 CA (perp.) speed} \\ \text{Spacecraft 2 CA (perp.) speed} \end{array} \qquad (5)$$

Figure 7 shows the resulting trajectories of both spacecraft. The coast speeds determined by the optimizer indicate that Spacecraft 1 travels substantially faster than Spacecraft 2, thus expending more fuel. Initial and final fuel levels of the two spacecraft are shown in Figure 8. Using the optimized trajectories, both spacecraft are left

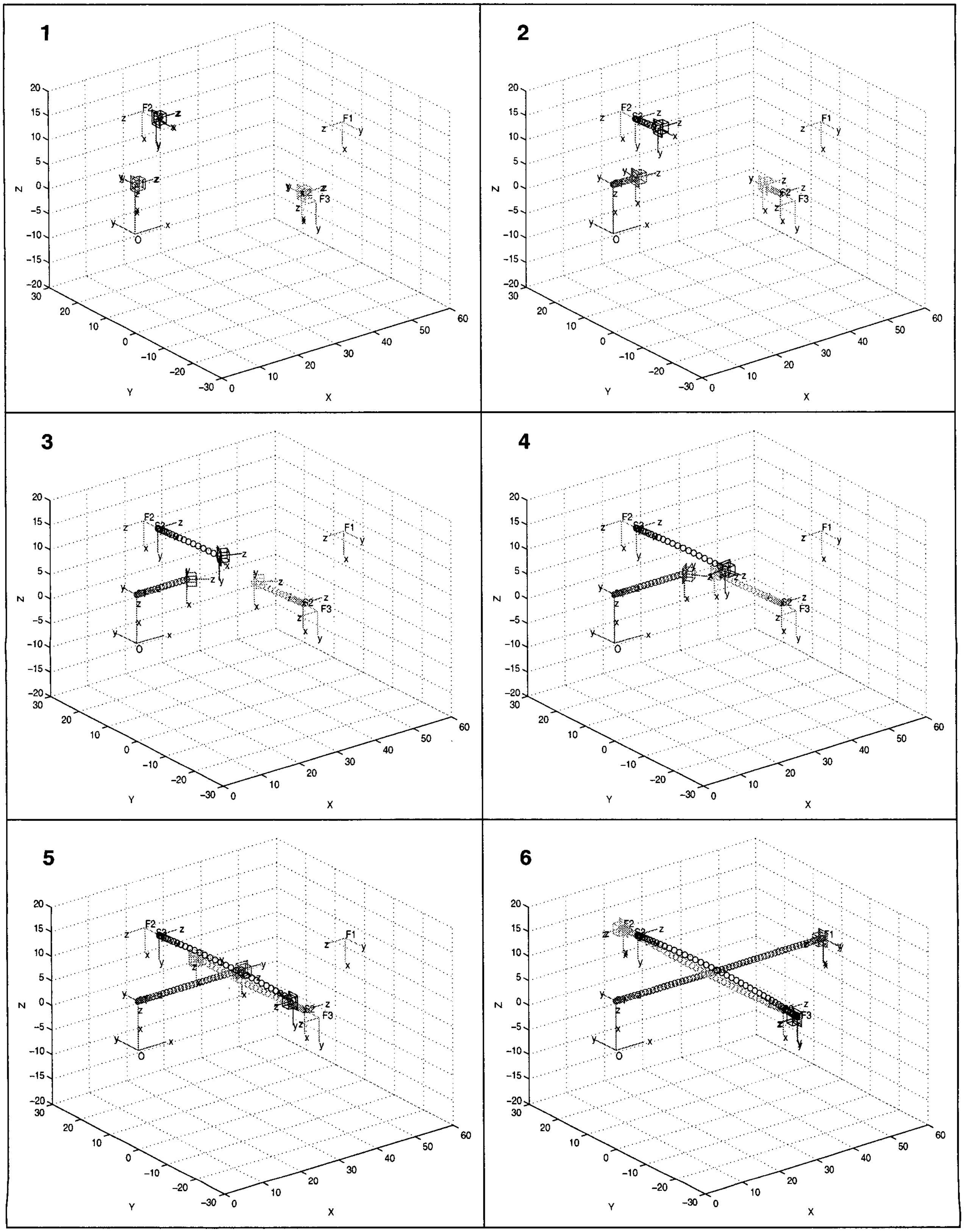

Figure 6: Case 1: Three-spacecraft Constellation Re-orientation.

with approximately the same amount of fuel at the end of the maneuver.

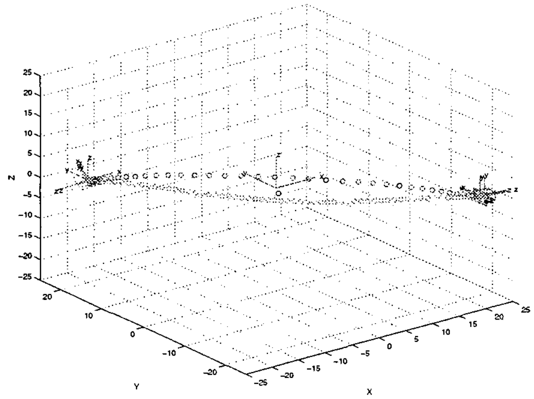

Figure 7: Case 2: Two Spacecraft with Fuel Equalization - 3-D View.

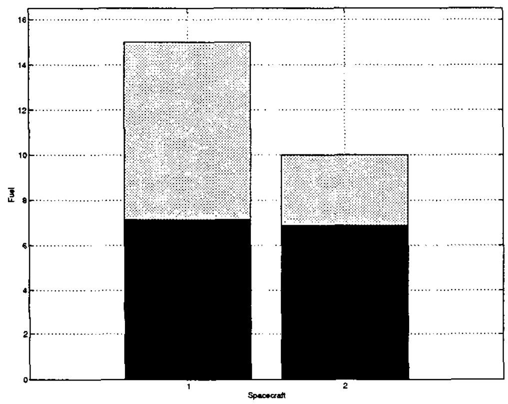

Figure 8: Case 2: Equalized Fuel Usage: yellow = initial, blue = final.

Case 3: N-spacecraft Collision Avoidance

Here, the number of spacecraft is increased to $N = 5$. Spacecraft tasks are to either exchange positions with one another or perform a maneuver with potential collisions with other spacecraft. Due to the number of spacecraft under consideration, this problem is fairly complicated and a good solution cannot be easily found by inspection. The lowest-cost solution generated by a preliminary GENESIS GA application is:

$$\begin{aligned} v_c &= \begin{bmatrix} 0.1058 & 0.0824 & 0.0684 & 0.2257 & 0.1369 \end{bmatrix} \\ v_q &= \begin{bmatrix} 0.0295 & 0.0406 & 0.0020 & -0.0319 & -0.0232 \end{bmatrix} \end{aligned}$$

Note that the solutions are indeed a combination of varying speeds, both along the LOS trajectories, and in the CA directions. Figure 9 shows the trajectories of all spacecraft. Distances between all spacecraft pairs $s_{ij}(t)$ for $i \neq j$ (Figure 10) are sufficiently far away from each other to avoid collisions throughout the entire constellation maneuver.

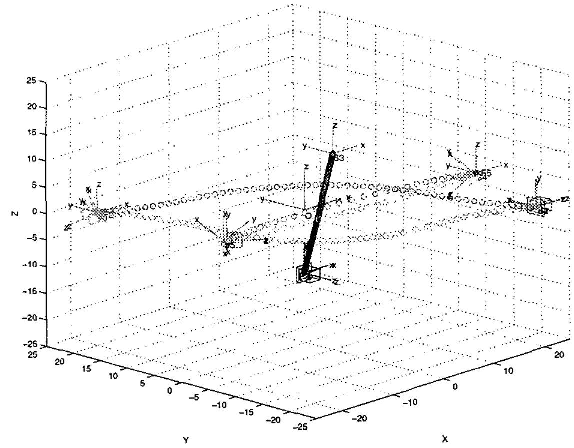

Figure 9: Case 3: $N = 5$ Spacecraft Maneuver Optimization - 3-D View.

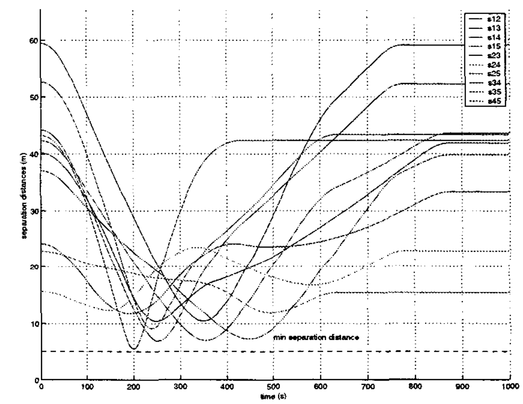

Figure 10: Case 3: Inter-spacecraft Separation distances.

6. Mission and Task Planning using GAs

The studies presented above use Genetic Algorithms primarily to solve a global optimization problem. However, the real advantage of GAs is in generating and refining a diverse population of solutions from which a higher-level decision maker can select. Realistically, the multiple spacecraft trajectory problem is a *multi-objective* optimization problem, and frequently several disparate solutions will need to be reviewed and evaluated prior to selecting a maneuver plan. Indeed, even after an initial plan has been executed, the possibility of dynamic uncertainties and system/environment changes means that a currently operational maneuver will often need to be modified or re-planned in order for a particular mission task to be achieved. Alternatively, disparate and often conflicting optimality criteria mandates trade-off evaluation of distinct solutions. In many cases, it is neither obvious nor apparent how a mission planner should weigh various optimality criteria in constructing a *single* objective function for a nominal optimization strategy. Our current research focuses on the development of Genetic Algorithms for multi-objective optimization in such mission and task-planning. In the following sections, we describe the basic concepts which we are currently investigating to implement multi-spacecraft mission and task optimization.

The Niched-Pareto GA

In the initial experiments presented above, the GA population is primarily as an artifact of the search process. That is, the population is a data structure exploited during the search, but, as the GA converges towards a single type of individual, the population becomes unimportant. Certainly, some remaining diversity in the GA population can provide several alternative solutions, but selective pressure in a typical GA often eliminates this diversity.

The Niched-Pareto GA is a slightly modified version of the standard GA, in which a diverse, steady state population is maintained as the final outcome of the search process. Importantly, the end population is steered to find disparate solution families, each being "optimal" with respect to one or a subset of all of the optimality criteria. Effectively, this type of GA simultaneously solves the multi-objective optimization problem with respect to each and every criterion. *Population niching* based on optimality criteria means that the solution is not influenced by an *ad hoc* choice of optimization weights. Instead, final results encapsulate multiple disparate solutions, which can be maintained and selected by higher level (perhaps non-deterministic) supervisory planners. When niching is exploited to maintain a diverse population of solutions, other GA possibilities open themselves up. In particular, one has the opportunity to consider several objective criteria simultaneously, through the concept of Pareto optimality.

Pareto Optimization

As an illustrative example, consider the familiar task of selecting a personal computer for purchase. Typically, such a selection involves at least two criteria: performance and price. Assume that there are a large number of PCs available, and that a price and quantified performance measure (e.g. FLOPS) are available for each PC. How might one to proceed to select the optimal PC, in terms of these criteria ?

Two approaches present themselves immediately. One is to select the PC with the highest "bang per buck" (e.g., FLOPS/dollar). Another is to optimize on a weighted combination of the two criteria (e.g., to assign a dollar value to FLOPS). Both approaches involve combining the two performance measures into one. In practice, such approaches prove inadequate, due to the complex interaction of performance needs and economic constraints on the purchaser. In other words, performance and price are criteria of distinct types, which are not easily combined. Multi-objective optimization problems with irreconcilable objectives are commonplace. Ultimately, a human being, not an automated process, usually resolves such problems. However, one must consider what optimization concepts can help in making this decision.

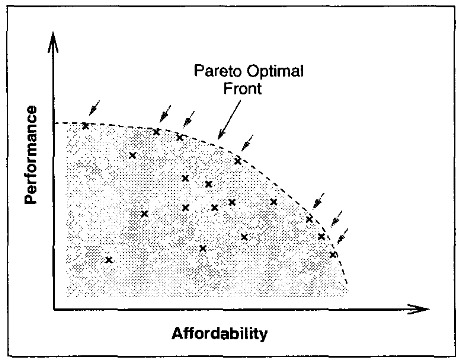

Figure 11: (Hypothetical) PC performance versus affordability (1/price).

From Figure 11, one can intuitively identify a limited set of points that should be considered as potential solutions (indicated by arrows), while eliminating the remaining alternatives. This is defined as the *non-dominated* solution set. Non-domination implies there are no other solutions that are superior in all criteria. When all possible solutions have been considered in such a problem, the complete set of non-dominated solutions is said to be on the *Pareto optimal front.*

The concept of Pareto optimization (that is, automatically searching for the Pareto optimal front) is particularly relevant to multi-spacecraft mission and task planning. Consider the previous formulation of objective criteria for the GA, which combined position/orientation endpoint errors; collision avoidance; minimization of path lengths; minimization of maneuvering times; minimization of fuel usage; and equalization of fuel loads across multiple spacecraft. In a Pareto optimization approach, each of these criteria could be considered separately (searching for a 6-dimensional Pareto optimal front), or in various, logically weighted combinations. Once the Pareto front is found, a mission planner could decide amongst the alternatives.

Using GAs in Pareto Optimization

A variety of GAs have been designed to address the Pareto optimization problem [7] The most promising schemes incorporate two features: a niched GA (to maintain population diversity) and a domination-based fitness function (to push the population towards the Pareto optimal front). Niching in a GA maintains a diverse population (while simultaneously searching for high-quality solutions) by using the biological analogy of resource sharing [5]. Simply stated, each individual in the population is forced to share its fitness with "neighboring" individuals. A GA designed in this fashion can be expected to drive a diverse population to the Pareto optimal front. Although there are careful considerations that must be taken into account in designing the exact operation of such GAs (for instance, the design of the sharing function), they show great promise at providing a set of Pareto optimal solutions, from amongst which a designer (e.g., mission planner) can select alternatives.

Clearly, an important area for future consideration in multi-spacecraft mission and task planning is Pareto optimization. Given its population basis, the GA is particularly well suited to this investigation.

7. Conclusions

This paper presents initial results of an investigation into using Genetic Algorithms for multiple spacecraft trajectory optimization. For the constellation initialization problem, a compact representation of point-to-point maneuvers was developed, which is directly applicable to thrusters and other saturation-type actuators. Preliminary results of applying standard GA algorithms to the N-spacecraft initialization problem have indicated the efficacy of GAs for global optimization. However, the multi-objective nature of this problem highlights the need for further development of techniques which adequately address the disparate and often conflicting nature of various mission and task objectives. The Niched-Pareto GA is identified as a candidate extension of the GA methodology, which is currently being used for development of a multiple-criterion global trajectory and resource optimizer for multiple spacecraft trajectory planning.

Acknowledgment

This research was supported by a NASA SBIR contract NAS5-99074 from the Jet Propulsion Laboratory (JPL). The authors would like to thank Drs. Fred Hadaegh and Mehran Mesbahi from NASA-JPL for their invaluable technical assistance and direction.

References

[1] Wang P.K. and Hadaegh F.Y. Coordination and control of multiple microspacecraft moving in formation. *Journal of the Astronautical Sciences*, 44(3), July-September 1996.

[2] Hadaegh F.Y. Wang P.K. and Lau K. Synchronized formation rotation and attitude control of multiple free-flying spacecraft. *AIAA-97-3753*, 1997.

[3] Ku W.M. Hadaegh F.Y. and Wang P.K. Adaptive control of formation flying spacecraft for interferometry. IFAC 1998, 1997.

[4] Pell *et al.* A remote agent prototype for spacecraft autonomy. In *SPIE Proceedings* Space Sciencecraft Control and Tracking in the New Millennium, volume 2810, August 1996.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[6] J. J. Grefenstette, L. Davis, and D. Cerys. *Genesis and OOGA: Two Genetic Algorithm Systems.* TSP, Melrose, MA, 1990.

[7] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3), 1999.