Evolutionary Computation Technologies for Space Systems

Richard J. Terrile, Christoph Adami, Hrand Aghazarian, Savio N. Chau, Van T. Dang, Michael I. Ferguson, Wolfgang Fink, Terrance L. Huntsberger, Gerhard Klimeck, Mark A. Kordon, Seungwon Lee, Paul von Allmen and Joseph Xu
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109
818-354-6158
Rich.Terrile@jpl.nasa.gov

Abstract—The Evolvable Computation Group,[1,2] at NASA's Jet Propulsion Laboratory, is tasked with demonstrating the utility of computational engineering and computer optimized design for complex space systems. The group is comprised of researchers over a broad range of disciplines including biology, genetics, robotics, physics, computer science and system design, and employs biologically inspired evolutionary computational techniques to design and optimize complex systems. Over the past two years we have developed tools using genetic algorithms, simulated annealing and other optimizers to improve on human design of space systems. We have further demonstrated that the same tools used for computer-aided design and design evaluation can be used for automated innovation and design. These powerful techniques also serve to reduce redesign costs and schedules.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. OVERVIEW 1
3. COMPUTER OPTIMIZED DESIGN 3
4. RESULTS 3
5. CONCLUSIONS 8
6. ACKNOWLEDGEMENTS 9
REFERENCES 9
BIOGRAPHY 10

1. INTRODUCTION

Complex engineering design problems are multi-parameter optimizations where physics models predict the outcome derived from a series of input parameters. Design, however, depends on desiring an outcome and deriving the necessary input parameters. Generally it is not feasible to invert the physics models to derive an optimal solution. Instead, by parallelizing the problem into a large population with varying input parameters and competing the results, we can extract favorable combinations of inputs. In the same way biological evolution functions, this process is repeated over many generations and uses the sophisticated biological operators of selection, mutation, and recombination to explore larger volumes of design space than could be examined by a human designer or by computational brute force (i.e., complete enumeration, exhaustive search, or other deterministic search algorithms).

The advantage of this approach is that it allows complex systems to be designed for and adapted to their environment (simulation or model) in the same way in which living and evolving systems adapt to natural environments. Knowledge of how the design should be optimized is not required. The necessary components are a good simulation of the design environment, a framework that allows evolution and knowledge of what requirements are desired (fitness). Competition and selection drive the systems to higher fitness and increased robustness. Computationally derived evolutionary designs have shown competitive advantages over human created designs in complexity, creativity and robustness. Our group has demonstrated this in the areas of power system design, low thrust trajectory optimization, robotic arm deployment path finding, MEMS micro-gyro calibration, mission planning and scheduling, neural network design, and avionics architecture design. We have also developed a framework for the rapid introduction and parallelization of optimization problems in an evolutionary environment using computer clusters.

These techniques offer an alternative approach to system engineering of complex systems by the imposition of design rules (Figure 1). Whereas this has been a successful approach for hardware systems that can rely on physics, mathematics, material science etc. as their foundation, software systems have largely failed to improve in robustness by the imposition of new design rules. The approach of evolutionary computation uses the same principles of variation and selection that have been so successful in the development of natural biological systems.

2. OVERVIEW

Evolutionary Computation Framework

The strength of evolutionary computation comes from the ability to utilize existing computer models and simulations that predict the results of multiple input parameters. These types of models are now common elements of computer-aided design (CAD) as well as scientific modeling and

[1] 0-7803-8870-4/05/$20.00© 2005 IEEE
[2] IEEEAC paper #1257, Version 26, Updated December 1, 2004

forecasting. In evolutionary computational techniques, a population of these models is created and input parameters are varied. The results are evaluated using fitness functions and a percentage of the highest fitness individuals from one generation is promoted to the next generation, while new models are created through variation (e.g., by mutation or cross-over).

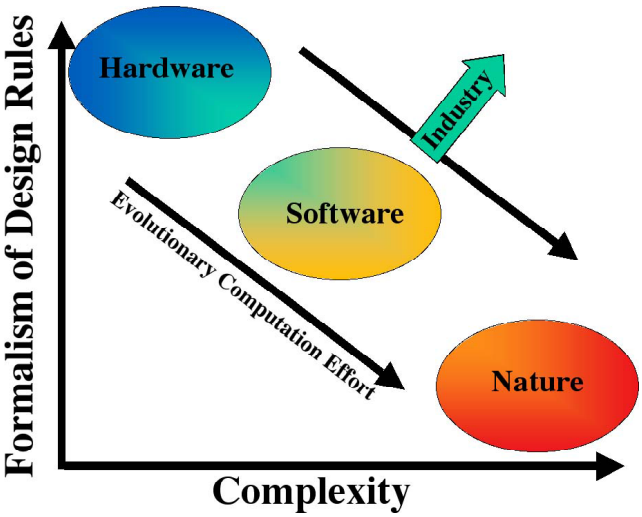


Figure 1. Complexity and Design Rules. As complexity increases from hardware systems to software to nature, the formalism and number of design rules decreases. Current efforts in software engineering are trying to move software systems to higher formalism. Our effort tries to explore the creation of complexity by removing formalism and using a biological evolutionary approach.

In order to rapidly adopt new problems into an evolutionary framework we have developed a graphical user interface that enables the parallel operation of genetic algorithms and other optimizers on a cluster computer. This framework enables the tie-in of a variety of different applications and the management of application inputs and outputs. The operation of a parallel genetic algorithm to optimize a variety of tasks has been demonstrated. This evolvable software system also includes a variety of algorithms useful for optimization such as local searches and simulated annealing. We have baptized the framework "Parallel, Evolvable, and Revolutionary Synthesis and Optimization Environment (PERSON)". The computational architecture of PERSON is based on a package created by D. Levin at Argonne National Lab [1]. The major contribution of PERSON is the scriptable front-end that turned the software into a tool. We have a scripted approach to the integration of applications into PERSON, which enables rapid integration of a variety of different physical models and a variety of different fitness functions without recompilation of the design environment. This architecture resembles a generalized optimization toolkit that can be applied to a large array of physical system simulation problems that require optimization in a huge search space. Note, however, that this system goes beyond the traditional optimization approach in that truly novel synthesis and design/selection may be performed by the genetic principles that govern the evolvable system algorithms. The software framework development for this evolvable system effort must be performed by an experienced team of software engineers, computer scientists, electrical engineers, biologists, and physicists in order to provide any hope for a reasonable outcome compared to the high mark targets.

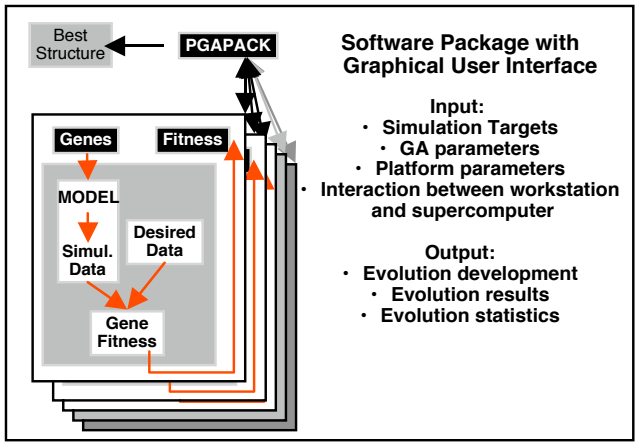


Figure 2. PERSON graphical evolutionary computational framework. Models can be introduced with existing tools and simulators and are adapted into a parallel evolutionary environment.

Genetic Algorithms

Genetic algorithms (GA), first introduced by John Holland and his colleagues [2], are search algorithms based on the mechanics of natural selection and sexual reproduction. GAs are theoretically and empirically proven to provide robust search in complex parameter spaces. Furthermore, they are not fundamentally limited by restrictive assumptions about the search space such as continuity and existence of derivatives.

The standard GA proceeds as follows: A possible solution of a given problem is encoded as a finite string of symbols, known as a *genome*. An initial population of the possible solutions called *individuals* is generated at random or heuristically. At every evolutionary step, known as a *generation*, the individuals in the current population are decoded and evaluated according to some predetermined quality criterion, referred to as the *fitness*. To form the next generation, parents are selected with a probability proportional to their relative fitness. This ensures that the expected number of times an individual is chosen is approximately proportional to its relative performance in the population. Thus, low-fitness individuals are more likely to disappear.

The parent selection process is followed by genetically inspired operators to form offspring. The most well known operators are mutation and crossover. The mutation operator is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space with some probability. Crossover is performed with a different probability between two selected parents, by

exchanging parts of their genomes to form two offspring. In its simplest form, substrings are exchanged after a randomly selected crossover point. This operator tends to enable the evolutionary process to move toward promising regions of the search space quickly, by recombining partial solutions. Genetic algorithms are stochastic, iterative processes that are not guaranteed to converge. The termination condition may be specified as some fixed, maximum number of generations or as the attainment of an acceptable fitness level. Further discussions of GAs have been published by this research group with respect to algorithm details and their application to nanoelectronic device designs [3], microelectronic device designs [4], automated circuit designs [5], quantum mechanical basis set selection [6], space craft power system design [7], low thrust orbit transfers [8], and neural network evolution [9].

This publication is focused on the application of efficient optimization techniques to a variety of space science systems. The method of operation is the repeated application of already sophisticated physics based models which predict reality. We would like to mention here that the availability of efficient optimization tools and the ability to explore large parameter spaces also enables the development of more sophisticated physics based models that predict "reality" better. One such concrete example is the development of an advanced model to treat the consequences of arbitrary mechanical strain distortions in semiconductor crystals [10]. This advanced model expands the physical parameter space dramatically, which could not have been usefully explored without the PERSON framework. GAs can therefore not only seek better engineering solutions, they can also help to refine our physical understanding of problems.

Simulated Annealing Algorithms

Simulated annealing (SA) is a widely used and well-established optimization technique especially for high-dimensional configuration spaces [11,12]. The goal is to minimize an energy function E (in the following three cases: 1) the squared distance from a start joint angle set to a target joint angle set for the rover arm; 2) the required flight time and propellant mass for low thrust trajectories; and 3) the frequency split of the MEMS micro-gyro), which is a function of N variables (in the three following cases: 1) the 4 joint angles for FIDO (Field Integrated Design and Operations) or the 5 joint angles for the MER (Mars Exploration Rover) rover platforms; 2) the Q-law parameters; and 3) the 4 bias voltages for the MEMS micro-gyros), with N being usually a large number. The minimization is performed by randomly changing the value of one or more of the N variables and reevaluating the energy function E. Two cases can occur: 1) the change in the variable values results in a new, lower energy function value; or 2) the energy function value is higher or unchanged. In the first scenario the new set of variable values is stored and the change accepted. In the second scenario, the new set of variable values is only stored with a certain likelihood (Boltzmann probability, including an annealing temperature). This ensures that the overall optimization algorithm does not get stuck in local minima too easily (greedy downhill optimization). The annealing temperature directly influences the Boltzmann probability by making it less likely to accept an energetically unfavorable step, the longer the optimization lasts (cooling schedule). Then the overall procedure is repeated until the annealing temperature has reached its end value, or a preset number of iterations has been exceeded, or the energy function E has reached an acceptable level.

3. Computer Optimized Design

We have demonstrated that evolutionary computational techniques can now be used for automatic innovation and design, using the same computer models that are employed to evaluate engineering designs. Five areas, described in this paper, demonstrate human competitive performance as described by Koza et al. [13]. Optimizations and designs using evolutionary techniques result in designs matching or exceeding the performance of those derived from traditional means by human designers. Metrics used for this performance evaluation include design time, robustness and fault-tolerance, cost, and comparison to accepted and flown designs. The areas described in this paper are the automatic design of power systems, robotic arm deployment path planning, the design of low-thrust trajectories, the automatic tuning of MEMS micro-gyroscopes, and the evolution of neural networks. We expect that future work will lead to further advances in computational engineering and in the development of Computer Optimized Design (COD) (Figure 3.).

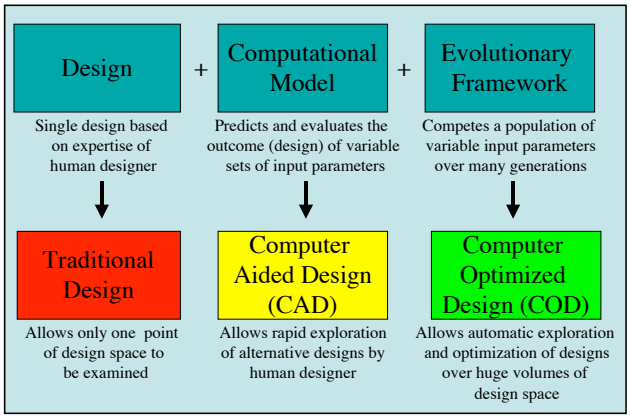


Figure 3. Elements of Computer Optimized Design (COD). The use of an evolutionary framework coupled to a computational simulation allows the extension of computer aided design to rapidly and automatically evaluate huge volumes of design space.

4. Results

Automatic Design of Power Sub-systems

Early in the formulation phase of a flight project the goals and objectives of the mission are defined and several

plausible mission concepts are created. These preliminary mission concepts will trade-off various elements in the design so that project managers can choose between different alternatives for mass, cost, performance and risk. This cycle of goal definition, mission concept creation and design trade study is repeated many times with each pass refining and improving the resolution of the design. The product of this process is a mission architecture characterized such that its effectiveness in achieving mission objectives can be properly evaluated.

More formally, a trade study is a process for seeking one or more optimal solutions when there are multiple, often conflicting, objectives. An optimal solution in this case means that if one objective improves, other objectives are compromised or traded off. The classic example of this is in car buying. Buyers must make a decision between cost and comfort since the less expensive cars are inevitably less spacious. This hypothetical trade-off is shown in Figure 4.

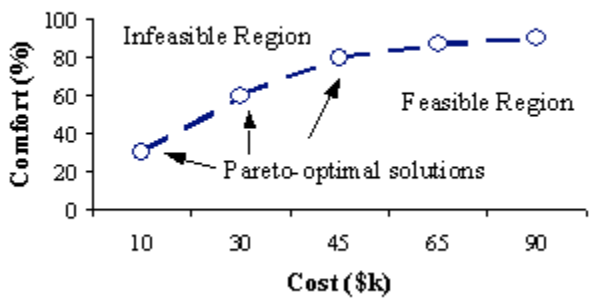


Figure 4 Hypothetical Car Buying Trade-Off [14].

To make the best decision, the buyer would want to consider solutions that are evenly distributed along the Pareto-optimal front. While the classical weighted sum approach is intuitive and easy to implement, creating these solutions using this method poses some problems. The most obvious is that since only one solution is generated at a time, users will need to perform multiple runs in order to obtain a set of possible solutions. More importantly, these solutions may not be evenly distributed, as illustrated in Figure 5. Furthermore, since evolutionary algorithms are stochastic the method may find an optimal solution only to lose it in later generations. Ideally, we would want to generate all of the solutions in one run, evenly distributing them first and then mapping them to the weightings afterward, as shown in Figure 6. Moreover, we would want to preserve the best solutions from both the parent and child populations. One way to achieve this is with a Niched-Elitist approach.

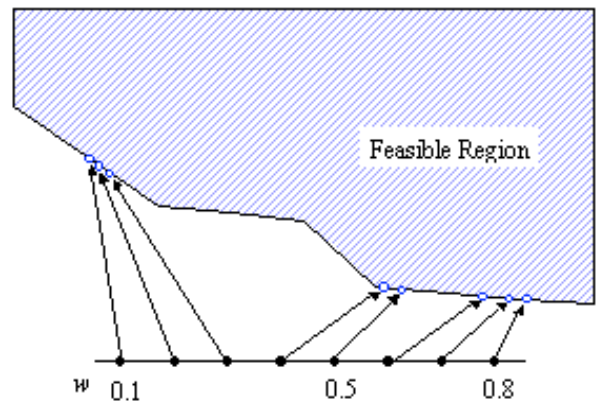


Figure 5. Weighted Sum Approach [14]

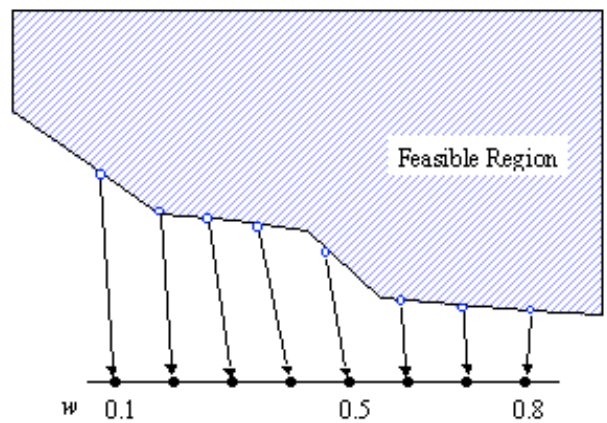


Figure 6. Ideal Approach [14].

This approach uses the same reproductive cycle as described in the Introduction. The main difference is that niches are used to help evaluate and replace the population members.

In evolutionary computing, niching refers to forming artificial subpopulations in the population to emphasize and maintain multiple solutions. This can be done in either the objective or parameter space. Since we wanted to maintain diversity in the solutions to trade-off regardless of how similar or different they are, we elected to niche by the objective space.

To evaluate the fitness of a particular design and implement the evolutionary cycle, we used a JPL power analysis tool called MMPAT within the PERSON evolutionary computing framework. MMPAT (Multi-Mission Power Analysis Tool) is a tool that models the behavior of a spacecraft's power sources and energy storage devices as they interact with the spacecraft loads and the environment over a mission timeline. It is currently used in Mars Exploration Rover (MER) operations to predict the power subsystem resources before a sequence of activities is uploaded. By using this tool in an evolutionary computing framework we were able to provide a set of optimized designs based on the anticipated performance of the subsystem rather than using worst-case estimates.

To verify the utility of this approach we used the mission plans from some current JPL missions and generated several alternative designs for each. One of these tests was an optimization of one of the Mars Exploration Rovers (MER), a NASA mission of two rovers that landed in January 2004.

To setup the analysis we gave the PERSON optimization framework some initial design parameters and a valid range of values. We also needed to define the cost and mass constants and create an activity plan. For the MER optimization, we varied the number of cells per string and the number of strings per segment for the six solar array segments, as well as the battery capacity. The PERSON framework chose the initial population based upon a random draw over a uniform distribution for each of the variable power subsystem design parameters before invoking MMPAT. As a starting point, the framework was instructed to use the actual MER rover solar array size and battery capacity.

The rover was placed at 14.95 degrees south latitude and given an activity plan that lasted 90 sols. This corresponds to the planned length of surface operations of MER-A at the Gusev Crater landing site. The activity plan consisted of applying a 50-watt load for six hours during local daytime and 8 watts the rest of the day. This simulated the load on the rover while it performed its duties during the day, and let it conserve battery power for the heaters at night.

Using a population of 200 we ran the analysis for 177 generations. This resulted in 35400 unique designs being evaluated. The optimization took 18 hours using 8 Intel(R) Xeon(TM) CPU 2.80GHz processors. As expected, this resulted in several credible alternative solutions being generated where each niche optimized their primary objective while compromising the others.

Rover Arm Path Planning

Current and future planetary exploration missions involving landers and/or rover-type vehicles such as Mars Exploration Rover (MER), Mars Science Laboratory (MSL), and subsurface access missions are and will be equipped with robotic arms with four or more joints, each joint having a high degree of freedom (e.g., an angle range of 100 degrees in 1 degree steps). Fast and efficient safe rover movement (e.g., legged rovers and cliff-climbing rovers) and rover arm deployment algorithms, taking rover position and surrounding ground obstacles (e.g., rocks) into account that can be executed with onboard CPU power, will tremendously enhance mission autonomy by cutting down on up-/downlink events, and thus increase the useful lifespan of a mission. This work is capable of increasing science return of future missions and enabling support of intelligent in-situ science experiments to be performed autonomously.

The calculation of a collision-free rover arm deployment path is a search in a high-dimensional configuration space. A rover arm consisting of N joints with, e.g., on average 100 angle positions per joint, spans a configuration space of $10^{(2*N)}$. With N>6, the number of possible configurations lies beyond exhaustive search in a timely manner. To increase the degree of complexity even more, the rover arm deployment requires the generation/calculation of a series of valid configurations, i.e., the safe arm deployment path.

We have created three separate software programs using a *modified simulated annealing algorithm*: 1) calculation of a safe, collision-free rover arm end configuration given a predetermined x-y-z end position of the instrument-carrying joint together with a surface normal at that point; 2) calculation of a safe, collision-free deployment path from a start rover arm configuration into the pre-calculated end configuration (1); and 3) optimization of safe, collision-free rover arm deployment path with respect to minimizing the overall absolute joint angle movement. The software is written in standard C and thus requires no special computing platform. It runs under Linux, Unix, Windows, DOS, and Mac OS X and requires less than 1.25MB of RAM. The time necessary to calculate a safe deployment path is now reduced from hours to hundreds of milliseconds (on a Macintosh PowerBook 800MHz G4).

We also applied a genetic algorithm [15] to the rover arm path planning problem, trying to mimic evolutionary principles used by nature in applying genetic operations such as point mutation, cross-over, and inversion to parameter strings, called “genes” or “chromosomes”, in order to evolve a set of parameters that achieve a high fitness as determined by a fitness function (similar to the energy function in simulated annealing). Our improved algorithm for the safe rover arm deployment problem uses the following seven-step process:

1. Start with a random initial population
2. Determine arm extent bounding volume to prune the search space
3. Define fitness function based on obstacle avoidance and goal orientation
4. Perform collision detection and prune population followed by goal orientation
5. Perform mutation with a probabilistic choice of small variation in state or segmented path mutation followed by another fitness evaluation
6. Promote top 10% of the survivors to the next arm extent bounding volume
7. Repeat steps 2 to 6 until placement point reached.

This approach will give an incremental path to the goal position, without having to search through the entire path from start position to end.

The simulated annealing-based rover arm path planning algorithm as well as the GA-based algorithm have been successfully tested both onboard the FIDO rover platform and on the FIDO software simulator at JPL (see Figure 7). The optimizer part came up with a novel, shortest (2-step) deployment path from the stowed to the safe rover arm position.

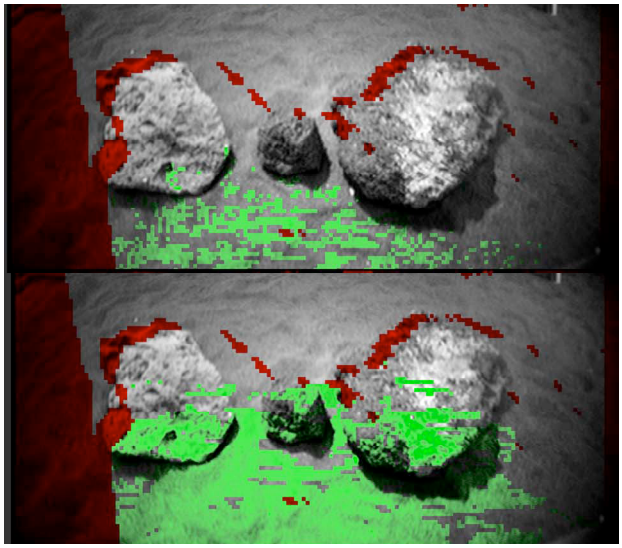


Figure 7. Reachability Map for FIDO Rover: Comparison of digital terrain maps showing reachability of targets with the FIDO robotic arm. Green (light) areas are reachable with arm path solutions. Grey areas are not reachable and red (dark) areas indicate no data available for a solution. Top image is default elbow up reachability derived from the FIDO arm path algorithm. This algorithm derives an un-optimized safe path to each target. The bottom image is the same terrain map analyzed with a genetic algorithm to find the safest paths to targets. The larger reachable area of the genetic algorithm is an indication of the power of the technique in not only providing a greater number of targets, but also in providing the fittest arm path solutions with respect to safety from arm (self and terrain) collisions.

We are in the process of deploying both algorithms on the MER rover software simulator and hardware platform, with possible real-world arm deployments during the extended NASA/JPL MER Mars Mission. Details of this work will be published in a future publication.

Optimization of Low-Thrust Trajectories

Future space missions DAWN and JIMO will use electric propulsion for inter-planetary cruise and orbital operations. The strength of electric propulsion is that in spite of its low thrust levels, the momentum transfer to the spacecraft per kilogram of expelled propellant is ten or twenty times greater than for chemical propulsion. However, the control of low-thrust spacecraft poses a challenging design problem because perturbative forces often dominate the thrust and a significant change of the orbit requires many revolutions. Here we address the problem of designing low-thrust orbit transfers between arbitrary orbits in an inverse-square gravity field by using evolutionary algorithms to drive parameter selection in a Lyapunov feedback control law (the Q-law).

The general goal of the design problem is to maneuver a spacecraft with a series of thrust arcs from orbit A to orbit B in the most fuel-efficient and simultaneously time-efficient manner. Since the fuel efficiency and the time efficiency often conflict, the goal of this design problem becomes to determine the Pareto front, which is the envelope in the objective space resulting from the trade-off between the optimal propellant mass and the flight time; each point along the Pareto front corresponds to one particular mission scenario. In order to access the Pareto front with reasonable accuracy and to provide the time history of the state variables and the thrust vector for any chosen point of the Pareto front, we have developed an efficient and efficacious method. A search for the Pareto-optimal trajectories is performed in two stages: 1) optimal thrust angles and thrust-arc locations are determined by the Q-law, and 2) the Q-law is optimized with two evolutionary algorithms: a modified simulated annealing algorithm (SA) and a genetic algorithm (GA) with non-dominated sorting [8].

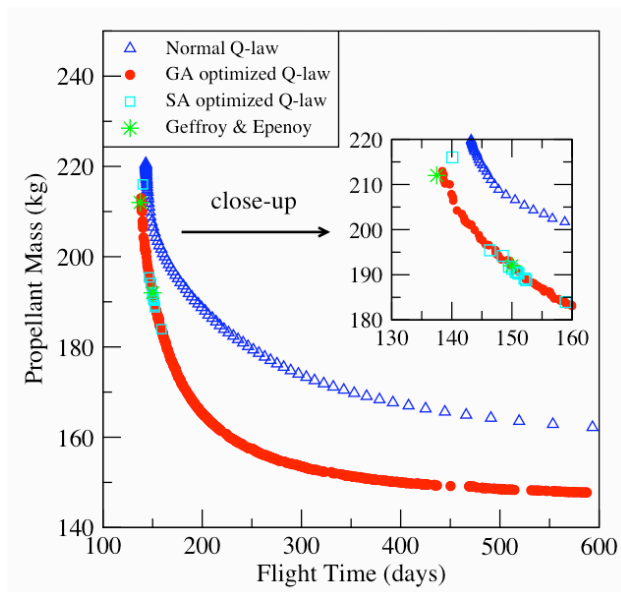


Figure 8. Pareto front for an orbit transfer from a slightly-inclined geostationary-transfer orbit to a geostationary orbit.

We applied our method to several types of orbit transfers around the Earth and the asteroid Vesta. Substantial improvements in both final mass and flight time over state-of-the-art are found in the calculation of the Pareto front. For example, for a low-thrust orbit transfer from a slightly-inclined geostationary-transfer orbit to a geostationary orbit we have obtained as much as a 15% propellant savings over the nominal Q-law. Furthermore, the resulting Pareto front contains the optimal trajectories found by other optimization algorithms such as a static/dynamic control algorithm [16] and an orbit averaging technique [17]. Figure 8 shows the substantial improvement in the estimation of the true Pareto front by the optimized Q-law with SA and GA over the nominal Q-law, and the comparable

performance of the optimized Q-law to other optimization techniques. Even more promising is that our method builds the Pareto front within a few hours of computation time, while other optimization algorithms require a comparable computational effort to acquire a single optimal trajectory. A more detailed description of our method and results is reported elsewhere [8].

Future plans comprise the direct optimization of low-thrust trajectories, i.e., determination of sequence of thrust arcs, both in space and time, and individual duration thereof, independent of human-prescribed control laws such as Q-law.

Automatic Tuning of MEMS Micro-Gyroscopes

The MEMS Micro-Gyro, developed by the MEMS Technology Group at JPL, is subject to an electro-static fine-tuning procedure, which is necessary due to unavoidable manufacturing inaccuracies. In order to fine-tune the gyro, 4 bias voltages, applied to 8 capacitor plates, have to be determined independently within a range of –60V to 15V. The fine-tuning directly correlates with the accuracy of the gyros in later use.

In order to fully automate the time-consuming (on the order of several hours) manual fine-tuning process, we have established a hardware/software test bed to the existing manual gyro-tuning hardware-setup using commercial-off-the-shelf (COTS) components, which includes four programmable power supplies, one offset power supply, and an (electronic) signal analyzer as well as driver and analyzing software.

We developed and implemented two algorithms for efficiently determining the bias voltages: 1) a modified *simulated annealing algorithm* and 2) a *dynamic hill-climbing algorithm*. Both have been incorporated into the hardware/software test bed. We were subsequently able to successfully fine tune both MEMS post-gyros and MEMS disk-resonating gyros within one hour for the first time fully automatically to a level of accuracy that is equal to or better than what can be accomplished manually (see Figure 9).

One of the key problems solved during the course of this research was that of determining the resonant frequencies along both axes of oscillation. In this closed-loop system, the resonant frequencies were determined by scanning through the range of likely frequencies to determine two peaks in the amplitude of vibration. The objective is to reduce the difference in resonant frequencies, also called the "frequency-split", to zero. The frequency split before tuning can be seen in Figure 10. The resonant frequencies are determined by fitting the data to two Lorentzian curves. The best-fit curves are seen in the inset. The fit parameters tell us the position of the peak, and hence the resonant frequency. Using this method we can accurately report the frequency split to a resolution below 0.06Hz, which is considerably better than the resolution determined by a human operator. The final tuned result can be seen in Figure 11.

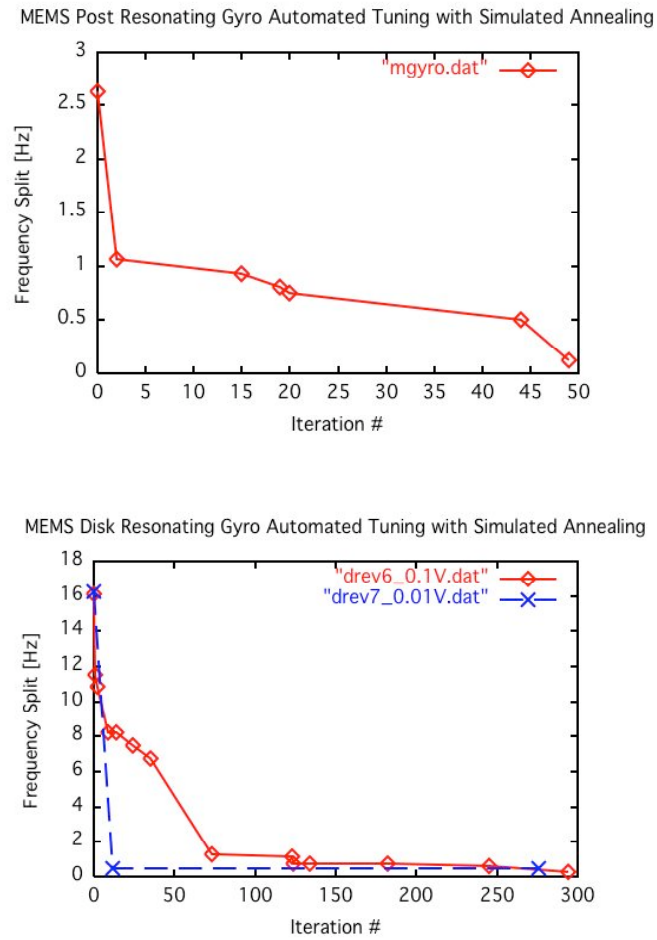


Figure 9. Frequency split as a function of Simulated Annealing Iterations: (top) for the MEMS post-gyro; (bottom) for the MEMS disk-resonating gyro.

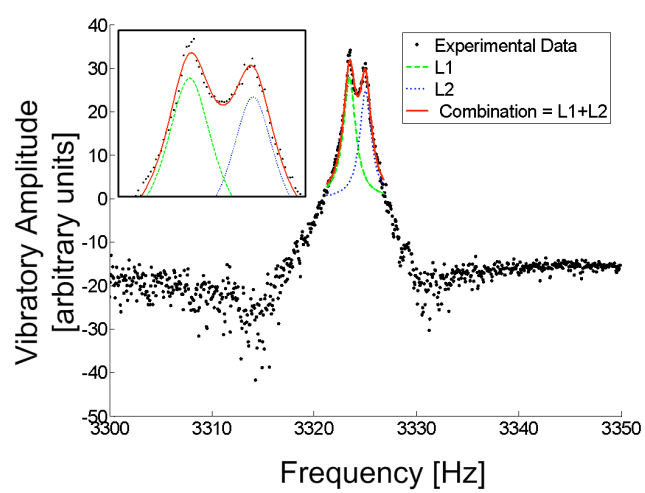


Figure 10. The frequency split before tuning. The two Lorentzian curves are shown, as dashed and dotted lines below the solid line indicating the sum of the curves. The inset shows the details of the peak data points.

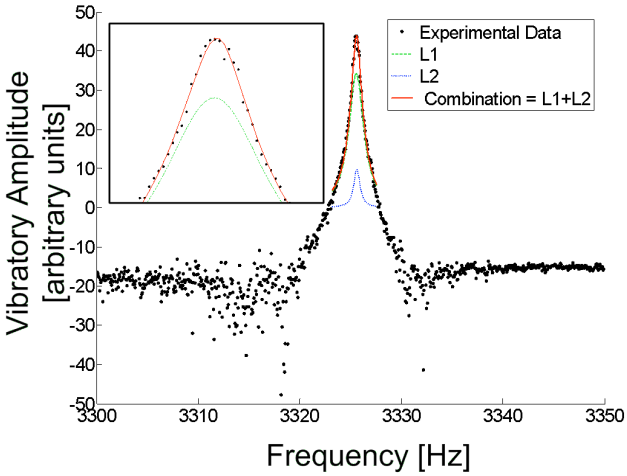


Figure 11. The frequency split after tuning is shown, reduced to approximately 0.05Hz.

The novel capability of fully automated gyro tuning enables ultra-low mass and ultra-low-power high-precision Inertial Measurement Unit (IMU) systems to calibrate themselves autonomously during ongoing missions, e.g., Mars Ascent Vehicle.

Evolution of Neural Networks

Spacecraft and rovers are conventionally controlled by software systems that determine attitude, position, direction, and speed as a function of environmental variables. To some extent, such controllers perform the same function as the nervous system of animals, a standard example being the fruit fly *Drosophila melanogaster*. Because of the analogy to nervous systems, robotic controllers can also be cast in the language of neural networks, which perform the computation of output variables as a function of input variables. Historically, neural networks have been designed according to standard rules from abstract neurons connected to each other. However, artificial brains based on standard neural network architectures failed to deliver on the initial promise, and have faded from use. We have started a research program in which not only the structure of neural networks is being evolved, but also the rules for growth and learning, within a software environment called "Norgev" (Neural Organism Evolution) [18]. This program is proceeding in four steps:

1. Proof of principle by evolving networks that perform complex logical input/output functions in a robust manner exceeding human-design standards

2. Evolution of neural networks that control simulated test robots (Khepera) better than human-designed controllers

3. Transfer and optimization of evolved controllers to real test robots

4. Application of technique to mission rovers and spacecraft.

We have finished step one, and were able to demonstrate the evolution *de novo* of sequences that encode extremely robust networks that perform complex logical functions. The structure of these networks (Figure 12) is unlike any designed by humans, and resembles instead the decentralized structure of animal nervous systems, such as that of the flatworm *Caenorhabditis elegans*. The network in Figure 12, because it is derived from a growth process, is reconstituted automatically even if more than half of the cells that make up its tissue are removed [9]. We are now proceeding with step 2, by evolving Khepera controllers within the Webots™ 4 simulation platform linked to the Norgev software.

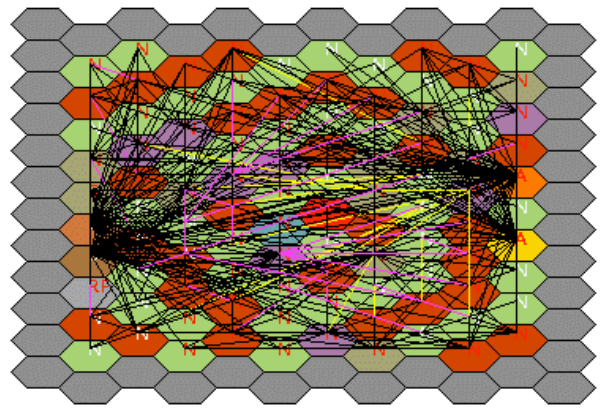


Figure 12: Evolved complex computational tissue solving a logical 3-input/2-output task [9]. Neurons are colored according to their level of expression of specific simulated chemicals. Connections are colored according to their activity.

.

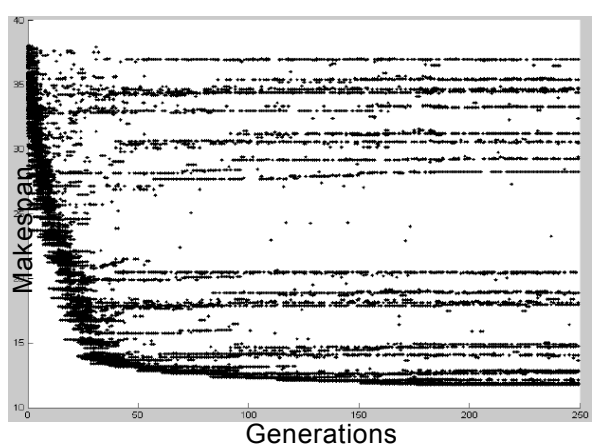


Figure 13: Plot of viable schedule at each generation and their makespan

Mission Operations Planning and Scheduling

The Mission Operations Planning and Scheduling problem is comprised of a large number of events that may be scheduled if the resources and constraints permit in an allotted time period, also referred to as the planning horizon. The goal of this study is to use the genetic algorithm (GA) technique to find the schedule that would globally minimize the amount of time it takes for all events to complete within the planning horizon and also to enable more high priority events to be scheduled. It was found that the GA technique alone does not enable one to achieve global optima. However, it is capable of quickly finding a population whose spans are sufficiently close to the global optima (Figure 13). In contrast, it is also observed that sequential quadratic programs (SQP) such as those found in Matlab using FMINCON are able to find the global optima more rapidly but require the search to start with a set of viable initial schedules. Therefore, the GA technique can be used in conjunction with SQP to find the whole set of globally optimal solutions such that the solutions from the GA can serve as viable initial guesses for the SQP. This coupling of GA with SQP can produce the global optima more quickly and accurately than just using one of the techniques alone. Details of this work will be published in another publication in the future.

Architecture Synthesis Tool

The purpose of the Architecture Synthesis Tool (AST) is to automatically generate functionally viable architectures for spacecraft Command and Data Handling (C&DH) subsystems that will adapt to frequently changing system requirements during the early phase of a flight project. These changes in requirements frequently lead to weeks of labor spent on redesign, and it would be highly beneficial to have a tool that is sensitive to such requirement changes and able to respond to these changes with viable architectures in a matter of hours.

The AST consists of an Architecture Generator and a Functionality Evaluator. The Architecture Generator employs genetic algorithms to generate populations of C&DH architectures and the Functionality Evaluator uses the availability of the components required by system functions to evaluate the fitness of these architectures. In preliminary tests, the AST has produced viable architectures that were optimal within the design space defined by the database inputs. Details of this work will be published in another publication in the future.

5. Conclusions

We have demonstrated that evolutionary computational techniques can be applied to the design and optimization of space systems. Generally, these applications offer better performance (in the range of at least 10%) than traditional techniques and show faster design times. Additionally, changing fitness requirements and redesign, which inevitably occurs in real systems and generally causes great fiscal and schedule disruption, can be accommodated at relatively low cost.

Our future work will consider the optimization of multiple sub-systems into full spacecraft optimizations. We are also evolving mission plans and schedules and hope to integrate this work into the co-design of a spacecraft optimized to the mission plan.

6. Acknowledgements

The work described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The research was supported by the JPL Research and Technology Development Program. Leveraged funding was provided by NASA ESTO-CT, ARDA, and ONR.

References

[1] D. Levin, Argonne National Lab, http://www-fp.mcs.anl.gov/ccst/research/reports_pre1998/comp_bio/stalk/pgapack.html

[2] J.H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, 1975.

[3] Gerhard Klimeck, Carlos H. Salazar-Lazaro, Adrian Stoica, and Tom Cwik, "Genetically Engineered" Nanostructure Devices, in "Materials in Space Science, Technology, and Exploration", MRS Symposium Proceedings, Vol. 551, pg. 149 (1999).

[4] Tom Cwik and Gerhard Klimeck, "Integrated Design and Optimization of Microelectronic Devices", Proceedings of 1999 Aerospace Conference, IEEE Volume: 5 , 1999 , Page(s): 131 -138.

[5] Didier Keymeulen, Gerhard Klimeck, Ricardo Zebulum, Adrian Stoica, and Carlos Salazar-Lazaro, "EHWPack: A Parallel Software/Hardware Environment for Evolvable Hardware", in Whitley Darrell (eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), July 8-12, 2000, Las Vegas, Nevada USA. San Francisco, CA: Morgan Kaufmann.

[6] Gerhard Klimeck, R. Chris Bowen, Timothy B. Boykin, Carlos Salazar-Lazaro, Thomas A. Cwik, and Adrian Stoica, "Si tight-binding parameters from genetic algorithm fitting", Superlattices and Microstructures, Vol. 27, No. 2/3, Mar 2000, pp. 77-88.

[7] M. Kordon, G. Klimeck, D. Hanks and H. Hua, "Evoluionary Computing for Spacecraft Power Subsystem Design Search and Optimization," IEEE Aerospace Conference Proceedings, Big Sky, MT., March 2004.

[8] S. Lee, P. von Allmen, W. Fink, A. Petropoulos, R. Terrile, "Design and Optimization of Low-thrust Orbit Transfers Using the Q-law and Evolutionary Algorithms" 2005 IEEE Aerospace Conference Proceedings, March 2005.

[9] A.N. Hampton and C. Adami, "Evolution of Robust Developmental Neural Networks", Proc. of *Artificial Life IX,* Boston, MA, Sep 12-15, 2004. J. Pollack, M.A. Bedau, P. Husbands, T. Ikegami, and R. Watson, eds., (Boston: MIT Press, 2004) pp 438-443.

[10] Timothy B. Boykin, Gerhard Klimeck, R. Chris Bowen, and Fabiano Oyafuso, "Diagonal parameter shifts due to nearest-neighbor displacements in empirical tight-binding theory", Phys. Rev. B 66, 125207 (2002).

[11] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, "Equation of State Calculation by Fast Computing Machines*," J. of Chem. Phys*., **21**, 1087–1091, 1953.

[12] S. Kirkpatrick, C.D. Gelat, M.P. Vecchi,, "Optimization by Simulated Annealing," *Science*, **220**, 671--680, 1983.

[13] J.R. Koza, F.H. Bennett III, D. Andre, M.A. Keane "Genetic Programming III: Darwinian Invention and Problem Solving," 1999, Morgan Kaufmann Publishers.

[14] D. Kalyanmoy *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd., 2001, p. 2, 172, 239, 173

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

[16] A.E. Petropoulos, "Simple Control Laws for Low-Thrust Orbit Transfers," *AAS/AIAA Astrodynamics Specialist Conference,* AAS Paper 03-630, 2003.

[17] G.J. Wiffen and J.A. Sims, "Application of a Novel Optimal Control Algorithm to Low-Thrust Trajectory Optimization," AAS/AISS Space Flight Mechanics Meeting, AAS Paper 01-209, 2001. [17] S. Geffroy and R. Epenoy, "Optimal Low-Thrust Transfers with Constraints – Generalization of Averaging Techniques," *Astronautica Acta*, 41, 1333-149, 1997.

[18] J. Astor and C. Adami, "A Developmental Model for the Evolution of Neural Networks". *Artif. Life* 6, 189-218, 2000.

BIOGRAPHY

Richard J. Terrile *created and leads the Evolutionary Computation Group at NASA's Jet Propulsion Laboratory. His group has developed genetic algorithm based tools to improve on human design of space systems and has demonstrated that computer aided design tools can also be used for automated innovation and design of complex systems. He is an astronomer, the Mars Sample Return Study Scientist, the JIMO Deputy Project Scientist and the co-discoverer of the Beta Pictoris circumstellar disk. Dr. Terrile has B.S. degrees in Physics and Astronomy from the State University of New York at Stony Brook and an M.S. and a Ph.D. in Planetary Science from the California Institute of Technology in 1978.*

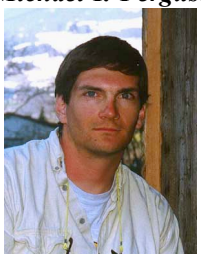
Christoph Adami *is a Visiting Scientist and former Principal Scientist in the Exploration Systems Autonomy Section of NASA's Jet Propulsion Laboratory, and a Professor of Applied Life Sciences at the Keck Graduate Institute in Claremont, CA. His interests concern the fundamental mechanisms involved in the evolution of complexity from simplicity, both in natural and artificial systems. He has pioneered the use of digital life in computational evolutionary biology, and is the author of over 70 peer-reviewed articles in theoretical physics and biology, as well as the textbook "Introduction to Artificial Life" (Springer). He has a B.A. and a Diplom in theoretical physics from Bonn University (Germany), as well as an M.A in physics and a Ph.D. in theoretical physics from SUNY Stony Brook.*

Hrand Aghazarian *is a Senior Member of the Engineering Staff in the Mobility and Robotics Section at the Jet Propulsion Laboratory in Pasadena, CA. He has extensive work experience in the design and development of real-time embedded software for a variety of avionics systems. Currently he is involved in providing software solutions in the area of software architecture, low-level drivers, motor control, user interfaces for commanding planetary rovers, and navigation algorithm for SRR and FIDO rovers. His research interests are Rover Navigation/Control and Real-Time Embedded software Design and Development. He received a dual B.S. degree in Applied Math and Computer Science and a M.S. degree in Computer Science from California State University in Northridge. He is a member of ACM and INNS.*

Savio Chau *is a principal engineer and group supervisor at the Jet Propulsion Laboratory of the California Institute of Technology. He has 24 years of industrial experience in both research and development. He is currently the lead engineer of the Command and Data Handling Subsystem of the Jovian Icy Moon Orbiter project. He is also leading two technology development projects in the applications of Genetic Algorithm and the advanced data bus architecture for spacecraft. He has been a lecturer in the Computer Science Department at the University of California, Los Angeles. He current research interests include modular, high performance, intelligent, and ultra long life flight systems that utilize techniques from reconfigurable and evolvable hardware, neural network, genetic algorithms, fuzzy logic, and other biology-inspired techniques. He has 4 refereed journal publications, 20 conference papers, 2 patents, and 1 pending patent. He was the program co-chair of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC2002). He earned his BS in electrical engineering from Loyola Marymount University, Master of Engineering from California State Polytechnic University, Pomona, and Ph.D. in computer science from University of California, Los Angeles. He is a member of Tau Beta Pi and Eta Kappa Nu.*

Van T. Dang *is an Associate Engineer in the Avionics Research Technology and Development Group at JPL. She holds both a B.A. and an M.S. in Applied Mathematics from Scripps College and Claremont Graduate University, respectively. Her mathematical interests include optimization, numerical analysis, and statistical analysis. She has worked at Beckman Coulter Inc. as a math analyst prior to this and hence has accumulated interests in modeling biological and chemical interactions.*

Michael I. Ferguson *is a member of the Technical Staff in the Bio-Inspired Technologies and Systems group. His focus is on evolutionary algorithm application to VLSI design and arithmetic algorithms. He is currently working on the application of GA to tuning MEMS micro gyros. His other projects include evolution of digital and analog circuits intrinsically and extrinsically using a variety of methods. He was the Local Chair of the 2003 NASA/DoD Conference on Evolvable Hardware. He received a M.S. in Computer Science from University of California at Los Angles and a B.S. in Engineering Physics from the University of Arizona e-mail:.fergie@jpl.nasa.gov.*



Wolfgang Fink *is a Senior Researcher at NASA's Jet Propulsion Laboratory, Pasadena, CA, Visiting Research Assistant Professor of both Ophthalmology and Neurological Surgery at the University of Southern California, Los Angeles, CA, and Visiting Associate in Physics at the California Institute of Technology, Pasadena, CA. His research interests include theoretical and applied physics, biomedicine, astrobiology, computational field geology, and autonomous planetary and space exploration. Dr. Fink obtained an M.S. degree in Physics from the University of Göttingen in 1993 and a Ph.D. in Theoretical Physics from the University of Tübingen in 1997. His work is documented in numerous publications and patents.*

Terrance L. Huntsberger *is a Principal Member of the Technical Staff in the Mobility and Robotics Section at the Jet Propulsion Laboratory in Pasadena, CA, where he is the Manager for numerous tasks in the areas of multi-robot control systems, rover systems for access to high risk terrain, and long range traverse algorithms for rovers. He is an Adjunct Professor and former Director of the Intelligent Systems Laboratory in the Department of Computer Science at the University of South Carolina. His research interests include behavior-based control, computer vision, neural networks, wavelets, and biologically inspired system design. Dr. Huntsberger has published over 120 technical articles in these and associated areas. He received his PhD in Physics in 1978 from the University of South Carolina. He is a member of SPIE, ACM, IEEE Computer Society, and INNS.*

Gerhard Klimeck *is the Technical Director of the Network for Computational Nanotechnology at Purdue University and a Professor of Electrical and Computer Engineering since Dec. 2003. He leads the development and deployment of web-based simulation tools that are hosted on http://nanohub.org a community website that is utilized by over 3,000 users annually. He was the Technical Group Supervisor for the Applied Cluster Computing Technologies Group and continues to hold his appointment as a Principal Member at the Jet Propulsion Laboratory on a faculty part-time basis. His research interest is in the modeling of nanoelectronic devices, parallel cluster computing, genetic algorithms, and parallel image processing. Gerhard originally initiated the development of the PERSON framework described in this publication to optimize nanoelectronic modeling problems, circuits and optical filter designs. He has been the lead on the development of NEMO 3-D, a tool that enables the simulation of multi-million atom quantum dot systems, and NEMO 1-D, the*

first nanoelectronic CAD tool. Previously he was a member of technical staff at the Central Research Lab of Texas Instruments. He received his Ph.D. in 1994 from Purdue University and his German electrical engineering degree in 1990 from Ruhr-University Bochum. Dr. Klimeck's work is documented in over 110 peer-reviewed publications and over 170 conference presentations. He is a member of IEEE (senior), APS, HKN and TBP. More information can be found at http://www.ece.purdue.edu/~gekco and htt;//nanoHUB.org.

Mark Kordon *is the Technical Group Supervisor for the Modeling and Simulation Technologies Group, and Task Manager for Multi-Mission Analysis Tools at the Jet Propulsion Laboratory. His research interests include modeling and simulation techniques, evolutionary computing, multi-agent systems and space systems. Mark conceived of and coordinated the Automated Design of Power Subsystem work described in this paper He received his Bachelor of Science in Computer and Systems Engineering from Rensselaer Polytechnic Institute.*

Seungwon Lee *is a member of the technical staff at the Jet Propulsion Laboratory. Her research interest includes genetic algorithms, low-thrust trajectory design, nanoelectronics, quantum computation, parallel cluster computation and advanced scientific software modernization techniques. Seungwon received her Ph.D. in physics from the Ohio State University in 2002. Her work is documented in numerous journals and conference proceedings.*

Paul von Allmen *is the supervisor of the Applied Cluster Computing Technologies group and a senior researcher at the Jet Propulsion Laboratory. Paul is currently leading research in quantum computing, thermoelectric and nonlinear optics material and device design, nano-scale chemical sensors, semiconductor optical detectors, and low-thrust trajectory optimization. Prior to joining JPL in 2002, Paul worked at the IBM Zurich Research Lab on semiconductor diode lasers, at the University of Illinois on the silicon device sintering process, and at the Motorola Flat Panel Display Division as manager of the theory and simulation group and as principal scientist on micro-scale gas discharge UV lasers. Paul received his Ph.D. in physics from the Swiss Federal Institute of Technology in Lausanne, Switzerland in 1990. His work is documented in numerous publications and patents.*

Joseph Xu *is an academic part time employee at JPL working in the Avionics Architectures group. He is an undergraduate student at California Institute of Technology, where he is pursuing a BS in computer science.*