CLOSING THE LOOP: PRECISION LANDING ON MARS USING EVOLUTIONARY COMPUTATION,
B. K. Birge, L-3 Communications/JSC, 1319 Saxony Lane, Houston, TX 77058, email: brian.k.birge@nasa.gov

Introduction: Various proposed solar system missions, such as the ever-proposed Mars Sample Return Mission (MRSR), International Lunar Network (ILN), and others such as perhaps the unnamed Russian contemplated Jupiter moon lander mission all will require precise re-entry terminal position and velocity states. This is to achieve mission objectives such as rendezvous with a previous landed mission, reaching a particular geographic landmark, and/or applying hazard avoidance to touch down in the midst of dangerous terrain.

The current state of the art footprint for Mars and farther is in the magnitude of kilometers [1].

Obstacles to reducing the footprint include trajectory dispersions due to initial atmospheric entry conditions, environment, deployment dynamics, and injection error.

A relatively recent method of evolutionary computation is Particle Swarm Optimization (PSO), which works well with multi-optimization problems of high dimensionality and non-linearity [2]. Continued refinement of PSO in the larger research community comes from attempts to understand human-human social interaction as well as analysis of its Emergent behavior.

A novel Particle Swarm Optimization trajectory generator capable of working in a non-stationary environment has been shown to reduce the generated reference trajectory footprint to under 10 meters for the Mars case [3].

It is proposed to examine the feasibility of "closing the loop" of this system using an Artificial Neural Network based guidance and control system. We wish to see how much generalization can be successfully performed from perturbed initial states as well as allowing on the fly state corrections either from changed mission directives or hazard avoidance concerns. Can we recalculate the reference trajectory fast enough to balance uncertainties in planetary atmosphere, dust cloud thickness, sensor loss, etc? Can we balance and dynamically choose optimization goals based on the health of the system such as. fuel conservation in the context of "expanded" science goals, among others? These answers have broad applicability to both manned and robotic future mission designs.

Approach: The big picture approach is to develop an artificial neural network trajectory generator coupled with online learning and particle swarm optimization that generates optimal trajectories. These trajectories then are integrated into a real-time control system for the simulated vehicle. Possible tools used could include Trick, Matlab, and/or POST II.

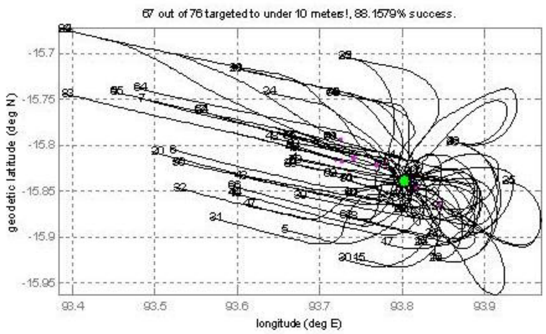
Trick is a simulation framework and library that allows rapid development of real time modeling as well as high fidelity analysis. Trick has been co-developed by NASA JSC and L-3 Communications since the late 1980s and has been adopted as the JSC standard for software simulation technology. It is used across the JSC in a wide variety of high fidelity simulation based facilities.

POST (Program to Optimize Simulated Trajectories) is a generalized point mass, discrete parameter targeting and optimization program originally developed at NASA's Langley Research Center (LaRC). POST provides the capability to target and optimize point mass trajectories for a powered or unpowered vehicle near an arbitrary rotating, oblate planet. It has been used to solve a wide variety of re-entry problems as well as orbital transfer. It has multiple phase simulation capability which features generalized and specific (via plug-in or user provided tables) planet and vehicle modes.

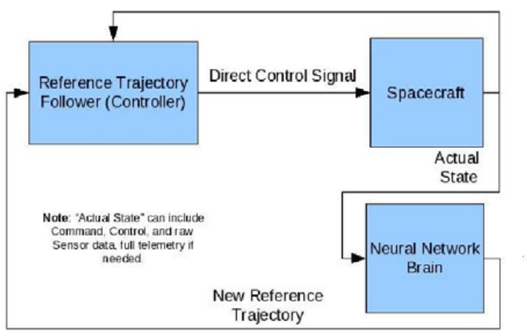
The Matlab family of software is used to rapidly prototype algorithms and visualize data analysis. It could be used within the optimization system to control runs of the core simulation by modifying input files at various stages of the trajectory. The output of the Trick and/or POST based simulation would then be fed back into the Matlab shell, driving convergence of the controller.

The optimal trajectories are ever changing based on current state of the simulated vehicle and can also be affected by real-time modification of mission objectives and parameters, fed directly into a cost function. The simulated vehicle control system and trajectory generator are then fed back into each other dynamically, creating the minimum distance from the target end state and avoiding hazards along the way.

Illustrations: The following graph shows optimized reference terminal descent trajectories made with PSO, using the above described strategy in an open loop sense and utilizing the Particle Swarm Toolbox [4]. Given a Monte Carlo derived cloud of likely entry states, it shows the terminal descent profile in local longitude and latitude coordinates of a lander on a simulated baseline Mars mission optimized for distance to target. Controls are only bank angle adjustments on a steerable parachute [5] but of course other specific control implementations depend on the mission design. For lower density atmospheres, other approaches such as powered descent may be used or a hybrid of the classic Mars entry strategies. The point being this is a flexible approach to generating realizable trajectories.



The following diagram shows the concept. We wish to close the loop of the control system. The neural controller should be based on error between predicted and actual.



References:

[1] R. Braun, et al., (1998) "The Mars Surveyor 2001 Lander: A First Step Toward Precision Landing", *IAF-98-Q.3.02, 49th AIF Congress*

[2] M. Clerc, (2006), "Particle Swarm Optimization", *ISTE, ISBN: 1-905209-04-5*

[3] B. Birge, (2008), "A Computational Intelligence Approach to the Mars Precision Landing Problem", *Doctoral Dissertation, ISBN: 978-0-549-55148-5*

[4] B. Birge (2003), "PSOt – A particle Swarm Optimization Toolbox for Use with Matlab", *IEEE 2003 Swarm Intelligence Symposium, 182-186*

[5] G. Walberg, B. Birge, (2000), "Terminal Guidance Techniques for a Mars Precision Lander", *AIAA 2000-5342*