The steps include setting up a region of interest, a start position, and a stop position, as well as initially random traversal waypoints. The optimization routine moves the way points around for each candidate solution and attempts to evolve the best path with regards to the reference cost function. The program calculates a path connecting all the waypoints from start to finish, and feeds this path to a cost function. The cost function determines various metrics such as length of path, collision with obstacles, work required to traverse the path, smoothness, weight on exploration of new territory vs. tracking the original outbound path, etc. The calculation of the optimal path is iterative; several rounds of feeding candidate solutions and using their associated costs to calculate new candidate solutions are required. The practical result of the pairing of this cost function strategy with PSO is that an optimal path is evolved much faster than random search, and completely forgiving of discontinuities.

The path planning prototype can be re-tasked on the fly and uses a unique "way point" optimization strategy. Unlike other optimization strategies, this one will work in a discontinuous environment with no modification necessary and is guaranteed to provide a continuous path from start to finish.

This work was done by Brian Birge of L-3 Communications for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24864-1

Smart-Divert Powered Descent Guidance to Avoid the Backshell Landing Dispersion Ellipse

The software and methods are valid for planetary or lunar powered descent.

NASA's Jet Propulsion Laboratory, Pasadena, California

A smart-divert capability has been added into the Powered Descent Guidance (PDG) software originally developed for Mars pinpoint and precision landing. The smart-divert algorithm accounts for the landing dispersions of the entry backshell, which separates from the lander vehicle at the end of the parachute descent phase and prior to powered descent. The smart-divert PDG algorithm utilizes the onboard fuel and vehicle thrust vectoring to mitigate landing error in an intelligent way: ensuring that the lander touches down with minimum-fuel usage at the minimum distance from the desired landing location that also avoids impact by the descending backshell.

The smart-divert PDG software implements a computationally efficient, convex formulation of the powered-descent guidance problem to provide pinpoint or precision-landing guidance solutions that are fuel-optimal and satisfy physical thrust bound and pointing constraints, as well as position and speed constraints. The initial smart-divert implementation enforced a lateral-divert corridor parallel to the ground velocity vector; this was based on guidance requirements for MSL (Mars Science Laboratory) landings. This initial method was overly conservative since the divert corridor was infinite in the down-range direction despite the backshell landing inside a calculable dispersion ellipse. Basing the divert constraint instead on a local tangent to the backshell dispersion ellipse in the direction of the desired landing site provides a far less conservative constraint. The resulting enhanced smart-divert PDG algorithm avoids impact with the descending backshell and has reduced conservatism.

This work was done by John M. Carson and Behcet Acikmese of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

The software used in this innovation is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47884.

Estimating Foreign-Object-Debris Density From Photogrammetry Data

John F. Kennedy Space Center, Florida

Within the first few seconds after launch of STS-124, debris traveling vertically near the vehicle was captured on two 16-mm film cameras surrounding the launch pad. One particular piece of debris caught the attention of engineers investigating the release of the flame trench fire bricks. The question to be answered was if the debris was a fire brick, and if it represented the first bricks that were ejected from the flame trench wall, or was the object one of the pieces of debris normally ejected from the vehicle during launch. If it was typical launch debris, such as SRB throat plug foam, why was it traveling vertically and parallel to the vehicle during launch, instead of following its normal trajectory, flying horizontally toward the north perimeter fence?

By utilizing the Runge-Kutta integration method for velocity and the Verlet integration method for position, a method that suppresses trajectory computational instabilities due to noisy position data was obtained. This combination of integration methods provides a means to extract the best estimate of drag force and drag coefficient under the non-ideal conditions of limited position data. This integration strategy leads immediately to the best possible estimate of object density, within the constraints of unknown particle shape. These types of calculations do not exist in readily available off-the-shelf simulation software, especially where photogrammetry data is needed as an input.