# Software

## G-DYN Multibody Dynamics Engine

G-DYN is a multi-body dynamic simulation software engine that automatically assembles and integrates equations of motion for arbitrarily connected multibody dynamic systems.

The algorithm behind G-DYN is based on a primal-dual formulation of the dynamics that captures the position and velocity vectors (primal variables) of each body and the interaction forces (dual variables) between bodies, which are particularly useful for control and estimation analysis and synthesis. It also takes full advantage of the spare matrix structure resulting from the system dynamics to numerically integrate the equations of motion efficiently. Furthermore, the dynamic model for each body can easily be replaced without re-deriving the overall equations of motion, and the assembly of the equations of motion is done automatically.

G-DYN proved an essential software tool in the simulation of spacecraft systems used for small celestial body surface sampling, specifically in simulating touch-and-go (TAG) maneuvers of a robotic sampling system from a comet and asteroid. It is used extensively in validating mission concepts for small body sample return, such as Comet Odyssey and Galahad New Frontiers proposals.

*This work was done by Behcet Acikmese, James C. Blackmore, and Milan Mandic of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.*

*This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47195.*

## Multibody Simulation Software Testbed for Small-Body Exploration and Sampling

G-TAG is a software tool for the multibody simulation of a spacecraft with a robotic arm and a sampling mechanism, which performs a touch-and-go (TAG) maneuver for sampling from the surface of a small celestial body. G-TAG utilizes G-DYN, a multi-body simulation engine described in the previous article, and interfaces to controllers, estimators, and environmental forces that affect the spacecraft. G-TAG can easily be adapted for the analysis of the mission stress cases to support the design of a TAG system, as well as for comprehensive Monte Carlo simulations to analyze and evaluate a particular TAG system design.

Any future small-body mission will benefit from using G-TAG, which has already been extensively used in Comet Odyssey and Galahad Asteroid New Frontiers proposals.

*This work was done by Behcet Acikmese, James C. Blackmore, and Milan Mandic of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).*

*This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47196.*

## Propulsive Reaction Control System Model

This software models a propulsive reaction control system (RCS) for guidance, navigation, and control simulation purposes. The model includes the drive electronics, the electromechanical valve dynamics, the combustion dynamics, and thrust. This innovation follows the Mars Science Laboratory entry reaction control system design, and has been created to meet the Mars Science Laboratory (MSL) entry, descent, and landing simulation needs. It has been built to be plug-and-play on multiple MSL testbeds [analysis, Monte Carlo, flight software development, hardware-in-the-loop, and ATLO (assembly, test and launch operations) testbeds].

This RCS model is a C language program. It contains two main functions: the RCS electronics model function that models the RCS FPGA (field-programmable-gate-array) processing and commanding of the RCS valve, and the RCS dynamic model function that models the valve and combustion dynamics. In addition, this software provides support functions to initialize the model states, set parameters, access model telemetry, and access calculated thruster forces.

*This work was done by Paul Brugarolas, Linh H. Phan, Frederick Serricchio, and Alejandro M. San Martin of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.*

*This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46978.*

## Licklider Transmission Protocol Implementation

This software is an implementation of the Licklider Transmission Protocol (LTP), a communications protocol intended to support the Bundle Protocol in Delay-Tolerant Network (DTN) operations. LTP is designed to provide retransmission-based reliability over links characterized by extremely long message round-trip times and/or frequent interruptions in connectivity. Communication in interplanetary space is the most prominent example of this sort of environment, and LTP is principally aimed at supporting "long-haul" reliable transmission over deep-space RF links.

Like any reliable transport service employing ARQ (Automatic Repeat reQuests), LTP is "stateful." In order to assure the reception of a block of data it has sent, LTP must retain for possible retransmission all portions of that block which might not have been received yet. In order to do so, it must keep track of which portions of the block are known to have been received so far, and which are not, together with any additional information needed for purposes of retransmitting part, or all, of the block. Long round-trip times mean substantial delay between the transmission of a block of data and the reception of an acknowledgement from the block's destination, signaling arrival of the block. If LTP postponed transmission of additional blocks of data until it received acknowledgement of the arrival of all prior blocks, valuable opportunities to use what little deep space transmission bandwidth is available would be forever lost.

For this reason, LTP is based in part on a notion of massive state retention. Any number of requested transmission conversations (sessions) may be concurrently "in flight" at various displacements along the link between two LTP engines, and the LTP engines must necessarily retain transmission status and retransmission resources for all of them. Moreover, if any of the data of a given block are lost en route, it will be necessary to retain the state of that transmission during an addi-