

Real-Time Dynamics Simulation of the Cassini Spacecraft Using DARTS

Part I. Functional Capabilities and the Spatial Algebra Algorithm

A. Jain and G.K. Man

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

Abstract

This paper describes the Dynamics Algorithms for Real-Time Simulation (DARTS) real-time hardware-in-the-loop dynamics simulator for the National Aeronautics and Space Administration's Cassini spacecraft. The spacecraft model consists of a central flexible body with a number of articulated rigid-body appendages. The demanding performance requirements from the spacecraft control system require the use of a high fidelity simulator for control system design and testing. The DARTS algorithm provides a new algorithmic and hardware approach to the solution of this hardware-in-the-loop simulation problem. It is based upon the efficient spatial algebra dynamics for flexible multibody systems. A parallel and vectorized version of this algorithm is implemented on a multiprocessor low-cost computer to meet the simulation timing requirements.

1. Introduction

The Cassini mission will be the first to conduct an in-depth study of the Saturnian system by sending a spacecraft and a probe to the planet. The planned launch date is in the mid 1990s with an arrival date in mid 2004. The major scientific goals of the mission are to obtain fundamental new information about the origin and evolution of the solar system, molecular evolution in space and its possible role in the origin of life, and astrophysical plasma dynamics and processes. There are twelve instruments on board and they can be grouped into three categories:

- Scanning-platform-mounted high-precision pointing and scanning instruments for imaging and spectroscopy in visual and infrared.
- Turntable-mounted instruments for plasma, charged particles, and magnetospheric imaging and dust detection.
- Basebody-mounted instruments such as the magnetometer and plasma/radio wave sensors on extended booms, radar mapper for imaging of Titan's surface using the spacecraft high gain antenna.

The spacecraft also carries the Huygens Probe, supplied by the European Space Agency, which contains six instruments to take science measurements as it enters Titan's atmosphere.

Figure 1 shows the deployed Cassini spacecraft. The probe is shown off to the left

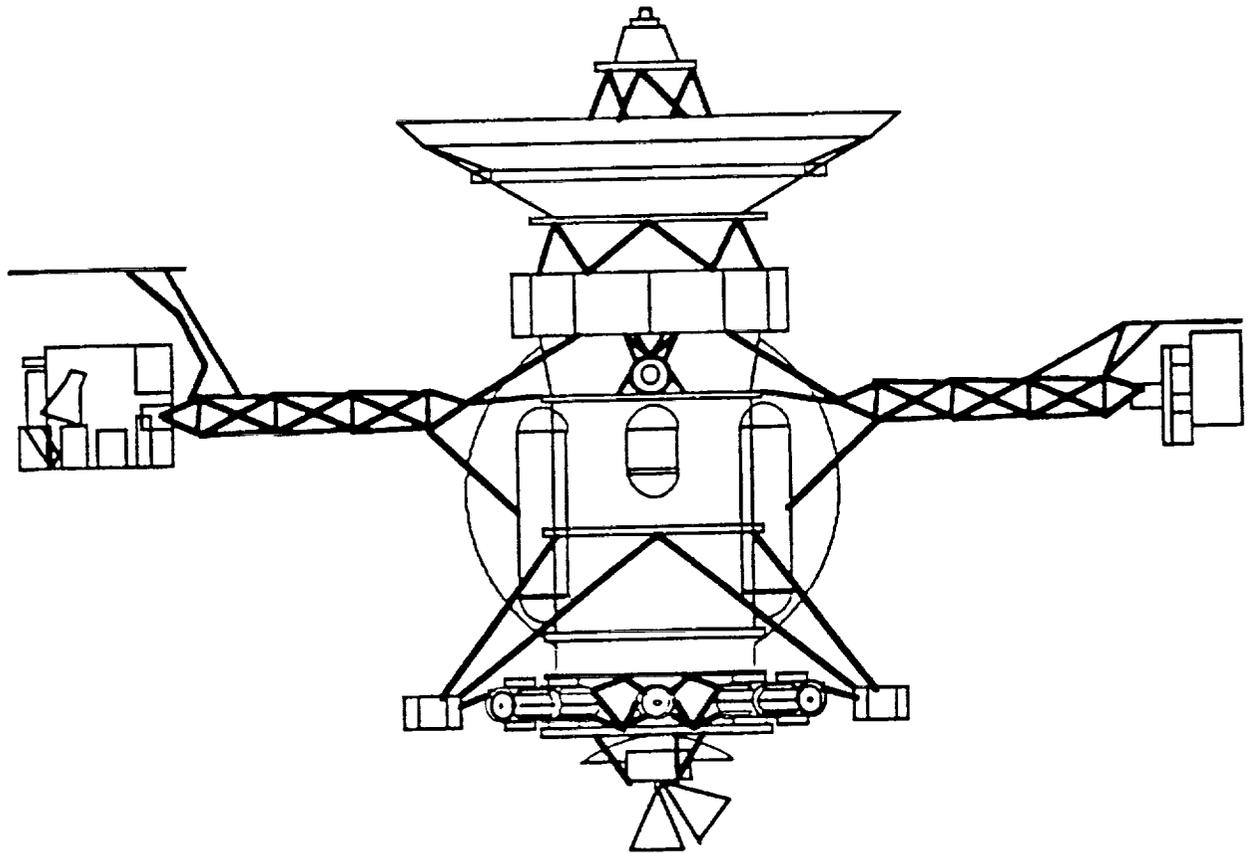


Figure 1: A schematic of the Cassini spacecraft

exposing the spin/eject device. At the top of the spacecraft is the high and low gain antenna. Three booms are attached to the upper equipment module. They carry the high-precision scanning platform (HPSP), the magnetometer and the 10-m plasma/radio wave antenna, and the turntable. The middle spacecraft structure contains the propulsion tanks carrying 68% of the spacecraft mass. At the bottom of the propulsion module is the lower equipment module which supports three radioisotope thermoelectric generators for spacecraft power, four reaction wheels, the articulated probe relay antenna, and the articulable main engine.

The HPSP articulates in two directions, one about the boom axis and one about an orthogonal intermediate axis. The turntable rotates continuously about the boom axis at 0.1, 1.0, or 3.0 rpm. The probe relay antenna has one degree of freedom (dof) about an axis

parallel to the turntable boom. The main engine can be articulated about two axes allowing velocity control along the high gain antenna beam direction.

There are a number of key attitude control functions the spacecraft must perform. The spacecraft must acquire the Sun and certain stars for inertial reference shortly after launch. Then it will maintain Earth/Sun pointed for ground communication and thermal control. There are a number of propulsive maneuvers for plane changes and orbit insertion. During the science phase of the mission when the Huygen Probe will be released into Titan, the data from the probe will be collected and relayed by the spacecraft back to Earth. The spacecraft will spend the next 4 years conducting intensive scientific investigations of the Saturnian system.

The primary attitude control sensors are the star sensors and the gyroscopes located on the HPSP. The key actuators are: (1) the electro-mechanical actuators for the HPSP, the turntable, the main engines, and the reaction wheels; and (2) the chemical propulsion thrusters for attitude control, and the main engines. During main engine firing, the gyroscopes are used as the control sensor and they are separated from the main engine gimbal actuators by the spacecraft bus and boom, which are nonrigid. Furthermore, the bus carries a large amount of liquid propellant. This sensor and actuator noncollocation problem is one that requires high fidelity dynamics simulation of the spacecraft for control design and testing. This simulation must also be used to develop all the control loops including the high precision control loop.

The simulation requirement is most stringent for real-time hardware-in-the-loop testing when flight hardware and software are integrated with a simulation of the spacecraft. In the mid 1980s, technology precluded the use of hardware-in-the-loop simulation for all but the simplest dynamic systems (typically single-axis rigid-body equations). Since the late 1980s, technology has advanced to a point that fairly complex spacecraft can be simulated in real-time but the cost of the computer hardware is very high. Part I of this paper presents a new algorithmic and hardware approach to this important hardware-in-the-loop simulation problem.

The Cassini spacecraft was simplified in June 1992 to cut mission costs. The new baseline eliminated the articulated HPSP and the turntable. The dynamics of the spacecraft were simplified and so were the simulation requirements. We look at the pre-June 1992 Cassini simulation needs and present the DARTS (Dynamics Algorithms for Real-Time Simulation) solution to this technically more challenging problem.

The DARTS algorithm is based upon the spatial algebra flexible dynamics algorithm [1]. Part I of this paper describes the functional capabilities and requirements of the DARTS simulator. It also contains an overview of the spatial algebra algorithm for flexible multibody dynamics. A parallel and vectorized version of this algorithm has been developed and implemented on a multiprocessor real-time computer consisting of a pair of SKYbolt i860 vector processors. These processors are high performance computers and are relatively inexpensive. Part II of this paper describes the parallel/vectorization of this algorithm and the real-time computing hardware and implementation.

2. Functional Capabilities of DARTS

The multibody model for the Cassini spacecraft is a *star-topology* dynamics model consisting of a central flexible “extended” bus body (denoted BUS in Figure 2) to which a number of articulated rigid-body appendages are attached. One such model for the spacecraft is shown in Figure 2. In this model, the bus body encompasses all bodies with significant structural

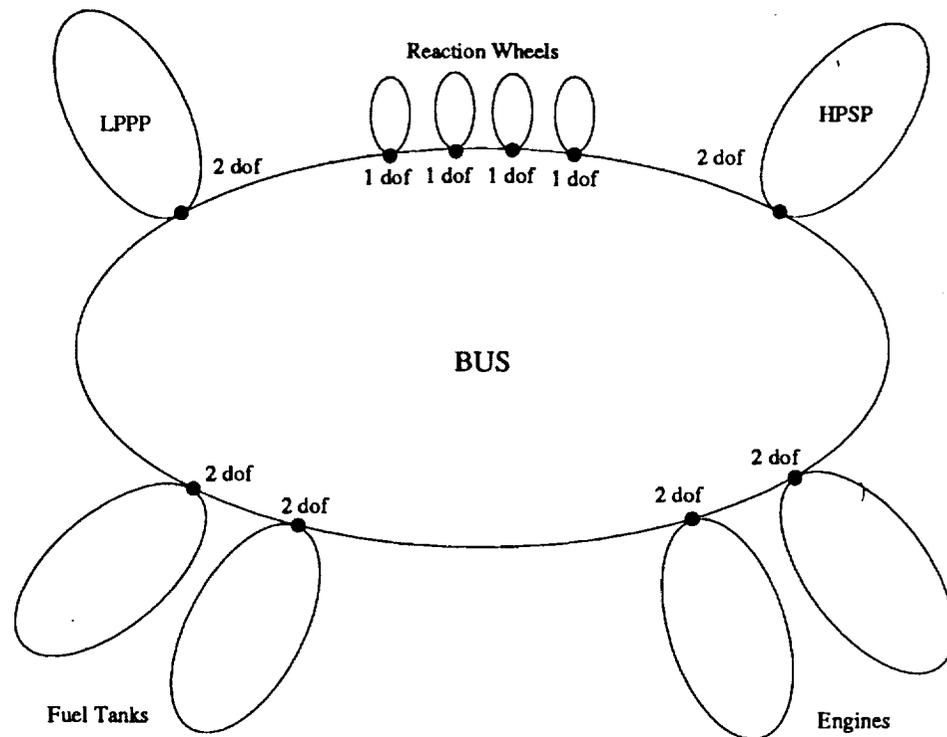


Figure 2: A star-topology multibody model for the Cassini spacecraft

flexibility such as the magnetometer boom and the platform truss structures. The articulated rigid-body appendages in the model include the high-precision scanning platform (HPSP) and the low-precision pointing platform (LPPP), four reaction wheels, two engine assemblies, and a pair of pendulum models for the fuel tanks.

The DARTS simulator has been designed to handle the flexible multibody dynamics of such a model for both the real-time and non-real-time simulation needs for the Cassini Project at the Jet Propulsion Laboratory. Some of the important features of this star-topology implementation of DARTS are described below.

The real-time DARTS was required to compute the generalized accelerations for the spacecraft within 5 ms. This timing requirement was for a model consisting of ten articulated appendages and five flexible assumed modes for the bus with overall 25 degrees of freedom.

During initialization, DARTS reads the data defining the spacecraft star-topology

model from a user specified model file (referred to herein as the MODEL file). Changes to the spacecraft model do not require any additional changes or recompilation of the DARTS software. Thus, spacecraft model changes, such as in the number of appendages, the mass and inertia properties, kinematical properties, the type of hinges, etc., require only the updating of the MODEL file. Evolutionary changes in the spacecraft model, as well as multiple models for early and late mission scenarios, can thus be easily handled using DARTS.

There are no restrictions within DARTS on the number of appendages or on the number of assumed modes used to model the bus flexibility in the spacecraft model. The assumed modes' data for bus flexibility is also provided in the MODEL file. This data includes the modal vectors for the various nodes which serve as attachment points for the appendages, actuators, and sensors on the bus. The modes are assumed to be eigen-modes and, therefore, the stiffness and damping matrices for the bus are diagonal matrices. Spacecraft models with fidelity ranging from rigid-body models to high-fidelity models with a large number of modes can be used by simply changing the MODEL file.

Pin, universal and gimbal rotational hinges, and one degree of freedom prismatic hinges between the appendages and the bus have been implemented in DARTS.

For the most part, during spacecraft simulations, the generalized force for the hinges is provided as an input, while the corresponding generalized accelerations are computed by DARTS. These hinges are referred to as "regular" hinges. DARTS also allows "prescribed motion" hinges - i.e., hinges for whom the generalized acceleration is provided as an input while the corresponding generalized forces are computed by DARTS. Prescribed motion hinge models are required for engine thrust vector control and for testing fault-recovery algorithms. DARTS allows multiple-degree-of-freedom hinges to be a hybrid combination of regular and prescribed motion degrees of freedom. The degrees of freedom can also be switched from regular to prescribed motion models (and vice versa) during run time. This feature is useful for simulating fault events such as hinge lockup as well as the dynamics of probe release.

DARTS allows for an arbitrary number of sources of external forces and moments on the spacecraft. Within DARTS, these external forces and moments are assumed to be applied by actuators located on the spacecraft. The actuators can include physical actuators such as thrusters, as well as pseudo-actuators, used to model disturbance forces from misalignment, comet dust, gravity gradients, etc. These actuators differ from the hinge actuators in that the latter only contribute to the generalized forces for the spacecraft. The external force actuators can be at arbitrary locations on the bus and the appendages. Data regarding the location and number of these actuators is provided in the MODEL file. The external force data is assumed to be defined in the actuator's reference frame.

DARTS allows for an arbitrary number of sensors on the spacecraft. Sensors can include accelerometers, cameras, gyroscopes, sun sensors, etc., and can be at arbitrary locations on the bus and the appendages. DARTS computes the orientation and location of each sensor with respect to the bus frame. The velocity and the acceleration of each sensor are also computed in its own reference frame. The number and location of the sensors are

once again specified through the MODEL file.

DARTS allows a limited number of changes to the spacecraft model during run time. It is possible to switch a hinge degree of freedom between prescribed to regular motion status during run time. There is also a limited (quasistatic) capability for handling changes to the mass and inertia properties of the appendages. This feature can be used to handle changes in fuel mass due to fuel depletion during engine burns.

The kinetic energy, the deformation potential energy, and the linear and angular momenta of the spacecraft are computed by DARTS. Also computed are the location and velocity of the center of mass of the spacecraft.

The structure of the DARTS software is in the form of a subroutine. It is completely portable across different computing platforms and can be used as a part of off-line simulations for control subsystem design as well as for hardware-in-the-loop real-time simulations.

3. The DARTS Spatial Algebra Algorithm

The DARTS dynamics algorithm is based upon the high-speed spatial algebra algorithm for flexible multibody dynamics described in reference [1]. The algorithm also incorporates the new techniques for handling prescribed motion described in reference [2]. A brief overview of the main developments that forms the basis of the DARTS algorithm is presented.

For the sake of brevity, a description of the nomenclature used here is omitted and, instead, details are described in reference [1]. The equations of motion of a (tree-topology) flexible multibody system can be written in the form

$$T = \mathcal{M}\dot{\chi} + \mathcal{C} \quad (3.1)$$

where the mass matrix \mathcal{M} and the vector of Coriolis and centrifugal forces \mathcal{C} are given as follows:

$$\mathcal{M} \triangleq \mathcal{H}\Phi M_m \Phi^* \mathcal{H}^* \in \mathfrak{R}^{N_{dof_s} \times N_{dof_s}} \quad \text{and} \quad \mathcal{C} \triangleq \mathcal{H}\Phi(M_m \Phi^* a_m + b_m + K_m \vartheta) \in \mathfrak{R}^{N_{dof_s}} \quad (3.2)$$

χ and T denote the vectors of generalized velocities and forces for the system. The definition of the spatial operators, such as \mathcal{H} , Φ etc., in Eq. (3.2) can be found in reference [1]. The expression for the mass matrix \mathcal{M} in Eq. (3.2) is referred to as its *Newton-Euler Operator Factorization*.

The following equation describes an alternative factorization, known as the *Innovations Operator Factorization* of the mass matrix:

$$\mathcal{M} = [I + \mathcal{H}\Phi K] D [I + \mathcal{H}\Phi K]^* \quad (3.3)$$

In this factorization, the factor $[I + \mathcal{H}\Phi K]$ is square, block lower triangular and nonsingular, while D is a block diagonal matrix. This factorization may be regarded as providing a

closed-form expression for the block LDL^* decomposition of \mathcal{M} . The following equation gives the closed-form operator expression for the inverse of the factor $[I + \mathcal{H}\Phi K]$.

$$[I + \mathcal{H}\Phi K]^{-1} = [I - \mathcal{H}\Psi K] \quad (3.4)$$

It follows from Eqs. 3.3 and 3.4 that the operator expression for the inverse of the mass matrix is given by:

$$\mathcal{M}^{-1} = [I - \mathcal{H}\Psi K]^* D^{-1} [I - \mathcal{H}\Psi K] \quad (3.5)$$

Once again, note that the factor $[I - \mathcal{H}\Psi K]$ is square, block lower triangular and nonsingular and so Eq. (3.5) may be regarded as providing a closed-form expression for the block LDL^* decomposition of \mathcal{M}^{-1} . The operator expression for the mass matrix inverse in Eq. (3.5) leads to the following operator expression for the generalized accelerations $\dot{\chi}$:

$$\dot{\chi} = [I - \mathcal{H}\Psi K]^* D^{-1} [T - \mathcal{H}\Psi \{KT + Pa_m + b_m + K_m \vartheta\}] - K^* \Psi^* a_m \quad (3.6)$$

This expression for the generalized accelerations directly leads to a recursive algorithm for computing the dynamics of the system. The structure of this algorithm is very similar in form to the articulated body algorithm for *rigid* multibody systems. The computational cost of this algorithm is reduced by separately processing the flexible and hinge degrees of freedom at each step in the recursion.

Based upon the unique features of the star-topology dynamics model for the Cassini spacecraft, the general spatial algebra algorithm has been simplified to obtain the DARTS algorithm. The primary simplifying features of the model are: (a) only the central bus body is a flexible body; (b) the model has only single body appendages; (c) all of the appendages are rigid bodies; and (d) the spacecraft is a free-flying multibody system. The computational cost of the algorithm for a 10-body spacecraft model is shown in Figure 3.

3.1 Algorithm Structure

Details of the computational steps of the DARTS algorithm can be found in reference [1]. It consists of three recursive sweeps whose structure is shown in Figure 4. Sweep 1 consists of an outward recursion (from the bus to the appendages) to compute the kinematics, velocities, and nonlinear Coriolis and centrifugal terms for all the appendages. This is followed by Sweep 2 which is an inward recursion from each appendage towards the bus. During this recursion, the articulated inertias and residual forces for each of the appendages are computed. The computation of the residual forces takes into account the external forces from any actuators on the appendage as well as the hinge torques for the appendage. The computations for each of the appendages are carried out independently of each other.

When these appendage computations are complete, the results from each appendage are used to compute the overall articulated body inertia and residual force for the bus. Unlike the case of the appendages, the computations here are more complex because of the structural flexibility of the bus. In addition to the rigid-body component, additional modal

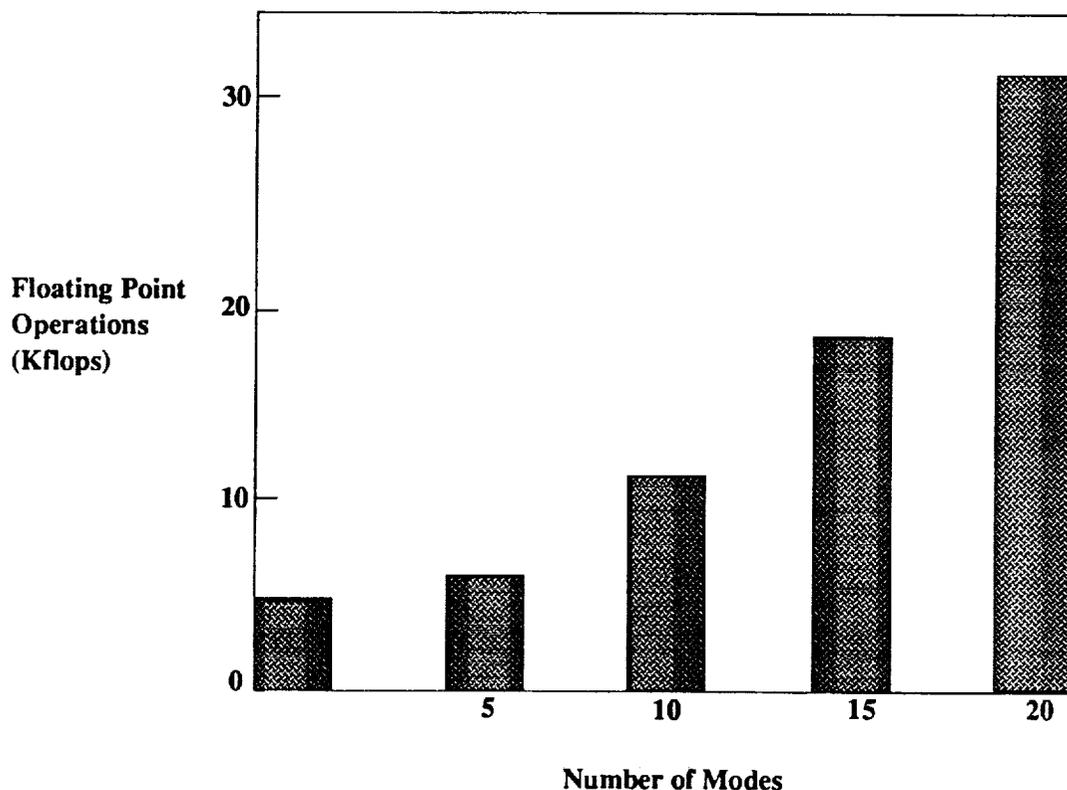


Figure 3: Computational cost versus the number of bus modes for the DARTS algorithm

components have to be computed. The bus articulated body inertia and residual force are used to compute the rigid-body acceleration and modal acceleration for the bus.

This step is followed by Sweep 3, which consists of independent outward recursions from the bus to each of the appendages. During this recursion, the hinge accelerations for each of the appendages are computed. Also as each body is processed, the attitude and velocity information for each sensor is also computed. This completes a single evaluation of the spacecraft dynamics.

3.2 Prescribed Motion

The above algorithm has been modified in order to handle prescribed motion hinges. While most of the hinge degrees of freedom on the spacecraft are “regular,” there are some hinge degrees of freedom, such as the engine gimbal assembly, that are modeled as undergoing prescribed motion. Moreover, fault-recovery algorithms are tested by simulating actuator

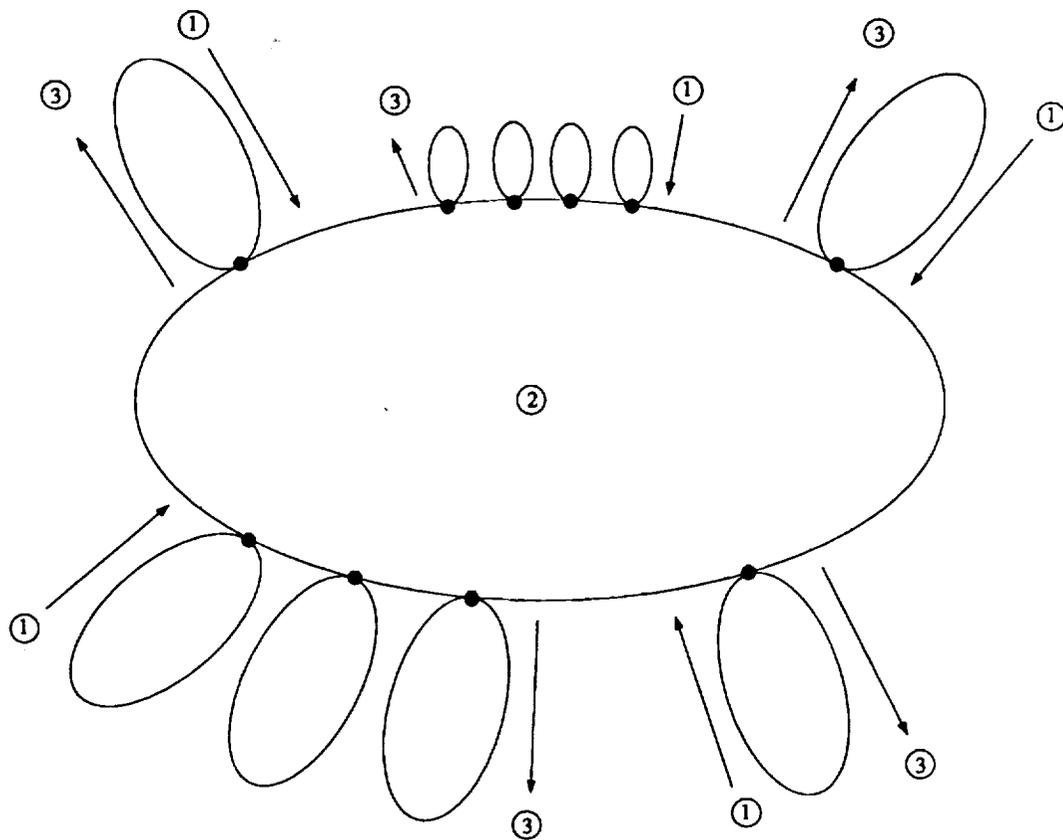


Figure 4: Structure of the star-topology DARTS algorithm

lock-up faults using prescribed motion models. This requires changing the status of the degree of freedom from regular to prescribed motion mode during run time.

With prescribed motion degrees of freedom, the dynamics problem is a “mixed” problem in that for each hinge degree of freedom either the generalized force or the generalized acceleration is known, and the complementary information for each degree of freedom needs to be computed. The prescribed motion features are implemented in DARTS using a recently developed spatial algebra algorithm for prescribed motion [2]. The structure of this algorithm turns out to be a simple variant of the regular dynamics algorithm and retains the same recursive structure. In the prescribed motion algorithm, during the articulated body recursion of Sweep 2, each hinge is checked for its regular/prescribed motion status. Depending on the status, the articulated body inertia and the residual force are computed differently using whichever of the generalized force or the generalized acceleration is known for the degree of freedom. The changes required to handle prescribed motion are entirely local to the hinge and do not affect the computations for any other hinge. The outward recursion of Sweep 3 is similar to Sweep 2 as well. Depending on the regular/prescribed motion status of a hinge, the unknown – the generalized acceleration or the generalized force – for the hinge is computed. Once again, the changes required to do the computational steps are completely local to the hinge and do not affect the computations at any other hinge. This algorithm allows the component degrees of freedom of multiple-degree-of-freedom hinges to

have arbitrary regular/prescribed motion status. This is especially useful for simulating a variety of actuator faults.

The local nature of the changes to the computations for a prescribed motion makes it simple to implement. Moreover, there is little computational overhead when the regular/prescribed motion status of a hinge is changed during run time. This is in contrast with the conventional prescribed motion algorithms which treat the prescribed motion as a global constraint on the dynamics.

4. Conclusions

Part I of this paper describes the functional capabilities of the DARTS flexible dynamics simulator for the Cassini spacecraft as well as the high-speed spatial algebra computational algorithms. The DARTS software is being used throughout the Cassini Project for control algorithm design and analysis, flight software integration and testing, and for real-time hardware-in-the-loop simulation. DARTS has been designed to be completely portable to run on different computing platforms. DARTS has also been designed to be data-driven. Thus DARTS can handle different spacecraft models to meet the various design and testing scenarios without any software modification. This feature considerably simplifies software maintenance since the same software is being used across the whole project. Due to the data-driven feature, the recent significant design changes to the spacecraft have not required any modifications to the DARTS software.

5. Acknowledgment

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] A. Jain and G. Rodriguez, "Recursive Flexible Multibody System Dynamics Using Spatial Operators," *Journal of Guidance, Control and Dynamics*, Nov. 1992. In press.
- [2] A. Jain and G. Rodriguez, "Recursive Dynamics Algorithm for Multibody Systems with Prescribed Motion," *Journal of Guidance, Control and Dynamics*, 1992. In press.