**Proceedings of The ASME 2024 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference**
**IDETC/CIE 2024**
**August 25-28, 2024, Washington, DC, USA**

**IDETC2024-142643**

# SEAMLESS INTEGRATION OF MULTIBODY AND ORBITAL DYNAMICS TO SUPPORT HUMAN SPACEFLIGHT SIMULATIONS

**Christopher C. Sullivan***
**Thomas A. Brain**

METECS
Houston, TX 77058
Email: csullivan@metecs.com, tbrain@metecs.com

**Leslie J. Quiocho**

NASA Johnson Space Center
Software, Robotics, and Simulation Division
Houston, TX 77058
Email: leslie.j.quiocho@nasa.gov

## ABSTRACT

*At the NASA Johnson Space Center (JSC), modeling capability requirements are often driven by the need to simulate upcoming mission scenarios for analysis, procedures development, and training. In preparation for free-flyer vehicle capture, maneuvers, and release, it was determined that the multibody dynamics capabilities provided by MBDyn and the orbital dynamics capabilities supplied by JEOD needed to be merged to properly simulate the dynamic behaviors expected on orbit. After initial attempts to combine the two packages proved cumbersome, it was decided that a more refined solution was required to facilitate the interface between these two packages. The resulting interface software package was named MBJEOD. The primary design goals for this new package were to leave MBDyn and JEOD untouched, to simplify and clarify simulation initialization for users, and to support any topology or on-orbit operational scenario that future missions might require. This paper provides a detailed overview of MBJEOD and the approach taken to seamlessly integrate the multibody dynamics of MBDyn with the orbital dynamics of JEOD. Additional attention is given to the numerical considerations required for MBJEOD and the simplifying assumptions leveraged to provide for real-time simulation needs. Verification and validation are addressed and current MBJEOD applications at the NASA JSC are discussed.*[1]

## INTRODUCTION

As has been previously discussed in [1], operations involving human spaceflight generally involve the complex application of multibody dynamics. Specific examples include the capture, maneuvering, and release of a spacecraft vehicle by a robotic manipulator or the articulation of solar arrays and other mechanical systems on an orbiting vehicle.

MBDyn is a software engine that calculates the dynamics states of kinematic, rigid, or flexible multibody systems [2-5]. An MBDyn multibody system may consist of multiple groups of articulated chains, trees, or closed-loop topologies. The algorithms have been optimized for efficiency and can be used for both non-real-time (NRT) and real-time (RT) simulation applications.

JEOD, on the other hand, is designed to simulate spacecraft trajectories in flight regimes ranging from low Earth orbit (LEO) to lunar operations, interplanetary trajectories, and other deep space missions [6]. This package can be used to simulate a stand-alone spacecraft dynamics trajectory and attitude state, taking advantage of several orbital environment models (e.g., atmosphere, ephemerides, gravity, and time) as well as force interaction models (e.g., aerodynamics, gravity torque, and radiation pressure). JEOD can also be interfaced with effector models of a given spacecraft's guidance, navigation, and control (GN&C) system.

While both MBDyn and JEOD are compatible with NASA JSC's Trick Simulation Environment [7-9], the simulation

*Address all correspondence to this author.

[1]This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Approved for public release; distribution is unlimited.

packages are designed with two very distinct sets of requirements. MBDyn is intended to solve complex dynamic equations of motion (EOMs) for articulated flexible multibody systems in an efficient manner. JEOD, conversely, implements configurable orbital environmental and interaction models on a single rigid body system or series of rigid bodies.

One of the driving operational scenarios that needed to be simulated for analysis, procedures development, and training, was the initial planned capture of the NASDA (now known as JAXA) HTV-1 by the CSA Space Station Remote Manipulator System (SSRMS) from the International Space Station (ISS) in 2009 (see Fig. 1) [10]. This mission, which represented HTV's first flight to the ISS, had two overall objectives: (1) to deliver supplies to the station and (2) to verify HTV's rendezvous capabilities and test onboard systems during the flight.

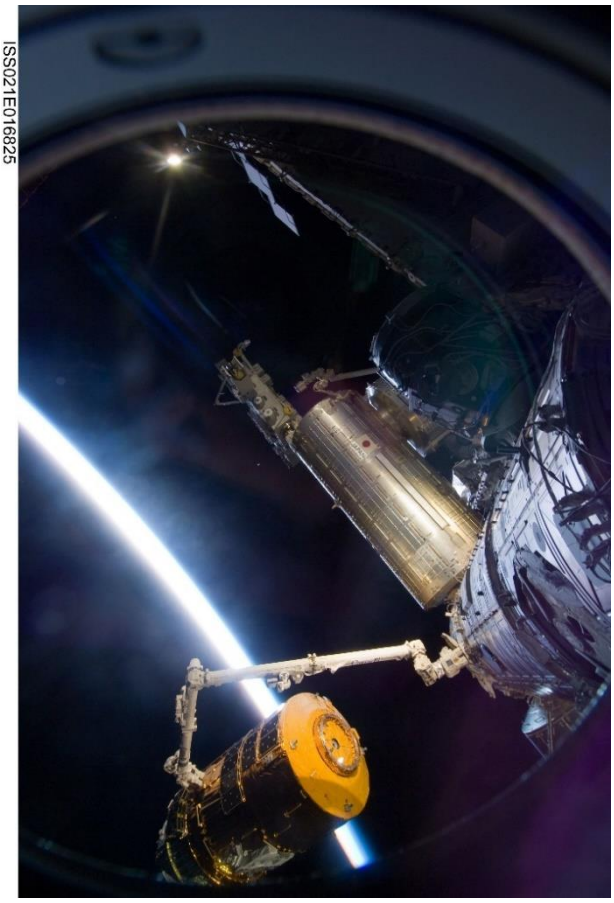



**FIGURE 1:** HTV CAPTURE BY SSRMS

Early attempts to combine multibody and orbital dynamics in support of this particular flight led to cumbersome and less than desirable usability for simulation developers. Since HTV-1 represented the first of many so-called "free flyer" captures and releases associated with ISS operations, including not only JAXA HTVs but also Orbital ATK (now Northrop Grumman) Cygnus (see Fig. 2) [11] and SpaceX Dragon vehicles (see Fig. 3) [12], there became a critical requirement to interface the capabilities of MBDyn and JEOD in a more effective and 'cleaner' manner. Moreover, as MBDyn and JEOD were undergoing significant revisions at the time, it was important to maintain compatibility of this new software interface to allow an integrated and fully functional dynamics capability that would evolve. At the time of these early attempts, software integration packages, such as the Functional Mock-up Interface [13], were less widely available and provided less customizability than an in-house solution. Therefore, a new package was developed allowing MBDyn and JEOD to work harmoniously together. It was entitled MBJEOD.



**FIGURE 2:** CYGNUS VEHICLE MANEUVER

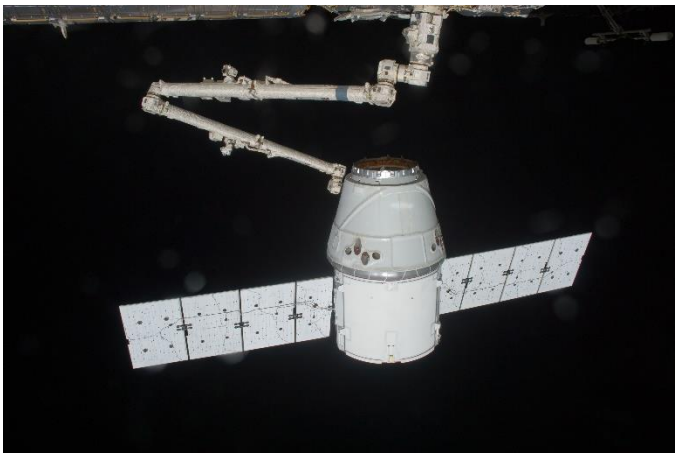


**FIGURE 3:** SSRMS GRAPPLES DRAGON

There were three original design goals of MBJEOD:

(1) To not change MBDyn or JEOD source code and delivered functions, i.e., to keep the released and independently tested packages as pure and pristine as possible.

(2) To develop a method for initializing data that did not require 'mixing' concepts. What this meant was that mass properties should be initiated using only one method as opposed to 'blending' mass initialization methods.

(3) To integrate the packages to support any topology or on-orbit operational scenario.

This paper provides a detailed overview of JSC's MBJEOD dynamics package and how it accomplishes these goals, along with the overall software approach taken and key capabilities of the interface. Specific example applications utilizing MBJEOD are discussed along with the related verification and validation (V&V) of these examples.

## APPROACH

In order to achieve these goals, MBJEOD needed to fuse the design paradigms of both MBDyn and JEOD in a manner that required the least change in input from users. MBJEOD accomplishes this through a set of interface classes which mirror the designs of both MBDyn and JEOD. The primary interface classes are MBJeodDyn, MBJeodBody, and MBJeodGroup.

MBJeodDyn is the executive MBJEOD class that controls the programmatic flow of initialization, gravitation, dynamics, and numerical integration. It calls the associated MBDyn and JEOD functions in the necessary order and stores information related to the system as a whole. For initialization, MBJEOD uses the input structures from either MBDyn or JEOD to calculate and populate the internal structures of both MBDyn and JEOD. Following initialization, the gravitation method handles calls to JEOD to calculate environmental and gravitational effects based on the states of the JEOD body structures, which MBJEOD then feeds into MBDyn's external force and torque structures in the dynamics method where MBDyn solves the EOMs. In the integration method, MBDyn's integrator is called to update the body states, which MBJEOD provides to the JEOD data structures via standard JEOD APIs, in order to prepare them for the next iteration's environmental and gravitational effects, thus completing the loop and coupling the software packages to harness the orbital capabilities of JEOD and the articulated multibody dynamics capabilities of MBDyn (see Fig. 4).

MBJeodBody incorporates elements of both MBDyn and JEOD by inheriting from JEOD's DynBody class and containing an instance of MBDyn's BODY_DATA class, thus retaining the fundamental concept of a body used by both software packages. This formulation allows for body input data to be provided in the structures from either MBDyn or JEOD, permitting users to convert simulations into an MBJEOD compatible format with limited changes. This is facilitated by a flag specifying the input source for the mass properties or the dynamic state and a collection of internal initializations and computations, ensuring all data is digested as expected by users of both MBDyn and JEOD and stored in the internal structures used for the simulation.

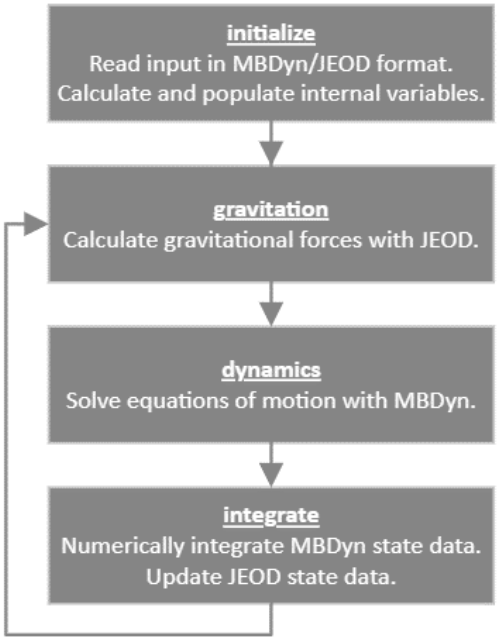



**FIGURE 4:** SIMPLIFIED MBJEOD WORKFLOW

In a standard JEOD simulation, the properties of a collection of bodies are recursively accumulated from the leaves of the topology to the root and remain unchanged through the duration of the simulation, as all bodies are rigidly attached. Table 1 demonstrates an example of the recursive accumulation of properties for the mass of the system shown in Fig. 5.

**TABLE 1:** STANDARD JEOD MASS PROPERTIES

| Body | Core Mass ($core_{body}$) | Composite Mass ($comp_{body}$) |
|---|---|---|
| n | $mass_n$ | $core_n + \sum comp_{child}$ |
| D | $mass_D$ | $core_D$ |
| C | $mass_C$ | $core_C + comp_D$ |
| B | $mass_B$ | $core_B$ |
| A | $mass_A$ | $core_A + comp_B + comp_C$ |

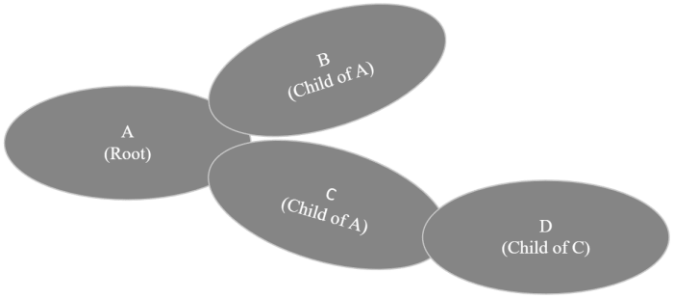



**FIGURE 5:** STANDARD JEOD TOPOLOGY

On the contrary, in a standard MBDyn simulation, the joints between bodies may permit movement, which means that the properties of the group of bodies changes over time (e.g., center of mass (CoM) of the composite group). Note that in Fig. 6, the system has articulating joints between the bodies on the inboard or proximal end of each body.

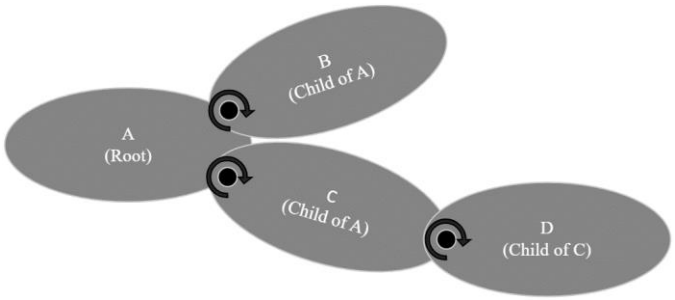



**FIGURE 6:** STANDARD MBDYN TOPOLOGY

Because MBJEOD utilizes the attachment and joint structure of MBDyn, the MBDyn attachment topology remains intact in MBJEOD applications, while JEOD's attachment structures are not retained, due to the articulating dynamics it is not capable of modeling itself. In essence, for MBJEOD applications, JEOD applies gravitation and environmental effects on each body as if it was independent from the rest (see Fig. 7). This is important to note because users of JEOD are accustomed to accessing the composite body state of JEOD, but in the context of MBJEOD, these composite states are invalid due to the aggregated CoM (and other properties) changing during simulation runtime.

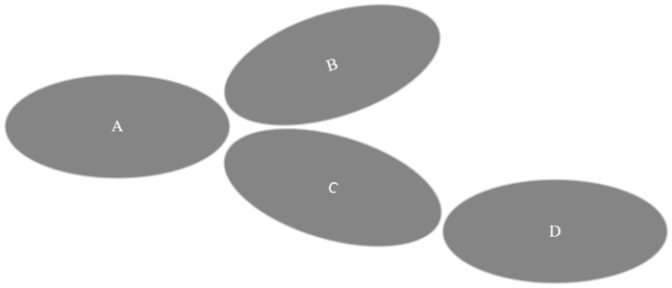



**FIGURE 7:** JEOD TOPOLOGY IN MBJEOD

MBJeodGroup behaves similarly to an MBDyn group in that it models an assembly of bodies. This involves tracking the list of connected bodies in the simulation, calculating system level properties, and managing the output of group-level data. A key feature of MBJeodGroup is that it interfaces JEOD gravity models with MBDyn body instances. Due to the complexities of combining orbital and articulating multibody dynamics, this is not a straightforward task and requires special numerical considerations, which MBJeodGroup handles. MBJeodGroup also offers options to apply simplifying assumptions, including linearized gravity, depending on the level of fidelity required by the current simulation.

With these interface classes, MBJEOD is capable of utilizing the multibody dynamics models of MBDyn and the orbital dynamics models of JEOD simultaneously; however, this alone is not enough to accurately model the behavior of multibody systems in orbit. Some of the environmental models in JEOD do not fully capture the dynamics of an orbital multibody system due to the rigid attachment assumptions made in JEOD. One example is the GravityTorque class, which models the torque caused by the misalignment of the gravity gradient and the inertial tensor of the system. It relies on the composite body data, which as mentioned previously, is not retained in MBJEOD, and additionally assumes that the torque should be calculated for, and applied to, the fully rigid system as a whole, rather than for each individual body. Therefore, the MBGravityTorque class was developed for MBJEOD to calculate this torque for each individual body in the system, allowing these torques to be transmitted across the rotating or translating joints, as allowed by the model. This capability is necessary to appropriately model the dynamics of an orbital multibody system with moving joints and potentially flexible bodies. For LEO, the impact of this approach is fairly small but for an oddly shaped asteroid body, the impact could be significant.

MBDyn provides a capability to track state data at any specified point on a body expressed in any frame defined in the system. These points are referred to as Points of Interest (POIs). When looking at the multibody subsystem of an MBJEOD model (and ignoring the orbital context), the gravitational acceleration is not always the primary concern, especially when considering relative accelerations of bodies in the system. To provide this data to users more readily, the NonGravAccels model is provided to calculate and store the non-gravitational acceleration of any POI. This is especially useful when modeling sensors like accelerometers, which in the freefall of orbit do not read the gravitational acceleration.

## NUMERICAL CONSIDERATIONS

The difference in the scale of joint accelerations between bodies in the system and accelerations of each body due to gravity requires special attention in order to produce a numerically sound simulation. The joint accelerations caused by motor torques, joint friction, or flexibility (given by $a_{multibody,n}$) are orders of magnitude smaller than those generated by gravitation (given by $a_{grav,n}$) as demonstrated in (1).

$$a_{grav,n} \gg a_{multibody,n} \tag{1}$$

This can cause numerical problems in which the smaller accelerations are lost due to precision limitations in floating-point data types (as has been observed in LEO sims of ISS and SSRMS). To address this issue, rather than applying the full gravitational acceleration at the level of the multibody accelerations, a differential gravity model is used as follows:

1. For all bodies, $n$, calculate the differential gravity (expressed as $\Delta a_{grav,n}$), which is the difference between the gravitational acceleration of the system (given by $a_{grav,sys}$) and the gravitational acceleration of the body (given by $a_{grav,n}$) as in (2):

$$\Delta a_{grav,n} = a_{grav,n} - a_{grav,sys} \quad (2)$$

2. Solve the EOMs for the acceleration (expressed as $a_n$) with the differential gravity applied at the level of the multibody accelerations (given by $a_{multibody,n}$) as in (3):

$$a_n = a_{multibody,n} + \Delta a_{grav,n} \quad (3)$$

3. Add the gravitational acceleration of the system to the acceleration of the root body (given by $a_{multibody,root}$) prior to state integration to solve for the total acceleration of the root body (expressed as $a_{root}$) as shown in (4):

$$a_{root} = a_{multibody,root} + a_{grav,sys} \quad (4)$$

In accounting for gravity in this way, the scale of the gravitational acceleration of the system as a whole does not prevent the smaller accelerations of the multibody subsystem from being accounted for.

An additional consideration is the choice of the numerical integrator. Orbital dynamics simulations and multibody dynamics simulations do not necessarily benefit from employing the same method of numerical integration, as each technique may be better suited for some situations more than others. To avoid the decision of selecting one integrator to maximize the performance of one aspect of the simulation at the expense of the other, Kuo and Nguyen developed a Taylor series-based integrator for propagating multibody translational states in an orbital setting while integrating joint states with an Euler-Cromer integrator [14]. This technique used one method to numerically integrate the orbital dynamics of the base body and another method to integrate the multibody dynamics of the remaining rigid and flexible degrees of freedom (DOFs). This allows MBJEOD to take advantage of numerical integration techniques that are time tested and trusted for both orbital dynamics and multibody dynamics without the need to compromise on stability in either aspect.

For the orbital DOFs, a third order Taylor series-based integration (with backwards difference of the acceleration ($\ddot{x}_o$) used as an approximation of the jerk) is used to solve for the position, ($x_o$) and the velocity ($\dot{x}_o$) of the base body in time ($t$) at a fixed time step size ($\Delta t$) as show in (5) and (6).

$$x_o(t+\Delta t) = x_o(t) + \dot{x}_o(t)\Delta t + \left[\frac{2}{3}\ddot{x}_o(t) - \frac{1}{6}\ddot{x}_o(t-\Delta t)\right](\Delta t)^2 \quad (5)$$

$$\dot{x}_o(t+\Delta t) = \dot{x}_o(t) + \frac{3}{2}\ddot{x}_o(t)\Delta t - \frac{1}{2}\ddot{x}_o(t-\Delta t)\Delta t \quad (6)$$

For the rigid and flexible DOFs, an Euler-Cromer integration is used to solve for the velocity ($\dot{x}_m$) and position ($x_m$) using the acceleration ($\ddot{x}_m$) as shown in (7) and (8).

$$\dot{x}_m(t+\Delta t) = \dot{x}_m(t) + \ddot{x}_m(t) * \Delta t \quad (7)$$

$$x_m(t+\Delta t) = x_m(t) + \dot{x}_m(t+\Delta t) * \Delta t \quad (8)$$

As each DOF is integrated independently, the order in which these computations are completed should have no effect on the results even though they are using different algorithms. Because the algorithms selected are more stable for each type of DOF and the step size is the same, the result of the integration is more consistent for all DOFs than selecting either of the integration methods alone.

## SIMPLIFYING ASSUMPTIONS FOR REAL-TIME APPLICATIONS

In order to provide real-time simulation for training applications, MBJEOD offers options for lower fidelity computation. By default, MBJEOD will call the JEOD gravitation function to calculate the gravity acceleration vector and gravity gradient matrix for each body in the group. If instead, the gravity gradient is assumed to be constant over the space occupied by the multibody system, the JEOD gravitation function can be called a single time at the system CoM and the acceleration at the rest of the bodies can be estimated by taking the product of the gravity gradient matrix (given by $\frac{\partial g}{\partial x_{sys}}$) and the position vector of the body CoM with respect to the system CoM (given by $\vec{r}_{CoM_n\ wrt\ CoM_{sys}}$) and summing this with the acceleration vector of the system CoM (given by $a_{grav,sys}$) as is expressed in (9).

For all bodies, $n$:

$$a_{grav,n} = a_{grav,sys} + \frac{\partial g}{\partial x}_{sys} * \vec{r}_{CoM_n\ wrt\ CoM_{sys}} \quad (9)$$

Because this calculation is a single matrix multiplication and vector addition, it reduces the computational demands of the simulation as compared to the complexity of JEOD's more intricate gravitational models. This simplifying assumption is generally safe in LEO for systems as large as the ISS, for which the gravitational acceleration and gradient do not vary much across its span.

Additional computational savings can be generated by decreasing the rate at which the JEOD states are updated. Rather than updating the JEOD states at every time step, the states can be updated at a slower rate, typically equal to that of the fastest environment model or the rate that data is being

logged. This simplification is reasonable because the states of each body do not change so much from one time step to the next that JEOD's environment models would produce drastically different results, especially if the environment model has not yet updated since its last iteration.

## VERIFICATION AND VALIDATION

JEOD has been independently verified and validated against empirical data from orbital vehicle flights [15], so results from JEOD simulations are treated as valid behavior (i.e., the 'truth model'), providing a baseline of comparison for MBJEOD.

MBDyn is also trusted as a source of truth, and because MBDyn is the dynamics engine behind MBJEOD, it can be understood that JEOD simply provides external forces and torques and MBDyn uses them with current non-specialized capabilities. Therefore, the multibody dynamics capabilities of MBJEOD are identical to that of MBDyn and do not require any special comparison beyond verifying by observation that the system of articulating bodies respond to the forces and torques applied by JEOD.

In order to rigorously test MBJEOD's orbital capabilities, a former JEOD-based tutorial simulation which demonstrates many of the core JEOD functionalities is selected as the basis of comparison. The simulation involves a vehicle in orbit around Earth with Sun and Moon third-body effects, non-uniform Earth gravity, gravity gradient torques, and atmospheric and radiation effects. For this V&V, we replace the mass properties of the example vehicle with those of a model of the ISS (I) with the SSRMS (S) and a payload (P) (referred to here as ISP) in a fixed configuration. By comparing the position of the CoM of a single rigid body in a pure JEOD simulation with that of an equivalent MBJEOD simulation, MBJEOD's performance for orbital dynamics can be assessed.

After establishing the performance of MBJEOD for a single rigid body, the next step is to verify the performance for a collection of rigidly connected rigid bodies utilizing the attachment capabilities of MBDyn in MBJEOD. For this simulation, we use a multibody model of the ISP with fixed joints and identical mass properties to the previous simulation. Once again, the orbital behavior of the CoM is the primary point of comparison for the correlation, as both models would theoretically produce identical behaviors.

With MBJEOD demonstrated to perform comparably to JEOD with single bodies and assemblies of rigidly connected bodies, the next step is to check the orbital behavior of assemblies of articulating bodies as compared to rigidly connected bodies. For this case, the rigidly attached ISP model from the previous case will be compared to an identical model with unlocked joints that are free to rotate. In making this comparison, it can be verified whether the articulating multibody dynamics adversely affect the orbital dynamics, confirming whether the orbital dynamics are retained for any topology or on-orbit operational scenario. These three comparisons for the V&V are summarized in Table 2.

**TABLE 2:** V&V COMPARISON TESTS CASES

| | Using this sim as a baseline … | And comparing it to this sim … | Proves that MBJEOD … |
|---|---|---|---|
| 1 | - JEOD<br>- Single body<br>- ISP mass properties | - MBJEOD<br>- Single body<br>- ISP mass properties | replicates JEOD's behavior for a single orbiting body |
| 2 | - MBJEOD<br>- Single body<br>- ISP mass properties | - MBJEOD<br>- Multiple bodies<br>- locked into position<br>- ISP mass properties | properly incorporates MBDyn's attachments between bodies |
| 3 | - MBJEOD<br>- Multiple bodies<br>- locked into position<br>- ISP mass properties | - MBJEOD<br>- Multiple bodies<br>- free to move<br>- ISP mass properties | can model multibody dynamics in an orbital setting |

## RESULTS AND DISCUSSION

Figure 8 shows the position of the CoM of the JEOD simulation with mass properties of ISS with SSRMS and payload (or ISP) described in the previous section. This is presented to provide a sense of scale for the differences that follow.

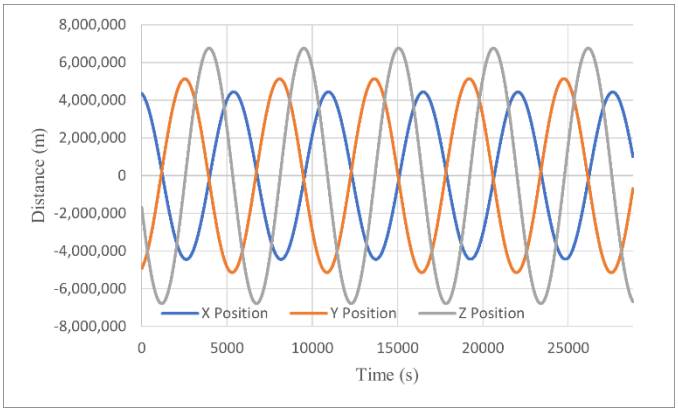



**FIGURE 8:** COM POSITION OF STANDARD JEOD SIMULATION WITH ISP MASS PROPERTIES

When simulating the equivalent case in MBJEOD, with a single body with identical mass properties, the difference in the CoM positions of the MBJEOD orbital system and the JEOD orbital system are plotted in Fig. 9. Over the course of 8 hours (i.e., approximately 5 orbits around Earth), the difference in each coordinate varies by less than 6 mm, which is more than acceptable when considering the absolute positions are measured in millions of meters. These differences in position can be explained by the use of entirely different numerical

integration techniques, which should be expected to produce different results. The good agreement between these simulations demonstrates that the orbital behavior of JEOD is maintained in MBJEOD for single body cases (Table 2 Case 1).

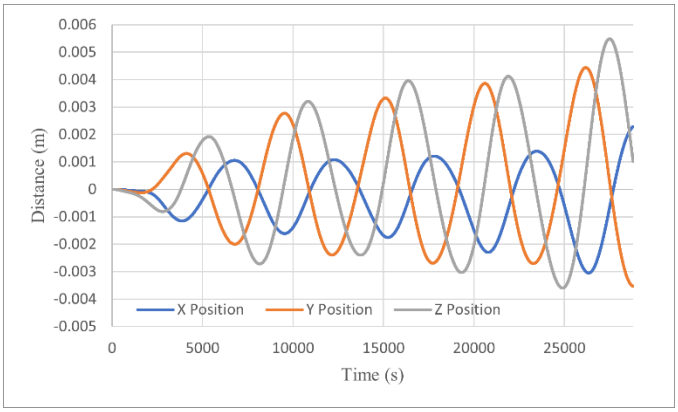



**FIGURE 9:** DIFFERENCE OF COM POSITION OF SINGLE BODY MBJEOD AND JEOD SIMULATIONS WITH IDENTICAL MASS PROPERTIES

When comparing the position of the CoM for the rigidly connected multibody case in MBJEOD and the equivalent mass single body case in MBJEOD used in the previous comparison, the differences are plotted in Fig. 10 below. Over the course of the 8 hours, the difference in each coordinate varies by less than 5 mm, which is once again completely acceptable. Because this case compares two MBJEOD models, the integrator is no longer a potential source of differences. In this case, the difference in behavior most likely come from numerical errors due to force and torque propagation through the chain of bodies. The minute differences between these two simulations demonstrate that the mass properties of interconnected MBDyn topologies, such as those used in ISS simulations with SSRMS and a payload (again ISP) are properly accounted for in the orbital calculations of MBJEOD (Table 2 Case 2).

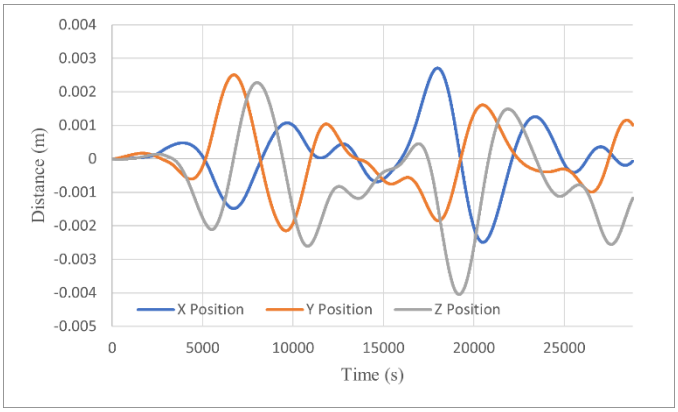



**FIGURE 10:** DIFFERENCE OF COM POSITION OF SINGLE BODY AND MULTIBODY MBJEOD SIMULATIONS WITH IDENTICAL MASS PROPERTIES

When simulating the equivalent case with freely rotating joints, the differences in the CoM positions are plotted in Fig. 11 below. Over the course of 8 hours, the distance between the two models is noticeably larger than the previous cases, with differences growing over time. Notably, the differences grow beyond 1 m after the first 4 hours of the simulation. While there is a larger difference in this case than the previous cases, it is important to note that the changing orientation of the bodies as the joints articulate affects the shape of the topology as a whole. This in turn affects the inertial properties of the system as well as the other environmental effects modeled by JEOD, such as the atmospheric effects and radiation effects, which may provide valid dynamical reasons for the differences seen here. Another contributing factor to the differences may come from the fact that for all of the simulations up until this point, the DOFs related to the multibody dynamics of the system have been entirely inactive, so the introduction of joint accelerations in this simulation may have introduced new numerical differences in the solution to the EOMs.

Regardless of the exact causes of the differences, they are still relatively small for short duration simulations and serve our purposes well (Table 2 Case 3). With this understanding, MBJEOD can be confidently used for multibody orbital dynamics.

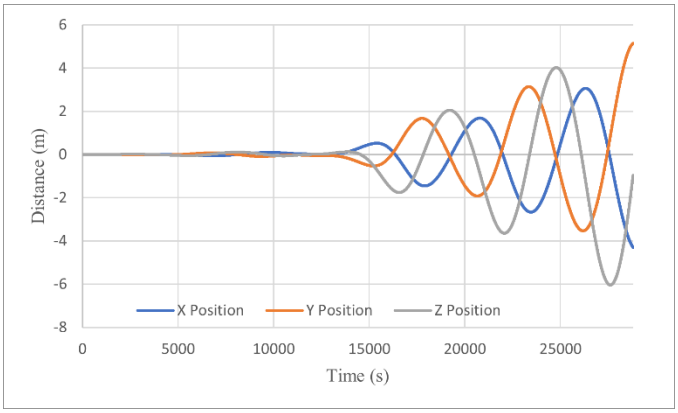



**FIGURE 11:** DIFFERENCE OF COM POSITION OF RIGIDLY CONNECTED AND FREELY ROTATING MBJEOD SIMULATIONS

## APPLICATIONS

MBJEOD is currently used to support a number of programs and projects at the NASA JSC, varying from engineering analyses to training. As was discussed previously, operational planning for HTV-1 was performed jointly by NASA, NASDA, and CSA. The development of the capture envelope along with analyzing SSRMS dynamics during berthing of the HTV Passive Common Berthing Mechanism (PCBM) and the ISS Node 2 Active Common Berthing Mechanism (ACBM) represented its own set of simulation challenges [16].

One of the principal users of the MBJEOD capabilities to date is Training Systems for the 21st Century (TS21) (see Fig.

12) [17]. As pointed out in [18], "TS21 provides the core capabilities for training flight controllers, crew, payload operators, international partners and commercial crew flight controllers ... and models ISS systems and environments with sufficient fidelity for training students on all aspects of ISS operations, including visiting vehicle rendezvous & docking." Moreover, TS21 provides capabilities to interface with commercial and International Partners like JAXA for joint mission simulations of their vehicles (i.e., HTV) with the ISS using Trick implementation of the Defense Modeling and Simulation Office's (DMSO) High Level Architecture (HLA) [19,20].



**FIGURE 12:** TS21 CUPOLA TRAINING

As simulations requirements expand beyond LEO, MBJEOD is now coming into play for applications associated with the Gateway [21] and the Human Landing System (HLS) [22] projects as part of the overarching Artemis Program. There are numerous integrated multibody and orbital dynamics needs for these systems ranging from articulating solar arrays, antennae, deployment mechanisms, and of course, robotic systems.

## CONCLUDING REMARKS

MBJEOD provides a clean interface to unify the strengths of MBDyn and JEOD to produce a package capable of orbital multibody dynamics modeling. It does this in a way that requires no changes to either of the base software packages, allows users to easily adapt input files for initialization with limited modification, and provides capabilities to support any topology or on-orbit operational scenario. MBJEOD is validated against JEOD for orbital behavior in cases with single bodies, multiple bodies with locked joints, and cases in which the joints are free to move, demonstrating that the multibody behavior does not interfere with the orbital behavior. MBJEOD has an ongoing role in supporting training, procedures development, and analysis for operations on the ISS as well as supporting the Gateway and HLS projects under Artemis as NASA returns to the moon.

## ACKNOWLEDGMENT

The work described in this paper was performed entirely within the Simulation and Graphics Branch of the Software, Robotics, and Simulation Division of the NASA JSC Engineering Directorate.

## REFERENCES

[1] Brain, TA, Kovel, EB, MacLean, JR, & Quiocho, LJ. "Advancement of Contact Dynamics Modeling for Human Spaceflight Simulation Applications." Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6: 13th International Conference on Multibody Systems, Nonlinear Dynamics, and Control. Cleveland, Ohio, USA. August 6–9, 2017. V006T10A007. ASME. https://doi.org/10.1115/DETC2017-67936

[2] Quiocho, LJ, Huynh, A, & Crues, EZ. "Application of Multibody Dynamics to On-Orbit Manipulator Simulations." Proceedings of the ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6: 5th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B, and C. Long Beach, California, USA. September 24–28, 2005. pp. 81-87. ASME. https://doi.org/10.1115/DETC2005-85545

[3] MacLean, JR, Huynh, A, & Quiocho, LJ. "Investigation of Boundary Conditions for Flexible Multibody Spacecraft Dynamics." Proceedings of the ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 5: 6th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B, and C. Las Vegas, Nevada, USA. September 4–7, 2007. pp. 17-25. ASME. https://doi.org/10.1115/DETC2007-35511

[4] Ghosh, TK and Quiocho, LJ. "Development and Evaluation of an Order-N Formulation for Multi-flexible Body Space Systems", EUROSIS 11th Annual Industrial Simulation Conference, Ghent, Belgium. 2013.

[5] Huynh, A, Brain, TA, MacLean, JR, & Quiocho, LJ. "Evolution of Flexible Multibody Dynamics for Simulation Applications Supporting Human Spaceflight." Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 6: 12th International Conference on Multibody Systems, Nonlinear Dynamics, and Control. Charlotte, North Carolina, USA. August 21–24, 2016. V006T09A055. ASME. https://doi.org/10.1115/DETC2016-60108

[6] National Aeronautics and Space Administration. "JSC Engineering Orbital Dynamics (JEOD) [Source Code]." 2023. https://github.com/nasa/jeod

[7] Paddock, E, Lin, A, Vetter, K & Crues, E. "Trick - A Simulation Development Toolkit." AIAA Modeling and Simulation Technologies Conference and Exhibit. August 2003. AIAA. https://doi.org/10.2514/6.2003-5809
[8] Penn, J, and Lin, A. "The Trick Simulation Toolkit: A NASA/Opensource Framework for Running Time Based Physics Models." AIAA Modeling and Simulation Technologies Conference. January 2016. AIAA. https://doi.org/10.2514/6.2016-1187
[9] National Aeronautics and Space Administration. "Trick Simulation Environment [Source Code]." 2013 https://github.com/nasa/trick
[10] https://images-assets.nasa.gov/image/iss021e016825/iss021e016825~orig.jpg
[11] https://images-assets.nasa.gov/image/iss063e101293/iss063e101293~orig.jpg
[12] https://images-assets.nasa.gov/image/iss031e070798/iss031e070798~orig.jpg
[13] Blochwitz, T, Otter, M, Arnold, M, Bausch, C, Clauß, C, Elmqvist, H, Junghanns, A, Mauss, J, Monteiro, M, Neidhold, T, Neumerkel, D, Olsson, H, Peetz, J, & Wolf, S. "The Functional Mockup Interface for Tool independent Exchange of Simulation Models." Proceedings of the 8th International Modelica Conference. 2011. https://doi.org/10.3384/ecp11063105.
[14] Nguyen, L. "A Numerical Integration Method for the Translational Orbital Dynamics." Internal L-3 Communications Memo. February 08, 2012.
[15] Jackson, AA and Thebeau, CD. "JSC Engineering Orbital Dynamics (JEOD) Top Level Document", 2023. https://github.com/nasa/jeod/blob/main/docs/JEOD.pdf
[16] Nguyen, PK, and Hiltz, M. "RMS Operations Support: From the Space Shuttle to the Space Station." Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001. https://robotics.estec.esa.int/i-SAIRAS/isairas2001/papers/Paper_AM088.pdf
[17] Miller, R. "View of the Cupola with the Mobile Servicing System Robotic Workstation Cupola Training Mockup Space Station Training Facility." 2019. https://scalar.chapman.edu/scalar/interior-space---a-visual-exploration-of-the-international-space-station-photographs-by-paolo-nespo/media/cupola-trainer
[18] National Aeronautics and Space Administration. "JSC Exceptional Software – Winner List." Accessed March 5, 2024. https://jscexceptionalsoftware.jsc.nasa.gov/WinnerList.cfm
[19] Crues, E, Lin, A, & Hasan, D. "Integrating HLA into the Trick Simulation Development Toolkit." AIAA Modeling and Simulation Technologies Conference and Exhibit. August 2003. AIAA. https://doi.org/10.2514/6.2003-5810
[20] National Aeronautics and Space Administration. "TrickHLA: An IEEE 1516 High Level Architecture (HLA) Simulation Interoperability Standard Implementation for Trick Base Simulations [Source Code]." 2020. https://github.com/nasa/TrickHLA
[21] National Aeronautics and Space Administration. "Gateway – NASA." Accessed March 5, 2024. https://www.nasa.gov/mission/gateway/
[22] National Aeronautics and Space Administration. "Human Landing Systems – NASA." Accessed March 5, 2024. https://www.nasa.gov/reference/human-landing-systems/