National Aeronautics and Space Administration

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

IDETC-CIE MSNDC
28-August-2024

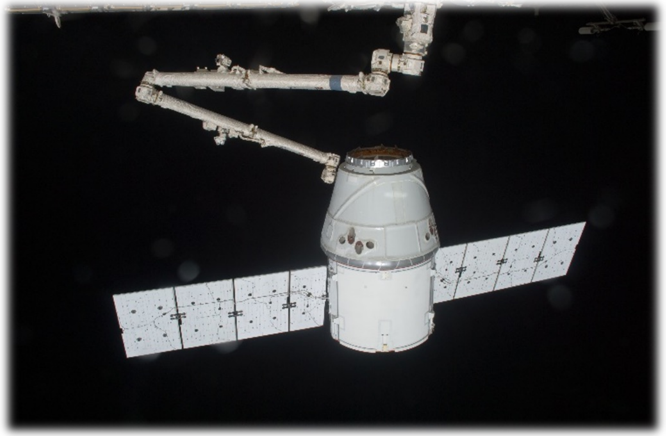
Christopher C. Sullivan – METECS

Thomas A. Brain – METECS

Leslie J. Quiocho – NASA Johnson Space Center

Orbital Multibody Dynamics Modeling Need

Northrop Grumman Cygnus



SpaceX Dragon

- Human spaceflight operations often involve complex multibody dynamics.
- The need for analysis, procedures development, and training for vehicle captures with the CSA Space Station Remote Manipulator System (SSRMS) from the International Space Station (ISS) drove the need for multibody orbital dynamics simulation.
- JSC had capabilities to model multibody and orbital dynamics independently, but no need to do this simultaneously before these operations.

JAXA HTV

Existing Orbital and Multibody Models

MBDyn – Multibody Dynamics
- Bodies – kinematic, rigid, flexible
- Topology – articulating chains, branched trees, or closed-loop
- Optimized for efficiency for real-time and non-real-time simulations

JEOD – JSC Engineering Orbital Dynamics
- Bodies – rigid
- Topology – locked into position (i.e., fixed)
- Trajectories – Low Earth Orbit (LEO), lunar operations, interplanetary, deep space
- Environment models, force interaction models, can interface with Guidance, Navigation, and Control (GN&C) system models

Early Attempts at Integrating the Models

Simplifying factors:
- Both work within JSC's Trick simulation environment framework.
- Both are developed in-house.

Complicating factors:
- Quickly became cumbersome (i.e., too ad hoc).
- Difficult to maintain because MBDyn and JEOD continued to be developed.
- At the time, interface packages (e.g., Functional Mock-up Interface) were less widely available and provided less customizability than an in-house solution.

MBJEOD Design Goals

1. To not change MBDyn or JEOD source code and delivered functions – 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. As an example, 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.

Design Approach – Core Paradigms

MBJEOD mirrors the design paradigms of both MBDyn and JEOD.

The primary interface classes:

- MBJeodBody
- MBJeodGroup
- MBJeodDyn

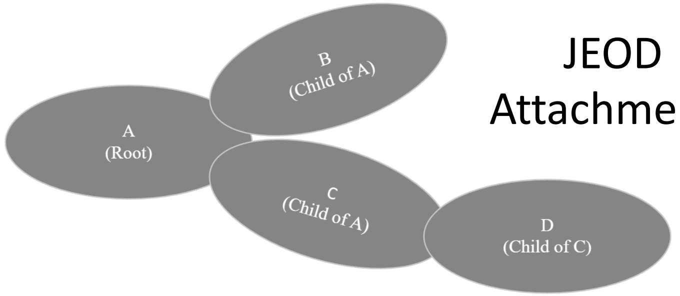
MBJeodBody:

- Inherits from JEOD's DynBody class and contains an instance of MBDyn's BODY_DATA class.
- Allows for body input data to be provided in the structures from either MBDyn or JEOD.
- Contains a flag specifying the input source for the mass properties and initial state.
- Internal initializations and computations ensure all data is digested as expected by users of both MBDyn and JEOD and stored in the internal structures.

Design Approach – Contrasting Body Attachments

JEOD

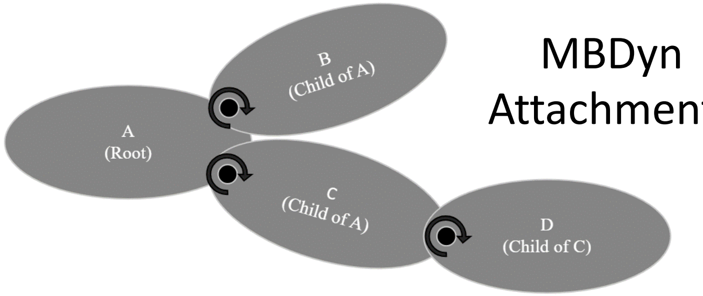
- Mass properties of a collection of bodies are recursively accumulated and remain constant.



JEOD Attachments

MBDyn

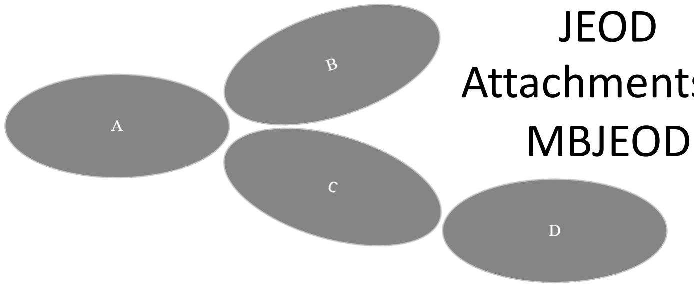
- Joints between bodies may permit movement, so mass properties of the group of bodies may change over time.



MBDyn Attachments

MBJEOD

- MBDyn’s attachment structure is maintained.
- JEOD’s attachment structure is not maintained.
- The JEOD model applies gravitation and environmental effects on each body as if it was independent.



JEOD Attachments in MBJEOD

Design Approach – Core Paradigms (cont.)

MBJeodGroup:

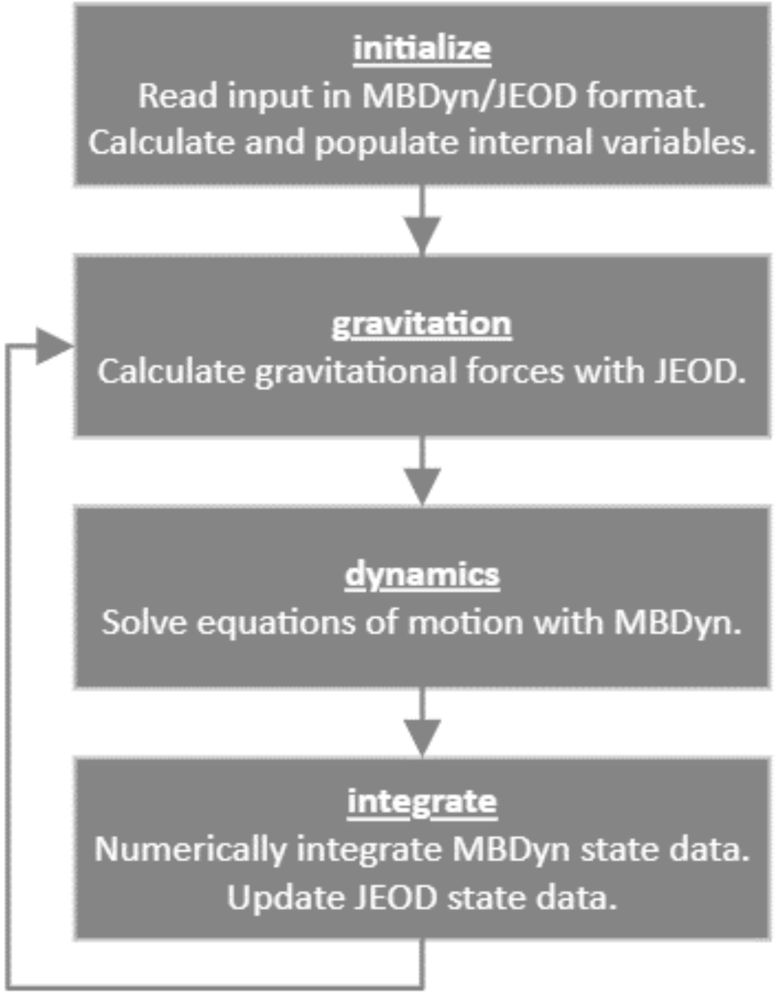
- Models an assembly of MBJeodBodies.
- Tracks the list of connected bodies in the simulation.
- Calculates system level properties.
- Manages the output of group-level data.
- Interfaces JEOD gravity models with MBDyn body instances.
- Requires special numerical considerations.
- Options to apply simplifying assumptions depending on the level of fidelity required.

Design Approach – Core Paradigms (cont.)

MBJeodDyn:

- Stores information related to the system as a whole.
- Executive MBJEOD class that controls the programmatic flow.

1. **initialize** – using either MBDyn or JEOD inputs, calculates and populates the internals of both MBDyn and JEOD.
2. **gravitation** – calls JEOD functions for environmental and gravitational effects using the states of the JEOD bodies.
3. **dynamics** – plugs environmental effects into MBDyn as external forces/torques and uses MBDyn's EOM solver.
4. **integrate** – calls MBDyn's integrator to update the MBDyn body states and uses them to update the JEOD body states.



Design Approach – Auxiliary Models

Some assumptions in JEOD and MBDyn models are invalid in an orbital multibody context.

JEOD's GravityTorque
- Models torque due to misalignment of the gravity gradient and the group's inertial tensor.
- Assumes the torque should be applied to the entire group at the root body.

MBJeod's MBGravityTorque
- Calculates the torque for each individual body in the group.

MBDyn's Points of Interest (POIs)
- Tracks state data at any point on a body expressed in any frame defined in the system.
- Assumes the total acceleration of a point is of interest.

MBJeod's NonGravAccels
- Removes the gravitational acceleration of the group from the POI, which can be useful when modeling sensors.

Numerical Considerations - Scale

The joint accelerations (e.g., motor torques, joint friction, flexibility, etc.) are orders of magnitude smaller than the gravitational accelerations, causing numerical problems due to precision.

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

Rather than applying the full gravitational acceleration to each body, a differential gravity model is used:

1. For all bodies, calculate the differential gravity:

$$\Delta a_{grav,n} = a_{grav,n} - a_{grav,sys}$$

2. Solve the EOMs with the differential gravity applied with the multibody accelerations:

$$a_n = a_{multibody,n} + \Delta a_{grav,n}$$

3. Add the gravitational acceleration of the system to the acceleration of the root body prior to state integration to solve for the total acceleration of the root body:

$$a_{root} = a_{multibody,root} + a_{grav,sys}$$

Numerical Considerations - Integration

Orbital dynamics and multibody dynamics simulations often require different integrators.

At NASA JSC, Kuo and Nguyen developed a specific integrator for this case:

- Orbital DOFs of the root body – third order Taylor series-based integration (with jerk approximated by backwards difference of acceleration)

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

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

- Remaining multibody DOFs – Euler-Cromer integration

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

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

Simplifying Assumptions for Real-Time Applications

Linearized gravity:

- Default – The gravitational acceleration and gravity gradient are calculated for every body.
- By assuming the gravity gradient is constant for the system, JEOD's gravitation function can be called once at the system CoM and the acceleration at the bodies can be estimated.

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

- This assumption is generally safe in LEO for systems as large as the ISS.

Reduced JEOD update rate:

- Default – JEOD states are updated at every time step.
- By assuming the output of JEOD's environment models is similar from one time step to the next, the states can be updated at a less frequent rate, resulting in faster frames.
- This assumption is safe because the bodies' states don't always change much from step to step, and some environment models update slower than the dynamic rate.

Verification and Validation Baseline

Multibody Dynamics

- MBJEOD is pure MBDyn with JEOD providing external forces/torques.
- Does not require special comparison beyond verification by observation.
- MBDyn has an extensive V&V history, so we only need to know if everything is plugged in correctly.

Orbital Dynamics

- JEOD is independently verified and validated against empirical data from orbital flights.
- Need to determine if orbital effects are accurate in MBJEOD by comparing to JEOD.
- Comparison simulation for correlation based on a former JEOD tutorial simulation:
  - One vehicle orbiting Earth with non-uniform gravity
  - Third-body effects from Sun and Moon
  - Gravity gradient torques
  - Atmospheric and radiation effects
  - Updated mass properties to match the ISS (I) with SSRMS (S) and a payload (P) (ISP) in a fixed configuration.

Verification and Validation Plan

1. Compare the position of the CoM of a single rigid body in a pure JEOD simulation with that of an equivalent MBJEOD simulation of a single rigid body with the same mass properties.

2. Compare the position of the CoM of a single rigid body in an MBJEOD simulation with that of an equivalent MBJEOD simulation with a multibody model of the ISP with fixed joints and identical mass properties.

3. Compare the position of the CoM of a multibody model of the ISP with fixed joints in an MBJEOD simulation with that of an equivalent MBJEOD simulation with unlocked joints that are free to rotate.

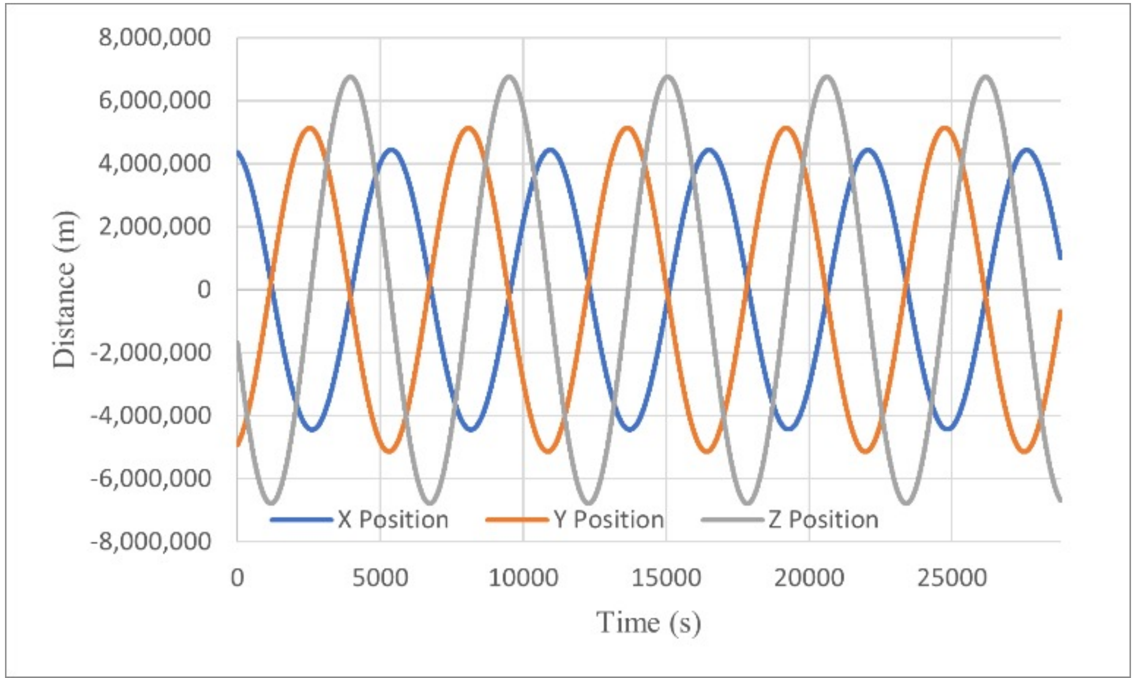
TABLE 2: V&V COMPARISON TESTS CASES

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

Baseline JEOD Simulation

The position of the CoM of the baseline JEOD simulation with mass properties of ISS with SSRMS and payload (ISP) in a fixed configuration.

CoM Position



Baseline JEOD Simulation vs MBJEOD with Single Body

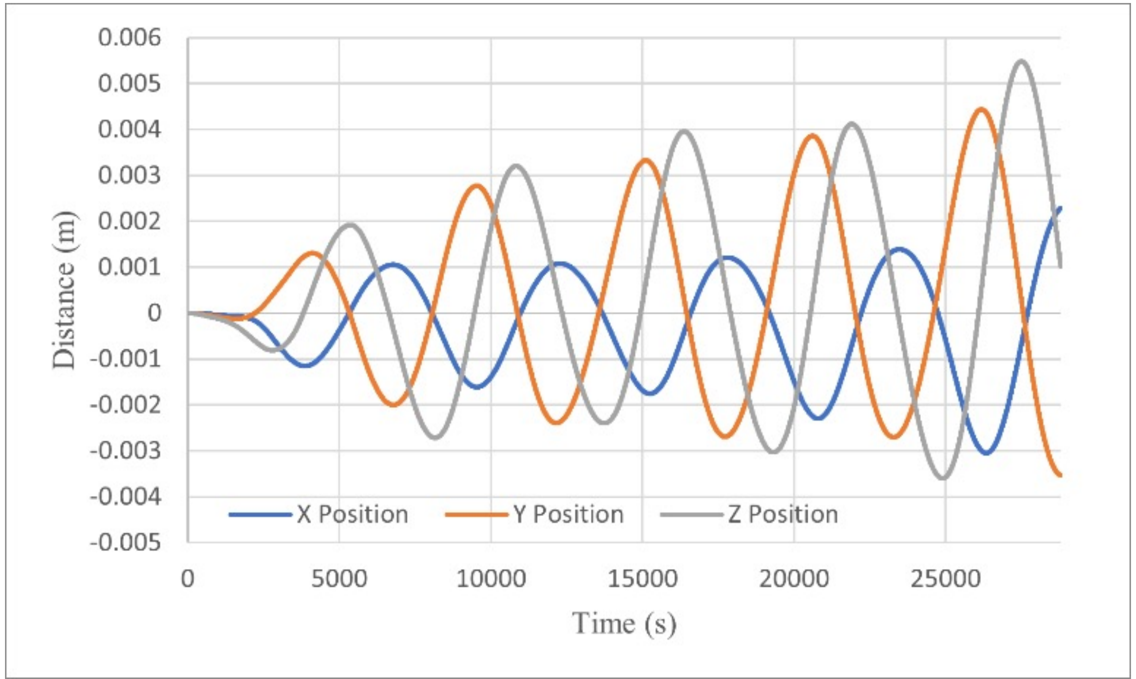
Magnitude of difference

- 6mm over the course of 8 hours (approximately 5 orbits).

Source of difference

- Different solvers and numerical integration techniques between JEOD and MBJEOD.

Difference of CoM Position



MBJEOD with Single Body vs MBJEOD with Multiple Bodies Locked

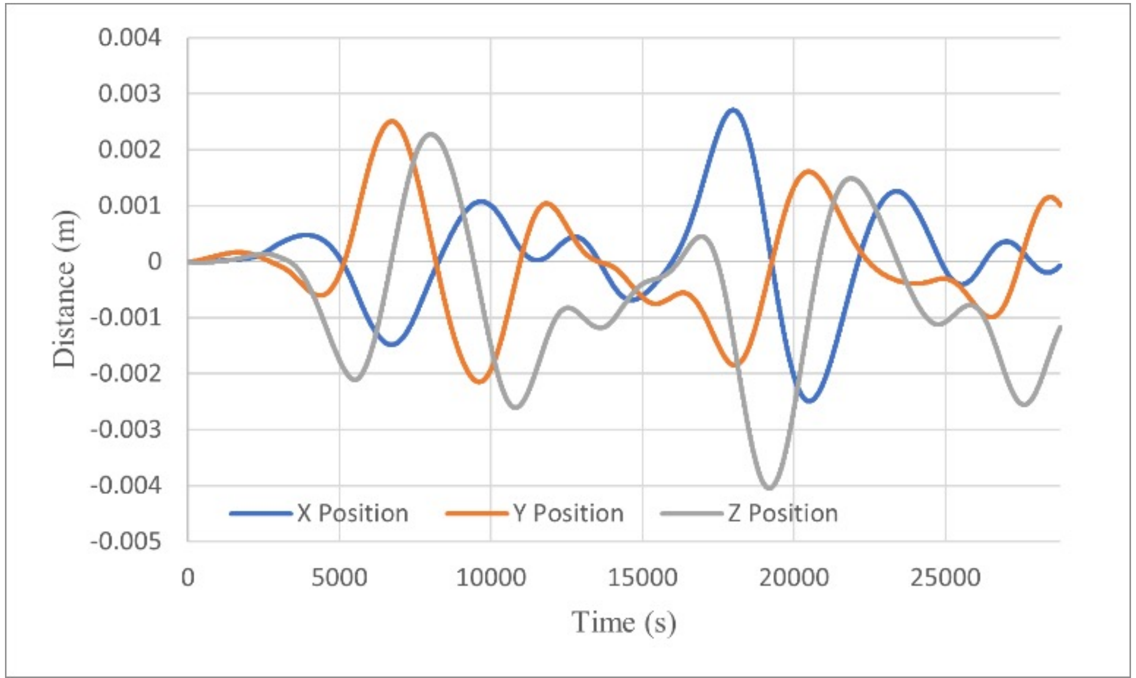
Magnitude of difference

- 5mm over the course of 8 hours (approximately 5 orbits).

Source of difference

- Most likely from numerical errors due to force and torque propagation through the chain of bodies.

Difference of CoM Position



MBJEOD with Multiple Bodies Locked vs Free

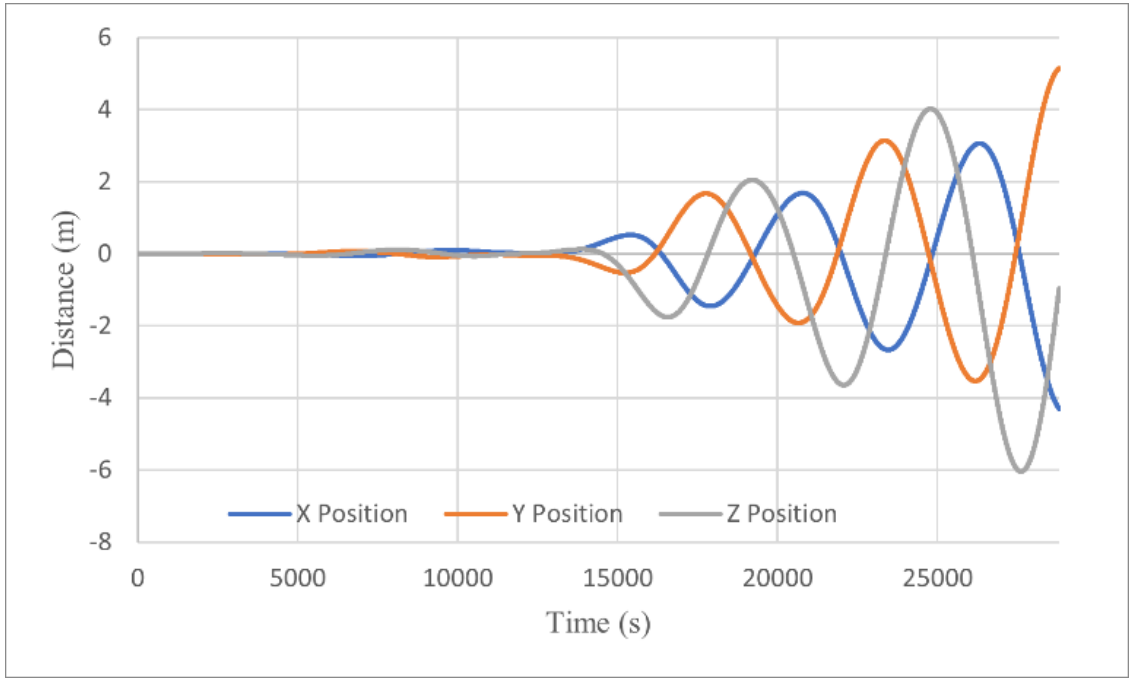
Magnitude of difference

- >1m after 4 hours (~2 orbits) and growing (acceptable for most uses).

Source of difference

- Changing orientation of the bodies (legitimate dynamical differences).
- Non-zero joint accelerations.

Difference of CoM Position



MBJEOD’s Applications

Operational planning for HTV-1

- Performed jointly by NASA, NASDA, and CSA.
- Development of the capture envelope.
- Analysis of SSRMS dynamics during berthing.

Training Systems for the 21st Century (TS21)

- Used for training flight controllers, crew, payload operators, international partners, commercial crew flight controllers.
- Interfaces with commercial and International Partners for joint mission simulations of their vehicles.

Artemis Program (in development)

- Gateway – articulating solar arrays, antennae, robotic manipulator systems
- Human Landing System (HLS) – Deployment mechanisms

TS21 Cupola Training

Conclusions

MBJEOD provides a clean interface between MBDyn and JEOD to produce a package capable of modeling multibody dynamics in the orbital setting.

It requires no changes to either of the base software packages, allowing users to easily adapt input files for initialization with limited modification, and provides capabilities to support any topology or on-orbit operational scenario.

It 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.

It 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.

Acknowledgement

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

Questions?