Techniques for Using ADAMS in Spacecraft Applications

Walter K. Daniel
Application Engineer
Mechanical Dynamics, Inc.
701 Brazos Street, Suite 500
Austin, TX 78701
United States of America

ADAMS is being used increasingly by spacecraft and launch vehicle manufacturers to study dynamic problems. The types of modeling done by these organizations can be quite different than projects in the automotive, general machinery, and aircraft industries.

The two primary uses of ADAMS for space applications are deployment analysis and control system simulation. Deployment mechanisms that are analyzed include solar arrays, antennas, payload fairings, and spacecraft separations; control system studies include large flexible satellites and space robotics. In addition to these two areas, other miscellaneous topics might be satellite handling fixture analysis, test setup verification, and on-orbit mechanisms anomaly resolution.

This paper covers ADAMS techniques for spacecraft deployment analysis. Topics include deployment springs, bumper modeling, latches, hinge releases, cable-pulley systems, and joint friction.

Kinematics for Model Building

Spacecraft deployables such as solar arrays and antenna reflectors are stowed into small packages for launch. From an ADAMS modeling perspective, building models of these mechanisms in the stowed position can be quite difficult. One approach to overcoming this difficulty is to build the models in the deployed configuration, add motions that will fold the device into the stowed position, run a kinematic simulation, then save the model in the stowed position. The motions then must be deleted from the stowed model and appropriate deployment items such as springs added.

A representative solar array with two booms and three panels is shown in Figure 1. The model was built in the deployed position with revolute joints between the components with the body of the spacecraft fixed to ground. The first hinge moves through 90 degrees while all others move through 180 degrees. Motions of 15 degrees per second (first hinge) and 30 degrees per second (all others) were added and the model run for 6.0 seconds. The configuration shortly before the end of the run is shown in Figure 2. The model was saved in the final stowed position, the motions were deleted, and deployment springs were added.
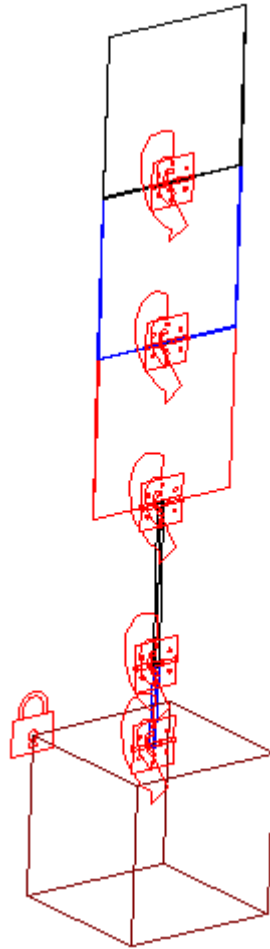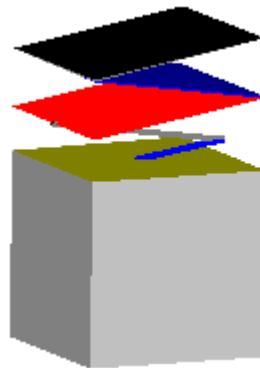
Figure 1. Solar Array in Deployed Configuration

Figure 2. Solar Array in Almost Stowed Configuration

Deployment and Pushoff Springs

Most satellite deployables are driven open by preloaded springs. These elements are straightforward to model in ADAMS/View because spring rate, damping rate, and

preload are displayed in the Modify Spring dialog box. Translational springs are sometimes used for booms but torsional springs are used in most applications. In the example from Figure 2, torsional springs were added at each hinge; after creating the springs, the user must modify each to enter the preload value. Users must carefully check the sign of each preload entry to ensure that the mechanism deploys in the correct direction. In some cases where latches and bumpers are not used, the preload value must be set so that the zero force/torque point occurs at the desired deployed position. Assuming that the deployment starts at zero displacement,

$$torque = k * angle - preload$$

For a 2.0 N-m/rad torsional spring to deploy to 90 degrees (1.57 rad), the preload must be 3.14 N-m. Damping must be present for the model to settle in the final position.

Another type of spring used in spacecraft deployables is the "pushoff" spring. This device is also a preloaded spring that initially pushes a mechanism then loses contact after reaching its free length. A typical ADAMS approach to modeling this behavior was to have a sensor call a script at the spring free length and deactivate the spring. Note that this approach does not directly work in ADAMS/View because a spring is a combination of a force/torque plus graphics; if using a script the best approach is to set the associated SFORCE equal to zero [Reference 1]. A more direct approach that does not involve simulation scripting is to create a force or torque that uses a STEP function to zero the force when the spring reaches the free length. For example, here is a function that returns 1.0 until two markers become 6 distance units apart [Reference 2]:

$$STEP(DM(1,2), 5.9, 1.0, 6.1, 0.0)$$

The DM function is used to determine the distance between markers 1 and 2. When the distance becomes greater than 6.0 units the function quickly drops to zero. A linear spring-type force is:

$$force = k * (x - free\_length)$$

The distance x is determined with the DM function and free length is a constant (perhaps a design variable). Here is a function that describes a spring with free length of 6.0, spring constant of 25.0, and distance measured between markers 1 and 2:

$$STEP(DM(1,2), 5.9, (25.0 * (DM(1,2) - 6.0)), 6.1, 0.0)$$

Pushoff springs are used for initial deployment only so no damping is necessary.

Bumpers

Satellite mechanisms often include bumpers to prevent hinges from rotating too far at the end of the deployment sequence. Bumpers are sometimes used to prevent underdeployment (the rotation of the hinge in the wrong direction). Although these

devices can be modeled in detail with contacts between bodies a more direct method is to use hinge torques with contact functions [Reference 3].

To model a bumper at the end of hinge travel, create a torque between the two bodies at the hinge location. Modify the torque and enter a function such as this one:

$$BISTOP(AZ(1,2), WZ(1,2), 0.0d, 90.0d, 1e5, 1.1, 100, 0.1d)$$

The AZ and WZ functions return the angle and angular velocity about the marker Z axes. This bumper is set at 100 degrees to prevent overdeployment of a 90 degree hinge by more than 10 degrees. The stiffness, exponent, damping, and ramp-up distance values are described in the ADAMS/View function builder documentation. In general, the damping should be fairly high since the bumpers are constructed of energy-absorbing material. Figure 3 shows a hinge angle plot for this example.
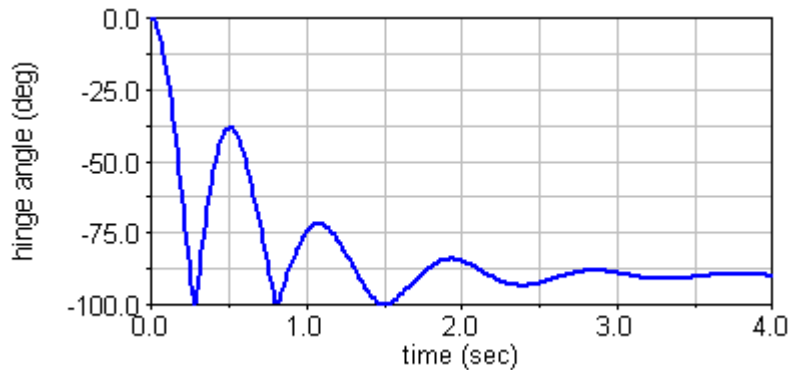

Figure 3. Hinge Angle with Bumper at 100 Degrees

The two-sided BISTOP contact function models bumpers at both ends of hinge travel. In the above example, the hinge angle cannot move below 0.0 degrees in addition to the upper limit of 100.0 degrees. The one-sided IMPACT function can be used in this application but is more difficult to set up than the two limits in the BISTOP.

Latches

Hinge latches are widely used to increase the stiffness of deployed devices. There are at least two approaches used to model latches in ADAMS: constraints and forces. The constraint approach requires simulation scripting while force-based approaches do not.

When the number of latches is small and the order of activation is known, simulation scripting is a straightforward method for modeling [Reference 4]. The user must create a motion on the joint that will be latched; the function used in this motion is not important since it will be replaced during the simulation. The user also needs to create a sensor on the joint that will activate when it reaches the angle at which the latch closes. With these

elements in place, the user must write a simulation script that deactivates the latching motion, starts the run, then activates the motion when the sensor trips. Note that the sensor must be deactivated by the script to prevent it from ending the simulation. A sample ADAMS/Solver script to implement one latch is listed in Table 1.

Table 1. Simulation Script for Latching

```
! Script to latch a joint with a motion of zero velocity
!
! Deactivate latching motion before starting run
DEACTIVATE/MOTION, ID=1
!
! Start simulation
SIMULATE/TRANSIENT, DURATION=0.5, STEPS=50
!
! Sensor has activated so script continues here
!
! Deactivate sensor so it won't keep activating
DEACTIVATE/SENSOR, ID=1
!
! Activate motion and assign velocity of zero
ACTIVATE/MOTION, ID=1
MOTION/1, VELOCITY, FUNC=0.0
!
! Continue simulation with another simulate command
SIMULATE/TRANSIENT, DURATION=0.5, STEPS=50
```

Simulation scripting with constraints becomes difficult when there are more than two latches or the order in which the latches activate is not known. A force-based approach to latching can overcome these difficulties. Much like the bumper modeling discussed previously, one concept is to use a STEP function for latch status and a BISTOP function for the torque.

STEP(AZ(1,2), 88.5d, 0, 89.0d, 1.0) * BISTOP(AZ(1,2), WZ(1,2), 89.5d, 90.5d, 1e5, 1.5, 1e3, 0.1d)

The STEP function evaluates the angle about Z between two markers at the hinge. When the angle increases to 88.5 degrees, the function rapidly increases to 1.0 to simulate the latch closing. The BISTOP function allows free motion between 89.5 and 90.5 degrees at which point the contact function begins to push the joint back towards the center angle of 90 degrees. Note that the BISTOP must have a large amount of damping to be effective. Note also that the problem with this simple formulation is that "unlatching" will occur if the angle moves below 89 degrees due to rebound or torques. Users should check models carefully to make sure that the desired latching remains in effect. It is possible to construct a more complex function to integrate the absolute value of the angular velocity of the joint to create a "one-way" latch.

Releasing Mechanisms

Mechanisms that release include launch vehicle payload fairings and instrument covers. In ADAMS, these devices can be modeled with constraints that are deactivated by simulation scripts [Reference 5].

Constraints that release at a specific time such as a payload fairing bolts require simple scripts. The only lines needed are those to start the simulation, deactivate constraints, and continue the simulation. For example, this script deactivates two joints after running for 0.5 second then continues the simulation for 10.0 seconds:

```
SIMULATE/TRANSIENT, DURATION=0.5, STEPS=50
DEACTIVATE/JOINT, ID=1
DEACTIVATE/JOINT, ID=2
SIMULATE/TRANSIENT, DURATION=10.0, STEPS=100
```

Instrument covers that swing open then detach can be modeled using a sensor with the script. Create a measure of the hinge angle and use it in the sensor. When the angle reaches the release point, the script simply deactivates the joint and restarts the simulation. Make certain that the initial run command has enough duration for the cover to reach the release angle. See Figure 4 for an example model shortly after the cover has been jettisoned.
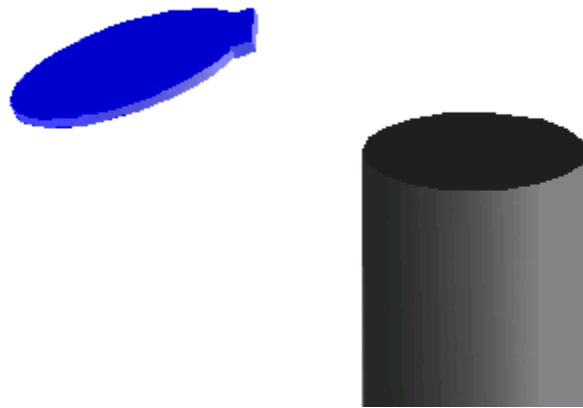

Figure 4. Instrument Cover after Hinge Deactivation

A sensor can be used to model a constraint that breaks under loading. For example, a fixed joint can be used to model a tiedown bolt such as the ones used to hold launch vehicles down on the pad. Set up the sensor to check the reaction force in the fixed joint and to pause the simulation when the breaking load is reached. Deactivate the joint in the script then issue another SIMULATE command to continue.

Cables and Pulleys

An alternative to preloaded spring deployment mechanisms are cable-pulley systems. A major advantage of cables and pulleys is that the deployment motion is slow thus minimizing loads on the structure. The slow motion makes modeling these mechanisms in ADAMS easier since the cable itself does not need to be represented. Two modeling approaches are couplers and forces/torques.

Coupler joints can be used to model very stiff cables that do not stretch. This type of modeling also implies that the cable is not slipping on the pulleys. Consider a solar panel with two hinges connected by a cable wrapped around rollers of the same radius at the hinges. If one hinge is driven open with an electrical actuator the other hinge will be pulled open at the same angular rate. The only ADAMS element needed is a coupler joint with 1:1 angular motion between the two hinges. More complex systems (additional pulleys, different roller radii, etc.) can be modeled using the appropriate scale factors. Be careful with the signs of motion of each joint—make sure that the motion takes place in the expected direction.

A more sophisticated cable-pulley modeling approach is to include stretching of cables. Cables are usually preloaded with springs so the effective stretching stiffness is actually the rate of the spring. Instead of using couplers which apply motion to a joint, this technique calls for applying forces/torques to represent the load that the cable imparts. In the solar panel example above the coupler is replaced with a torque on the outer hinge:

$$M_{outer} = kR_{outer}(R_{outer}\theta_{outer} - R_{inner}\theta_{inner}) + F_{preload}R_{outer}$$

The angles at the two hinges are defined as measures for use in functions. This torque-based technique was used successfully to model solar array deployment on a NASA astronomy spacecraft in 1998 [Reference 6]. As always, be careful with the signs of measures, forces, and torques when developing models.

Joint Friction

Joint friction can play a significant role in spacecraft mechanism performance especially when ground testing is used to predict flight motion. Although translational, cylindrical, and spherical joints have applications, most spacecraft mechanisms are comprised of revolute joints. The friction model derives torque from three sources: radial preload, axial preload, and reaction force. In general, radial preload will be zero since interference fit of hinge pins is rarely used in satellites. Axial preload might be present due to springs holding pins in place. Reaction forces that are present during ground tests will not be present on orbit. As a result, ground test motion will be different than flight motion. The best practice is to build an ADAMS model of the ground test configuration (with gravity and supports) and a separate model of the flight configuration (no gravity). Make certain the ground test model accurately reflects motion in the lab before attempting to predict on-orbit motion.

To illustrate the effects of joint friction, examine the plot in Figure 5. This plot is from an ADAMS model of a two-panel solar array deployment with the inner hinge driven by an actuator and the outer hinge opened by a cable-pulley system [Reference 6]. The cable was tensioned with a spring and modeled with a torque on the outer hinge. Note that joint friction causes the outer hinge to stop several times during the deployment, motion that would not be predicted by a coupler joint representation of the cable-pulley mechanism.
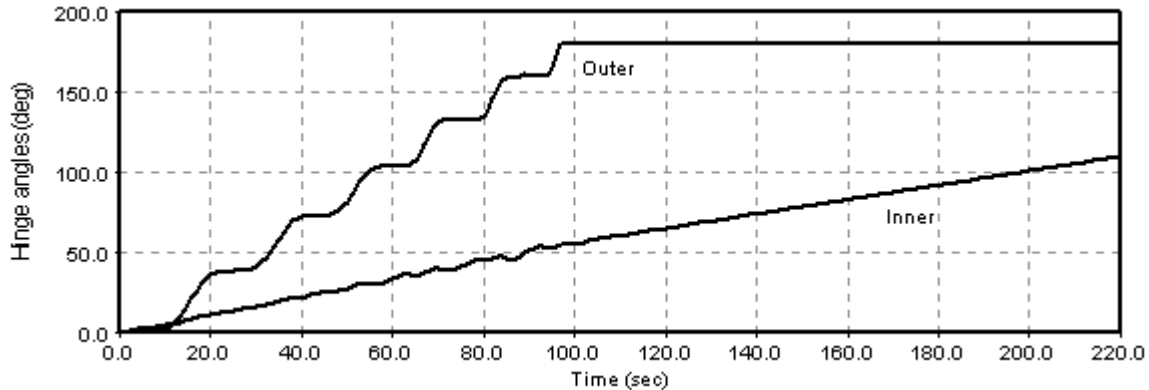

Figure 5. Plot of Hinge Angles for Cable-Pulley Solar Array Deployment Model

References

1. "Methods to 'turn off' springs," http://support.adams.com/ask/askview.asp?ID=3002

2. "Methods to deactivate springs," http://support.adams.com/ask/askview.asp?ID=3185

3. "Adding limits to revolute joint rotation using contact functions,"
http://support.adams.com/kb/faq.asp?ID=kb7707.html

4. "When markers reach angle=PI, make revolute act like a fixed joint,"
http://support.adams.com/kb/faq.asp?ID=kb7291.html

 5. "Turning off joint when sensor is triggered,"
http://support.adams.com/kb/faq.asp?ID=kb7577.html

6. "A Flight Prediction for Performance of the SWAS Solar Array Deployment Mechanism," Gary Sneiderman and Walter K. Daniel, 33rd Aerospace Mechanisms Symposium (NASA/CP-1999-209259), May 1999, pages 43-56