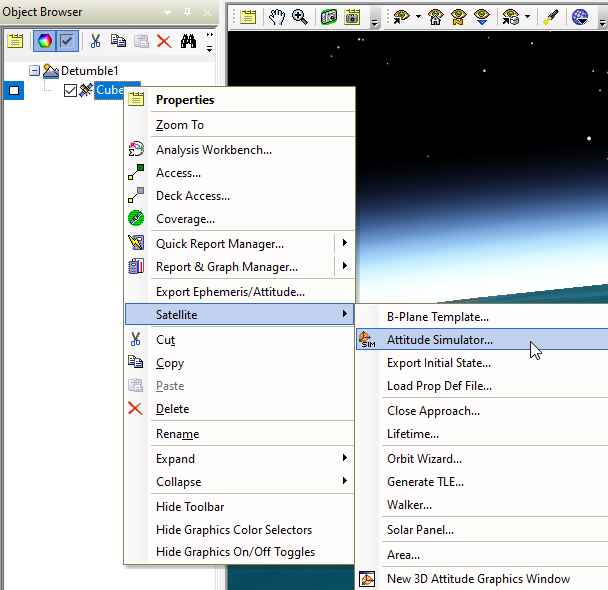
This folder contains four scenarios that demonstrate use of the Attitude Simulator in STK. The Attitude Simulator is available for any satellite object and can be accessed by right clicking on the satellite in the object browser, and selecting Satellite - Attitude Simulator.



The tool allows the user to incorporate their own torque models and momentum biases to implement and test custom attitude control laws. The custom torque models must be written in MATLAB, Perl, Python, or VBScript, and placed into any of the following directories

* <STK install folder>/STKData/Scripting/Attitude
* <STK user area>/Scripting/Attitude
* <STK all users area>/Scripting/Attitude
* <scenario folder>/Scripting/Attitude

The following examples have VBScript and MATLAB scripts located in the scenario folder. To run the MATLAB plugins, you will need to download and install the MATLAB connectors which can be found on the AGI website here.

<https://support.agi.com/downloads/>

A few steps are involved in correctly configuring an Attitude Simulator run. On the main page of the tool, configure the start and stop times of the simulator run. By default, the start and stop times for the run will be the same as the satellite’s ephemeris time interval, which is often far longer than necessary. Be sure to only run the simulator for the interval necessary to conduct or test the attitude sequence. Next the Initial Conditions must be considered. If you would like to use the satellite’s current attitude and rates, hit the “Initialize from current attitude” button. Otherwise, specify the initial orientation and body rates. Momentum bias can also be configured on this main page. The Output section is where the output attitude file can be configured. It is convenient to select “Use for attitude definition”, which ensures that the attitude file created will immediately be loaded for the target satellite after the simulator run has completed.

Most importantly, the user must specify which plugin scripts will be used by the simulator. This can be done in the Configuration page. The user can select Initialization scripts (to be run once before simulation begins), Simulation scripts, and Post Processing scripts (to be run once at the end of the simulation). More information about the format of these scripts can be found in the Help under Attitude Simulator Plugin Points. In general, it is most important to specify a Simulator script, which outputs a torque vector at each compute cycle. Multiple simulator scripts can be used to calculate and output separate torques, i.e. torque due to drag, torque due to solar radiation pressure, and torque due to attitude controls. The attitude simulator will numerically integrate the attitude state using all the torque components. The integration routine, step sizes and tolerances can be configured in the Integrator page. After properly configuring the simulator run, hit the Run button. A progress bar will be displayed in the Status Bar in STK (if this option is selected in the Advanced page). Any errors encountered during the process will be reported in the Message Viewer.

**Description of Example Scenarios**

**DetumbleCubeSat** – This example demonstrates how one might model a simple magnetic torquer used to detumble a CubeSat. The scenario consists of a simple CubeSat in low earth orbit. Right click on the satellite and select Satellite – Attitude Simulator. Notice that the initial conditions for the satellite and large Body Fixed Rates (wx, wy, wz = 14.3, -8.59, 8.59), the CubeSat will tumble at these rates unless an outside torque is applied. Click on the “Configuration…” button, currently VB\_DetumbleTorque.vbs is selected as the Simulation script. If you prefer to use MATLAB, hit Edit, and select the MATLAB\_DetumbleTorque.m script. If you inspect these scripts, you will see that the script brings in the IGRF Magnetic Field vector as an input, and then uses that to compute a torque to counter the rate of change relative to the magnetic field. After the simulation has completed, reset the animation and play forward. You will see how the CubeSat initially tumbles at a high angular velocity, but then gradually the magnetorquer will slow the rate of tumble to a steady state.

**FeedbackControl** – This demonstrates a simple feedback control law. The scenario consists of a satellite in low earth orbit, where the goal is to slew the attitude to a sun pointing profile. Open the Attitude Simulator and notice the initial conditions for the satellite are fixed in the inertial frame. Open the Configuration page and add the script of your choice to the Simulation section. The script brings in the current satellite attitude and the orientation of the SunPointing axes, and calculates a torque to drive the current attitude towards SunPointing. This attitude maneuver happens very quickly, so notice that the simulation time interval is only 10 minutes. Click on the Integrator page, notice that the max step size is 1 second. It is sometimes necessary to configure the integrator if very rapid attitude response is expected. Finally, run the simulation. Reset the animation and play forward with an animation time step of 0.1 seconds or less.

**FeedbackControl\_Targeting** – This scenario consists of two LEO satellites and a number of targets on the ground. The PerfectPointing satellite has a multi-segment attitude profile where it has been configured to slew to target each of the ground locations, and incorporate sun pointing and nadir pointing profiles. Because PerfectPointing uses the pointing models from STK, the slewing will be perfect, with no overshoot or dampening. The AttitudeSimulator satellite has the same orbit as PerfectPointing but will use the Attitude Simulator to integrate the attitude state to try and match the attitude of PerfectPointing. The plugin scripts use the same feedback control law as in the FeedbackControl example, except that the PerfectPointing Body Axes are used as the goal.

**Tracking** – This example is similar to FeedbackControl\_Targeting, in that there is a reference satellite with perfect pointing, and a satellite which uses the Attitude Simulator to try and match the reference satellite’s pointing profile. In this case, the goal is to track a ballistic missile. The simulator plugin scripts are written in MATLAB, and there is one Initialization script which is used to set up the gain values used during simulation.