**Introduction**

This is a simple example that shows how Simulink can integrate with STK, where Simulink controls the simulation clock and pulls analysis data from STK.

The STK analysis derives from a scenario containing a LEO satellite with an attached sensor and transmitter, an area target, and a ground station with an attached receiver. From this base scenario, access is calculated to determine when the satellite is in the sun, when the satellite’s sensor can view the area target, and when the satellite’s transmitter can communicate with the ground station w/ a Bit Error Rate less than 1e-8. The Simulink model consists of a source clock block, a Level-2 S-Function block for the STK component, and three display blocks to view the output from STK. Simulink passes the current simulation time to the s-function block, where STK is queried to determine the current state of access for the three analyses previously mentioned. If access is true, a value of 1 is passed to the display block for that case (Sunlight, Experiment, or CommLink).

**Setup**

The SimulinkToSTK folder contains two MATLAB scripts and one Simulink model. From the command prompt in MATLAB, browse to the folder and open SimulinkToSTK.slx (browsing to the folder ensures that your MATLAB path is correct and it will find the s-function script). Double click on the STK block to see that the STK\_SatelliteAccess\_sFunction.m is selected as the S-function. You can select Edit to view this in more detail. This function will be called at each time update to check the current state of access in the STK scenario (see the Outputs() callback).

The STK scenario must be created before the simulation begins. This is done in the InitFcn callback of the STK S-Function block. Right click on the STK block, select Properties, select the Callbacks tab, and click on InitFcn. You will see that this code starts STK, creates the scenario, and populates all the necessary objects. A cell array called stkParameters is used to store the handle to the STK application, the IAgStkObject root, and the three access calculations. stkParameters is then stored in the UserData parameter for the block so it can be used again in the s-function script.

Run the Simulink model. STK will start and the objects for analysis will be created. As Simulink passes the current simulation time to the STK s-function block, STK’s animation time will be updated, one point access will be calculated, and the result of the access will be shown in the Display blocks (1=access, 0=no access).

**Notes**

Included in the folder is the MATLAB script used to test and debug the InitFcn callback for the s-function block, InitFcn\_TestCode.m. This can be run without any setup.

The concepts demonstrated in this example should be applied for any Simulink to STK integration. Use a Level-2 S-Function block to host and interact with STK. Use the s-function’s InitFcn callback to initialize the connection to STK, create or load the scenario, and then store the important objects in the ‘UserData’ parameter. In the s-function script, retrieve the UserData parameter and then run the appropriate analysis in STK.

This approach was necessary because the s-function’s DWork vector is not able to handle complex types, like COM objects. You can read a more in depth discussion of this approach here.

This example could comprise one block of a more complicated state flow model like the one pictured below:

