

**High Level Architecture Distributed Space System  
Simulation for Simulation Interoperability Standards  
Organization Simulation Smackdown**

Zuqun Li

NASA Johnson Space Center

Major: Aerospace Engineering

USRP Spring Session

Date: April 12, 2011

# High Level Architecture Distributed Space System Simulation for Simulation Interoperability Standards Organization Simulation Smackdown

Zuqun Li<sup>1</sup>

*Pennsylvania State University, University Park, PA 16802*

Modeling and Simulation plays a very important role in mission design. It not only reduces design cost, but also prepares astronauts for their mission tasks. The SISO Smackdown is a simulation event that facilitates modeling and simulation in academia. The scenario of this year’s Smackdown was to simulate a lunar base supply mission. The mission objective was to transfer Earth supply cargo to a lunar base supply depot and retrieve He-3 to take back to Earth. Federates for this scenario include the environment federate, Earth-Moon transfer vehicle, lunar shuttle, lunar rover, supply depot, mobile ISRU plant, exploratory hopper, and communication satellite. These federates were built by teams from all around the world, including teams from MIT, JSC, University of Alabama in Huntsville, University of Bordeaux from France, and University of Genoa from Italy. This paper focuses on the lunar shuttle federate, which was programmed by the USRP intern team from NASA JSC. The shuttle was responsible for provide transportation between lunar orbit and the lunar surface. The lunar shuttle federate was built using the NASA standard simulation package called Trick, and it was extended with HLA functions using TrickHLA. HLA functions of the lunar shuttle federate include sending and receiving interaction, publishing and subscribing attributes, and packing and unpacking fixed record data. The dynamics model of the lunar shuttle was modeled with three degrees of freedom, and the state propagation was obeying the law of two body dynamics. The descending trajectory of the lunar shuttle was designed by first defining a unique descending orbit in 2D space, and then defining a unique orbit in 3D space with the assumption of a non-rotating moon. Finally this assumption was taken away to define the initial position of the lunar shuttle so that it will start descending a second after it joins the execution. VPN software from SonicWall was used to connect federates with RTI during testing and the Smackdown event. HLA software from Pitch Technology and MAK Technology were used to edit and extend FOM and provide HLA services for federation execution. The SISO Smackdown event for 2011 was held in Boston, Massachusetts. The federation execution lasted for one hour, and the event was very successful in catching the attention of university students and faculties.

## Nomenclature

DCM	=	Direction Cosine Matrix
DSSS	=	Distribute Space System Simulation
FOM	=	Federation Object Model
GUI	=	Graphic User Interface
HLA	=	High Level Architecture
IEEE	=	Institute of Electrical and Electronics Engineers
JSC	=	Johnson Space Center
MIT	=	Massachusetts Institute of Technology
NASA	=	National Aeronautics and Space Administration
RTI	=	Runtime Infrastructure
SISO	=	Simulation Interoperability Standards
SSL	=	Secure Socket Layer
USRP	=	Undergraduate Student Research Program
VPN	=	Virtual Private Network

---

<sup>1</sup> Undergraduate Student Research Intern (USRP), Software, Robotics, & Simulation Division, ER-7, NASA Johnson Space Center, Pennsylvania State University.

NASA USRP – Internship Final Report

$a_{des}$	=	semi-major axis of descending orbit
$e_{des}$	=	eccentricity of descending orbit
$h_{des}$	=	specific angular momentum of descending orbit
$\theta_{Land,des}$	=	true anomaly of landing site in descending orbit
$i$	=	inclination
$\Omega$	=	Right ascension of ascending node
$\omega$	=	argument of periapsis
$M$	=	mean anomaly
$dM$	=	changing mean anomaly
$E$	=	eccentric anomaly
$\theta$	=	true anomaly
$\ddot{\vec{r}}_{LS}$	=	acceleration vector of the lunar shuttle
$\vec{r}_{LS}$	=	position vector of the lunar shuttle
$\vec{r}_{land,fix}$	=	position vector of landing site in moon centric fixed frame
$\vec{r}_{land,perifocal}$	=	position vector of landing site in perifocal frame
$r_{LS}$	=	distance of the lunar shuttle from center of the moon
$r_{LS,cir}$	=	distance of the lunar shuttle from center of the moon in circular orbit
$r_{a,des}$	=	apoapsis distance of descending orbit
$r_{Land,des}$	=	distance of landing site in descending orbit
$\vec{v}_{LS}$	=	velocity vector of the lunar shuttle
$v_{a,des}$	=	speed at apoapsis point of descending orbit
$v_{LS}$	=	speed of the lunar shuttle
$v_{LS,cir}$	=	speed of the lunar shuttle in circular orbit
$TOF$	=	time between first and second descending burns
$Long$	=	Longitude
$Lat$	=	Latitude
$h$	=	elevation
$q_0$	=	scalar element of quaternion
$q_1, q_2, q_3$	=	vector elements of quaternion
$DCM_{peri\ to\ inertial}$	=	direction cosine matrix from perifocal frame to inertial frame
$DCM_{inertial\ to\ fixed}$	=	direction cosine matrix from moon centric inertial to moon centric fixed frame
$DCM_{fixed\ to\ inertial}$	=	direction cosine matrix from moon centric fixed to moon centric inertial frame
$\mu_M$	=	gravitational parameter of the moon
$R_M$	=	mean radius of the moon

## I. Introduction

Modeling and simulation is a critical element in engineering design and analysis. It identifies risks, verifies system performance based on requirements, and helps engineers to find solutions for potential problems. Moreover, modeling and simulation will also give university students an opportunity to practice course materials that they learned from classes, such as orbital dynamics, thermodynamics, and physics.

The SISO Simulation Smackdown is an event that facilitates modeling and simulation in academia. Participating teams will program one or more HLA based distributed simulation for a space system. This paper focuses on the lunar shuttle federate that was built by the NASA –JSC intern team. The lunar shuttle is responsible for providing lunar orbit and lunar surface transportation for a lunar base supply mission. The scenario of the SISO Smackdown for 2011 will be discussed in detail in section two, Project Overview, and the detail of the lunar shuttle federate will be discussed in section four, Lunar Shuttle Federate.

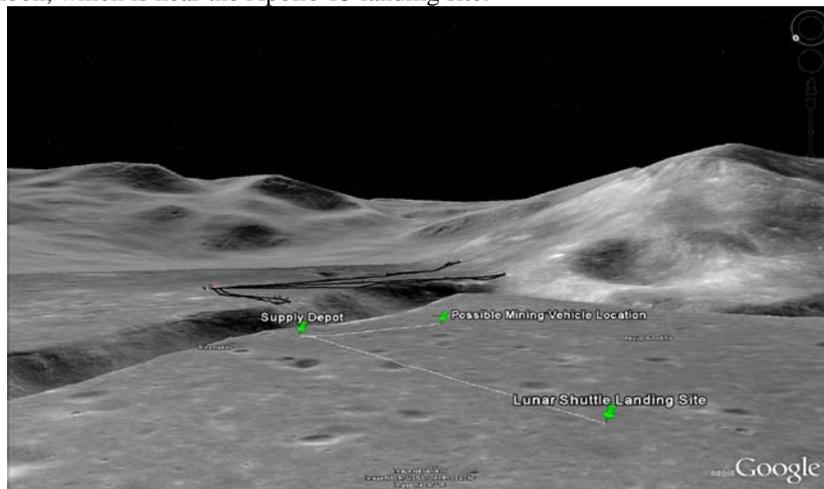
## II. Project Overview

The major objective of this project is to facilitate the interest in modeling and simulation at university undergraduate and graduate levels, and allow students to understand both the importance and complexity of modern modeling and simulation. Additionally, it prepares students to meet the high demanding market for modeling and simulation skills.

### A. Scenario

The scenario for the SISO Smackdown in 2011 was to simulate a lunar base supply mission. The mission objective was to transfer Earth cargo, which could be fuel, parts, robots, etc., to a lunar base supply depot, and bring back Helium-3 to Earth.

An Earth-Moon transfer vehicle carries Earth cargo to lunar orbit, and transfers the cargo to the lunar shuttle, which is orbiting around the moon in a circular orbit. The transfer vehicle will also refuel the lunar shuttle before undocking. Once undocked with the transfer vehicle, the lunar shuttle will descend down to the moon’s surface in a designed descending trajectory and land on a predefined landing location. Once the lunar shuttle lands, the lunar rover will come to the lunar shuttle and carry the Earth cargo from the shuttle to the lunar base supply depot for storage. Then the rover will refuel the shuttle and transfer Helium-3, which was mined by the mobile ISRU plant, to the lunar shuttle. The shuttle will then ascend to lunar orbit and dock with the Earth-Moon transfer vehicle. After docking is completed, Helium-3 will be transferred to the Earth-Moon transfer vehicle, and the lunar shuttle will undock with the transfer vehicle. Finally, the transfer vehicle will carry Helium-3 back to Earth. The cycle will repeat as the second transfer vehicle arrives. Figure 1, from Google Moon, shows that the location of this mission took place on the moon, which is near the Apollo 15 landing site.



**Figure 1: Lunar Base Supply Mission Location (Google Moon)**

## B. Participating Teams

Many teams from government, industries, and universities participated in this project for the SISO Smackdown during 2011.

NASA JSC provides two core federates, which includes the Environment federate and the Earth-Moon transfer vehicle federate. The environment federate provides state information for the Sun, Earth, and moon, which include the position and attitude of these planets. It also defines simple data types for different variables.

The University of Bordeaux, from France, and the University of Genoa, from Italy, formed a team and provided the Earth Cargo Depot and Lunar Supply Depot federates. These federate serve as resource storage.

The University of Alabama in Huntsville provides two Lunar Communication Satellite federates, which orbit around the moon and publish their position and velocity.

The team from MIT provided the High-mobility Scouting Hopper federate and the Mobile ISRU Plant federate. The High-mobility Scouting Hopper is used to analyze resource concentration at potential mining sites and sends resource report to the Mobile ISRU Plant. By analyzing the resource report, the Mobile ISRU Plant mines and processes resource in the concentrated area.

Keio University from Japan also participated in this project. However, the team decided to drop out due to the earthquake cataclysm. However, they will participate in the Smackdown event for 2012.

Louisiana State University acted as observers this year but hope to participate in the 2012 Smackdown.

The NASA intern team from JSC provided the lunar shuttle federate and the lunar rover federate. The lunar Shuttle federate provides transportation between lunar orbit and the lunar surface. The lunar rover federate provides transportation on lunar surface. Its responsibility is to transfer resources between the Mobile ISRU Plant, lunar supply depot, and lunar shuttle federates.

The team from ForwardSim Incorporation provided a 3D Viewer federate, which gave a virtual view of activities of all other federates on lunar surface. Figure 2, below, shows a snap shot from 3D Viewer federate.

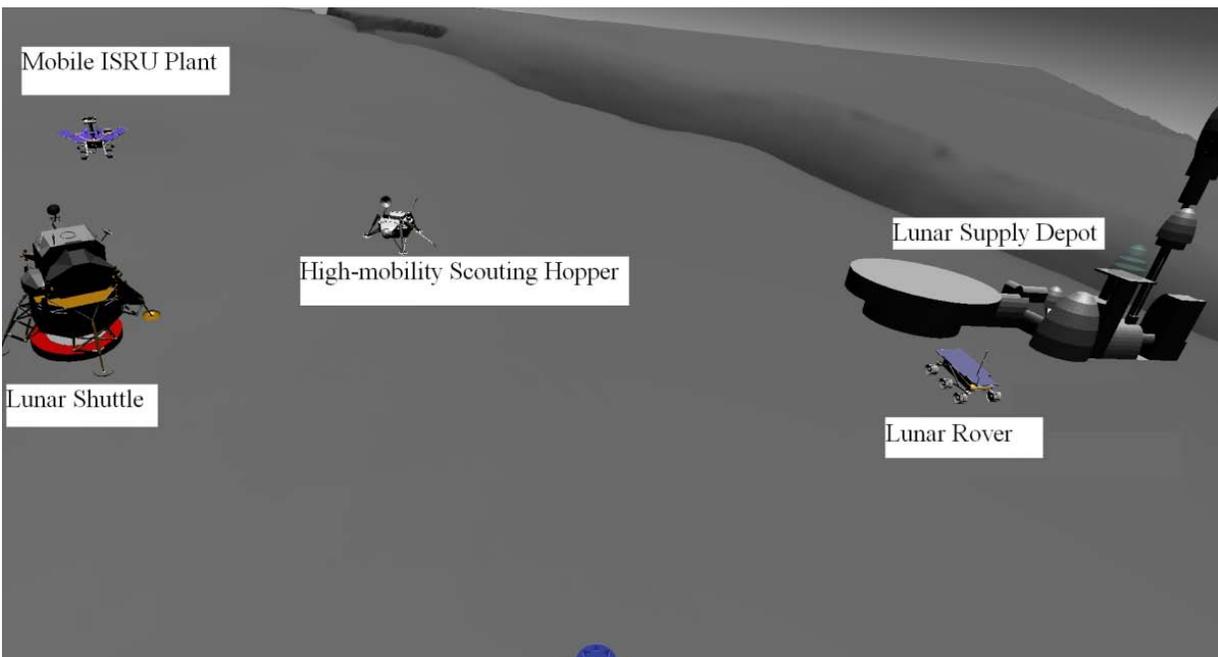


Figure 2: Snapshot from 3D Viewer Federate (ForwardSim Inc.)

### III. Software Infrastructure

#### A. VPN Software

NetExtender from SonicWall Incorporation was used to provide a network connection for all federates to connect with the RTI. It is SSL VPN that provides security and flexibility connections. Figure 3, below, shows the graphic user interface of NetExtender.



**Figure 3: NetExtender GUI (SonicWall Incorporation)**

During testing, federates connected to the RTI through NetExtender VPN from all around the world. Figure 4 shows the participating team for 2011 connecting through the NetExtender during tests.

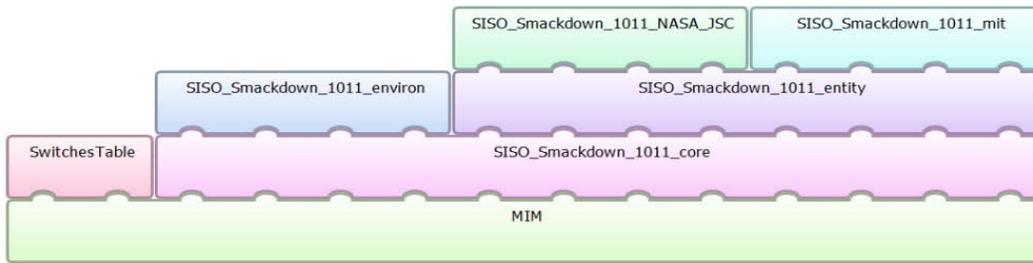


**Figure 4: Distrubted Simulation Network during Testing**

#### B. HLA Software

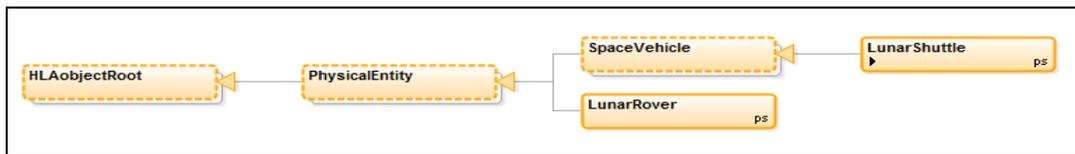
##### 1. FOM Editor

Pitch Visual OMT, from Pitch Technologies, was used to create, edit, and extend Federation Object Model (FOM) for federates. Figure 5 shows the FOM Modules for this project, and Figure 6 give the example of an extension module.



**Figure 5: Smackdown FOM Modules (Pitch Technologies)**

Figure 5 shows the dependence of each FOM Modules. Switches Table and SISO Smackdown 1011 core (Core) both depend on the MIM modules, then SISO Smackdown 1011 entity (Entity) and SISO Smackdown 1011 environ depend on the Core module, and finally, SISO Smackdown 1011 NASA JSC and SISO Smackdown 1011 mit depend on the Entity modules.



**Figure 6: JSC Intern Team Extension Module**

As showed in Figure 6, both Space Vehicle class and Lunar Rover class are inherited from Physical Entity class, which is also inherited form HLA object Root class. Lunar Shuttle class is inherited from Space Vehicle class. Inheritance provides reusability; it allows classes to acquire the properties of objects of other classes while also being able to create its own attributes. For example, Lunar Shuttle is inherited from Space Vehicle; therefore, besides all the properties that Space Vehicle has, Lunar Shuttle also has an attribute that only belongs to itself, which was called “request fuel mass”.

**2. RTI**

RTI is what connect all federates together. RTI provides common service to the simulation system, such as federation management, declaration management, object management, ownership management, data distribution management, and time management. Two RTI softwares, one from VT MAK Technologies and one from Pitch Technologies, were used for this project. Both of them met the requirements defined in IEEE 1516 standard for HLA RTI. (IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, 2010)

RTI software from Pitch Technologies is called Pitch pRTI Evolved, which connects all federates together during the federation execution. Figure 7 shows the graphical view of joined federates during a federation execution test.

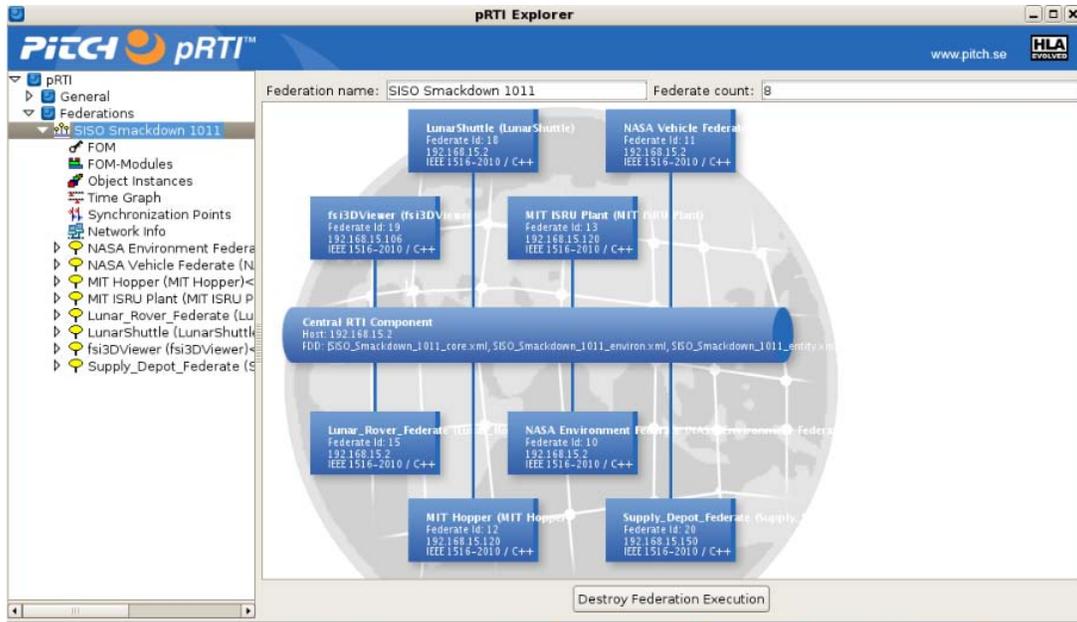


Figure 7: Graphical View of Federates during Federation Execution

RTI software from TV MAK Technologies is called MAK RTI. It provides same functionality that Pitch pRTI has with exception of not having a graphical view of joined federates.

Both RTIs provided outstanding service for federation execution during testing and during the Smackdown event.

### C. NASA Simulation Software

Two Simulation packages were used to program the lunar shuttle federates. Both packages were developed at NASA JSC, they are Trick and TrickHLA.

#### 1. Trick

Trick was used to develop the simulation model for the lunar shuttle. It supports C and C++ programming language. Trick has various features that allow the users to build a reliable and accurate simulation. Some of Trick features include unit conversion, real time and non-real time running, data logging, graphics, and math utilities.

Unit conversion allows users to specify the units of variables, so that they do not have to worry about unit inconsistency. It also allows users to input value in units that is different than what is defined in the program. Trick will automatically convert the input value into the unit that is defined in the program.

Trick has a control panel GUI that allows users to run the simulation in both real time and non-real time. It also gives users the ability to freeze and step the running for debugging. Moreover, the control panel consists of a status message window, which displays debugging messages while running the simulation. Figure 8 shows the control panel GUI of Trick.

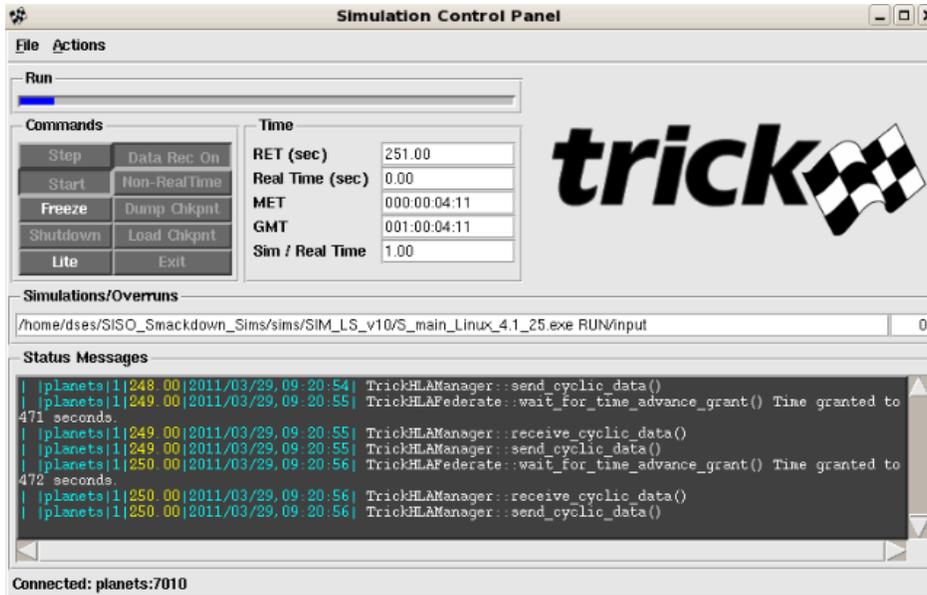


Figure 8: Trick Control Panel GUI

Trick data logging allows users to record, view, and change variables while running the simulation. A Trick GUI called data recording editor (dre) was used to record data for any variables defined in the simulation for both real time and non real time running. The selected variables can be plotted with the Trick data products GUI. Another useful tool called Trick View (TV) was used to view and change data for real time running. Figure 9 shows the snap shot of the Trick View GUI.

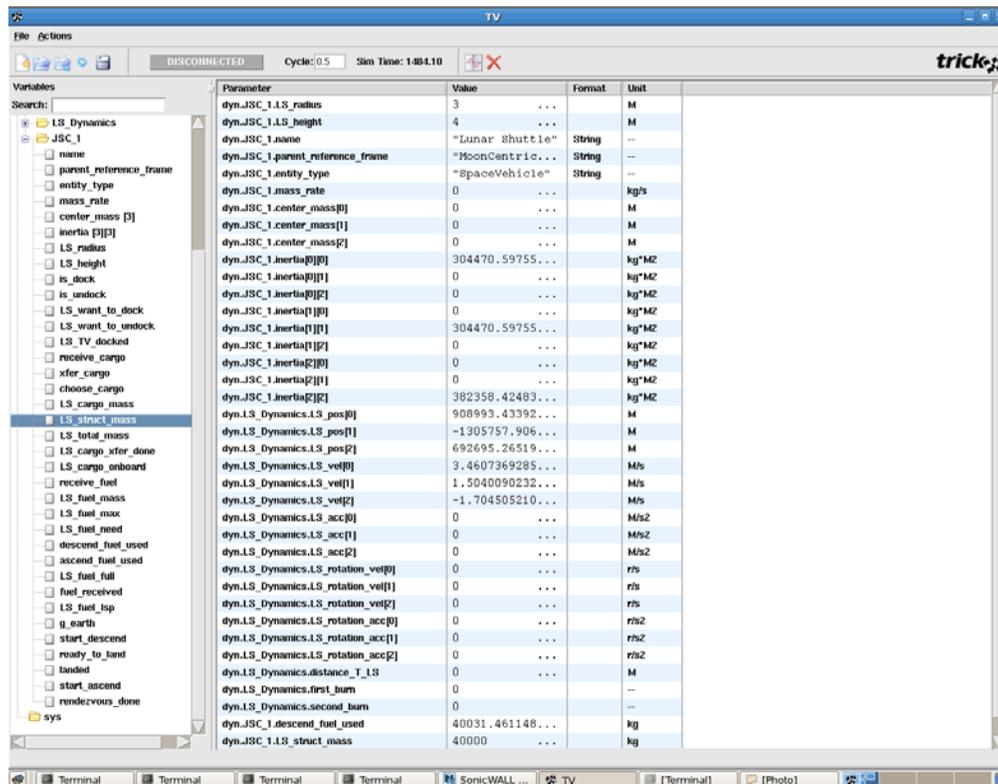
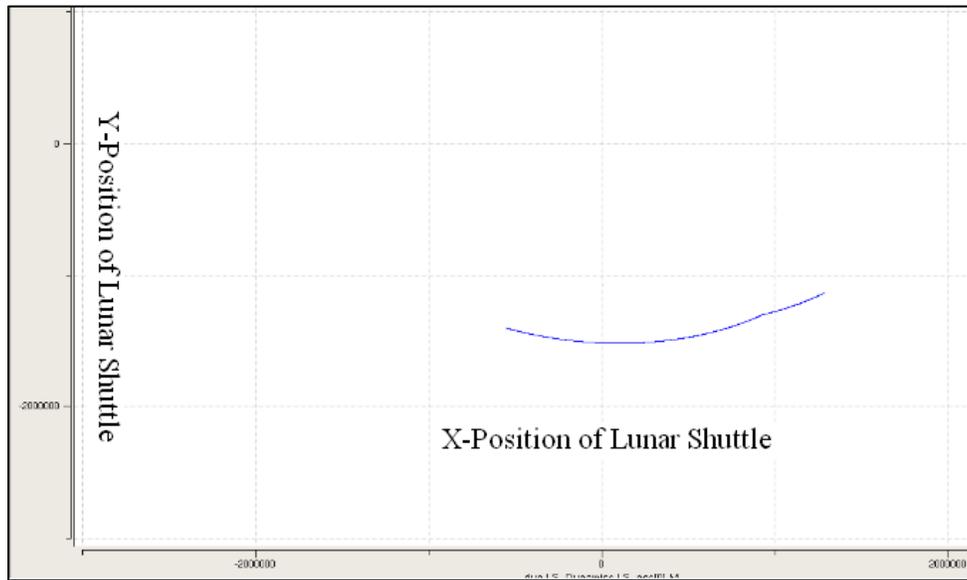


Figure 9: Trick View GUI

Trick View also allows users to plot any variables in a strip chart. Figure 10 below, shows a plot of y-position with respect to x-position for the lunar shuttle federate.



**Figure 10: X-Position vs. Y-Position for Lunar Shuttle**

Last but not least, Trick provides powerful math utilities that include complex vector and matrix operations, which save a lot of programming time and allow users to focus on model development.

## 2. *TrickHLA*

TrickHLA was developed by Daniel Dexter at NASA JSC that is specifically used to expand Trick models with HLA functions without having to understand all the technical detail of HLA. Many HLA features were added to the lunar shuttle federate with TrickHLA. These features include receiving and sending attributes, interaction handling, and data packing.

TrickHLA allows the lunar shuttle federate to publish and subscribe attributes to/from RTI. Attributes such as position, which constantly changes during the execution, will be published to RTI and made available for other federates' use. An example of this is that lunar rover federate subscribes to the lunar shuttle's position so that it can move to the shuttle's location for cargo transfer and refueling.

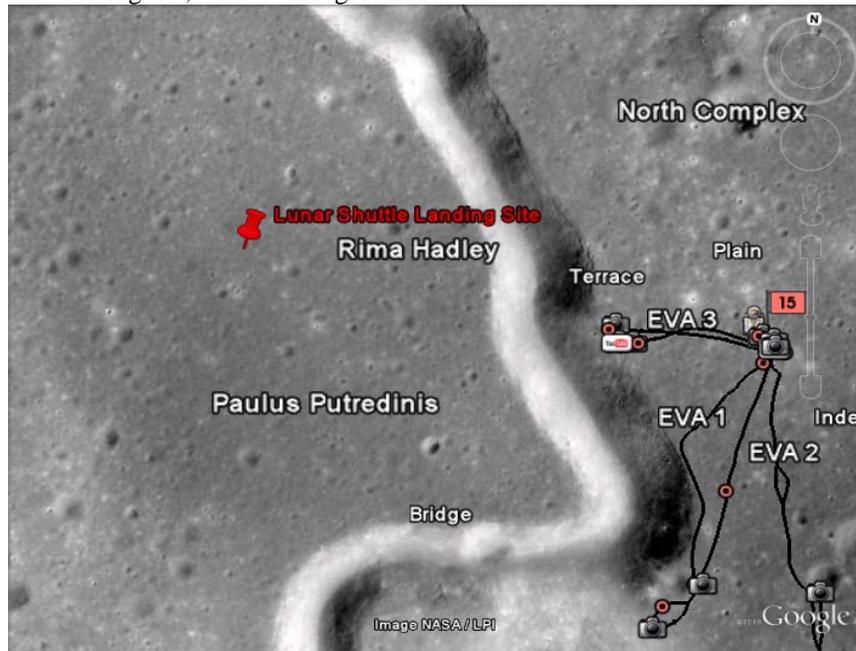
Interaction capability was also added to the lunar shuttle federate. Unlike attributes, which constantly update during the execution, interactions can only be send when predefined conditions are satisfy. Interactions can be use as an event flag. An example of interaction is when the lunar shuttle arrived, it sent out an arrival interaction so the rover would notice the arrival of the lunar shuttle and prepare to move to the shuttle for cargo transfer and refueling. This can reduce the amount of data transfer during the execution.

In order to send and receive non basic data types to and from other federates, federate has to be able to pack and unpack data. By expanding the TrickHLA packing class, the lunar shuttle is able to pack fixed record data before it publishes it and unpack them before it uses this data. An example of using a packing class is the lunar shuttle subscribing to the rotational state of the moon from the environment federate. Rotational state of the moon is a type of Reference Frame Rotation, which is a fixed record data type and consisted of one scalar and two vectors. Expanding the TrickHLA packing class allows the lunar shuttle federate to create a buffer with size equal to the size of Reference Frame Rotation, then extract scalar and vectors from the attribute, and finally convert the quaternion to a DCM that was used to transfer vectors from Lunar Centric Inertial frame to Lunar Centric Fixed frame.

## IV. Lunar Shuttle Federate

The mission objective for the lunar shuttle is to carry Earth supply cargo down to the lunar surface and bring back Helium-3 to the lunar orbit, where it docked with the Earth-Moon transfer vehicle and transferred Helium-3 to it.

The lunar shuttle landing site with Longitude of  $3^{\circ}23'28.35''\text{E}$ , Latitude of  $26^{\circ}09'14.4''\text{N}$ , and Elevation of  $-1848\text{m}$ , located near Apollo 15 landing site, is shown in Figure 11.



**Figure 11: Lunar Shuttle Landing Site (Google Earth)**

The lunar shuttle federate was modeled with three degrees of freedom. The dimension of the lunar shuttle was assumed to be a uniform cylinder with radius of three meters and height of four meters. The structure mass of the lunar shuttle was set to be 20000 kg, fuel mass was set to be 40000 kg, and payload mass was set to be 5000 kg. These numbers were randomly chosen, since they were not the major focus for this project.

Starting from lunar orbit, the lunar shuttle docked with the transfer vehicle and received Earth supply cargo from the transfer vehicle. Then the shuttle will undock with the transfer vehicle and descended down to the lunar surface. Once the shuttle landed, it notices the lunar rover, and the rover moves to the shuttle for the Earth supply cargo. After the rover received Earth supply cargo and stored it in the lunar supply depot, the rover will refuel the shuttle and transfer Helium-3 to the shuttle. Once the shuttle received both fuel and Helium-3, it will lift off from the surface and ascend to lunar orbit. The lunar shuttle will then rendezvous and dock with Earth-Moon transfer vehicle for Helium-3 transfer, however, this part was not modeled in the federate due to the time limitation of the Smackdown event.

### A. Simulation Development

In order to satisfy the objective, the lunar shuttle has to be able to perform docking, resource transferring, fuel receiving, descending, and ascending. A top-down design was used for designing the lunar shuttle federate. First, a class name LunarShuttle, which defined the functionalities of the lunar shuttle federate, was created with methods including, dock\_undock, cargo\_xfer\_receive, fuel\_receive, descend, ascend, LS\_Controller, and LS\_Reset. Then, three interaction handler classes were created to send and receive docking, resource transfer, and maneuver state interactions. Moreover, LSOribitalDynamics class was created to model the dynamics of the lunar shuttle federate, including orbit propagating, descending, and ascending. Finally, two packing classes were created to pack and unpack fixed record data for receiving the moon centric fixed quaternion from the environment federate and sending the lunar shuttle attitude to RTI. The flow diagram in Figure 12 shows the structure of the lunar shuttle federate.

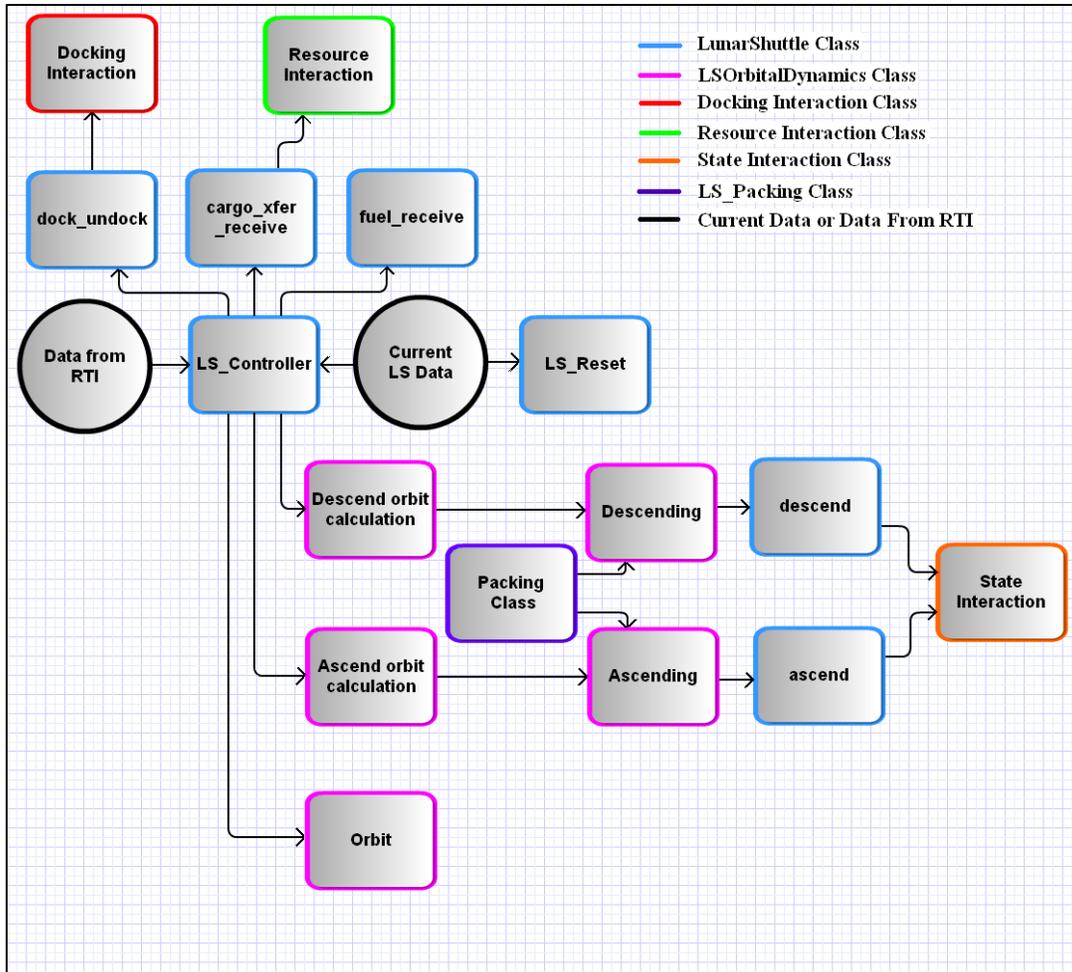


Figure 12: Structure of Lunar Shuttle Federate

As showed in Figure 12, the LS\_Controller from the LunarShuttle class controlled which method to call based on the current data of the lunar shuttle federate and the data received from the RTI. The controller has switches for methods that connect to it, and the controller was responsible for turning these switches on or off. When the switch is off, statements in the method are not executed; likewise, when switch is on, the statements are executed. Methods such as dock\_undock, cargo\_xfer\_receive, and fuel\_receive are responsible for updating the status data for the lunar shuttle federate and call send interaction methods if necessary. Interaction classes, such as docking, resource, and state interaction, are responsible to pack/unpack any fixed record data and send/receive interaction to/from the RTI. The controller also connects to the LSOribitalDynamics class, which has a method that models the dynamics of the lunar shuttle. Orbit method propagates the state of the shuttle while it is in orbit, descending, and ascending. Descend orbit calculation and Ascend orbit calculation methods are responsible for calculating the descending trajectory and ascending trajectory in moon centric fixed frame, then they call the Descending and Ascending methods. Descending and Ascending methods from the LSOribitalDynamics class will get DCM, which converts vectors from moon centric fixed frame to moon centric inertial frame, from the packing class and perform velocity changes to the shuttle’s dynamics. Descending method also give control of slowing down the shuttle when it is less than 8km above the landing site. The Descending and Ascending method in the LSOribitalDynamics class will then turn on the Descend and Ascend methods in the LunarShuttle class to update the status of the lunar shuttle, which will then call the state interaction to send the interaction to the RTI.

## B. Dynamics

### 1. State Propagation (Vallado, 2001)

The shuttle was modeled with three degrees of freedom, which is the translational motion in three dimensional space. The two body dynamics assumption was used to propagate the lunar shuttle’s position and velocity state during orbiting, descending, and ascending by integrating the acceleration equation show below.

$$\ddot{\vec{r}}_{LS} = -\frac{\mu_M}{r_{LS}^3} \vec{r}_{LS} \quad (1)$$

Where  $\mu_M$  is the gravitational parameter of the moon.

The lunar shuttle was initially in a circular orbit with altitude of 100 km, and the speed of the shuttle was calculated by equation 2, show below.

$$v_{LS,cir} = \sqrt{\frac{\mu_M}{r_{LS,cir}}} \quad (2)$$

2. *Descending* (Vallado, 2001)

One of the major parts of the lunar shuttle’s dynamics is descending. Few assumptions were used for descending trajectory design in this project. These assumptions are show below.

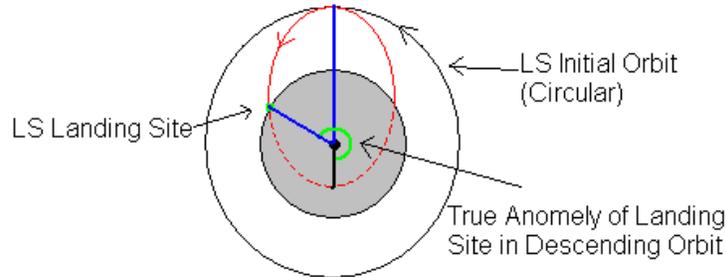
- Lunar shuttle is initially in a circular orbit
- Initial orbital plane of the lunar shuttle is same as descending orbital plane
- Two instantaneous burns for descending
- Lunar shuttle starts descending at the periapsis of descending orbit

The following is a list of known variable for calculating the descending orbit

- Landing site longitude, latitude, and elevation
- Lunar shuttle initial orbit radius

With these assumptions and known variables, the descending trajectory for the lunar shuttle can be derived in three steps.

First step is to define a unique elliptical descending orbit in two dimensional space. Figure 13 shows a schematic drawing of descending orbit.



**Figure 13: Descending Orbit in Two Dimensional Space**

As show in Figure 13, the radius of apoapsis of descending orbit is equal to the radius of the lunar shuttle’s initial orbit. Knowing only two variables,  $r_{a,des}$  and  $r_{land,des}$ , is not enough to define an unique orbit in two dimensional space. Therefore, a third known variable is required. Fortunately, the third known variable can be chosen for mission requirements. The federation execution time for the Smackdown event was limited to one hour. Base on the speed of lunar rover and the distance of the shuttle landing site, the lunar shuttle will stay on the moon’s surface for about 15 minutes. Therefore the time left for descending and ascending is about 45 minutes, and descending time should be around 23 minutes or less. Equation 3 below was used to calculate time of flight, which is the time between first descending burn and second descending burn.

$$TOF = \frac{dM * h_{des}^3}{[\mu_M^2 * (1 - e_{des}^2)^{3/2}]} \quad (3)$$

$e_{des}$  is the eccentricity of descending orbit, which is given by equation 4

$$e_{des} = \frac{(r_{a,des} - r_{Land,des})}{[r_{Land,des} * \cos(\theta_{Land,des}) + r_{a,des}]} \quad (4)$$

$h_{des}$  is specific angular momentum of descending orbit, and can be calculate with equation below

$$h_{des} = r_{a,des} * v_{a,des} \quad (5)$$

And velocity of the lunar shuttle at any given distance from lunar center was calculated by equation 6.

$$v_{LS} = \sqrt{\mu_M \left( \frac{2}{r_{LS}} - \frac{1}{a_{des}} \right)} \quad (6)$$

Where  $a_{des}$  is major semi-axis of the descending orbit, and given by equation below

$$a_{des} = \frac{r_{a,des}}{(1 + e_{des})} \quad (7)$$

$dM$  is the change of mean anomaly and is given by

$$dM = M_2 - M_1 \quad (8)$$

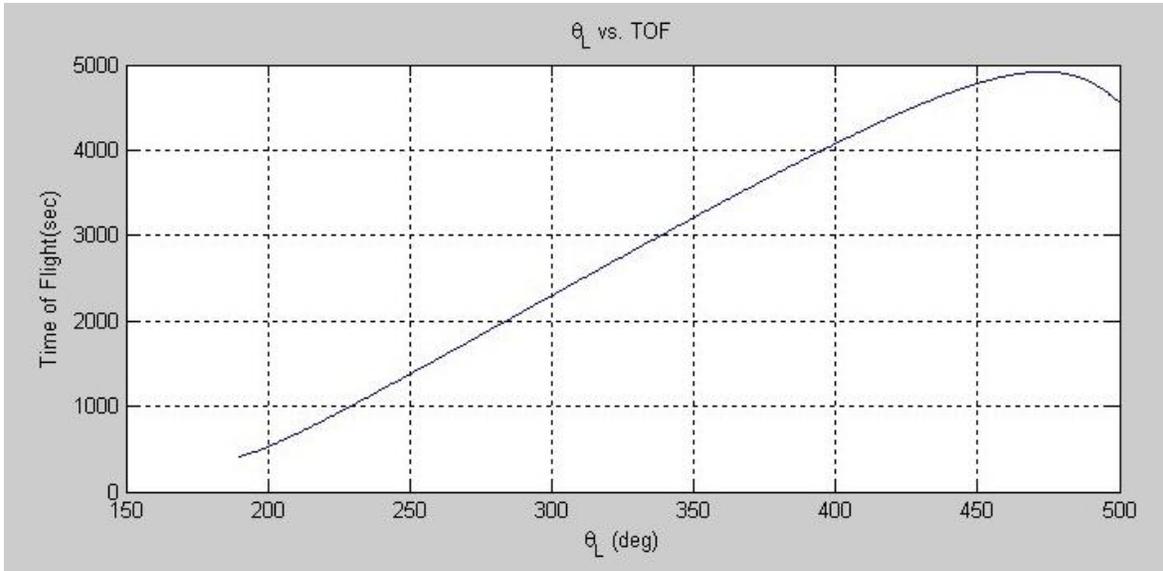
And

$$M = E - e_{des} * \sin(E) \quad (9)$$

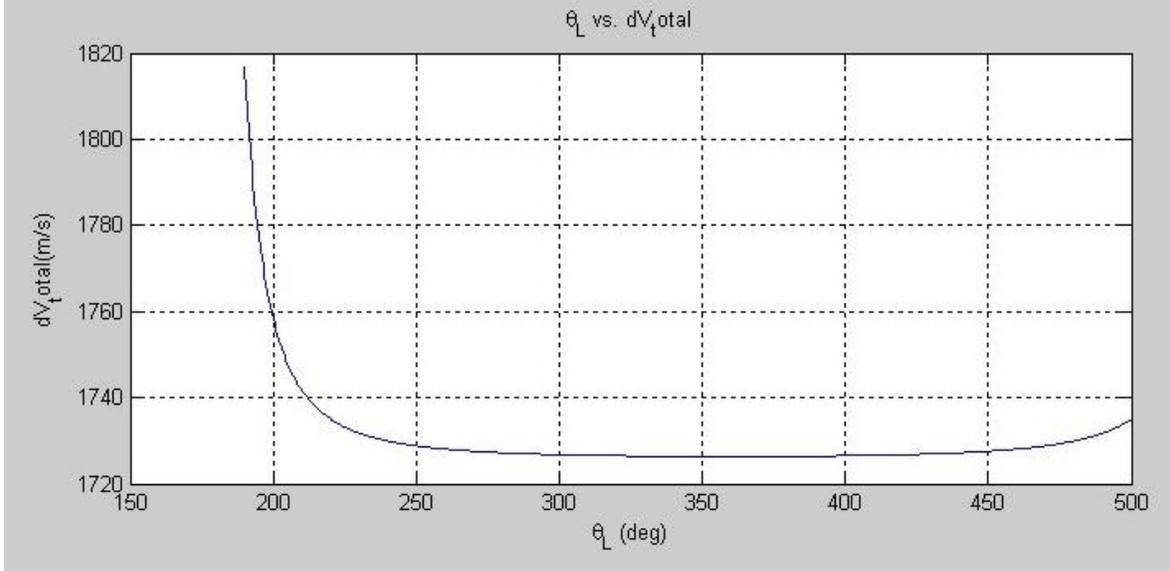
Where  $E$  is eccentric anomaly. Equation 10 shows the relationship between eccentric anomaly and true anomaly

$$\tan\left(\frac{\theta}{2}\right) = \sqrt{\frac{(1 + e_{des})}{(1 - e_{des})}} * \tan\left(\frac{E}{2}\right) \quad (10)$$

By varying the true anomaly of the landing site in descending orbit, its relationship between time of flight and total descending velocity change were calculated and are shown in Figure 14 and 15.



**Figure 14: Time of Flight vs. True Anomaly of Landing Site in Descending Orbit**



**Figure 15: Total Change Velocity vs. True Anomaly of Landing Site in Descending Orbit**

From Figure 14 and 15, true anomaly ranging from 210 degree to 240 degrees will satisfy the time requirement without performing large amount of velocity change. To narrow down the range, another criteria was considered during the design. That is the lunar shuttle should perform the second burn above the landing site and slowly decrease its altitude. In reality, no rocket engine could perform the large instantaneous velocity change. The assumption of instantaneous velocity change gives a simple start for this first SISO Smackdown, and this assumption will not be included for next year’s design. An altitude of 8 km was chosen for the lunar shuttle to perform the second burn. A few mountains around the landing site, such as Mountain Mons Hadley, have elevation of about 2 km, and the landing site has elevation of about -2 km. a safety factor of 2 was used; therefore, an altitude of 8 km was chosen. When the shuttle is 8 km above the landing site, it will slowly decrease its altitude with a speed of 25 m/s. This will take about 5 minutes. Therefore the time of flight for the shuttle should be around 1000 seconds, where the true anomaly is about 230 degrees. With the third known variable defined, then a unique descending elliptical orbit in two dimensional space can be calculated. Position and velocity vector of the lunar shuttle in perifocal frame was calculated with the equations show below.

$$\vec{r}_{LS} = r_{LS} * \cos(\theta) \hat{p} + r_{LS} * \sin(\theta) \hat{q} + 0 \hat{w} \quad (11)$$

$$\vec{v}_{LS} = \frac{\mu_M}{h_{des}} * [-\sin(\theta) \hat{p} + (e_{des} + \cos(\theta))\hat{q} + 0 \hat{w}] \quad (12)$$

Where

$$r_{LS} = \frac{h_{des}^2 / \mu_M}{[1 + e_{des} * \cos(\theta)]} \quad (13)$$

The changing-velocity vectors for the first and second burns were also calculated in perifocal frame as:

$$\vec{dv}_1 = \vec{v}_{a,des} - \vec{v}_{LS,cir} \quad (14)$$

The second step is to define a unique descending orbit in three dimensions. Three more orbital parameters are needed, and they are inclination ( $i$ ), RAAN ( $\Omega$ ), and argument of periapsis ( $\omega$ ). To reduce the complexity, another assumption was made during this step of design, which is assuming the moon is not rotating. This assumption will be excluding in the third step of the design. Base on this assumption, the Moon Centric Fixed Frame will always align with Moon Centric Inertial Frame.

Setting the inclination equal to the latitude of landing site will make the shuttle directly above the landing site as it orbiting the moon. Therefore  $i = \text{Latitude of landing site}$ .

Equation 15, shown below, was used to calculate the RAAN (See Appendix for derivation of this equation).

$$\Omega = (\text{Longitude of landing site in degree}) - 90^\circ \quad (15)$$

Then argument of periapsis was calculated using the DCM that convert vectors from perifocal frame to moon centric inertial frame (same as moon centric fixed frame in here). The equation below shows the relationship between DCM and  $i$ ,  $\Omega$ , and  $\omega$ .

$$DCM_{peri\ to\ inertial} = \begin{bmatrix} \cos(\Omega) \cos(\omega) - \sin(\Omega) \sin(\omega) \cos(i) & -\cos(\Omega) \sin(\omega) - \sin(\Omega) \cos(\omega) \cos(i) & \sin(\Omega) \sin(i) \\ \sin(\Omega) \cos(\omega) + \cos(\Omega) \sin(\omega) \cos(i) & -\sin(\Omega) \sin(\omega) + \cos(\Omega) \cos(\omega) \cos(i) & -\cos(\Omega) \sin(i) \\ \sin(i) \sin(\omega) & \sin(i) \cos(\omega) & \cos(i) \end{bmatrix} \quad (16)$$

The landing site position vector in perifocal frame was calculated from equation 11 and 13.

$$\vec{r}_{land,perifocal} = x \hat{p} + y \hat{q} + 0 \hat{w}$$

Landing site position vector in moon centric fixed frame was calculated using equations shown below.

$$\vec{r}_{land,fix} = X \hat{I} + Y \hat{J} + Z \hat{K} \quad (17)$$

Where

$$X = (R_M + h) \sin(90^\circ - Lat) \cos(Long) \quad (17a)$$

$$Y = (R_M + h) \sin(90^\circ - Lat) \sin(Long) \quad (17b)$$

$$Z = (R_M + h) \cos(90^\circ - Lat) \quad (17c)$$

And  $R_M$  is the mean radius of the moon;  $h$  is the elevation of landing site.

Equation 18 can be used to convert the position vector from perifocal frames to inertial frames using the DCM shown in equation 16.

$$\vec{r}_{land,fix} = DCM_{peri\ to\ inertial} * \vec{r}_{land,perifocal} \quad (18)$$

Since both  $\vec{r}_{land,fix}$  and  $\vec{r}_{land,perifocal}$  are known, then the argument of periapsis ( $\omega$ ) can be calculated (see Appendix for detail calculation). Then the direction cosine matrix from perifocal frame to inertial frame (fixed frame) was found.  $DCM_{peri\ to\ inertial}$  was then used to convert change velocity vector from perifocal frame to inertial frame.

The third step was to take away the non-rotating assumption of the moon. An attribute that published by the environment federate describe the rotation of the moon using a quaternion, which include one scalar ( $q_0$ ), a vector ( $q_1$  to  $q_3$ ), and an angular velocity vector.

$$quaternion = [q_0, (q_1, q_2, q_3)]$$

A DCM that converts a vector from Moon Centric Inertial Frame to Moon Centric Fixed Frame was calculated using equation 19, shown below, with quaternion from the environment federate

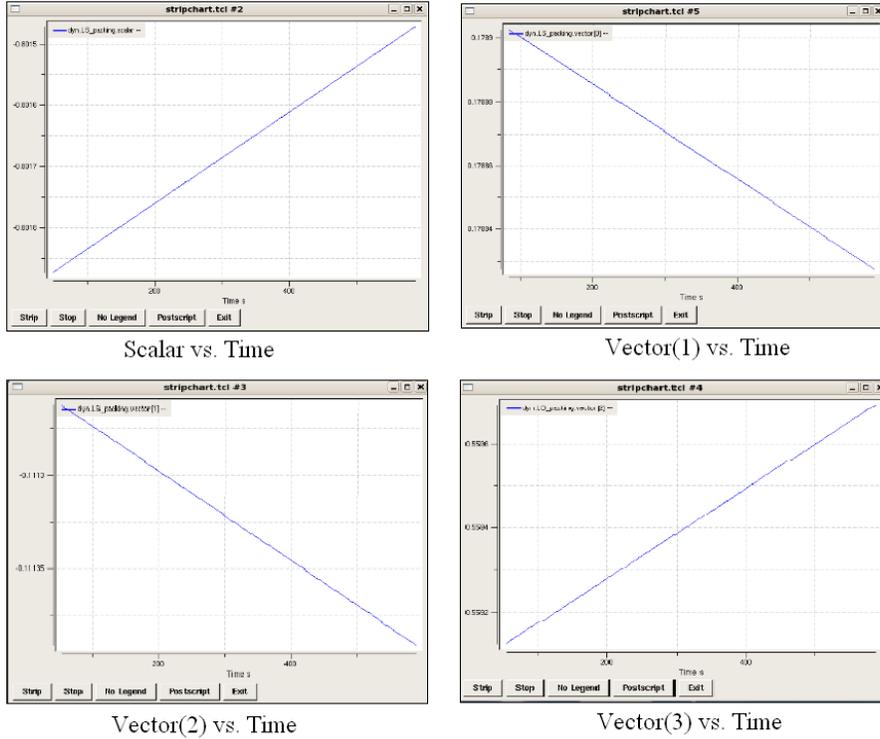
$$DCM_{inertial\ to\ fixed} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_0^2 & 2 * (q_1 * q_2 + q_3 * q_0) & 2 * (q_1 * q_3 - q_2 * q_0) \\ 2 * (q_1 * q_2 - q_3 * q_0) & -q_1^2 + q_2^2 - q_3^2 + q_0^2 & 2 * (q_2 * q_3 + q_1 * q_0) \\ 2 * (q_1 * q_3 + q_2 * q_0) & 2 * (q_2 * q_3 - q_1 * q_0) & -q_1^2 - q_2^2 + q_3^2 + q_0^2 \end{bmatrix} \quad (19)$$

A special property of DCM is that the inverse of a DCM is equal to the transpose of that DCM. Therefore:

$$DCM_{fixed\ to\ inertial} = transpose(DCM_{inertial\ to\ fixed})$$

In order for the shuttle to hit the target, a future DCM must be used, since as the moon rotates, the descending orbital plane also rotates with the moon. The goal for the shuttle is to be in the descending orbital plane when it performs the second burn. Therefore the first burn must inject the shuttle into the descending plane when time is equal to current time plus time of flight so when shuttle perform second burn, the moon will rotate to the correct point. In

order to accomplish this, prediction of future quaternion and angular velocity vector must be made. Figure 16 shows the relationship between quaternion and time.



**Figure 16: Quaternion vs. Time**

Fortunately all elements of the quaternion have a linear relationship with time, and the slope of  $q_0$  is about  $0.7486e-6s^{-1}$ , slope of  $q_1$  is about  $-0.1495e-6$ , slope of  $q_2$  is about  $-0.2390e-6$ , and slope of  $q_3$  is about  $1.0657e-6$ . Moreover, the angular velocity vector of the moon’s centric fixed frame remains constant with respect to the moon’s centric inertial frame.

When the future DCM was found, the changing velocity vector can then be converted into the inertial frame, and the shuttle will inject itself into the correct descending orbital plane. The initialization of the shuttle’s position and velocity vector were done using the relative position with respect to the landing site.

*3. Ascending*

Ascending model of the lunar shuttle is similar to its descending model. The difference between them is the true anomaly of landing site in the ascending orbit is equal to  $360^\circ$  minus true anomaly of landing site in the descending orbit. The lunar shuttle will perform its second burn when it reaches the apoapsis of the ascending orbit, and it will continue its initial orbit with altitude of 100 km but with different RAAN and argument of periapsis.

*4. Post-Descending or Ascending*

After the shuttle had landed or docked with the transfer vehicle, the lunar shuttle federate will calculate the fuel consumption for descending or ascending using equation 20.

$$m_{fuel} = m_{LS} * \left( 1 - e^{\frac{-\Delta V}{Isp * g}} \right) \quad (20)$$

Then the shuttle will update its mass property and recalculate the mass moment of inertia using equation 21a and 21b.

$$I_x = I_y = \frac{1}{12} m (3 * r_{cyl}^2 + h_{cyl}^2) \quad (21a) \quad I_z = \frac{m * r_{cyl}^2}{2} \quad (21b)$$

## **V. SISO Smackdown**

This project was presented in the Smackdown event hosted by SISO in Boston, Massachusetts on April 6<sup>th</sup>, 2011, where all federates were joined together to perform a federation execution. The time limit for this execution was one hour. The lunar shuttle started descending once it joined the execution, and the time for descending was about 22.5 minutes, which is very close to the predicted value. There is about a 0.35% error for descending, and this error was corrected by the slow descending method when the shuttle is 8 km above the landing site so the shuttle landed exactly on the landing point. There is also an error of 0.3% for ascending, and the result of this put the shuttle into an elliptical orbit instead of circular orbit. These errors can be introduced by the prediction of the future quaternion. However, more studies must be conducted to find out what caused the error in order to improve the model for next year's design.

## **VI. Conclusion**

The SISO Smackdown is an event that promotes modeling and simulation at the university level and raises the interest of modeling and simulation down into the K – 12 grades, since modeling and simulation is a critical tool for engineering and science. Advancing the students knowledge in modeling and simulation will prepare undergraduate and graduate students for their future careers in engineering and science.

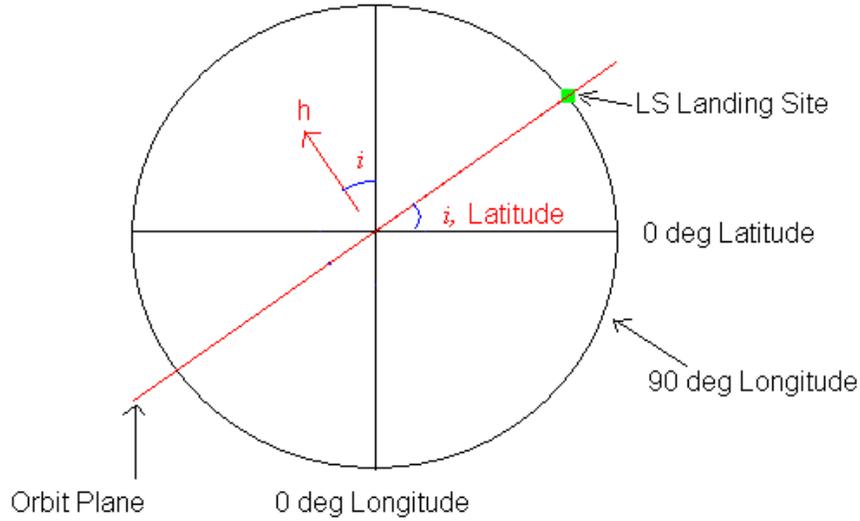
This year's Smackdown event was very successful, and caught the attention of university students and faculty. Participating teams did an excellent job in putting everything together, and students have learned many things related to modeling and simulation. Participating teams will continue to be involved in next year's Smackdown. The author will lead a team from Pennsylvania State University to participate in the Smackdown event in 2012.

The SISO Smackdown event not only improved the student's modeling skills, but also gave the student the chance to practice their engineering knowledge, such as orbital dynamics, physics, thermal, and other subjects in engineering and science.

## Appendix

### A. Derivation of equation 14:

Assume the landing site has latitude of  $x$ , and longitude of  $90^\circ$ . With  $i$  equal to  $x$ , the orbit plane will be orientated as show in figure below.



**Figure 17: Orbit Plane Orientation for Latitude =  $x$ , and Longitude =  $90^\circ$**

From Figure 17, the RAAN is equal to  $0^\circ$  or Longitude -  $90^\circ$ . Therefore, for a general case, if inclination of the orbit plane equals to latitude of the landing site, then the RAAN equals the Longitude (degree) -  $90^\circ$ . Therefore:

$$\Omega = (\text{Longitude of landing site in degree}) - 90^\circ$$

### B. Sample Calculation of Descending Orbit:

Known:

- $r_{LS,cir} = 1837100 \text{ m} = r_{a,des}$
- $r_{land,des} = 1735252 \text{ m}$
- $\theta_{Land,des} = 230^\circ$
- $Latitude = 26^\circ 09' 14.4'' \text{ N}$
- $Longitude = 3^\circ 23' 28.35'' \text{ E}$
- $\mu_M = 4.9027779e12 \text{ m}^3/\text{s}^2$

1. Define unique descending orbit in 2D space

$$e_{des} = \frac{(r_{a,des} - r_{Land,des})}{[r_{Land,des} \cdot \cos(\theta_{Land,des}) + r_{a,des}]} = 0.141122$$

$$a_{des} = \frac{r_{a,des}}{(1 + e_{des})} = 1609906.7 \text{ m}$$

$$v_{a,des} = \sqrt{\mu_M \left( \frac{2}{r_{a,des}} - \frac{1}{a_{des}} \right)} = 1513.98 \text{ m/s}$$

$$h_{des} = r_{a,des} * v_{a,des} = 2781334572.1 \text{ m}^2/\text{s}$$

$$E_{a,des} = 180^\circ = \pi \text{ rad}$$

$$\tan\left(\frac{\theta_{land,des}}{2}\right) = \sqrt{\frac{(1+e_{des})}{(1-e_{des})}} * \tan\left(\frac{E_{land,des}}{2}\right) \quad \rightarrow \quad E_{land,des} = 236.52^\circ = 4.1280 \text{ rad}$$

$$M_{a,des} = \pi \text{ rad} = 180^\circ$$

$$M_{land,des} = E_{land,des} - e_{des} * \sin(E_{land,des}) = 4.2457 \text{ rad} = 243.26^\circ$$

$$dM = M_{land,des} - M_{a,des} = 1.1041 \text{ rad} = 63.26^\circ$$

$$TOF = \frac{dM * h_{des}^3}{[\mu_M^2 * (1 - e_{des}^2)^{3/2}]} = 1018.5 \text{ sec}$$

$$\begin{aligned} \vec{r}_{Land,peri} &= r_{Land,des} * \cos(\theta_{Land,des}) \hat{p} + r_{Land,des} * \sin(\theta_{Land,des}) \hat{q} + 0 \hat{w} \\ &= -1115398.49 \hat{p} - 1329280.15 \hat{q} + 0 \hat{w} \text{ (m)} \end{aligned}$$

$$\vec{v}_{a,des} = \frac{\mu_M}{h_{des}} * [-\sin(\theta_{a,des}) \hat{p} + (e_{des} + \cos(\theta_{a,des})) \hat{q} + 0 \hat{w}] = 0 \hat{p} - 1513.98 \hat{q} + 0 \hat{w} \text{ (m/s)}$$

$$\vec{v}_{land,des} = 1350.34 \hat{p} - 884.31 \hat{q} + 0 \hat{w} \text{ (m/s)}$$

$$v_{LS,cir} = \sqrt{\frac{\mu_M}{r_{LS,cir}}} = 1634.5 \text{ m/s}$$

$$\vec{v}_{LS,cir} = 0 \hat{p} - 1634.5 \hat{q} + 0 \hat{w} \text{ (m/s)}$$

$$\vec{v}_{final} = 0 \hat{p} + 0 \hat{q} + 0 \hat{w} \text{ (m/s)}$$

$$\vec{dv}_1 = 0 \hat{p} + 120.52 \hat{q} + 0 \hat{w} \text{ (m/s)}$$

$$\vec{dv}_2 = -1350.34 \hat{p} + 884.31 \hat{q} + 0 \hat{w} \text{ (m/s)}$$

## 2. Define unique orbit in 3D Space for non-rotating moon

Set  $i$  = Latitude of landing site =  $26.152^\circ$

$\Omega$  =  $90^\circ$  - (Longitude of landing site in degree) =  $-86.608^\circ$

$$\vec{r}_{land,fix} = (R_M + h) \sin(90^\circ - Lat) \cos(Long) \hat{I} + (R_M + h) \sin(90^\circ - Lat) \sin(Long) \hat{J} + (R_M + h) \cos(90^\circ - Lat) \hat{K} = 1554856.03 \hat{I} + 92136.07 \hat{J} + 764873.65 \hat{K} \text{ (m)}$$

$$\begin{bmatrix} 1554856.03 \\ 92136.07 \\ 764873.65 \end{bmatrix} = \begin{bmatrix} \cos(\Omega) \cos(\omega) - \sin(\Omega) \sin(\omega) \cos(i) & -\cos(\Omega) \sin(\omega) - \sin(\Omega) \cos(\omega) \cos(i) & \sin(\Omega) \sin(i) \\ \sin(\Omega) \cos(\omega) + \cos(\Omega) \sin(\omega) \cos(i) & -\sin(\Omega) \sin(\omega) + \cos(\Omega) \cos(\omega) \cos(i) & -\cos(\Omega) \sin(i) \\ \sin(i) \sin(\omega) & \sin(i) \cos(\omega) & \cos(i) \end{bmatrix} \begin{bmatrix} -1115398.49 \\ -1329280.15 \\ 0 \end{bmatrix}$$

Solve for unknown  $\omega$ ,  $\omega = 3.8397 \text{ rad} = 220^\circ$

So DCM convert from perifocal to moon centric fixed is

$$DCM_{peri\ to\ inertial} = \begin{bmatrix} -0.62128 & -0.64838 & -0.44001 \\ 0.73057 & -0.68234 & -0.02607 \\ -0.28333 & -0.33766 & 0.89761 \end{bmatrix}$$

Then changing velocity in fixed frame is

$$\vec{dv}_1 = -78.114 \hat{i} - 82.205 \hat{j} - 40.680 \hat{k} \text{ (m/s)}$$

$$\vec{dv}_2 = 265.57 \hat{i} - 1589.92 \hat{j} + 84.0 \hat{k} \text{ (m/s)}$$

### 3. Rotating Moon

Defining the starting position and velocity of the lunar shuttle for a rotating moon requires the instantaneous value of the quaternion from environment federate. Follow the process described in the third step of section IV-B-2.

### **Acknowledgments**

The author Zuqun Li would like to extend my profuse appreciation to his mentor Edwin Zack Crues for providing excellent guidance throughout this project. Author would also like to thank co-mentor Daniel Dexter for his help on Trick and TrickHLA. Appreciation is also expressed to Michael Red and Zena Perryman for their help on the travel to Boston for the SISO Smackdown. Zuqun would also like to thank both USRP coordinators, Veronica Seyl and Courtney Crooks, for providing the internship opportunity. Last but not least, author would also like to thank Tung Pham, Joseph Hubbard, and Meghan Daley for making this internship a wonderful memory.

### **References**

IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. (2010, August 18). New York, NY, USA: The Institute of Electrical and Electronics Engineering, Inc.

Vallado, D. A. (2001). *Fundamentals of Astrodynamics and Application 2nd Edition*. Kluwer Academic Pub.