Ground Contact Modeling for the Morpheus Test vehicle Simulation

Luis Cordova
L-3 Communications, Houston, TX 77004

Luis Cordova

Morpheus Simulation Technical Lead

L-3 STRATIS Division
NASA JSC Engineering Directorate
Software, Robotics, and Simulation Division
Spacecraft Software Systems Engineering Branch

The Morpheus vertical test vehicle is a robotic lander being developed at the NASA Johnson Space Center (JSC) to test autonomous landing and hazard detection technology. The existing ground contact simulation model for this project was not very realistic and needed improvement. The first development cycle added capability to define vehicle contact points (CP) and to keep track of their states in the lander reference frame (LFRAME). These states are used with a spring damper model to compute a CP contact force. The lateral force is then overwritten, if necessary, by the Coulomb static or kinetic friction force. The second development cycle added capability to use the PolySurface class as the contact surface. The class can load terrain data in STL (Stereo Lithography) format for use in line of sight (LOS) intercept computations. A polygon frame (PFRAME) is computed from the intercept facet normal and used to convert the CP state to PFRAME. Two flat plane tests validate the transitions from kinetic to static and vertical impact. The hazard terrain test and a graphical display program were used to visually validate the model. The improved model is numerically inexpensive, robust, and produces results that are reasonable.

I. INTRODUCTION

The Morpheus vertical takeoff and landing (VTOL) test vehicle is being developed at the NASA Johnson Space Center (JSC). The vehicle is an autonomous robotic lander with two liquid oxygen tanks, 2 liquid methane tanks, a single gimbaled orbital main engine (OME), and several reaction control system (RCS) jets. The purpose is to use the vehicle to test new hazard detection technologies and green fuel propulsion systems. In late 2011, the Autonomous Landing and Hazard Avoidance Technology (ALHAT) project joined forces with Morpheus to pursue these common goals. The plan is to mount the ALHAT sensors and navigation system onto the Morpheus vehicle for testing purposes.

The Morpheus project has been using Lean Development practices to save time and money. The team has used existing mature software packages for simulation and flight software whenever possible. The Core Flight Software (CFS) system, developed by NASA Goddard Space Flight Center, is being used as the backbone of the Morpheus Flight Software (FSW). The CFS package is a mission and platform independent FSW environment with a reusable Core Flight Executive (CFE). On the simulation side, we are using the JSC Trick Simulation Environment. This package is a generic simulation toolkit that allows users to rapidly develop, integrate, and operate simulations. The JSC Engineering Orbital Dynamics (JEOD) package is used for its reusable and validated set of orbital dynamics models. In addition, we have the Valkyrie package for generic simulation models and the Morpheus package for project specific simulation models.

As the Morpheus project was operating under a lean budget, the initial GroundContact class was not very robust or realistic. The vehicle state while on the ground was fixed not through dynamics, but by overriding the state itself. There was a simple kinetic friction model and no static friction model. The model was not sufficient for several reasons. Detection of the post landing state is very important for the Morpheus vehicle. Landing detection is used by the FSW to shut down the engine. Also, the purpose of the project is to deal with hazardous terrain. A ground contact model that can handle complicated terrain would be advantageous. For these reasons, it was decided to improve the model with the caveat that it should not be too computationally expensive.

II. 1ST DEVELOPMENT CYCLE

A. Points and Frames

The first update was to allow the user to specify any number of contact points (CP) on the vehicle in the structure frame. The class name for these objects within GroundContact is ContactPoint. For the current simulation, four points are specified at the bottom of the legs and four for the top outer edge of the tanks. The relative states of these points are computed in LFRAME (L for lander). This frame is a planet fixed frame with the origin below the initial vehicle center of mass (CM), at the level of the bottom of the legs. It is oriented in the Up, East, North direction. Alternatively, if the vehicle is initially in flight, the origin of the LFRAME is still beneath the initial vehicle CM, but at the Earth radius level.

B. Spring Damper Model

The transition between static and kinetic friction is simplified by keeping track of modes with the "friction_flag" variable. The options are NONE, STATIC, and KINETIC. If friction_flag is at NONE, but the CP is below the LFRAME YZ plane, the friction_flag is set to STATIC. In addition, the CP LFRAME position is saved. If friction_flag is at STATIC or KINETIC, a delta vector from the saved CP position to the current CP position is computed. This delta vector is used to compute a spring force. The CP velocity vector is used to compute a damper force. It is important to note that the direction is opposite the CP velocity vector, and not along the position delta

American Institute of Aeronautics and Astronautics

vector. The last step is to null the vertical component of the spring damper force if it is negative. It is assumed that the ground can only apply a positive pushing force. In the following equation, F is the friction force, k is the spring coefficient, x is the delta vector, c is the damping coefficient, and v is the CP velocity vector.

C. Coulomb Static Friction

Coulomb friction assumes that the force of friction is directly proportional to the normal load and is independent of the visible area of contact between the two surfaces. As such, the CP structure needs a position but no area. If the friction_flag is in STATIC mode, additional steps are taken. First, the maximum possible lateral static friction force is computed based on the vertical normal force computed in the previous step. If the previous lateral force magnitude is less than this maximum static force, no additional step is taken. If the previous lateral force magnitude is larger than the maximum static force, then the lateral components of the force are overwritten by the maximum static friction force. In addition, the friction_flag is changed to KINETIC. Equation 2 was used to compute the maximum static friction force possible. In this equation, N is the vertical normal force and μ_s is the coefficient of static friction [Ref. 1].

D. Coulomb Kinetic Friction

If the friction_flag is in KINETIC mode, the maximum lateral kinetic friction force is computed. This force is based solely on the kinetic friction coefficient and the vertical normal force computed from the spring damper model. It is not dependent on the velocity magnitude, but is in the opposite direction of the CP velocity vector (projected onto the LFRAME YZ plane). If in KINETIC mode, the lateral force is always overwritten by the maximum kinetic friction force. In addition, if the CP velocity is within a certain small magnitude, the friction_flag is changed to STATIC. In Equation 3, f is the lateral kinetic friction force, N is the vertical normal force, and μ_k is the coefficient of kinetic friction [Ref. 1]. Figure 1 shows the typical relationship seen between force and velocity in Coulomb or "slip-stick" friction [Ref. 2].
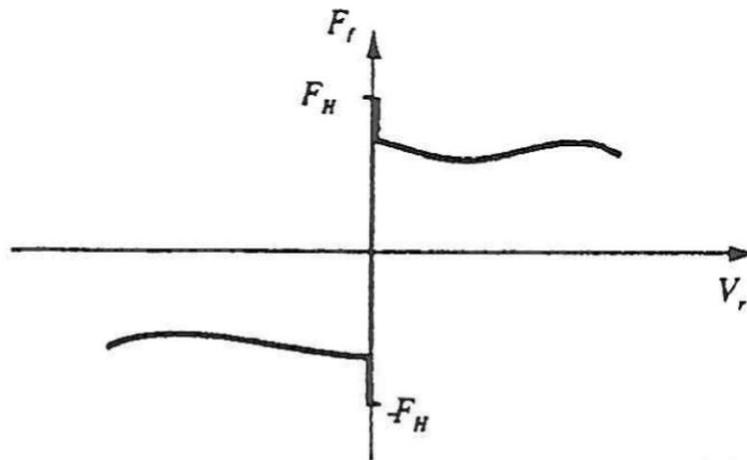

Figure 1: Standard slip-stick friction model

American Institute of Aeronautics and Astronautics

E. Force Collection

After going through the friction flag modes, the resulting LFRAME force on the CP point is crossed with the delta vector from the vehicle CM to the CP point to compute a torque, also in LFRAME. Within the GroundContact class, the forces and torques from the CP points are summed together and converted to structure frame for JEOD force and torque collection, where they are applied at the vehicle CM location.

III. 2ND DEVELOPMENT CYCLE

A. Limitations of Model

The LFRAME ground contact model was sufficient for the JSC based simulations and testing that was being run at the time. The terrain in Houston is relatively flat and there was no need at the time for a complicated terrain model. However, there were limitations. The LFRAME ground contact model is essentially a flat Earth contact model. The further the vehicle landed from its initial take off point, the larger the discrepancy. Another problem had to do with the ALHAT altimeter and velocimeter sensor models. These models have line of sight (LOS) sensors that interface with our existing PolySurface class object to compute range and range rate to polygon intersection. The PolySurface object that we have been using consists of four triangle facets forming a square. The central point is below the vehicle at ground level. The other four points are in the north, east, south, and west direction from that point, each 2000 meters apart. The distance from the Earth center is the same for all five points. Therefore, the four triangle facets do not form an exact flat plate. This non flat PolySurface object created a discrepancy with the flat LFRAME ground contact model. For these reasons and more, it was decided to improve the ground contact model by using the PolySurface class object as the contact surface.

B. PolySurface Class

The PolySurface class is an existing Valkyrie package model that has been used for other projects at JSC. It contains a load STL (Stereo Lithography Format) method. STL is a common file format used on many 3D CAD software packages. The method loads an STL file and saves the data into a list of TriangleFacets. The convert facet method uses a scale, vector offset, and Direction Cosine Matrix (DCM) to transform the point data into a different reference frame if necessary. The purpose is to convert the point data into a reference frame in which the coordinates do not change with time. In our case, the points are defined in Earth Centered Earth Fixed (ECEF) coordinates in the STL file and do not need to be changed. Internally, normals for each TriangleFacet are computed based on the right hand rule and the order of the points. The normal vectors should point away from the inner space of a closed PolySurface object, or in our case, in the general up direction. The facet normals are important because they are used to reject ray intersections that hit the "back" side of the facets. The ray intersect method takes a LOS origin and ray and returns the facet that is intercepted by the ray and closest to the LOS origin. Finally, the PolyOctree object within PolySurface is used to efficiently make use of the ray intercept method.

The PolyOctree class is a tree data structure in which each node has exactly 8 children. In this case, it is used to partition a 3 dimensional bounding box into 8 octants. Each octant box can then be recursively partitioned into another 8 octants. The smallest bounding box is called a leaf and contains a list of TriangleFacets. A facet that is partially in the box is included entirely in the box. The purpose of the PolyOctree is to efficiently find the facet that is intersected by the LOS ray and that is closest to the LOS origin. It works by first finding the bounding box that is intercepted by the ray. Once the box is found, it recursively goes up that branch until it finds the leaf and finally the facet.

C. PolySurface Ground Contact Integration

To begin, the GroundContact class was given a PolySurface pointer object as a member. The ContactPoint class was given three new members: LOS_origin_struc, LOS_vector_struc, and LOS_origin2ap_range. The LOS_vector_struc is defined by the user in a data file. It holds the LOS ray direction in structure frame. One must be defined for each of the CP points. The LOS_origin2ap_range is hard coded to be 0.8 meters. It is used to derive the LOS_origin_struc vector. The purpose is to make sure that the LOS origin never goes into the PolySurface. If it does, no intercept will be found.

The GroundContact compute_T_pfix_NEN() method was created to compute the DCM from ECEF to LFRAME as before. The NEN stands for Normal, East, North frame. It takes a normal vector in planet fixed (ECEF) coordinates as input. This new method is also used to compute the DCM from ECEF to PFRAME. The PFRAME is a planet fixed frame that shares the same origin as the LFRAME. However, what is important is the orientation. This frame is orientated in the intercept facet Normal, East, North frame. If the facet is horizontal to the ground, then the PFRAME is the same as the LFRAME. The purpose is to convert the CP state and delta vector coordinates from LFRAME to PFRAME. Once this is done, the spring damper and coulomb friction models can be used just as before.

Several other changes had to be done to integrate the PolySurface into GroundContact. First, the LFRAME state is no longer used to decide whether the CP point is below the ground, thus initiating the friction_flag mode actions. Instead, the LOS range to the PolySurface is computed. It is compared to the hard coded CP LOS_origin2ap_range value. If the LOS range is less than LOS_origin2ap, then the CP point is below the ground. The delta vector is also slightly different. Before, upon touching the ground, the CP position was saved. Now, the computed intercept position on the facet is saved. This is used together with the current CP position to compute the delta vector used for spring force computation. The final update was the cushion pad on each of the CP points. In effect, the CP point is extended along the LOS direction by a user defined cushion distance. The purpose is to keep the actual CP point from sinking too much into the ground.

D. KSC Hazard Terrain

A 100m x 100m area of simulated lunar terrain was constructed at the NASA Kennedy Space Center (KSC) for the ALHAT project. The terrain consists of artificial craters, hills, and rocks. The end goal is to have the ALHAT navigation sensors mounted onto the Morpheus vehicle so that they can navigate over this hazard terrain and autonomously find a safe place to land.

In order to load this terrain into our simulation we had to create a new method to load AC3D CAD data. The original CAD data is in a local surface frame coordinates. An offset vector places the model in the correct location and a DCM converts the data into planet fixed coordinates. The CAD data along with the same offset vector and DCM are used to load the terrain model into NASA's Engineering DOUG Graphics for Exploration (EDGE) graphical display program. After some troubleshooting, we were able to observe that our simulation contact model and the graphical representation of the model were aligned correctly.

E. Ground Contact Settings

There are several ground contact parameters that need to be set to reasonable values to properly simulate the Morpheus vehicle landing. These include the coefficients of static and kinetic friction, the spring coefficient, and the damping coefficient. Unfortunately, there is currently no Morpheus vehicle friction test data to validate the ground contact model. Parameters values were set to reasonable values. For example, the coefficients of static and kinetic friction are set to 0.8 and 0.3. The spring and damping coefficients were derived from Equation 4 and 5 [Ref. 3]. In these equations, m is one fourth of the vehicle total mass (4 legs), g

is the gravity coefficient, h is the desired sink height, and d is the damping ratio. Ideally, h should be as small as possible, but the problem with making it too small is excessive vibration. This vibration shows up in our inertial measurement unit (IMU) sensor models. One solution is to reduce the integration time step, but that increases our simulation run time. A compromise of 2 cm sink height was chosen. For the damping ratio, we chose 0.1. This was chosen by dropping the vehicle 20 meters and observing the resulting bounce using EDGE.

IV. SIMULATION EXAMPLES

A. Static to Kinetic Friction Transition Test

For the Static to Kinetic Friction Transition Test, the terrain file contains four facets that form a flat plane. The test begins with the vehicle on the ground. The aerodynamics model is turned off to eliminate drag effects. The vehicle body frame is aligned with the LFRAME (Up, East, North direction). Vehicle Legs 1, 2, 3, and 4 are in the northeast, southeast, southwest, and northwest direction, respectively. A linearly increasing external force is applied along the lateral north direction.

As shown in Figure 2, the lateral friction force increases at the same rate as the applied force. The model exerts static friction on all four legs up until 24.3 seconds. Around that time, as shown in Figure 4, Legs 2 and 3 transition from static to kinetic friction. The transition happens earlier for these legs because the applied lateral force causes the vehicle to lean a fraction of a degree in the north direction. This increases the normal force on the northern legs and decreases the normal force on the southern legs. Therefore, the maximum static friction the contact model can apply on the southern legs decreases. These contact points transition into weaker kinetic friction. This creates a momentary dip at 7750 N in Figure 2. Between 24.3 and 32.0 seconds, the vehicle still manages to stay in the same location because although the friction force on Leg 2 and 3 decreases, the friction force on Leg 1 and 4 increases. This can be seen in Figure 5. At 32.0 seconds, the applied force overcomes the static friction from Leg 1 and 4. Once all four legs are in kinetic friction, the vehicle starts to slide in the north direction and rotate along the vertical direction. Figure 3 shows Coulomb static and kinetic friction in action. This kinetic friction force is not constant because the four facets did not form a perfectly flat plane and because the vehicle slid along the boundary between two facets.
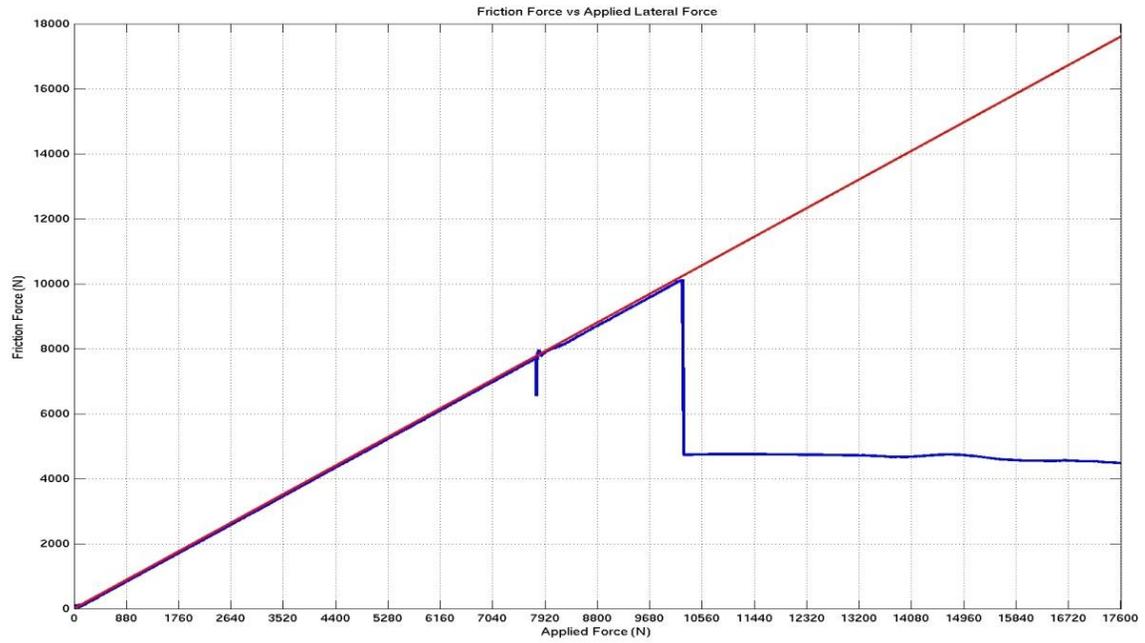
American Institute of Aeronautics and Astronautics

Figure 2: Lateral Friction Force versus Applied Force
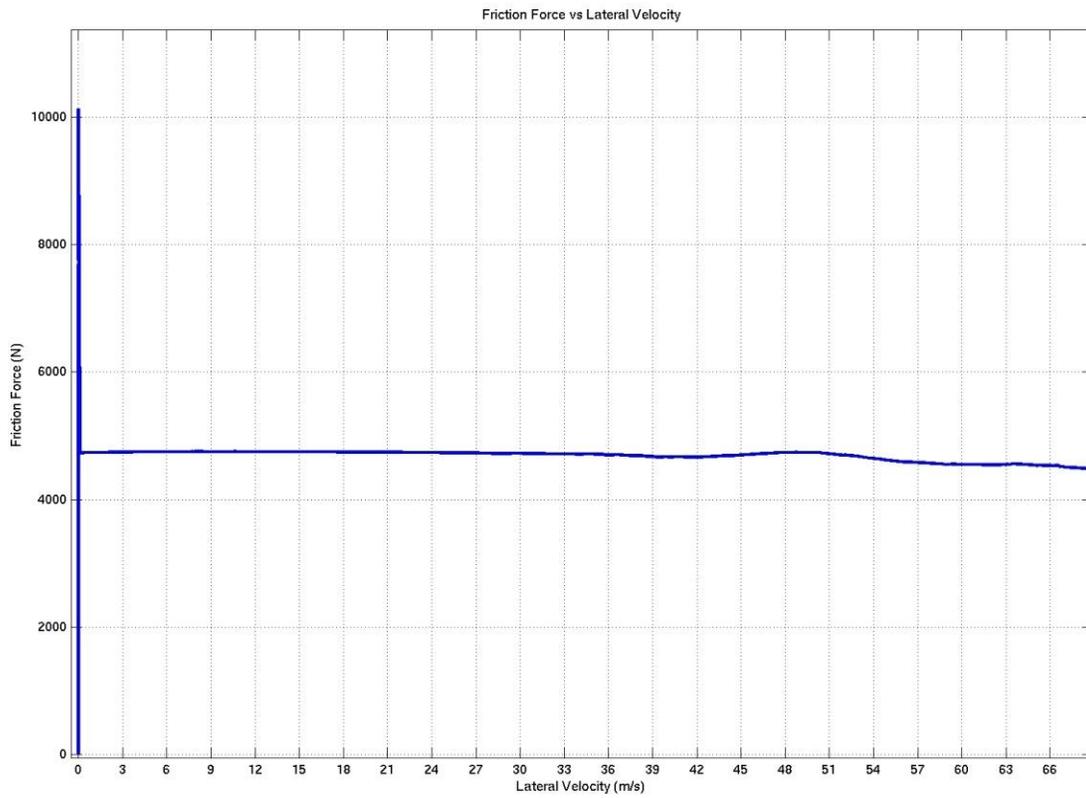
Figure 3: Lateral Friction Force versus Velocity

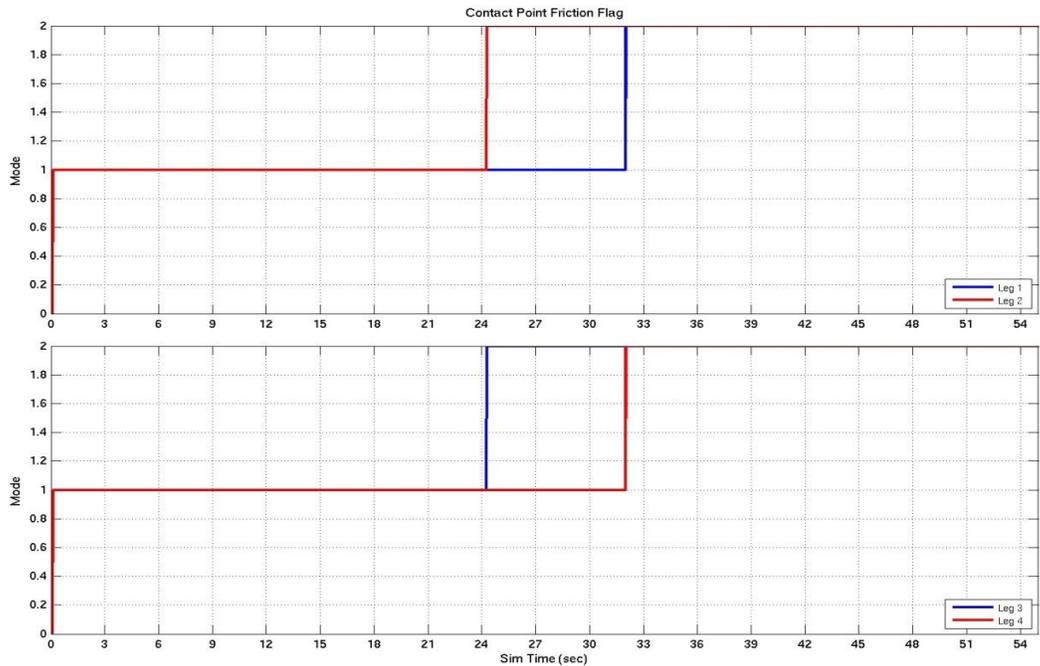American Institute of Aeronautics and Astronautics
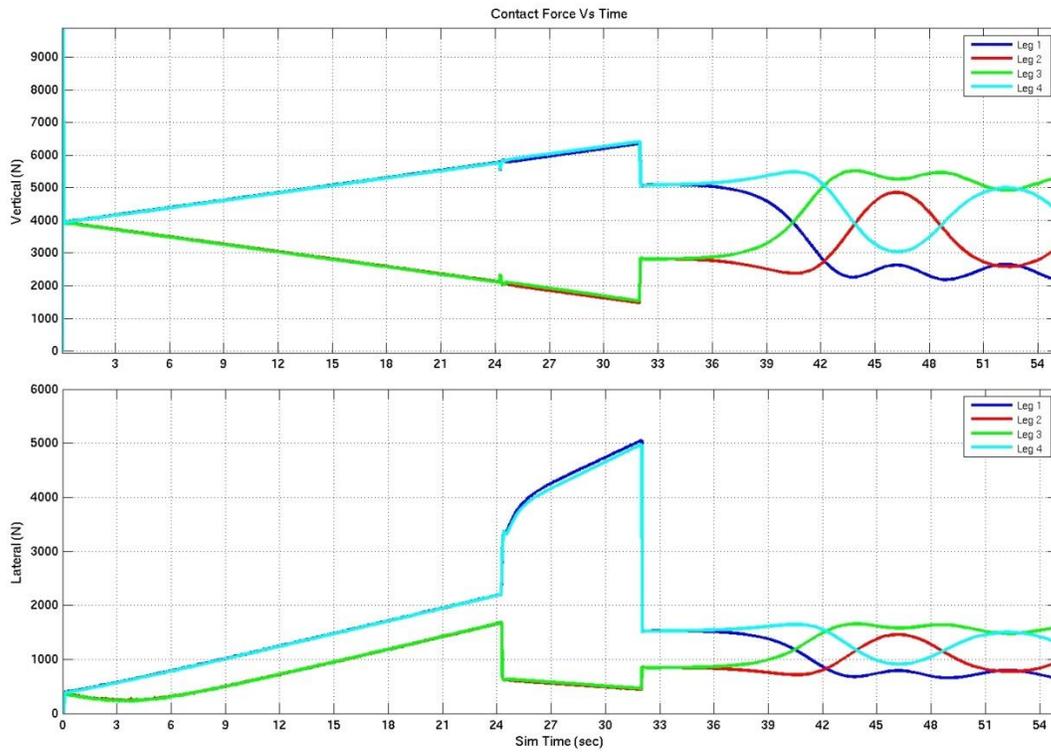
Figure 4: Friction Modes

Figure 5: Contact Force versus Time

8

B. Vertical Drop Test

The Vertical Drop Test uses the same terrain file, orientation, and aerodynamics mode as the previous test. However, the initial altitude is 20 m, and the linearly increasing external force is no longer applied. Also missing are the 8 contact points at the bottom of the legs and top of the tanks. Instead, only one contact point at the vehicle center of mass is used. Because of model limitations, the same point is used 6 times, with each instance having a different LOS direction. These are in the +X, -X, +Y, -Y, +Z, and –Z body directions, respectively. Since the vehicle body is aligned with the LFRAME, this translates to up, down, east, west, north, and south at the beginning of the simulation. The LFRAME origin in this case, is at the vehicle CM at the beginning of the simulation.

When the simulation begins, the vehicle falls into the ground and bounces several times before coming to a rest. The amount of spring and damping is configurable as were the coefficients of static and kinetic friction in the Static to Kinetic Friction Transition Test. Figure 6 shows the results. Figure 7 shows the friction modes for each contact point/direction. Specifying six directions as described above ensures that at every moment in time, between 1 and 3 contact point directions will hit the ground contact plane. The vehicle appears to settle at around 21.3 m. That is because in the nominal case when the vehicle is on the ground, the vehicle CM is 1.3 m above ground.
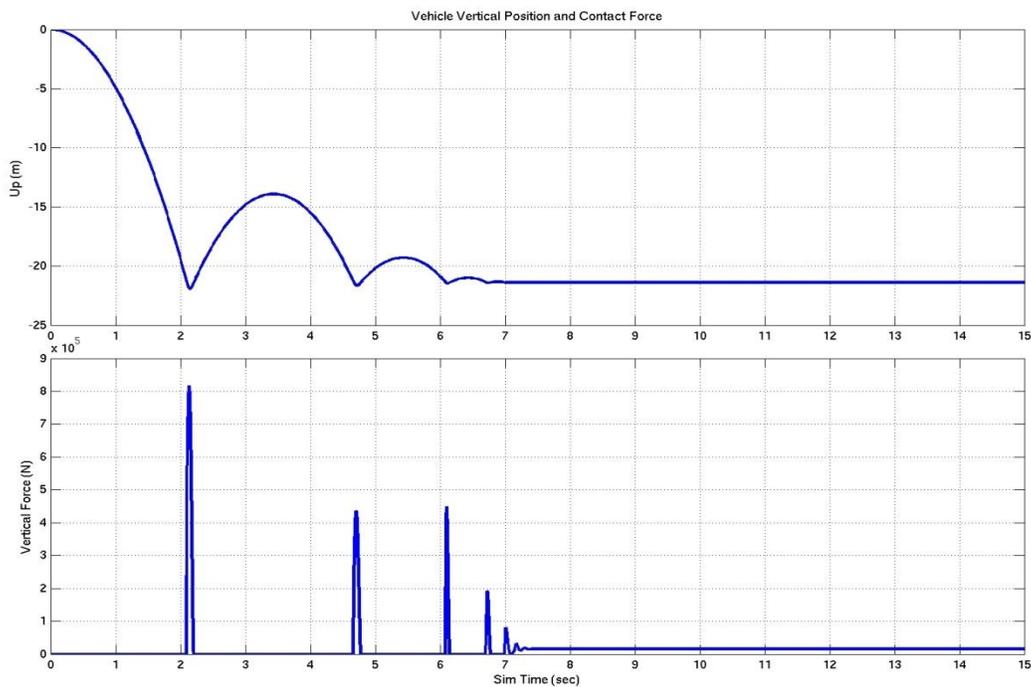

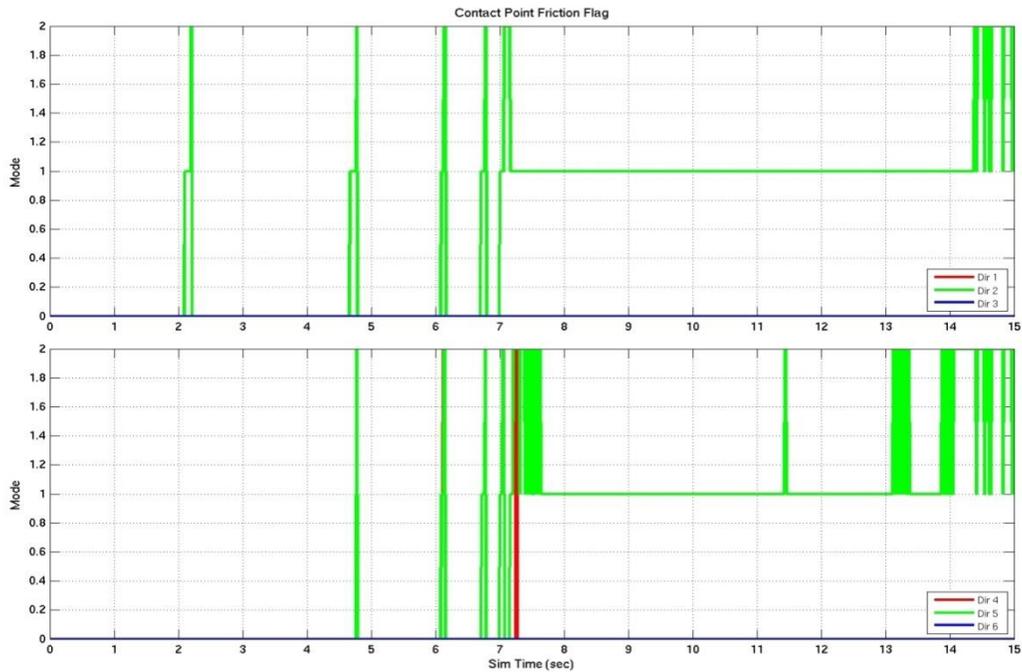**Figure 6: Vehicle Vertical Position and Contact Force**

American Institute of Aeronautics and Astronautics

Figure 7: Friction Modes

C. Hazardous Terrain Visual Test

The Hazardous Terrain Visual Test was used to verify that the simulation contact model and the visual representation of the hazard field are aligned correctly. The original hazard field CAD data was loaded into the simulation and into EDGE, along with the same offset vector and DCM. The initial tests showed that the simulation and EDGE graphics did not match. After some troubleshooting, it was discovered that the vehicle CAD model used in EDGE was out of date and undersized. After proper scaling, the simulation contact model and EDGE graphics came into alignment. The hazard field was inclined by 45 degrees in both the simulation and in EDGE. The vehicle was then allowed to tumble down the hazard field. The results were as expected. The vehicle bounced off rocks and momentarily sank into craters. Figure 7 shows the Morpheus vehicle and hazard field displayed on EDGE.
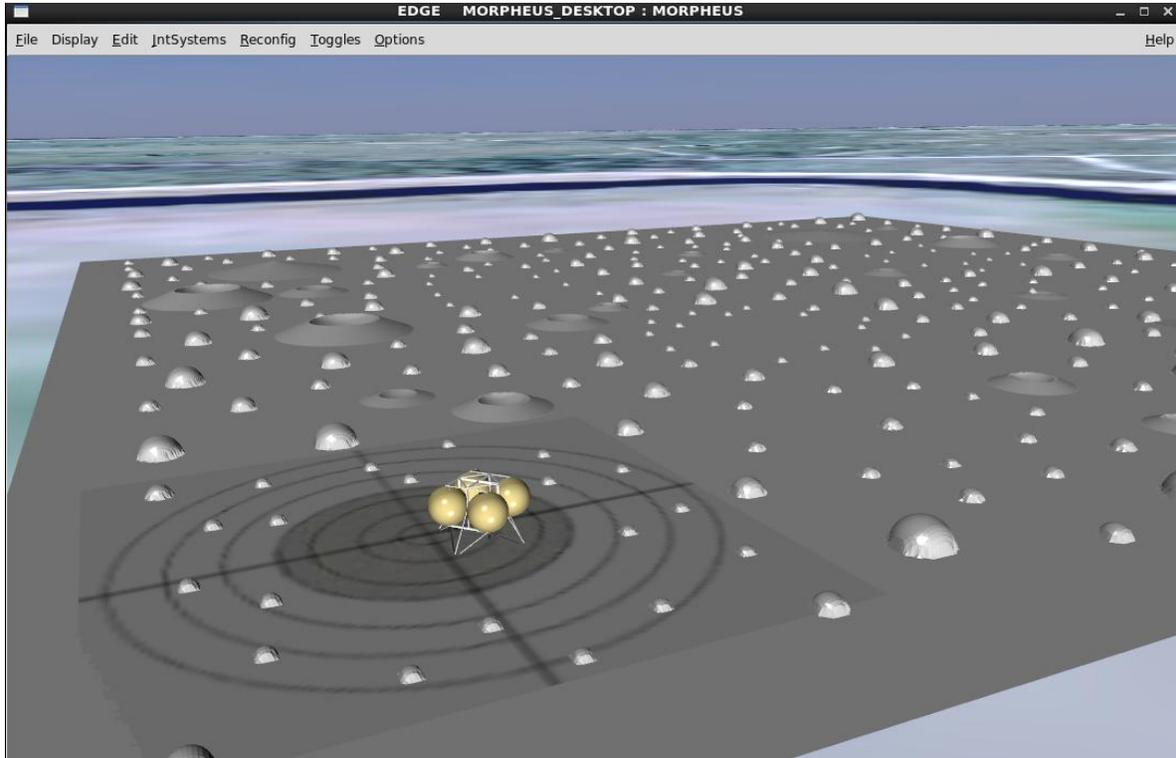
American Institute of Aeronautics and Astronautics

Figure 7: Morpheus Vehicle and Hazard Field displayed on EDGE

V.   CONCLUSION

The Morpheus ground contact model has been updated to include contact with polygon surfaces. The improved model is numerically inexpensive, robust, and produces results that are reasonable. In addition, it brings the contact model into alignment with the ALHAT altimeter and velocimeter sensor models. These models use ray tracing to a polygon surface to compute measurements. Now the same terrain data can be used for both. The new model allows us to test the behavior of the Morpheus vehicle while landing on an inclined plane or crater. It will specifically allow us the ability to simulate and test the Morpheus vehicle on the KSC hazard field before the actual KSC flight tests.

American Institute of Aeronautics and Astronautics

VI. REFERENCES

[1] Bedford, A., and Fowler, W., *Engineering Mechanics: Statics, Second Edition*, Addison Wesley Longman, Menlo Park, CA, 1999 Addison Wesley Longman.

[2] Haessig Jr, D. A., and Friedland, B., "On the modeling and Simulation of Friction", *Journal of Dynamic Systems, Measurements, and Control*, Vol 113, September 1991, pp. 355 to 362.

[3] Raiszadeh, B., and Way, D., "Ground Contact Model for Mars Science Laboratory Mission Simulations", *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, MN, 2012.