Design and Analysis of the Deorbit and Earth Entry Trajectories for SPORE

Author:
Matthew Nehrenz

Advisor:
David Spencer

May 2, 2012

Design and Analysis of the Deorbit and Earth Entry Trajectories for SPORE

Matthew T. Nehrenz and David A. Spencer
Georgia Institute of Technology, Atlanta, GA, 30332

Small Probes for Orbital Return of Experiments (SPORE) provides on-orbit operation and recovery of small payloads. The flight system architecture consists of a service module for on-orbit operations and deorbit maneuvering, and an entry vehicle for atmospheric entry, descent, and landing. Prior to approximating a landing footprint with a Monte Carlo analysis on the entry trajectory, the entry state uncertainties must be characterized. These uncertainties arise from errors induced by the guidance system and thruster pointing control during the deorbit maneuver. In order to capture the effect that these errors have on the entry state uncertainty, the service module's attitude determination and control system (ADCS) and guidance system were both modeled in Matlab. By incorporating the ADCS loop into the guidance loop, the effect of pointing errors during the deorbit trajectory combined with errors in the guidance system can be assessed. A Monte Carlo analysis is performed on this 3+3 DOF deorbit simulation (which terminates at entry interface), resulting in an entry state covariance. The analysis is performed on the three orbits under consideration for SPORE: ISS, LEO, and GTO. Finally, the resulting entry state covariance from the deorbit simulation is used as input for an entry, descent, and landing trajectory Monte Carlo analysis. Landing footprint, heating, and g-loading are considered for trajectories targeting Woomera Test Range in Australia.

Nomenclature

d = thruster moment arm
F = force
g = gravitational constant
$\boldsymbol{H}$ = thruster mapping matrix
K_P = proportional gain
K_D = derivative gain
M = moment
q_e = quaternion error
r_e = position error
T = commanded torque
t = time
u = thruster force
ΔV = delta-v
v = velocity
v_e = velocity error
γ = flight path angle
$\boldsymbol{\tau}$ = control command
ω_e = body rate error

I. Introduction

The Small Probes for Orbital Return of Experiments (SPORE) flight system architecture provides a scalable, modular approach to the return and recovery of multi-purpose probes from orbit. Capable of accommodating payload volumes ranging from the 1-unit (1U) dimensions of 10x10x10 cm to 2U and 4U payloads, SPORE is targeted to carry flight experiments related to thermal protection system (TPS) performance validation, biological science, and materials science missions. SPORE is also designed to accommodate the return of small payloads from the International Space Station (ISS).

SPORE consists of an entry vehicle attached to a service module. The service module provides power, thermal control, communications, propulsion, attitude control, and deorbit guidance while SPORE is on orbit. After the vehicle has spent sufficient time in orbit, the service module performs a deorbit maneuver, allowing for the entry vehicle to reenter the Earth's atmosphere and touch down at a specified landing site. Figure 1 shows the baseline sequence of events for a typical SPORE mission.
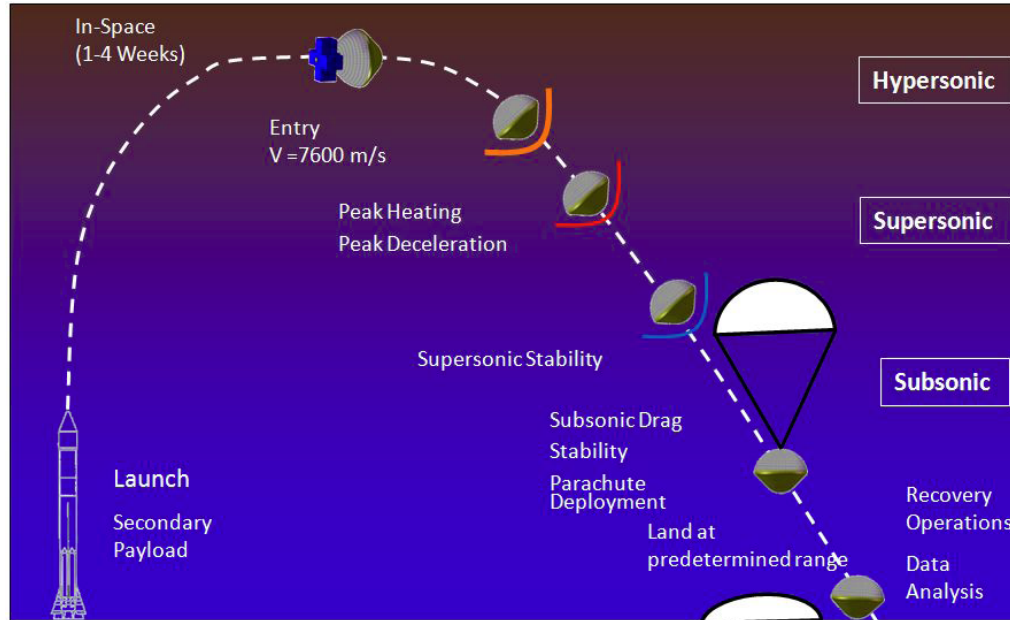
Figure 1: Sequence of events for the SPORE mission

The deorbit trajectory will target a certain entry state that will allow the entry, descent, and landing (EDL) trajectory to meet thermal, g-loading, and landing ellipse constraints.

A common method for analyzing EDL trajectories is to use a Monte Carlo analysis to determine whether or not the vehicle will meet the constraints during reentry. However, in order to perform a meaningful Monte Carlo analysis, statistical uncertainties on the entry state must be known. In this case, these uncertainties are a function of the errors of the guidance system and the attitude determination and control system (ADCS). Inaccuracies in the GPS sensors and thruster pointing errors during the deorbit burn will dictate covariance of the state of the vehicle at atmospheric interface. The entry state covariance is calculated by performing a Monte Carlo analysis on the deorbit trajectory.

This paper describes the service module's ADCS and guidance system and explains how they are incorporated into a deorbit simulation. Then, using this simulation, a deorbit Monte Carlo analysis is used to determine SPORE's entry state uncertainties. Finally, an EDL Monte Carlo analysis is performed using the previously attainted entry state uncertainties to show that the trajectory passes all constraints.

II. ADCS

The service module performs all ADCS tasks on orbit and during deorbit. Characterizing the errors in the ADCS is pivotal for understanding how ADCS inaccuracies affect the entry state uncertainty. During deorbit, the ADCS will point the main engine along the desired thrust vector that is provided by the guidance system. Thus, any pointing error will lead to variation in the vehicle's entry state at the end of the deorbit trajectory.

To control the vehicle, the service module utilizes thruster-only control with 12 thrusters for 3-axis stabilization. The thrusters were sized using an approximate size and mass of the vehicle as well as quantifying the expected disturbance torques on the spacecraft. Due to the low thrust necessary to stabilize the vehicle, cold gas thrusters will be used. Since the on orbit mission lifetime is on the order of a few weeks, the amount of fuel for ADCS needed will be low.

The service module's ADCS currently utilizes a proportional-derivative control law to command the necessary torque M_i for vehicle pointing and stabilization, and is given as

$$M_i = -K_{P_i}q_{e_i} - K_{D_i}\omega_{e_i} \qquad \text{for } i = 1,2,3 \qquad (1)$$

where K_p and K_d are system gains, ω_e represents the body rates, and q_e is the error between the current quaternion and desired quaternion. However, the ADCS must apply this torque with some combination of the 12 thrusters. At the same time, fuel needs be minimized. To solve for the combination of thruster firings given a certain torque while minimizing fuel (referred to here as the thruster allocation logic), the simplex algorithm is used.[1,2] The simplex algorithm solves the linear optimization problem,[3] and for the following simulation, a built-in Matlab function is used to perform the optimization. The following expressions are the ADCS command input and the thruster command, respectively.

$$\tau = \begin{bmatrix} F_x & F_y & F_z & M_x & M_y & M_z \end{bmatrix}^{\text{T}} \qquad (2)$$

$$u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_1 & u_{10} & u_{11} & u_{12} \end{bmatrix}^{\text{T}} \qquad (3)$$

In the case of the service module ADCS, only moments will be present in the command input since no translational actuation is required; the translational dynamics are addressed in the deorbit guidance section. The thruster command is a vector of thrusts from each of the 12 thrusters, satisfying the following expression,

$$0 \le u_i \le u_{max} \qquad \text{for } i = 1, \dots, 12 \qquad (4)$$

where u_{max} is the maximum thrust output for each one of the 12 cold gas thrusters. To map the thruster command to the ADCS torque command, the geometry of each thruster's placement on the vehicle must be characterized. Figure 2 shows the thruster placement on the service module.
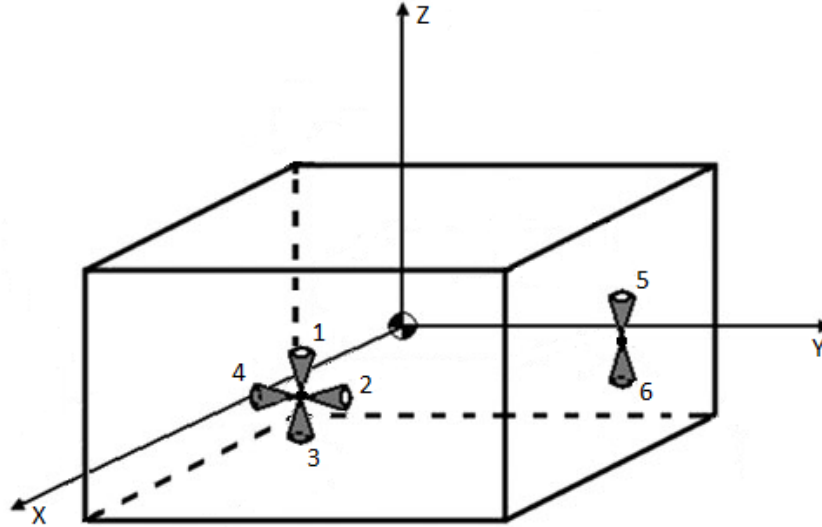
Figure 2: ADCS thruster placement

Only two sides are shown above for clarity; the thruster cluster on the –X face is identical to the +X face, and the thruster cluster on the –Y face is identical to the +Y face. With this thruster placement, the thruster command can be mapped to the ADCS torque command as follows

$$\tau = Hu \qquad (5)$$

3

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -d_y & d_y & 0 & 0 & 0 & 0 & d_y & -d_y \\ d_x & 0 & -d_x & 0 & 0 & 0 & -d_x & 0 & d_x & 0 & 0 & 0 \\ 0 & -d_x & 0 & d_x & 0 & 0 & 0 & d_x & 0 & -d_x & 0 & 0 \end{bmatrix} \tag{6}$$

where H is the thruster mapping matrix with 12 columns, each representing a thruster. The top three rows describe the direction the thrusters are pointed, while the bottom three rows describe the moment each thruster imparts on the vehicle. In this case, the ADCS will not provide a net force on the vehicle, only moments as follows: $\tau = \begin{bmatrix} 0 & 0 & 0 & M_x & M_y & M_z \end{bmatrix}^{\mathrm{T}}$. Therefore, the top three rows of the mapping matrix are not necessary. The entries in the bottom three rows are moment arms, or the distance from the vehicle's center of mass. The above mapping matrix assumes the thrusters are placed equidistant from the center of mass; however, this assumption is for clarity in the thruster placement figure and is not necessary for the convergence of the thruster allocation algorithm.

The minimization problem is written as follows

$$\text{minimize} \sum_{i=1}^{12} u_i$$

$$\text{subject to:}$$
$$\tau = Hu$$
$$0 \le u_i \le u_{max}$$

(7)

The above formulation minimizes the total fuel used over all 12 thrusters while satisfying the thruster mapping constraint and the thrust magnitude constraint. Matlab's built in linear programming function solves this problem quickly and accurately, so the 12 thrust values for the ADCS simulation can be computed easily each ADCS cycle.

A Matlab program was written to simulate the service module's ADCS. Figure 3 is a block diagram illustrating the 3 DOF ADCS simulation.
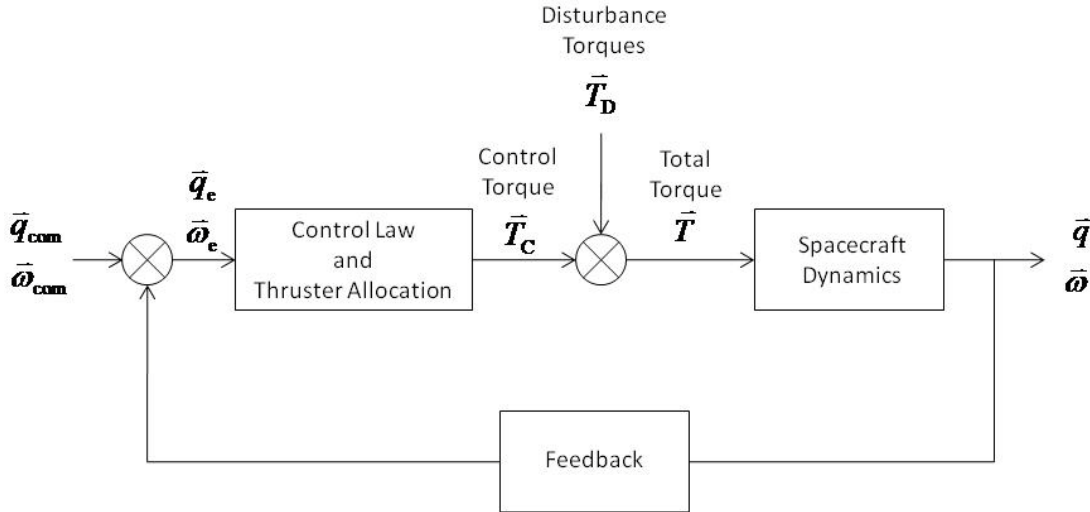

Figure 3: ADCS block diagram

The block diagram shows how the control law commands a torque that the thruster allocation logic must convert into the 12 thruster outputs. Then, these outputs are fed to the system dynamics along with the following disturbance toques: gravity gradient, solar torque, aerodynamic torque, and main engine misalignment. The expressions for the disturbance torques are left out here, and the system dynamics relies on Euler's equations of rotational motion.[4] Thruster output from each thruster over time is shown in Figure 4.
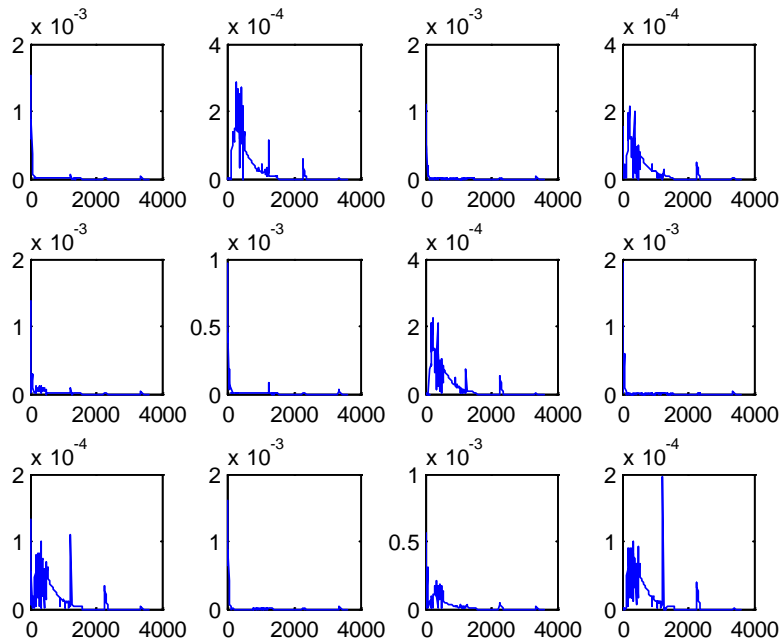
Figure 4: Each of the 12 thruster outputs in Newtons

The above plots represent an hour of on-orbit operation; for a mission lasting 4-6 weeks, the fuel used for on-orbit operations and deorbit amounts to approximately 3 kg. With a functional ADCS simulation, errors can be added into either the attitude determination sensors or the control actuation to produce a desired pointing error. In this case, ADCS errors are added such that the overall pointing error in the system is 1°. Figure 5 shows the pointing error over a sample deorbit burn.
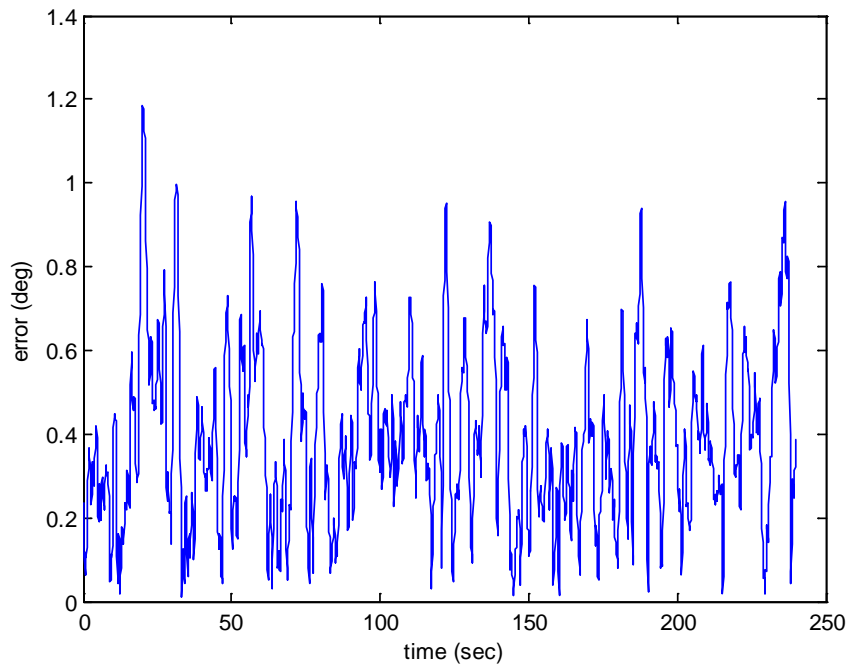
Figure 5: Pointing error during deorbit burn

As seen in the above figure, the ADCS simulation mimics a control system with a pointing error of approximately 1° during a deorbit burn. In the following section, the ADCS loop is inserted into the guidance loop in order to ascertain the effect that this pointing error has on the guidance system and the entry state uncertainty.

III. Deorbit Guidance

When the vehicle needs to deorbit, the service module's guidance system will command a thrust direction for the main engine. A proportional-derivative guidance law is used to track a predetermined, nominal deorbit burn trajectory. The guidance law calculates a thrust vector, T, for the main engine to apply that will track the nominal trajectory; it takes the form

$$T_i = -K_{P_i} r_{e_i} - K_{D_i} v_{e_i} \qquad \text{for } i = 1,2,3 \tag{8}$$

where r_e and v_e are the position and velocity errors, respectively. The position and velocity errors are simply the difference between the vehicle's state sensed by GPS sensors and the state at which the vehicle should be on the nominal deorbit trajectory at that specific time. This gains associated with this guidance law will affect the magnitude of the commanded thrust; however, the service module's main engine has a constant thrust engine, so the only purpose of the gains will be to weight the importance of either the position error or the velocity error. The ADCS will align the vehicle's thrust vector with the commanded thrust direction from the guidance law. A block diagram for the combined system with the ADCS running within the guidance loop is shown in Figure 6.
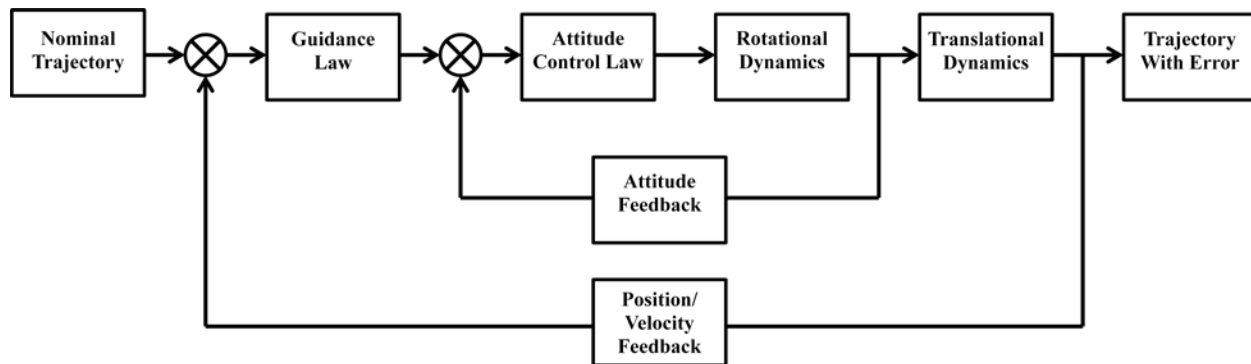

Figure 6: Integration of ADCS into the guidance system block diagram

Since the ADCS loop runs within the guidance loop, the ADCS must have a sufficiently faster cycle than the guidance system; currently, the ADCS frequency is 10 Hz and the guidance system frequency is 1 Hz. This gives the ADCS enough time to align the vehicle's thrust vector with the commanded thrust vector from the guidance system. A Matlab simulation combining the ADCS with the guidance system was written for the purpose of characterizing the entry state uncertainty, which will be discussed in the following section. In addition to the error added into the ADCS loop, position and velocity errors in the GPS sensors are added in as well. The simulation decouples the rotational dynamics from the translational dynamics: for each cycle of the outer guidance loop, the ADCS portion runs the rotational 3 DOF simulation and saves the resulting thrust vector to be applied over that particular guidance system cycle. The resulting simulation is not 6 DOF since the rotation dynamics are decoupled from the translational dynamics, but it is a 3+3 DOF simulation.

As mentioned previously, the guidance system tracks a predetermined nominal trajectory. The nominal trajectory should be fuel optimal in order to minimize fuel mass on board the vehicle. A Hohmann transfer is an optimal transfer;[5] however, this type of transfer assumes an impulsive burn, and in reality, the service module will perform a fixed length finite burn. In general, spacecraft orbit transfers are treated as impulsive because the length of the burn is negligible compared to the orbital period.

For this analysis, two different nominal trajectories will be created. One trajectory will be the result of thrusting in the opposite direction of the vehicle's velocity vector, referred to in this paper as "opposite to velocity thrusting." The other trajectory will be created by thrusting in the direction of the impulsive Hohmann transfer burn for the duration of the deorbit burn, referred to as "Hohmann direction thrusting." This direction will be tangent to the near-circular orbit at the beginning of the burn and will stay constant in the inertial frame for the duration of the burn. Figure 7 illustrates the two approaches.
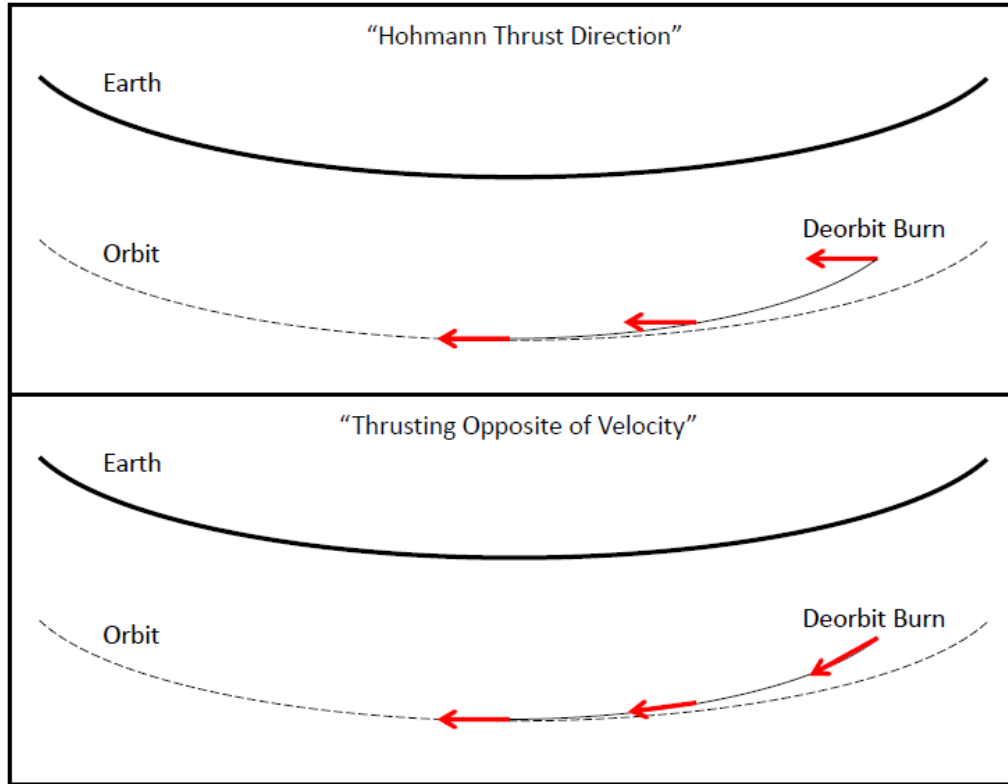
**Figure 7: The two deorbit approaches under consideration**

For each approach, the length of the burn will be varied until the desired vehicle state at entry interface is obtained. Since the service module's engine is constant thrust, the burn time will be the only parameter to vary. To target certain entry conditions—in this case a specific flight path angle is targeted—a bisection root solver is wrapped around the trajectory code. The root solver calculates the duration of thrust necessary to target a given flight path angle at entry interface.

The two approaches will be compared based on the fuel needed to target a given flight path angle. Each approach suffers non-impulsive burn losses: the Hohmann direction thrusting velocity approach suffers gravity losses according to

$$\Delta V_g = \int_{t_i}^{t_f} g \sin \gamma \, dt \tag{9}$$

Since the direction of thrust remains fixed with the Hohmann direction thrusting, the flight path angle is zero at the start of the burn but is non-zero and steadily grows after the start of the burn. On the other hand, the opposite to velocity thrusting approach keeps the flight path angle fixed at zero, effectively making the gravity loss term zero for all time; however, the thrust direction deviates from the Hohmann thrust direction thereby losing the optimal attribute of the Hohmann transfer burn direction. To compare the two approaches, the ISS orbit case will be considered because it is the lowest orbit and therefore requires the longest deorbit burn. For each approach, a range of entry flight path angles were targeted with the root solver mentioned earlier. Figure 8 shows the burn time needed to reach a given entry flight path angle for each approach.
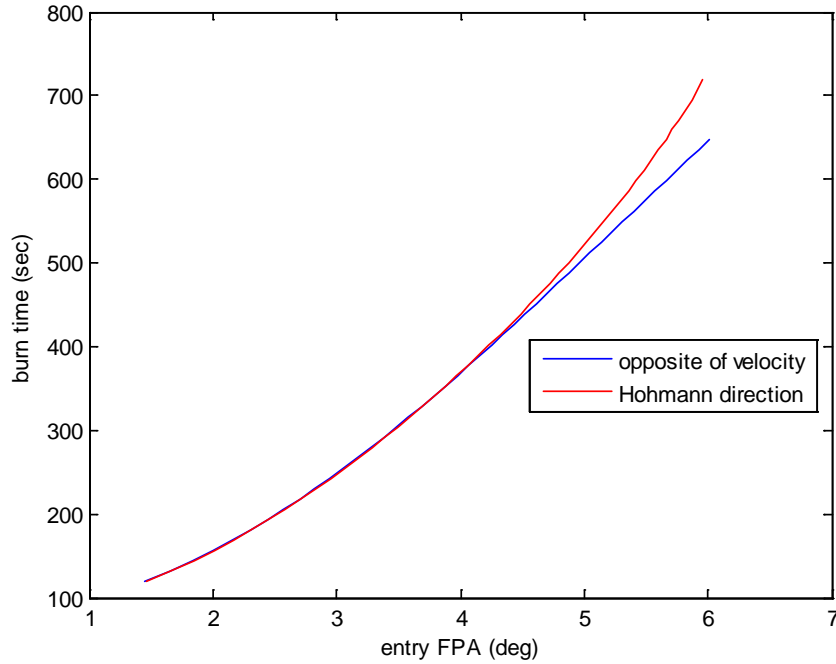
Figure 8: Comparison of the two deorbit burn approaches

While there is not much difference between the two approaches at lower flight path angles, after an entry flight path angle of approximately 4.5°, the Hohmann thrusting direction approach requires a shorter burn and therefore less fuel, since the engine is constant thrust. Again, this analysis was performed on the ISS orbit case because it required the longest burn, thus having the worst approximation to an impulsive burn. The GTO burn occurs so far from Earth that the burn time is on the order of a few seconds, which is very close to an impulsive approximation.

IV. Entry State Uncertainty

To calculate entry state uncertainty, the Matlab simulation that incorporates the ADCS error and the guidance system error was wrapped in a Monte Carlo analysis. The errors included in the analysis are ADCS pointing error, GPS position error, and GPS velocity error. These errors are summarized in Table 1. The errors are the inputs to a Monte Carlo analysis for each of the three reference orbits in question: LEO, ISS, and GTO. For each orbit, 1000 Monte Carlo runs are performed. Table 2 shows the resulting 3-σ uncertainties for each of the three orbits. Entry altitude, or atmospheric interface, is defined here as 125 km.

From Table 2, it is apparent that entry flight path angle uncertainty decreases as the burn time increases. As discussed in the previous section, the lower orbits, ISS and LEO, require a longer burn to achieve enough delta-v for deorbit. As the deorbit burn is performed, a long burn time provides the guidance system with more time to perform trajectory corrections. Additionally, from GTO, there is a much greater distance to coast after the deorbit burn which allows the vehicle to deviate from the nominal trajectory more. In contrast to the trend in entry flight path angle uncertainty, the entry velocity uncertainty increases as burn time increases. With longer burn time, there is increased opportunity for velocity errors due to the ADCS not pointing the main engine exactly along the commanded thrust vector.

**Table 1: Input errors for deorbit Monte Carlo analysis (3-σ)**

ADCS Pointing Error	1°
GPS Position Error	1.8 m
GPS Velocity Error	0.03 m/s

Table 2: 3-σ entry state uncertainties

	LEO	GTO	ISS
Longitude (°E)	0.1663	0.0121	0.4993
Latitude (°N)	0.2401	0.0235	0.1176
Velocity (m/s)	0.3393	0.0032	0.0379
Flight path angle (°)	0.0053	0.0111	0.0036
Heading (°)	0.0181	0.004	0.0189

8

V. Entry, Descent, and Landing

With the entry state uncertainties obtained in the previous section, a Monte Carlo analysis can be performed on the EDL segment of the mission. A separate Matlab simulation was written for this purpose, utilizing atmospheric relative, 3 DOF equations of motion with bank angle modulation.[6] Bank angle modulation allows for a lift vector to be rotated, effectively applying a force on the vehicle. The SPORE entry vehicle generates no lift; however, wind force is added into the simulation via the lift vector and bank angle modulation. Atmospheric winds and all atmospheric uncertainties are modeled using EarthGRAM (Global Reference Atmospheric Model) to randomly generate a table of 1000 atmospheres. These atmospheres were used in the Monte Carlo simulation and represent the uncertainties in density, temperature, and winds from 0 to 125 km altitude. Plots of the 3-σ variation in the atmospheres are shown in Figure 9.
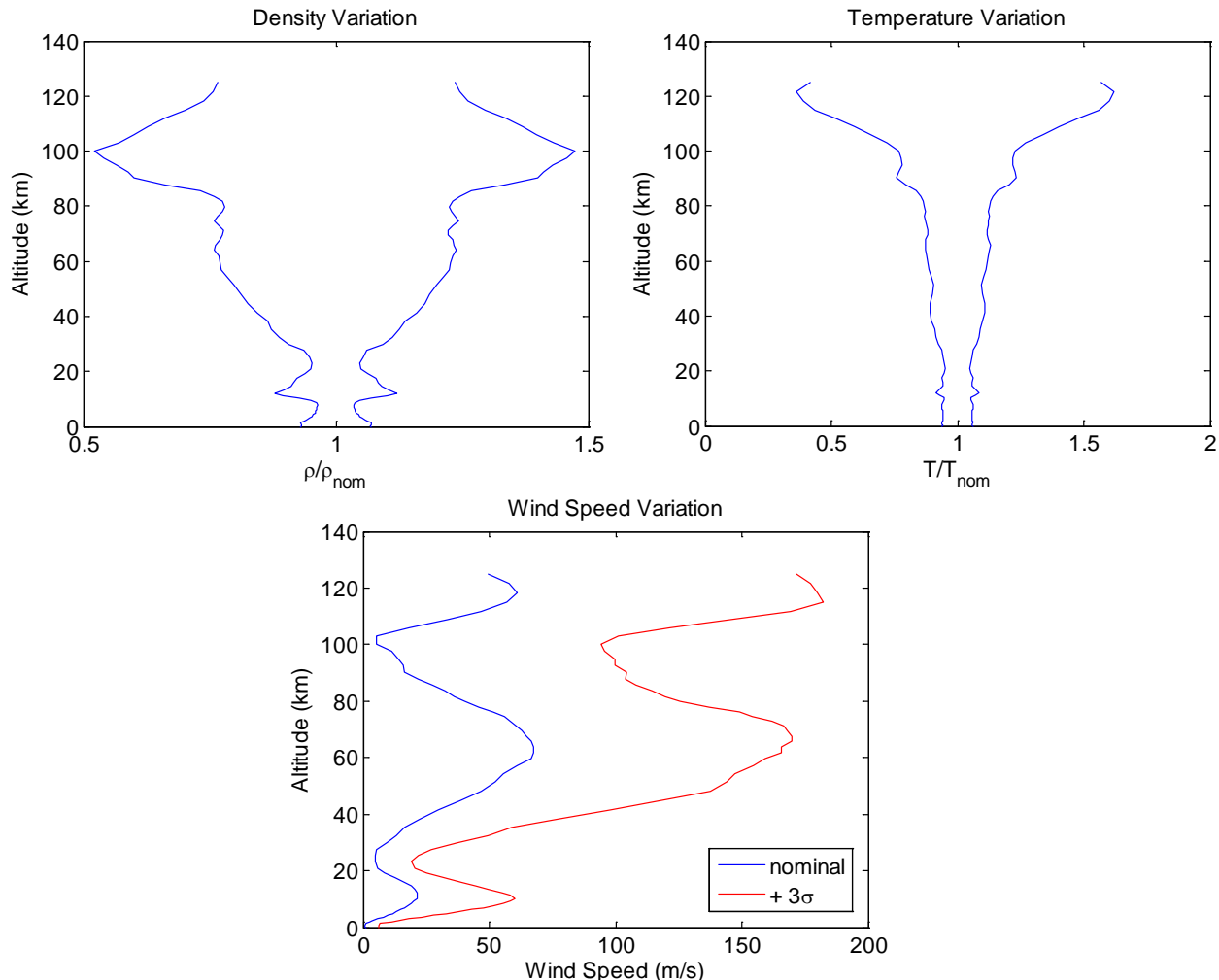

Figure 9: Variation in the atmospheric model

Entry state uncertainties are listed in Table 2 in the previous section. The 3-σ uncertainties for drag coefficient are 0.03 for Mach numbers greater than 10 and 0.1 for Mach numbers less than 5.

The nominal entry states for each orbit were chosen to target Woomera Test Range in Austalia. Nominal flight path angle was the variable that was tuned in order to meet specific g-loading and heating requirements. The nominal entry states, masses and diameters are given in Table 3. The LEO and GTO cases are 1U

Table 3: Nominal entry states

	LEO	GTO	ISS
Longitude (°E)	137.65	127.98	126.78
Latitude (°N)	-16.65	-36.40	-43.75
Velocity (m/s)	7780.0	9964.4	7282.0
Flight path angle (°)	-5.00	-6.72	-5.00
Heading (°)	267.10	32.20	53.70
Mass (kg)	8.74	8.74	28.70
Diameter (mm)	406	406	630

configurations while the ISS orbit case is a 2U configuration. The drag area of each vehicle is calculated using the diameter of the vehicle and the drag coefficient, C_D, which is a function of Mach number. This relationship can be found in the aerodynamic database for Mars Microprobe.[7] The Sutton-Graves relationship is used for stagnation point heating, and peak decelerations are calculated to make sure that payload loading requirements are not violated. Currently, parachute deployment is modeled as occurring at an altitude of 5 km, but in the future will be modeled as a G-switch. A deployment altitude of 5 km was chosen as it falls between the 3.1 km main deployment altitude of Stardust and 10 km main deployment altitude of the Hayabusa spacecraft.[8,9] At this altitude, the entry vehicle is traveling at a low subsonic speed at a near 90° flight path angle. The nominal trajectories are illustrated below in Figure 10 with an altitude versus velocity plot.
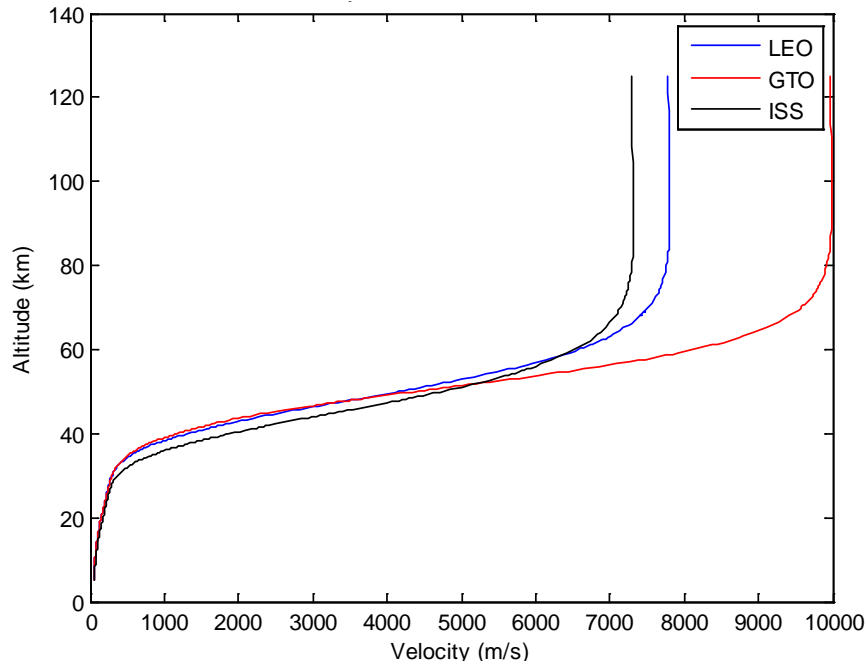

Figure 10: Nominal trajectories for the three orbit cases

Each trajectory begins at 125 km with the entry states defined in Table 3 and ends at 5 km, where the parachute is deployed. At this point, the model switches to a parachute mode.

In the parachute portion of the trajectory, two types of inflation are modeled. For the Monte Carlo simulations, an instantaneous inflation is modeled to save on run time. For the nominal simulations, a linear inflation profile is assumed using Knacke's inflation time relationship.[10] This inflation profile, combined with a time delay for line stretch, allows for re-contact analysis to be performed, ensuring that any deployed fragments of the backshell will not intersect the entry vehicle trajectory. Atmospheric wind contributions to the entry vehicle landing dispersions are modeled, assuming the parachute-vehicle system trims to the relative wind vector. Logic iss also present in the code to ensure that the maximum dynamic pressure parachute constraints are not violated. 1000 Monte Carlo cases were run in order to capture the extreme off-nominal points of the landing ellipse, heating, and g-loading. The results for the 1U LEO and GTO and the 2U ISS cases are shown in Table 4. The landing ellipse for LEO and a plot of maximum heat rate and maximum deceleration are shown in Figure 11.
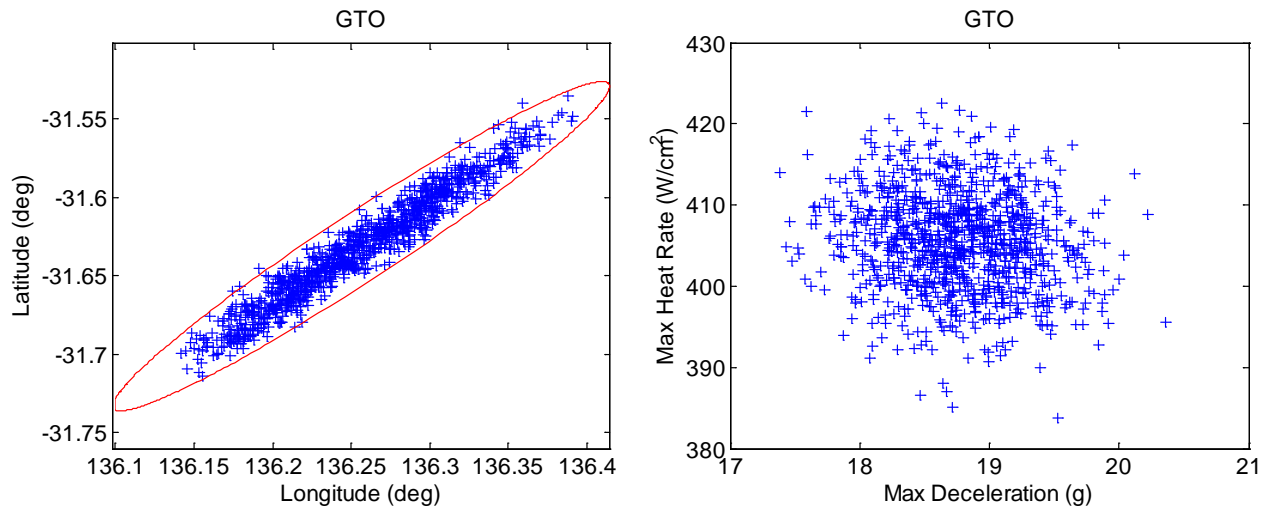
Table 4: EDL Monte Carlo results

|	LEO	GTO	ISS
Worst Case Heat Flux (W/cm²)	267.2	514.8	237.37
Worst Case Heat Load (J/cm²)	12857	22227	12192
Maximum G's	15.65	20.33	14.59
Landing Ellipse Downrange (km)	59	37	84
Landing Ellipse Crossrange (km)	34	5	31
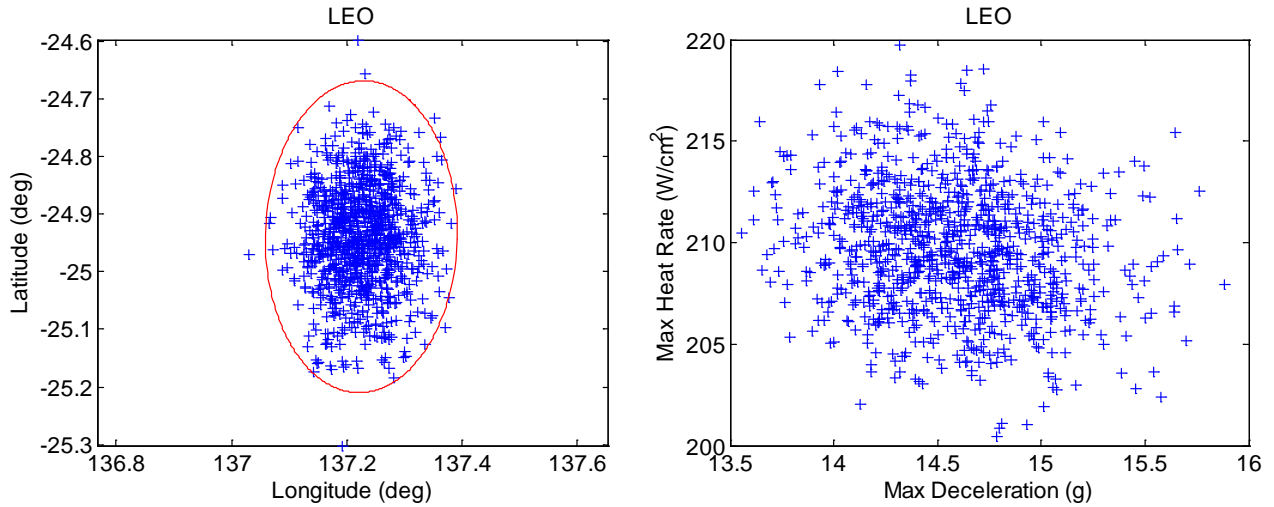
Figure 11: Monte Carlo results for LEO

The landing ellipse dimensions are found in Table 4; the size of the ellipse is acceptable for the Woomera Test Range. From a LEO orbit, maximum heat rate and maximum deceleration remain reasonably low because of the shallow -5° entry flight path angle. Results for entry from GTO are given in Figure 12.

Figure 12: Monte Carlo results for GTO

A contrast to LEO, the trajectory from GTO has a smaller ellipse that is much narrower. The reason the ellipse is smaller is because the entry flight path angle is steeper, creating an entry trajectory that is more direct to the ground. This trajectory has less time for uncertainties to affect the dispersion on the ground. The reason the ellipse is much longer downrange than cross range is because the uncertainty on entry flight path angle is higher for the GTO case. The ellipse is very narrow because the heading angle uncertainty from the GTO deorbit Monte Carlo results is much lower than for the other two orbits. Because the entry flight path angle is steeper and the entry velocity is much higher from GTO, the maximum heat rate and maximum deceleration are much higher. Since the 1U configuration is a test bed for new heat shield materials, the elevated maximum heat rate is desirable. Finally, Figure 13 shows results for the ISS case.

11
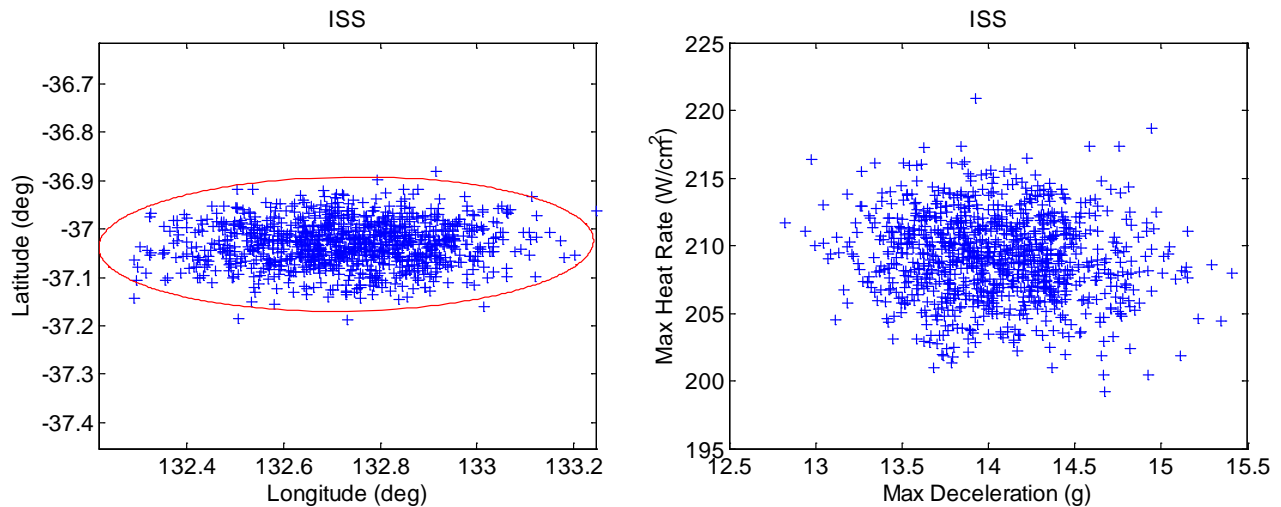
Figure 13: Monte Carlo results for ISS orbit

Given equal entry flight path angles and a smaller entry flight path angle uncertainty for the ISS case than for the LEO case, one would expect a smaller landing ellipse for the ISS case. However, the nominal entry velocity for ISS is lower, allowing for uncertainties to have a longer time to affect the dispersions. Similar to the LEO case, the maximum heat rate and maximum deceleration are relatively low; since the ISS case presented here is a 2U vehicle that will possibly contain a biological payload, the reduced heating and deceleration is necessary.

The g-loading requirements for a biological payload are: do not exceed 16 g's for 20 seconds, and do not exceed 9 g's for 100 seconds. There are no specific heat rate requirements; however, there are biological payload temperature requirements. During a collaborative design session, a thermal modeling tool used heat rate information from this trajectory analysis to make sure the payload did not overheat. Also, if any dispersed trajectories failed to satisfy g-loading or heating requirements, than the nominal entry states were varied until 100% of the dispersed trajectories met the requirements.

VI. Conclusion

It has been shown how SPORE's ADCS and guidance system are incorporated into a deorbit simulation. This simulation is used for a Monte Carlo analysis that produces the vehicle's state covariance at entry interface. The entry state covariance is then used as the input to an EDL trajectory Monte Carlo analysis to gain insight on landing footprint, heating, and g-loading. These tools will aid in the mission design process for future SPORE missions.

Since SPORE is currently in the initial design phase, these simulation tools will need to be updated throughout the design process. The attitude and guidance control laws may change; this analysis assumed standard proportional-derivative control in order to obtain initial uncertainties and results. Additionally, when the service module design is further along, more uncertainties should be included. Namely, a thrust magnitude error should be quantified through testing and added into the deorbit Monte Carlo analysis. As the SPORE design develops, more collaborative design sessions with the other disciplines will be performed to ensure the design still meets all requirements.

References

[1]Curti, F., Romano, M., and Bevilacqua, R., "Lyapunov-Based Thrusters' Selection for Spacecraft Control: Analysis and Experimentation," *Journal of Spacecraft & Rockets*, Vol. 33, No. 4, July-Aug. 2010, pp. 1143-1160.

[2]Cho, D., Jung, D., and Tsiotras, P., "A 5-dof Experimental Platform for Spacecraft Rendezvous and Docking," *Infotech at Aerospace Conference*, Seattle, WA, April 6-9, 2009, AIAA Paper 2009-1869

[3]Vanderplaats, G. N., *Multidiscipline Design Optimization*, 1st ed., Vanderplaats Research & Development, Inc., Monterey, CA, 2007.

[4]Wie, B., *Space Vehicle Dynamics and Control*, AIAA Education Series, AIAA, Washington, DC, 1998.

[5]Bate, R. B., Mueller, D. D., and White, J. E., *Fundamentals of Astrodynamics*, Dover Publications, New York, 1971.

[6]Vinh, N., Busemann, A., and Culp, R. D., *Hypersonic and Planetary Entry Flight Mechanics*, University of Michigan Press, 1980.

[7]Mitcheltree, R. A., Moss, J. N., Cheatwood, F. M., Greene, F. A., and Braun, R. D., "Aerodynamics of the Mars Microprobe Entry Vehicles," AIAA Paper 97-3658, *Atmospheric Flight Mechanics Conference*, New Orleans, LA, August 11-13, 1997.

[8]Desai, P. N., Lyons, D. T., Tooley, J., and Kangas, J., "Entry, Descent, and Landing Operations Analysis for the Stardust Re-Entry Capsule," AIAA-2006-6410, *AIAA/AAS Astrodynamics Specialists Conference and Exhibit*, Keystone, Co., Aug. 2006.

[9]Hinada, Motoki, et. al., "Parachute System of MUSES-C Reentry Capsule," *CEAS/AIAA Aerodynamic Decelerator Systems Technology Conference, 15th,* Toulouse, France, 8-11 June 1999, Collection of Technical Papers (A99-30926 07-03).

[10]Knacke, T. W., *Parachute Recovery Systems Design Manual*, 1st ed., Para Publications, 1992.