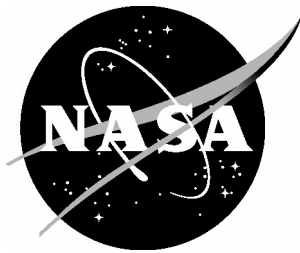


NASA/TM-2005-213779



POST II Trajectory Animation Tool Using MATLAB, V1.0

*Behzad Raiszadeh
Langley Research Center, Hampton, Virginia*

July 2005

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

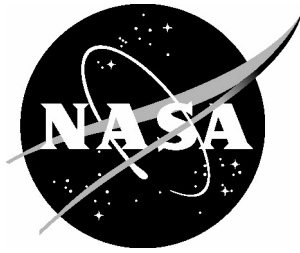
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Phone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/TM-2005-213779



POST II Trajectory Animation Tool Using MATLAB, V1.0

*Behzad Raiszadeh
Langley Research Center, Hampton, Virginia*

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

July 2005

The use of trademarks or names of manufacturers in the report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Table of Contents

	<u>Page</u>
1 Introduction.....	1
2 Method.....	1
3 Step-by-step instructions.....	2
3.1 Step 1: Installation.....	2
3.1.1 Windows installation.....	2
3.1.2 Linux installation.....	2
3.2 Step 2: Setting up the local directory	2
3.3 Step 3: POST II trajectory data	3
3.3.1 POST II input file	3
3.3.2 POST II Matfile	3
3.3.3 POST II output file	3
3.4 Step 4: Setting up master_setup.txt	4
3.5 Step 5: Setting up geom_data	7
3.6 Step 6: Setting up camera_properties.txt.....	9
3.7 Step 7: Run animation_driver.exe.....	10
References.....	11

POST II Trajectory Animation Tool Using MATLAB, V1.0

1 Introduction

Typically the output from trajectory simulation programs is analyzed using two-dimensional time history plots. It is often desired to animate the trajectory where position and orientation of the bodies in flight are put into motion. The following is a MATLAB based tool which generates a movie from the trajectory data. The animations generated using this tool serve as an engineering analysis tool to gain further insight into the dynamic behavior of flight vehicles. This tool is able to animate a single body as well as multiple vehicles in flight, and has been tailored for output generated from POST II simulations.

2 Method

This tool has been designed such that no code modification is required by the user. The animation tool obtains most of the required inputs from the POST II input file, POST II output file, and the Matfile (Figure 1). The user is required to provide master_setup.txt, geom_data and camera_properties.txt files for additional parameters.

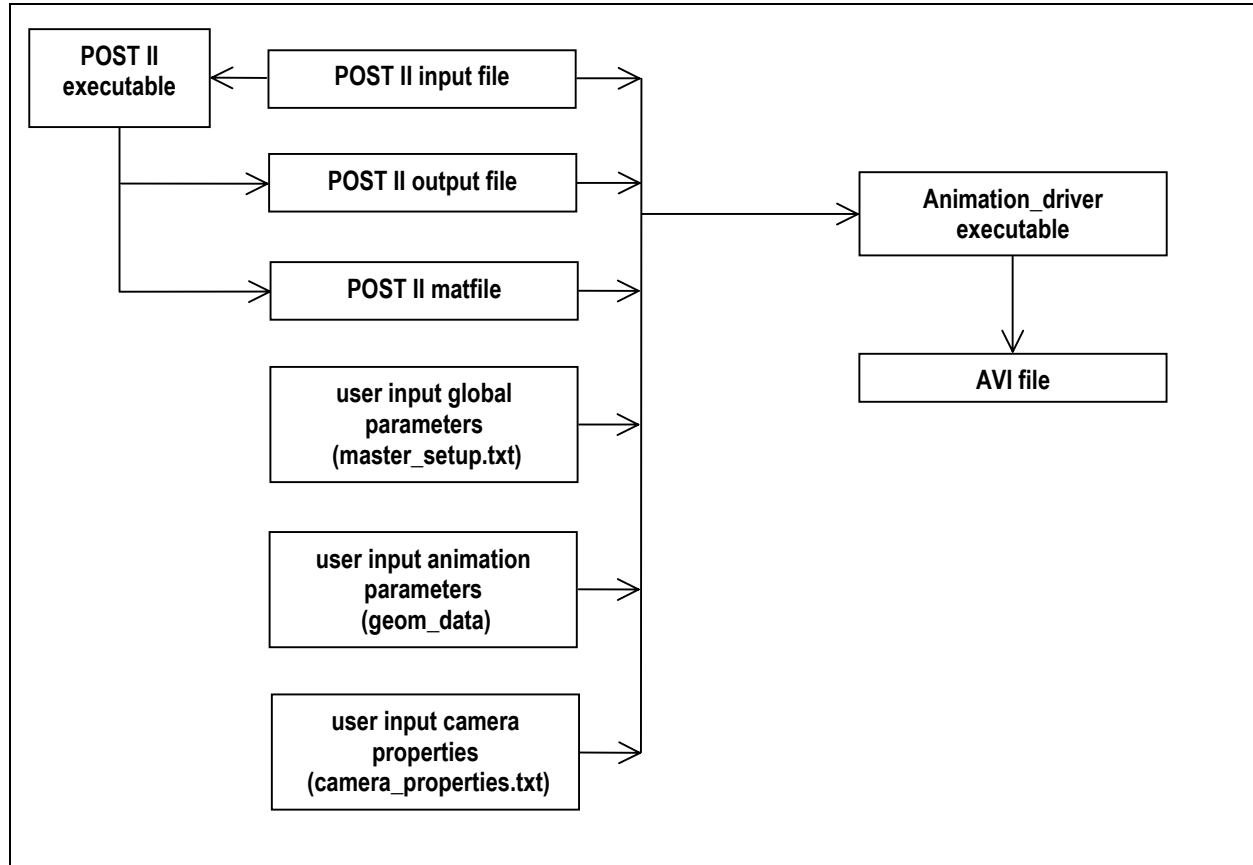


Figure 1 Animation tool flow diagram

3 Step-by-step instructions

This section outlines, step-by-step, the procedure to make a trajectory animation. The animation tool has been developed in MATLAB. All the source files have been compiled using the MATLAB Compiler. The compiled executable, named `animation_driver.exe`, is a stand-alone application and is used to generate trajectory animations in AVI format. The AVI file is compatible with both PCs and Macs.

3.1 Step 1: Installation

MATLAB Component Runtime (MCR) module needs to be installed on the host machine. A MATLAB license is not required for the MCR installation. The MCR installer utility program (`MCRInstaller.exe`) will be provided as part with the animation tool. The MATLAB Component Runtime installation procedure is provided in MATLAB Compiler User's Guide [2], page 4-6. It is also available on the web. The instructions are provided for the Windows and Linux platforms.

<http://www.mathworks.com/access/helpdesk/help/toolbox/compiler/999353>

3.1.1 Windows installation

The following components make up the animation application on Windows platforms.

<code>MCRInstaller.exe</code>	Self-extracting MATLAB Component Runtime library utility; Platform-dependent file that must correspond to the end user's platform.
<code>animation_driver.exe</code>	Animation application
<code>animation_driver.ctf</code>	Component Technology File archive; Platform-dependent file that must correspond to the end user's platform.

Copy `animation_driver.exe` and `animation_driver.ctf` files to the local drive in the same directory as the trajectory data. The animation program is provoked by double-clicking on the `animation_driver.exe` file.

3.1.2 Linux installation

For the Linux operating system, the following components will be provided.

<code>MCRInstaller.zip</code>	Self-extracting MATLAB Component Runtime library archive; Platform-dependent file that must correspond to the end user's platform.
<code>unzip</code>	Utility to unzip <code>MCRInstaller.zip</code> (optional). The target machine must have an unzip utility installed.
<code>animation_driver</code>	Animation application
<code>animation_driver.ctf</code>	Component Technology File archive; Platform-dependent file that must correspond to the end user's platform.

3.2 Step 2: Set up local directory

The user needs to set up a local directory containing all the case specific files, such as the POST II input file, output file, Matfile, and other setup files. These files contain mission specific data such as the

trajectory, movie setup (frame rate, frame size, background color, lighting, etc), and links to where the geometries for the vehicles are stored. The following is a listing of local files:

```
POSTfile.inp
POSTfile.out
POSTfile.mat
master_setup_inputs.txt
camera_properties.txt
geom_data
```

Samples of these files can be found in the following directory:

```
/usr/people/v3u2/vab/rais/animation_local_dir
```

3.3 Step 3: POST II trajectory data

The POST II input file, output file, and the Matfile need to be copied to the local directory where they are processed and prepared for animation.

3.3.1 POST II Input file

Some parameters are extracted from the POST II input file. This includes planet rotation rate, mass of each vehicle, vehicles activation/deactivation, event sequencing, line activation/deactivation, line attach point locations, etc. The input parser also looks through all the included files to gather information. The user needs to make sure the path to the included files remains accurate if the input file and the include files are copied from their original directory. If the input file contains a “* milestone read” statement, it should be replaced by “* include milecreate_fname.inp”, where “milecreate_fname.inp” is the input file that created the binary milestone.

3.3.1 POST II Matfile

The animation tool expects the following variables in the POST II Matfile for each active vehicle. Make sure they are included in the profile. For POST II variable definitions see Reference 1. The burden of choosing a proper time interval between the data points is placed on the user. Choosing a proper time interval is a function of the desired movie frame rate and the apparent speed of the movie. See section 3.4 for further discussion.

```
time - trajectory time
longi - inertial longitude
decln - declination
ib matrix - inertial to body direction cosine matrix (9 elements)
xi, yi, zi - cartesian coordinates of the vehicle CM in inertial frame
gammar - planet relative flight path angle
azvelr - azimuth of the planet relative velocity vector
xcg, ycg, zcg - location of cg with respect to the BR frame
```

3.3.3 POST II output file

The event numbers were obtained from the POST II input file, and the corresponding event times are extracted from the POST II output file.

3.4 Step 4: Setting up master_setup.txt

Next, the user needs to fill in appropriate data for the `master_setup.txt` input file. This file lets the user specify data that is global in nature, such as the background color, number of movie frames per second, lighting parameters, movie name, and link to of various input files. The variable names are descriptive for convenience. All colors are input as three integers representing the Red, Green, and Blue (RGB) intensities ranging from 0 to 255. The entries in **bold** are required entries, and the entries in *italic* are informational and will be displayed with a dial or as a bar graph. Display of informational parameters is completely optional. Other parameters in plain font are required parameters but will default to the values listed below if the user does not provide input.

```
post_input_file      = c1_animation.inp      // mandatory entry
post_output_file    = c1_animation.out    // mandatory entry
post_matfile       = c1_animation_25fps.mat // mandatory entry
movie_name           = mera_take2.avi      // default output.avi if not input
frames_per_second  = 25                // mandatory entry
projection            = perspective        // other option: orthographic
num_pixels_horizontal = 600                // movies always 3 X 2 aspect ratio

light_color(1)       = 255 255 255        // by default, only one light source
light_color(2)       = 210 210 160        // additional optional light source
light_position_ned(1) = 0 0 -500           // default position of light 1
light_position_ned(2) = -4 30 -100        // position of additional light
background_color     = 0 0 0              // background black by default
gauge_color          = 150 150 150        // all gauges in gray by default

starting_index       = 1                  // starting index defaulted to 1
ending_index         = 3516               // defaulted to the length of trajectory
step                 = 1                  // use higher numbers to reduce run time

altimeter_var        = gdalt              // altitude up to 100 km
radar_altimeter_var  = hgtagl            // altitude up to 500 m
G_meter_var          = asmg              // acceleration in Earth Gs
mach_meter_var       = mach              // mach meter
airspeed_var         = velr              // airspeed gauge
horiz_vel_var        = N_vel E_vel       // horizontal velocity gauge
fuel_gaguge_var      = wprop             // fuel usage
fuel_capacity        = 234               // amount of fuel when full

altimeter_assigned_spot = 5              // default altimeter spot
radar_altimeter_assigned_spot = 6        // default radar altimeter spot
G_meter_assigned_spot  = 7              // default G meter spot
mach_meter_assigned_spot = 8            // default mach meter spot
airspeed_assigned_spot = 9              // default airspeed gauge spot
horiz_vel_assigned_spot = 10            // default horizontal vel gauge spot
fuel_gaguge_assigned_spot = 11          // default fuel usage spot

display_post_variable_name(1) = gammar_1 // displayed as a bar graph
display_post_variable_name(2) = velr_1   // displayed as a bar graph
display_post_variable_put_right(1) = 1    // displayed on the right side

AmbientStrength      = 0.4
DiffuseStrength      = 0.6
SpecularStrength     = 1.1
SpecularColorReflectance = 1.0
SpecularExponent     = 20
```

animation_setup.txt listing

The `frames_per_seconds` parameter specifies the number of frames that are combined to make one second of animation. If the reciprocal of this parameter is equal to the time interval provided by the Matfile, then the movie will appear to be in real time. Other combinations of frame rate and output time increment can be used to simulate slow motion or fast-forwarding effects. For example, if `frames_per_seconds = 20` and Matfile time interval is 0.1, the movie will appear to be twice the normal speed. The recommended values for `frames_per_seconds` are 20 and 25. The movie will be choppy for small frame rates around 10, and the monitor displays cannot keep up with frame rate of 50 and higher.

`Starting_index`, `ending_index` and `step` specify a subset of the trajectory to be animated. If left out, `starting_index` will default to one, `ending_index` will default to the length of the trajectory, and `step` equals one. By default, the entire trajectory is animated.

`Display_post_variable_name(n) = 'POST variable'` fields are optional. The POST II variables specified in this field are displayed on the bottom of the screen as moving bar graphs. The variable names need to be exactly as they appear in the POST II Matfile. By default, the bar graphs move horizontally from left to right when increasing in value. Bars graphs are stacked from bottom to top if more that one is requested. If desired, the bar graphs can be displayed on the right of the screen. This is accomplished by exercising the `display_post_variable_put_right(n) = 1` option.

The animation tool supports the display of some parameters by dials and gauges. The following parameters are available for dial display: time, net velocity, horizontal velocity, acceleration in Gs, mach number, altitude up to 100 km, altitude up to 500 m, spent fuel. Dials and gauges are automatically assigned a spot on the screen, but the spot assignment can be overridden by the user in the `animation_setup.txt` file. Figure 2 shows the default dial and gauge spot assignments.

5 altimeter	9 airspeed	10 horizontal velocity	11 fuel gauge	1 clock
6 radar altimeter				2 not used
7 G meter				3 not used
8 mach meter	12 not used	13 not used	14 not used	4 not used

Figure 2 Default gauge locations

Figure 3 is an example of all the gauges being used on a Mars entry animation. All the gauges are placed in their default location in this example.

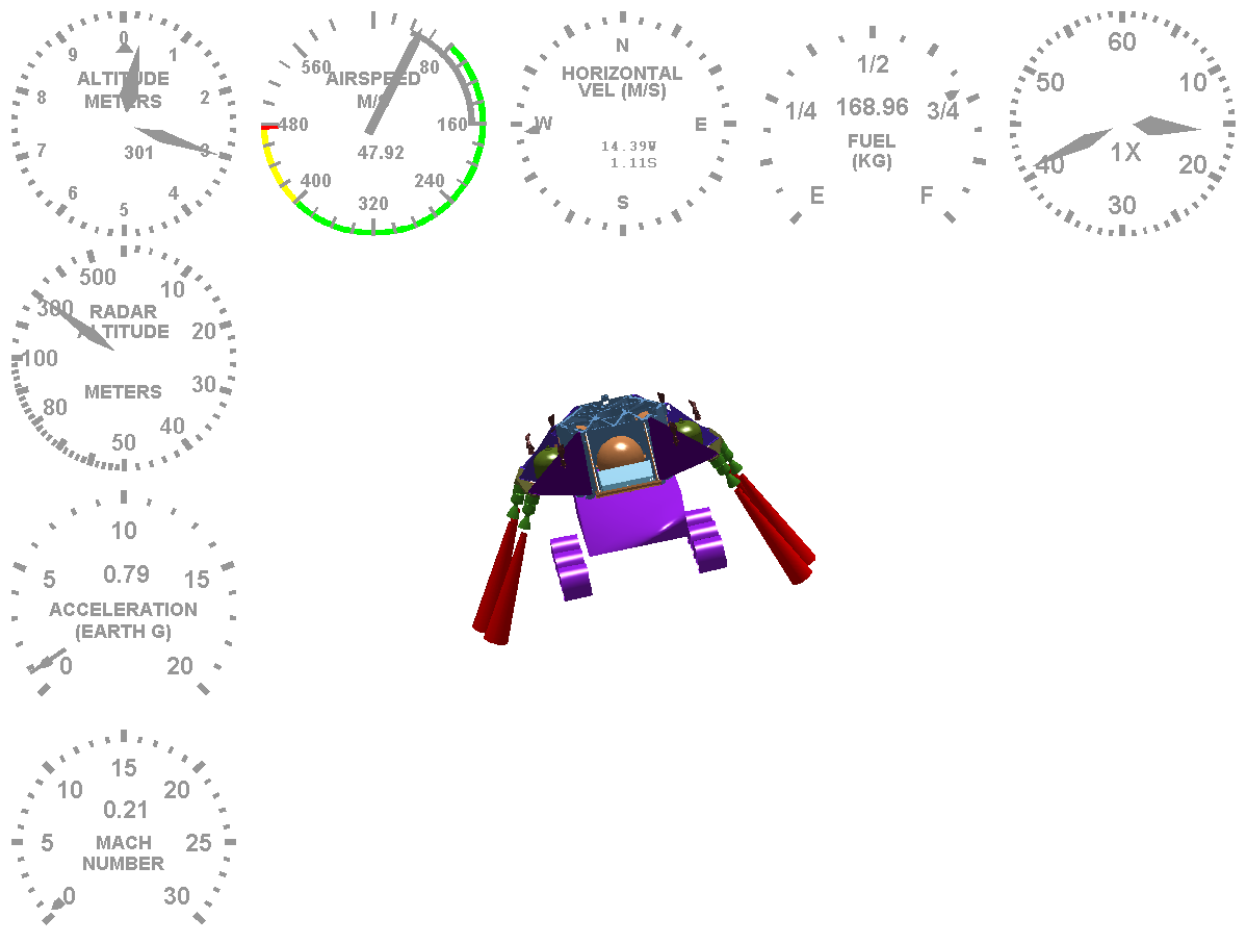


Figure 3 Example of gauges

3.5 Step 5: Setting up geom_data

The `geom_data` text file is responsible for setting up the geometrical representation of vehicles in flight. The syntax of `geom_data` is similar to the POST II input deck. Vehicle geometries are input at the start of POST II events. The animation tool expects a geometry input for each vehicle when first activated. Geometry input for an existing vehicle overwrites the previous input. Overwriting previous geometry can be used to reflect vehicle appearance changes such as jettisoning of spent stages, heatshield separation, parachute deployment, etc. The `geom_data` file is integrated into other animation inputs by adding the following line to the end POST II input file: `* include geom_data`. The animation tool processes this file along with other POST II included file.

```
event          = 1
vehicle        = 1
veh_geom_file  = patch_data/capsule      / relative path to vehicle geometry
incl_cg_calc   = 1                      / include in composite cg calculation
incl_body_axes = 1                      / display body axes
eng_scale      = 1 1 2 2                / scaling of rocket engine plumes
c
event          = 45
vehicle        = 2
* declare Rc 4,2215
veh_geom_file  = patch_data/mer_parachute / relative path to vehicle geometry
incl_cg_calc   = 1                      / include in composite cg calculation
incl_body_axes = 1                      / display body axes
eng_scale(2)   = 3                      / scale rocket engine 2 by factor of 3
```

geom_data sample

The geometry is input as a collection of MATLAB patch elements. For more information on patches refer to MATLAB graphics manual. Patches are ideal for visualizing 3D objects such as aerospace vehicles and spacecrafts of arbitrary shapes. The vertices for the patches must be input in POST II Body Reference (BR) frame. The animation tool makes appropriate transformations to shift the geometrical representation of the vehicles to a planet relative coordinate system. The animation tool loads the geometrical data file specified on the right hand side of the `veh_geom_file` entry above. The file that is loaded contains an array of structures stored in `geom_data` with each element of the array making up a part. Each element of the `geom_data` array contains the coordinates of the vertices, polygons and color information. Color can be input in two ways. The first method is to specify a uniform color for a part using `geom_data(n).color = [R G B]` option. MATLAB also accommodates for each facet of the geometry to have its own color. This is accomplished by providing an additional `color_array` matrix of data in `geom_data` structure. The number of entries in `color_array` field must correspond to the number of entries in the `faces`, with each entry being the RGB intensity. The wireframe model of the Mars Exploration Rover entry capsule shown in Figure 4 contains 1152 vertices, and 1104 polygons all having the same color.

```
geom_data(1).vert0: [1152x3 double]
geom_data(1).faces: [1104x4 double]
geom_data(1).color: [226 204 190]
```

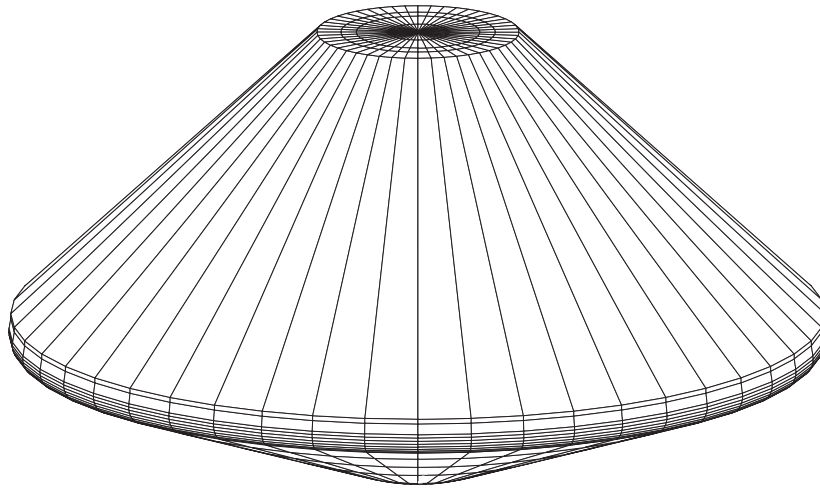


Figure 4 MER entry capsule model

Camera position and target are specified with respect to the composite vehicle center of mass. For a single vehicle in flight, composite vehicle center of mass is the same as the vehicle center of mass, but for multiple vehicle trajectory simulation, the composite vehicle center of mass is calculated at each time step. The user can exclude a vehicle from composite center of mass calculation by setting the `incl_cg_calc` flag to zero in the `geom_data` file. This is useful when a body is being jettisoned, where the desired focus of the camera is the main vehicle, not the jettisoned body.

3.6 Step 6: Setting up camera_properties.txt

The camera special effects are accomplished by manipulating the `camera_properties.txt` file. This file contains the camera position and camera target with respect to the composite vehicle center of mass as a function of time. The camera generally moves through space along with the vehicle. The user specifies where the camera is located with respect to the composite vehicle center of mass. The camera position and target points are input in the geographic frame or in the relative velocity frame. North, east, and down directions form the Cartesian X, Y, and Z axes in the geographic frame. In the velocity frame, X-axis is in direction of the relative velocity vector; Y-axis is in the local horizontal plane, and Z-axis completes the right-handed coordinate system. The camera viewing angle is also input in `camera_properties.txt` file. The camera viewing angle can be used to simulate the effect of zooming in and out. All the properties are input as a function of trajectory time. The animation tool linearly interpolates in between discreet time intervals, so all camera movements appear smooth. The following is sample of the `camera_properties.txt` file:

```
% Variables:
% CVA          : Camera viwing angle
% posi_cs     : camera position coordinate system
%              ned -> Cameral location defined in NED frame with respect to composite CG
%              fix -> Camera location stays fixed in ECR frame
%              vel -> Camera location defined in relative velocity frame
% camera_position : Camera position, if posi_cs = fix camera_position is in NED frame
% targ_cs     : Camera target coordinate system
%              ned -> Camera target defined in NED frame with respect to composite CG
%              fix -> Camera target stays fixed in ECR frame
%              vel -> Camera target defined in relative velocity frame
% camera_target : Camera target coordinatates in targ_cs coordinate system
% CFT         : Camera fixed time. Used only when posi_cs or targ_cs = fix
%
% time      CVA  posi_cs  <- camera_position ->  targ_cs  <- camera_target ->  CFT
%-----
% 0.000    45   ned      0.0   -5.0   -20.0   ned      0.0   0.0   0.0   0.0
% 300.000  45   ned      0.0   -5.0   -20.0   ned      0.0   0.0   0.0   0.0
% 320.000  45   ned     -50.0   0.0   -15.0   ned      0.0   0.0   0.0   0.0
% 1135.000 45   vel     -10.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1140.000 45   vel     -50.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1143.000 45   vel     -50.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1144.000 10   vel     -50.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1149.000 10   vel     -50.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1150.000 45   vel      15.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1155.000 45   vel      15.0  10.0   0.0    vel      0.0   0.0   0.0   0.0
% 1157.000 45   vel     -50.0   0.0   0.0    vel      0.0   0.0   0.0   0.0
% 1170.000 45   vel     -50.0   0.0   0.0    vel      0.0   0.0   0.0   0.0
% 1171.000 40   ned     -50.0   0.0  -30.0   vel     -10.0   0.0   0.0   0.0
% 1177.000 40   ned     -50.0   0.0  -30.0   vel     -17.0   0.0   0.0   0.0
% 1178.000 40   ned     -70.0   0.0   20.0   vel     -17.0   0.0   0.0   0.0
% 1185.000 40   ned     -70.0   0.0   20.0   vel     -17.0   0.0   0.0   0.0
% 1186.000 5    fix     -20.0   0.0   0.0    vel     -17.0   0.0   0.0  1192.0
% 1205.000 5    fix     -20.0   0.0   0.0    vel     -17.0   0.0   0.0  1192.0
% 1206.000 27   ned     -70.0   0.0  -90.0   vel     -25.0   0.0   0.0   0.0
% 2000.000 27   ned     -70.0   0.0  -90.0   vel     -25.0   0.0   0.0   0.0
```

Sample camera_properties.txt file

In the camera position options above, the camera location and target points are defined with respect to the composite center of mass. The animation tool also supports a fixed option for the camera where either the camera location or camera target or both remain fixed in space. This feature is activated by specifying the “fix” option in the `camera_position` and/or `camera_target` coordinate system flag. When this option is chosen, the system also looks up the Camera Fixed Time (CFT) field in the table. The

CFT field is the time when the vehicle position is looked up as the fixed location for the camera. This option simulates the effects of the vehicle flying past the camera, with the camera tracking. Setting the position and target coordinate systems to fix (columns 3 and 7) simulates the effects of a completely stationary camera.

3.7 Step 7: Run animation_driver

The animation tool is now ready to build all the data structures for generation of the movie. Double-click on the `animation_driver` executable to initiate the program. The `animation_driver` executable opens a DOS window and goes through the POST II input files, `master_setup.txt`, `geom_data`, `camera_properties.txt`, POST II output file, and the POST II Matfile to build up appropriate data structures. The movies are made in AVI format, and are compressed using Cinepak codec on the PCs. Figure 5 is a sample frame taken from one of the Mars entry animations.

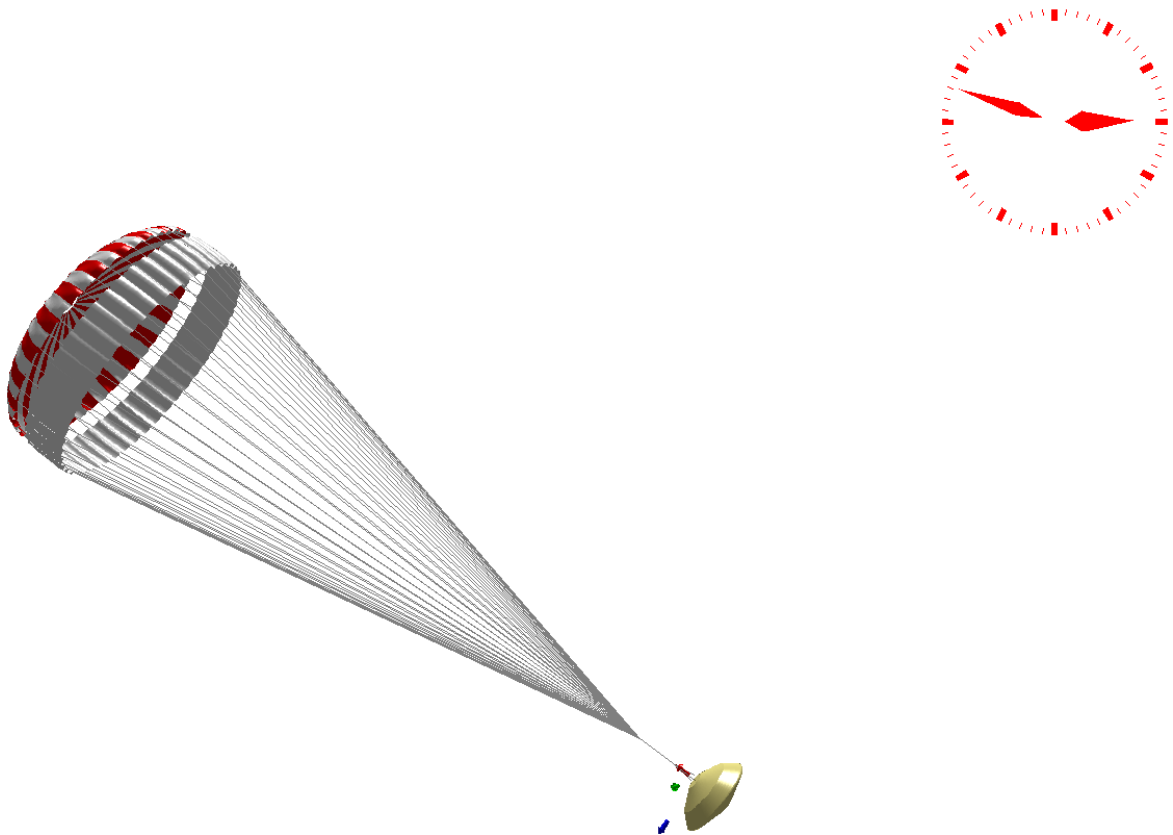


Figure 5 A sample frame from Mars entry simulation

References

- [1] Program to Optimize Simulated Trajectories: Volume II, Utilization Manual, prepared by: R.W. Powell, S.A. Striepe, P.N. Desai, P.V. Tartabini, E.M. Queen; NASA Langley Research Center, and by: G.L. Brauer, D.E. Cornick, D.W. Olson, F.M. Petersen, R. Stevenson, M.C. Engel, S.M. Marsh; Lockheed Martin Corporation, Version 1.1.1.G, May 2000.
- [2] MATLAB Compiler User's Guide, Version 4. October 2004

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
01- 07 - 2005		Technical Memorandum			
4. TITLE AND SUBTITLE POST II Trajectory Animation Tool Using MATLAB, V1.0				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Raiszadeh, Behzad				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER 23-979-20-10	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199				8. PERFORMING ORGANIZATION REPORT NUMBER L-19094	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001				10. SPONSOR/MONITOR'S ACRONYM(S) NASA	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/TM-2005-213779	
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 12 Availability: NASA CASI (301) 621-0390					
13. SUPPLEMENTARY NOTES An electronic version can be found at http://ntrs.nasa.gov					
14. ABSTRACT A trajectory animation tool has been developed for accurately depicting position and the attitude of the bodies in flight. The movies generated from This MATLAB based tool serve as an engineering analysis aid to gain further understanding into the dynamic behavior of bodies in flight. This tool has been designed to interface with the output generated from POST II simulations, and is able to animate a single as well as multiple vehicles in flight.					
15. SUBJECT TERMS MATLAB; Animation; Multibody; Simulation; Trajectory					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	18	19b. TELEPHONE NUMBER (Include area code) (301) 621-0390