

**2004**

**NASA FACULTY FELLOWSHIP PROGRAM**

**MARSHALL SPACE FLIGHT CENTER  
THE UNIVERSITY OF ALABAMA  
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE  
ALABAMA A&M UNIVERSITY**

**INTEGRATION OF LAUNCH VEHICLE SIMULATION/ANALYSIS  
TOOLS AND LUNAR CARGO LANDER DESIGN (PART 1/2)**

Prepared By:	Yeu-Sheng Paul Shiue George Brian DeJean, Accompanying Student, Undergraduate Senior Jeffrey King, USRP Student, Undergraduate Senior
Academic Rank:	Professor
Institution and Department:	Christian Brothers University Mechanical Engineering Department
NASA/MSFC Directorate:	Engineering
MSFC Colleague:	Mr. Bill Pannell

## Introduction

Simulation and analysis of vehicle performance is essential for design of a new launch vehicle system. It is more and more demand to have an integrated, highly efficient, robust simulation tool with graphical user interface (GUI) for vehicle performance and simulations. The objectives of this project are to integrate and develop launch vehicle simulation and analysis tools in MATLAB/Simulink under PC Platform [2, 3], to develop a vehicle capable of being launched on a Delta-IV Heavy Launch Vehicle which can land on the moon with the goal of pre-implanting cargo for a new lunar mission, also with the capability of selecting other launch vehicles that are capable of inserting a payload into Trans-Lunar Injection (TLI).

The vehicle flight simulation software, MAVERIC-II (Marshall Aerospace Vehicle Representation In 'C'), developed by Marshall Space Flight Center was selected as a starting point for integration of simulation/analysis tools [1]. The goals are to convert MAVERIC-II from UNIX to PC platform and build input/output GUI's in the MATLAB environment, and then integrate them under MATLAB/Simulink with other modules. Currently, MAVERIC-II has been successfully converted from UNIX to PC using Microsoft Services for UNIX subsystem on PC. Input/Output GUI's have been done for some key input/output files. Calling MAVERIC-II from Simulink has been tested.

Details regarding Lunar Cargo Lander Design are described in Part 2/2 of the paper on page X-1.

## Roadmap to the Launch Vehicle Simulation (LVSim)

The possible modules for integration using MATLAB/Simulink with GUI for parameter inputs and outputs are shown in Figure 1.

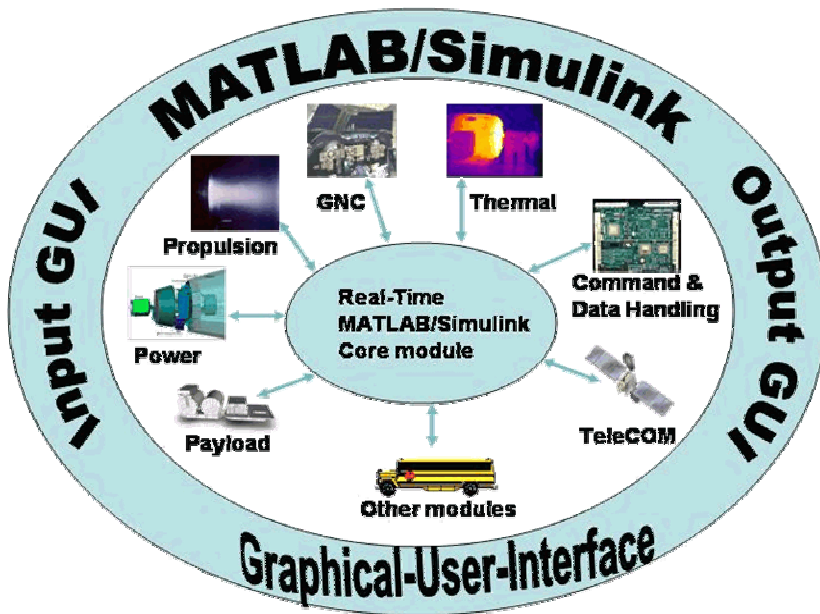


Figure 1: Integration of Launch Vehicle Simulation Tools

The optimal goal is to have all separate modules such as propulsion, GNC (guidance, navigation, and control), thermal, CDH (command & data handling), telecommunication, power, payload, and other modules, developed in MATLAB/Simulink and then integrate with input/output graphical user interface. Currently, propulsion module has been developed by the Transportation Directorate in Marshall Space Flight Center. MAVERIC-II, which was also developed by the Transportation Directorate, is mainly command-line simulation software that includes many modules that are available for integration. Therefore, conversion of MAVERIC-II to PC becomes the first step of this project. Next will be integrating propulsion module with MAVERIC-II. After that, other modules will need to be developed or extracted from MAVERIC-II to make the whole package as native MATLAB/Simulink software.

### Conversion of MAVERIC-II to PC

MAVERIC-II is a UNIX-based ANSI C 3- or 6-DOF Vehicle Flight Simulation Program. It can accommodate any vehicle configuration and flight phases. The main numerical integration engine is the TFrames software. It has become one of NASA's primary tools for simulating space transportation vehicle designs. The source codes of MAVERIC-II consist of 87 program files (77 C, 2 C++, and 8 FORTRAN) and 152 user-defined header files [4,5]. Figure 2 shows a screen shot of MAVERIC-II data screen.

```

Command Prompt - mav
MAVERIC < Marshall Aerospace Vehicle Representation in "G">
Project: GENERIC          Case: Orb_28.5nom      Version: 1.8.4 6/08/04
FltPhase.A: BST_ascent    FltPhase.B: None      Insertion: 0
time: 0.000000e+00        thrust.1: 5.289898e+06 thrust.2: 0.000000e+00
alt_Ft.B: 1.999995e+01    alphaDeg.B: -4.500000e+01 fEngineX.1: -5.284023e+06
headDeg.B: 0.000000e+00  betaDeg.B: 3.526439e+01  oxRem.1: 2.182622e+06
latDeg.B: 2.866168e+01    phibkDeg.B: 1.501402e+02  fuelRem.1: 8.083780e+05
lonDeg.B: 2.794500e+02    alphaCmdDeg: 0.000000e+00 oxRem.2: 5.871430e+05
vRelMagFt.B: 5.682581e-12 betaCmdDeg: 0.000000e+00 fuelRem.2: 9.785700e+04
mach.B: 5.090372e-15     phibkCmdDeg: 0.000000e+00 Stage_sep: 0
windSpeed.B: 5.682581e-12 qBarPsf.B: 3.835552e-26  Stage_sep: 0
fpaDeg.B: 9.000000e+01   qAlpha.B: -1.725998e-24  eng1_pwr.1: 1
eng2_pwr.1: 1            eng3_pwr.1: 1          eng4_pwr.1: 1
eng1_pwr.2: 0            eng2_pwr.2: 0          eng3_pwr.2: 0
fEngineX.1: -5.284023e+06 fEngineY.1: 0.000000e+00 fEngineZ.1: 2.492463e+05
axAccel.B: 1.169997e+00  qBeta.B: 1.352584e-24   azimToTAEM.: 9.454096e+01
normAccel.B: -5.518852e-02 alt_Nm.B: 3.291569e-03  rangeToHAC.: 4.911030e+03
totalMass.B: 4.516272e+06 heatLoad.B: 0.000000e+00 rangeToGoNM.: 4.911030e+03
speedSound.: 1.116339e+03 thetaSensed: -6.361109e-15 psiSensed.B: 1.908333e-14
phiSensed.B: -9.541664e-15 thetaNED.B: 8.999992e+01  mrCmd.1: 2.700000e+00
phiNED.B: -9.000000e+01 throttleCmd: 1.000000e+00 mrCmd.2: 6.000000e+00
aeroForceX.: -5.589382e-23 aeroForceY.: 0.000000e+00 aeroForceZ.: -8.625764e-23

Select -> Next Run Scroll Labels Quit Display Set Page 1 of 1

```

Figure 2: MAVERIC-II Data Screen

Since UNIX environment and PC Windows environment are completely different, the authors tried to use different approaches for conversion. MinGW 3.1 was first selected as a compilation tool which includes GNU C and FORTRAN compilers. However, the standard libraries and header files in MinGW are different from those on UNIX. Therefore, compilation was not successful. Cygwin was installed to emulate the UNIX environment on top of MinGW during the next try. Some head files are still not quite compatible. Finally, Microsoft Services for UNIX (SFU) 3.5 was selected as a subsystem on PC [6, 7]. The SFU is an interoperability

designed to streamline interaction between the Windows Server 2003 Family, Windows XP Professional and the Windows 2000 Family of operating systems and UNIX or Linux systems in heterogeneous networks. The Interix SDK that comes with SFU provides support for UNIX threaded applications and updated UNIX utilities and APIs.

With some changes on the MAVERIC-II source codes, all MAVERIC-II source codes were recompiled, linked, and run on PC under Windows XP Professional with SFU installed. Since the standard ASCII text file format on UNIX and PC are different, the source codes for input from a data file had to be modified. The comparisons of computation speed for MAVERIC-II to run on UNIX via Telnet and PC standalone are shown in Figure 3.

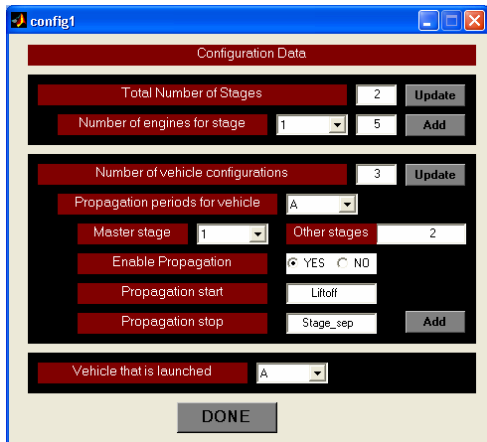
	1st Menu Case Selection	Popup Data Screen	1st Run up to 165.17 sec. w/ 1 sec. refresh	Continue Run up to 437.3695 sec.
MAVERIC on UNIX using Telnet	Immediate	1 sec.	15 sec.	18 sec.
MAVERIC on PC standalone	Immediate	1 sec.	12 sec.	17 sec.
MAVERIC on PC under MATLAB/Simulink	Immediate	1 sec.	12 sec.	18 sec.

Figure 3: Calculation Speed Comparisons

At the time of testing, MAVERIC-II was running on DEC ALPHA with eight 560MHz processors under Compaq Tru64 UNIX v5.1 operating system and Pentium 4 3.06GHz laptop with 512MB memory. The testing results will vary depending on the machines.

**Input/Output GUI**

Input and output graphical-user-interface for key data files has been developed. These GUIs can be standalone or integrated together in MATLAB/Simulink environment. These GUI modules will create exactly the same data files for MAVERIC-II setup and configuration. User can read in existing data and perform modifications. Figure 4 shows an example of input GUI for configuration file and its data file.



```

num_stages 2 # Total number of stages in this simulation.
num_engines 5 3 # Number of engines on each stage respectively.

num_vehicles 3 # Number of vehicle configurations in this simulation
# A Vehicle Configuration is a stack of stages
# to be propagated (can be a single stage).

```

```

-----
I N P U T #2
-----
#
# Vehicle Configurations and Propagation Periods
#
# In the following table, specify the stages that comprise each
# Vehicle Configuration. The first stage listed will be the "Master"
# stage. This is the stage that will be propagated and whose variables
# will be updated during propagation.
#
# Also specify in this table when the trajectory for each vehicle
# configuration is to be propagated. The Name of a mission events
# (see data/events.dat) or a literal time value can be used to indicate
# the propagate start and stop times. When the integration time reaches
# the Start time, the propagation of the stage stack will be triggered
# IF the enable flag is set to "Y" or "1".
#
#
#
#

```

PROPAGATION PERIODS FOR EACH CONFIGURATION					
	Stages	Enable	Propagation	Propagation	Propagation
			START	STOP	
vehicle_A	1 2	Y	Liftoff	Stage_sep	\$
vehicle_B	2	Y	Stage_sep	ORB_meco	\$
vehicle_C	1	N	Stage_sep	BST_impact	\$

NOTES: The first stage listed is the "Master" stage for that configuration.  
The Mission Events are defined in dataw/events.dat .

Figure 4: Example Input for Configuration and Data File

Since output data from MAVERIC-II are basically multiple columns data for different sets of output variables. An output GUI was developed to read a certain output data file. It will extract the header of the data file and allow user to select different variables to plot against each other. Figure 5 shows an example of output GUI and its data file.

FltPhase.A	FltPhase.B	Insertion time	thrust.1	thrust.2	alt_Ft.B	alphaDeg.B	f
9999999	9999999	0.000000e+00	0.000000e+00	5.289999e+06	0.000000		
9999999	9999999	0.000000e+00	1.000000e+00	5.925964e+06	0.000000		
9999999	9999999	0.000000e+00	2.000000e+00	5.920170e+06	0.000000		
9999999	9999999	0.000000e+00	3.000000e+00	5.920518e+06	0.000000		
9999999	9999999	0.000000e+00	4.000000e+00	5.921010e+06	0.000000		
9999999	9999999	0.000000e+00	5.000000e+00	5.921494e+06	0.000000		
9999999	9999999	0.000000e+00	6.000000e+00	5.922436e+06	0.000000		
9999999	9999999	0.000000e+00	7.000000e+00	5.923372e+06	0.000000		
9999999	9999999	0.000000e+00	8.000000e+00	5.924461e+06	0.000000		
9999999	9999999	0.000000e+00	9.000000e+00	5.925703e+06	0.000000		
9999999	9999999	0.000000e+00	1.000000e+01	5.927100e+06	0.000000		
9999999	9999999	0.000000e+00	1.100000e+01	5.928653e+06	0.000000		
9999999	9999999	0.000000e+00	1.200000e+01	5.930361e+06	0.000000		
9999999	9999999	0.000000e+00	1.300000e+01	5.932222e+06	0.000000		
9999999	9999999	0.000000e+00	1.400000e+01	5.934240e+06	0.000000		
9999999	9999999	0.000000e+00	1.500000e+01	5.936411e+06	0.000000		
9999999	9999999	0.000000e+00	1.600000e+01	5.938737e+06	0.000000		
9999999	9999999	0.000000e+00	1.700000e+01	5.941216e+06	0.000000		
9999999	9999999	0.000000e+00	1.800000e+01	5.943855e+06	0.000000		
9999999	9999999	0.000000e+00	1.900000e+01	5.946641e+06	0.000000		
9999999	9999999	0.000000e+00	2.000000e+01	5.949581e+06	0.000000		
9999999	9999999	0.000000e+00	2.100000e+01	5.952671e+06	0.000000		
9999999	9999999	0.000000e+00	2.200000e+01	5.955911e+06	0.000000		
9999999	9999999	0.000000e+00	2.300000e+01	5.959298e+06	0.000000		
9999999	9999999	0.000000e+00	2.400000e+01	5.962829e+06	0.000000		
9999999	9999999	0.000000e+00	2.500000e+01	5.966502e+06	0.000000		
9999999	9999999	0.000000e+00	2.600000e+01	5.970316e+06	0.000000		
9999999	9999999	0.000000e+00	2.700000e+01	5.974266e+06	0.000000		
9999999	9999999	0.000000e+00	2.800000e+01	5.978350e+06	0.000000		
9999999	9999999	0.000000e+00	2.900000e+01	5.982569e+06	0.000000		
9999999	9999999	0.000000e+00	3.000000e+01	5.986904e+06	0.000000		
9999999	9999999	0.000000e+00	3.100000e+01	5.991367e+06	0.000000		
9999999	9999999	0.000000e+00	3.200000e+01	5.995949e+06	0.000000		
9999999	9999999	0.000000e+00	3.300000e+01	6.000646e+06	0.000000		
9999999	9999999	0.000000e+00	3.400000e+01	6.005452e+06	0.000000		
9999999	9999999	0.000000e+00	3.500000e+01	6.010363e+06	0.000000		
9999999	9999999	0.000000e+00	3.600000e+01	6.015374e+06	0.000000		

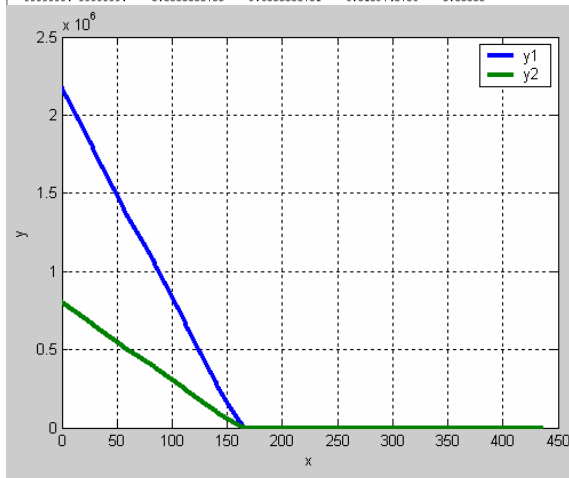
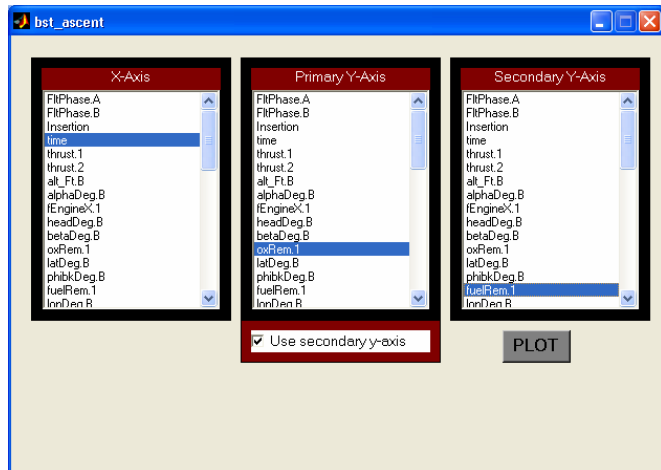


Figure 5: Example Output GUI and Data File

## **Conclusions**

As a result of this project, key input GUI modules for input data files were developed under MATLAB. MAVERIC-II was successfully converted and run on PC. Comparison of the calculation speed has been performed. The testing the existing propulsion module in Simulink and calling MAVERIC-II in Simulink has been done. This project shows great potential for future vehicle simulation and analysis. It provides user-friendly GUIs for input and output. After integrating all modules in Simulink, users can run real-time simulation. Also, data output could be streamed to Satellite ToolKit (STK) for visualization, communications and power analyses. Other modules such as Thermal, GNC, Telecommunication, and Data Handling will be converted to MATLAB/Simulink when they are available. It is found that MATLAB/Simulink is a great tool to create user friendly GUI for real-time simulations in different levels.

Clearly, Microsoft Services for UNIX has great advantages in migrating from UNIX to PC. It provides an UNIX subsystem for PC that will not sacrifice the performance of the PC.

Future works that lead to final integration include testing MAVERIC-II on other PCs, linking propulsion Simulink module to MAVERIC-II, acquiring other Simulink modules to be integrated, and continuing improvement of Input/Output GUI. The process of integration will take place piece-by-piece when necessary modules are available. Details regarding Lunar Cargo Lander Design are presented on page X-1.

## **Acknowledgements**

The authors would like to thank Mr. Bill Pannell, Mr. Bert Dolerhie, Mr. Geoffrey Beech, and all personnel in Engineering Directorate for providing a quiet but stimulating work environment and support. Also, the authors would like to thank the MAVERIC group in Transportation Directorate for their technical support.

## **References**

1. Jim McCarter, "Marshall Aerospace Vehicle Representation In 'C' (MAVERIC-II) User's Guide", 2<sup>nd</sup> Generation, NASA-MSFC.
2. James B. Dabney, "Mastering SIMULINK", Prentice Hall, 2004.
3. Stephen J. Chapman, "MATLAB Programming for Engineers", 2<sup>nd</sup> edition, Brooks/Cole, 2002.
4. Samuel P. Habison III, "C, A Reference Manual", 5<sup>th</sup> edition, Prentice Hall, 2002.
5. Peter Aitken and Bradley, "Teach Yourself C in 21 Days", Sams Publishing, 1994.
6. "Windows Services for UNIX 3.5 Reviewer's Guide", Microsoft Corp., 2004.
7. "Windows Services for UNIX 3.5 White Paper", Microsoft Corp., 2004.