NASA FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER

THE UNIVERSITY OF ALABAMA
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE
ALABAMA A&M UNIVERSITY

Java-based Graphical User Interface for MAVERIC-II

Prepared By:	Suk Jai Seo
Academic Rank:	Assistant Professor
Institution and Department:	Middle Tennessee State University Computer Science Department
NASA/MSFC Directorate:	Transportation
MSFC Colleague:	Greg Dukeman

Introduction

A computer program entitled "Marshall Aerospace Vehicle Representation in C II, (MAVERIC-II)" is a vehicle flight simulation program written primarily in the C programming language. It is written by James W. McCarter at NASA/Marshall Space Flight Center. The goal of the MAVERIC-II development effort is to provide a simulation tool that facilitates the rapid development of high-fidelity flight simulations for launch, orbital, and reentry vehicles of any user-defined configuration for all phases of flight.

MAVERIC-II has been found invaluable in performing flight simulations for various Space Transportation Systems. The flexibility provided by MAVERIC-II has allowed several different launch vehicles, including the Saturn V, a Space Launch Initiative Two-Stage-to-Orbit concept and a Shuttle-derived launch vehicle, to be simulated during ascent and portions of on-orbit flight in an extremely efficient manner. It was found that MAVERIC-II provided the high fidelity vehicle and flight environment models as well as the program modularity to allow efficient integration, modification and testing of advanced guidance and control algorithms. In addition to serving as an analysis tool for techno logy development, many researchers have found MAVERIC-II to be an efficient, powerful analysis tool that evaluates guidance, navigation, and control designs, vehicle robustness, and requirements.

MAVERIC-II is currently designed to execute in a UNIX environment. The input to the program is composed of three segments: 1) the vehicle models such as propulsion, aerodynamics, and guidance, navigation, and control 2) the environment models such as atmosphere and gravity, and 3) a simulation framework which is responsible for executing the vehicle and environment models and propagating the vehicle's states forward in time and handling user input/output. MAVERIC users prepare data files for the above models and run the simulation program. They can see the output on screen and/or store in files and examine the output data later. Users can also view the output stored in output files by calling a plotting program such as gnuplot. A typical scenario of the use of MAVERIC consists of three-steps; editing existing input data files, running MAVERIC, and plotting output results.

The three steps described are conducted by a sequence of numerous UNIX commands, and it requires users' considerable amount of knowledge of MAVERIC program and data files. We developed a graphical user interface (GUI) so that it makes MAVERIC much more user-friendly and it allows MAVERIC users more productive. Users select necessary steps required to conduct a flight simulation from the menus that are provided by the GUI instead of invoking multiple, separate UNIX commands. The GUI we developed is written in Java programming language, which has many advantages over other programming environment. Java is free and it is available from java.sun.com for all modern platforms with no license restrictions. It is object-oriented and there are a huge amount of pre-written packages available, which are ready-to-use or easily extendable. It is platform independent, so the same program can be run on any environment without recompiling. Java is consistent with traditional MAVERIC philosophy, i.e., all GUI source code is available (provides low-level insight into the GUI) and modifiable by TD54 developers so that improvements can be made.

User Interaction With MAVERIC (command-based)

Many input data files are needed to run simulation using MAVERIC. These data files are composed of three segments; the vehicle models such as propulsion and aerodynamics, the environment models such as atmosphere and gravity, and a simulation framework that specifies simulation options. Interaction with MAVERIC is currently command-line based (UNIX commands). Typically MAVERIC users change/modify contents of these input data files before the simulation, execute MAVERIC, and view the simulation results by invoking many different UNIX commands as shown in Figure 1. It requires user's knowledge of name and location of data files and type of data in each data file. Users also should know how to invoke different programs.

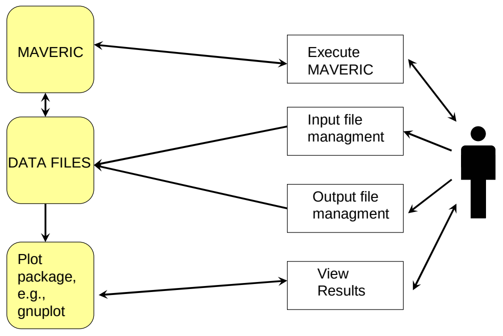


Figure 1: Command-based MAVERIC

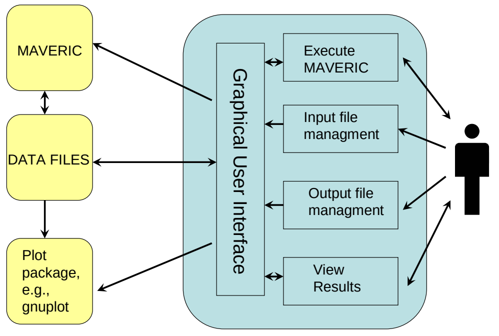


Figure 2: GUI-based MAVERIC

GUI Architecture(GUI-based MAVERIC)

The GUI we developed provides a more user-friendly environment for the MAVERIC users. Users do not have to invoke any UNIX commands; the GUI is a menu- and mouse-based application. Different menu items in GUI replace the functionalities of most of UNIX commands that used be required for the flight simulation using MAVERIC. Users can manage input/output data files, execute MAVERIC, and view the simulation results by selecting options that are provided in GUI as shown in Figure 2.

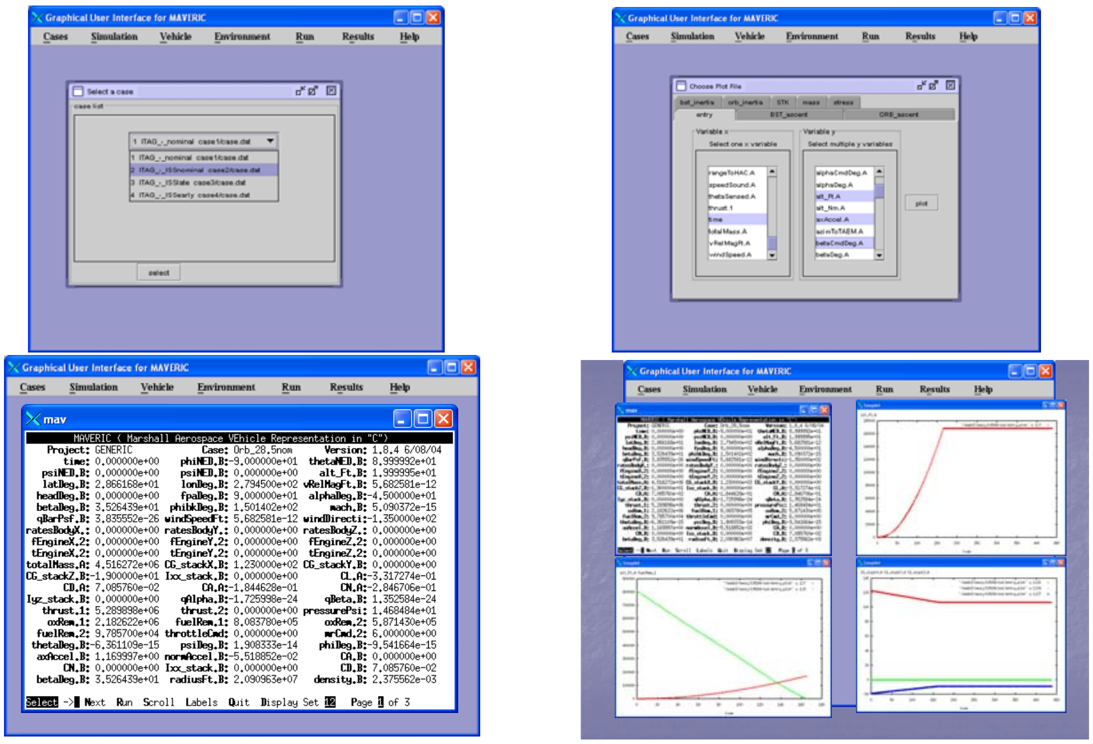
GUI Design

There are three menu categories provided in the GUI; editing, running MAVERIC, and results. The edit menu allows the user to change any input data before the simulation by guiding through the editing process. The run menu provides the user a various options to execute the flight simulation program, MAVERIC. After or during the simulation the user can not only watch the screen output from MAVERIC but also plot the output data that is stored in the output data files produced by running MAVERIC. All of the above functionalities are provided directly from the GUI menu options.

Features of GUI

In addition to the functionalities described above, the GUI contains the following features. Some of the features are shown in Figure 3.

- Find the appropriate input data files for the case selected by the user
  - Users are not required to know the location of input or output data files
  - Check out (co) files for the user if necessary
- Allows multiple cases of flight simulations
  - GUI dynamically adapts to a new case so that the user doesn't have to remember the 'case' directory or the 'case' file name – important because the these names are arbitrary and not always easy to remember
  - Allows user to modify new case data
  - Dynamically updates GUI menu-items
- Provides very useful menu-items for plotting output data
  - Show the output files created by the most recent simulation
  - Easily switch among different output files
  - Show the variables to plot in alphabetical order
  - Invoke plotting program for the user, continually update graphs while MAVERIC is running
- Shows useful message dialog boxes for the user
  - Provides helpful information on MAVERIC
  - Provides appropriate, informative error messages


Graphical User Interface for MAVERIC
Cases
Simulation
Vehicle
Environment
Run
Results
Help
Select a case
case list
1 ITAG_-_nominal case1/case.dat
2 ITAG_-_ISSnominal case2/case.dat
3 ITAG_-_ISSlate case3/case.dat
4 ITAG_-_ISSearly case4/case.dat
select
Choose Plot File
entry
Variable x
Select one x variable
Variable y
Select multiple y variables
plot
mav

```
        MAVERIC ( Marshall Aerospace VEhicle Representation in "C")
   Project: GENERIC                Case: Orb_28.5nom       Version: 1.8.4 6/08/04
      time: 0.000000e+00    phiNED.B:-9.000000e+01  thetaNED.B: 8.999992e+01
  psiNED.B: 0.000000e+00    psiNED.B: 0.000000e+00    alt_Ft.B: 1.999995e+01
  latDeg.B: 2.866168e+01    lonDeg.B: 2.794500e+02 vRelMagFt.B: 5.682581e-12
 headDeg.B: 0.000000e+00    fpaDeg.B: 9.000000e+01 alphaDeg.B:-4.500000e+01
 betaDeg.B: 3.526439e+01  phibkDeg.B: 1.501402e+02      mach.B: 5.090372e-15
 qBarPsf.B: 3.835552e-26 windSpeedFt: 5.682581e-12 windDirecti:-1.350000e+02
ratesBodyX.: 0.000000e+00 ratesBodyY.: 0.000000e+00 ratesBodyZ.: 0.000000e+00
fEngineX.2: 0.000000e+00  fEngineY.2: 0.000000e+00  fEngineZ.2: 0.000000e+00
tEngineX.2: 0.000000e+00  tEngineY.2: 0.000000e+00  tEngineZ.2: 0.000000e+00
totalMass.A: 4.516272e+06 CG_stackX.B: 1.230000e+02 CG_stackY.B: 0.000000e+00
CG_stackZ.B:-1.900000e+01 Ixx_stack.B: 0.000000e+00        CL.A:-3.317274e-01
      CD.A: 7.085760e-02        CA.A:-1.844628e-01        CN.A:-2.846706e-01
Iyz_stack.B: 0.000000e+00    qAlpha.B:-1.725998e-24     qBeta.B: 1.352584e-24
  thrust.1: 5.289898e+06    thrust.2: 0.000000e+00 pressurePsi: 1.468484e+01
   oxRem.1: 2.182622e+06   fuelRem.1: 8.083780e+05     oxRem.2: 5.871430e+05
 fuelRem.2: 9.785700e+04 throttleCmd: 0.000000e+00     mrCmd.2: 6.000000e+00
thetaDeg.B:-6.361109e-15    psiDeg.B: 1.908333e-14   phiDeg.B:-9.541664e-15
 axAccel.B: 1.169997e+00 normAccel.B:-5.518852e-02        CA.B: 0.000000e+00
      CN.B: 0.000000e+00 Ixx_stack.B: 0.000000e+00        CD.B: 7.085760e-02
 betaDeg.B: 3.526439e+01  radiusFt.B: 2.090963e+07   density.B: 2.375562e-03

Select -> Next  Run  Scroll  Labels  Quit  Display Set 12   Page 1 of 3
```

Figure 3: Snap shots of GUI

Resources

The GUI was developed in Java programming language, which can be run on any platforms. However, the GUI uses the gnuplot plotting program, so currently the GUI can be run in UNIX only. MAVERIC is also run in UNIX now, but in the future when MAVERIC is ready to be run on other platforms, plotting option should be replaced by a Java-based plotting package.

Conclusion/Future Works

The Java-based GUI we developed makes MAVERIC-II, an invaluable simulation tool for various Space Transportation Systems, much more user-friendly and much more useful. The project of this summer was focused on reducing the input/output file management burden on the user, and adding features that reduce the learning curve of MAVERIC is recommended in the future. Instead of invoking gnuplot program to view/plot the output, incorporating Java-based plotting package into the GUI will provide GUI developers/maintainers more control over plotting functions and make the GUI more platform-independent. This will be much easier with data file structure enhancements being undertaken by TD54, i.e., any enhancements to data file structure will directly benefit the GUI design.

Acknowledgements

The author would like to thank Greg Dukeman of TD54 for providing an opportunity to participate in the space vehicle simulation area of the Marshall Space Flight Center's Transportation Directorate and many hours of valuable discussion and guidance. The author also appreciates other team members' support on the research including Jim McCarter, John Hanson, and Curtis Zimmerman.

References

[1] McCarter, J. (2004), "MAVERIC-II Vehicle Flight Simulation Software," NASA-MSFC, Space Transportation Directorate, Vehicle & Systems Development Dept, GN&C Group/TD54, Guidance & Trajectory Team.

[2] McCarter, J. (2004), "MAVERIC-II User's Guide," NASA-MSFC.

[3] Deitel & Deitel. (2003), *Java How To Program*, Prentice Hall.

[4] Sun Microsystems (2004), "The Java Tutorial", http://java.sun.com/docs/books/tutorial/

[5] Sun Microsystems (2004), "The Source for Developers", http://java.sun.com/docs/